Recent Experiments with INQUERY

James Allan, Lisa Ballesteros, James P. Callan, W. Bruce Croft, and Zhihong Lu
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, Massachusetts 01003, USA

Past TREC experiments by the University of Massachusetts have focused primarily on ad-hoc query creation. Substantial effort was directed towards automatically translating TREC topics into queries, using a set of simple heuristics and query expansion. Less emphasis was placed on the routing task, although results were generally good. The Spanish experiments in TREC-3 concentrated on simple indexing, sophisticated stemming, and simple methods of creating queries.

The TREC-4 experiments were a departure from the past. The ad-hoc experiments involved "fine tuning" existing approaches, and modifications to the INQUERY term weighting algorithm. However, much of the research focus in TREC-4 was on the routing, Spanish, and collection merging experiments. These tracks more closely match our broader research interests in document routing, document filtering, distributed IR, and multilingual retrieval.

The University of Massachusetts' experiments were conducted with version 3.0 of the INQUERY information retrieval system. INQUERY is based on the Bayesian inference network retrieval model. It is described elsewhere [7, 5, 12, 11], so this paper focuses on relevant differences to the previously published algorithms.

1 Description of Ad-Hoc Experiments

For the ad-hoc retrieval experiments, the major change to the system was a new estimation technique for term weighting. We also continued to refine our analysis techniques for the TREC topics, our use of passage retrieval, and query expansion using InFinder[1].

1.1 Query Processing

TREC topics 201–250 differ from earlier TREC topics in that the <title> fields were removed. This change makes the TREC topics even more dissimilar from user queries in an online system than in the past. The TREC topics observe the niceties of grammar, punctuation and, especially, polite periphrasis. In an online system, users typically discard grammar, punctuation and any non-functional verbiage in an effort to get the information they want.

The removal of the <title> field created a set of topics that resembled essay questions. Much of our TREC processing this year focussed on creating queries that more closely resemble "real" online queries, by stripping off the polite circumlocution and its accompanying grammar. As a result, in addition to the standard "stop-phrase" program distributed with INQUERY, which removes the occasional polite circumlocution, we resurrected an old program for removing additional verbiage that is likely to be content-free, especially in questions. For example, in topic 201

[1] Formerly called PhraseFinder.

> "What procedures should be implemented to insure that proper care is given to children placed under the au pairs' responsibility?"

only the phrase "au pair" is actually useful. Our second stage stop-phrase processing removes "what procedures should be implemented," which gives a small improvement in performance, but it is unable to strip the topic down to just the single phrase.

Besides intensified stop-phrase processing, the only innovations this year were slightly improved part-of-speech tagging (we used JTAG [14]), and removal of references to the U.S. (in the past we have alternately removed them or downweighted them).

The complete topic-to-query process consisted of the following processes in the order specified.

1. Produce words:
   (a) Remove stop-phrases.
   (b) Remove additional stop-phrases.
   (c) Remove references to U.S., generalize references to U.K, U.N., U.S.S.R and turn two-word country names into #PHRASE.
2. Produce phrases:
   (a) Tag for part of speech.
   (b) Remove stop-phrases.
   (c) Remove additional stop-phrases
   (d) Make noun groups into #PHRASES: N1 N2 N3 ⟶ #PHRASE(N1 N2) #PHRASE (N2 N3).
   (e) Remove references to U.S., generalize references to U.K, U.N., U.S.S.R and turn two-word country names into #PHRASE.
   (f) Remove company suffixes, such as "Inc."
   (g) Discard anything that does not form a #PHRASE.
3. Combine words and phrases for each query, using a #SUM operator.

1.2 InFinder

InFinder[2] is a technique for corpus-based query expansion [8, 1, 4]. For TREC-4, a subset of 30% of the adhoc document set was used to build the InFinder database. None of the Federal Register and Patent documents were used. Noun phrases were defined as being all single, doubles, and triples of adjacent nouns as determined by the JTAG part of speech tagger [14]. Concepts that occurred less than 16 times or greater than 3,000 times were eliminated. Terms that co-occured with a noun phrase more than 20,000 times were also eliminated.

Queries, created as described above, were used to retrieve phrases from the InFinder database. The 30 most highly ranked noun phrases were added to the queries. When adding noun phrases, each phrase was enclosed in a #3 operator and given a weight that reflected whether all terms in the noun phrase occured in the original query (called a "duplicate" phrase) or contained some or all new terms (called a "novel" phrase).

[2] Formerly called PhraseFinder.

Duplicate phrases were given a weight of $1 - \frac{1}{30} \cdot D$, where D was the number of duplicate phrases that preceded it in the phrase ranking. Thus, the first duplicate phrase was given a weight of 1.0, the second was 0.967, the third 0.933 and so forth.

Similarly, novel phrases were given a weight of $0.3 - \frac{0.3}{30} \cdot N$, where N was the number of novel phrases that preceded it in the phrase ranking. Thus, the first novel phrase was given a weight of 0.3, the second was 0.29, the third was 0.28 and so forth.

The weights used were a function of rank with respect to phrases of the same type (novel or duplicate) only. For instance, the first novel phrase would have a weight of 0.3 even if it followed 29 duplicate phrases in the overall ranking. Only the top 30 phrases were added, regardless of type.

1.3 Passage Retrieval

In [3, 4], we reported experiments that showed significant improvements in retrieval effectiveness when document rankings based on the entire document text are combined with rankings based on the best passages in the documents. The TREC-4 ad-hoc document retrieval experiments tested a new approach to using passages in retrieval. The queries used for retrieval were of the form

#SUM (#SUM (Q') #PASSAGE200 (Q))

where Q is a query created from the TREC topic (Section 1.1) and Q' is the expanded form of Q (Section 1.2).

In the past, passage-level evidence was weighted more heavily than document-level evidence. However, the new term-weighting formula (described below) improved the quality of document-level evidence sufficiently that even weighting seemed more appropriate.

1.4 Manual Modifications

These queries were generated by simulating some of the modifications a user might make to an initial query in an interactive environment. The starting point for this experiment was the automatically produced queries (INQ201, described above). Changes to these initial queries were limited to the deletion of spurious (in the user's opinion) words and phrases, modification of weights based on perceived relative importance, and adding proximity restrictions such as are often used in Boolean systems. The user spent an average of 2-3 minutes per query (2 hours for 50 queries).

1.5 Estimation

INQUERY relies on a *tf.idf* formula for estimating the probability that a document is about a concept. The estimation formulae have been used for several years, with only minor modifications [1, 5, 12, 11]. In TREC-3, we began experimenting with new formulae for the adhoc runs. The TREC-3 formulae offered only minor improvements over the more traditional formulae [1], so they were discarded.

Experiments prior to TREC-4 suggested that the Okapi treatment of tf [9] was most effective with common terms, while the INQUERY treatment of tf was most effective with infrequent terms. For TREC-4, we adopted a term weighting formula that is a combination of the two.

$$\begin{aligned} idf &= \frac{\log(\frac{C+0.5}{df})}{\log(C+1.0)} \\ T_i &= \frac{\log(tf+0.5)}{\log(max\_tf+1.0)} \end{aligned}$$

$$
\begin{aligned}
T_o &= \frac{tf}{tf + 0.5 + 1.5 \cdot \frac{L}{avg\_doclen}} \\
ntf &= d_t + (1 - d_t) \cdot (idf \cdot T_i + (1.0 - idf) \cdot T_o) \\
d_t &= \frac{\log(avg\_doclen)}{24} \cdot e^{-5 \cdot \frac{df}{C}} \\
\mathrm{bel}_{\mathrm{term}}(Q) &= 0.4 + 0.6 \cdot ntf \cdot idf
\end{aligned}
$$

where

L = the length of the document (in words, including stopwords),
tf = the frequency of term t in the document,
max_tf = the frequency of the most frequent term in the document,
df = the number of documents in which term t occurs, and
C = the number of documents in the collection.

T_i is the usual INQUERY tf weight, while T_o is the Okapi tf weight.

d_t is the minimum term frequency component when a term occurs in a document. d_t was set to 0.4 in the past. However, the appropriate value appeared to be collection-dependent, and rarely 0.4. A goal of our research was to identify automatic methods of determining d_t for a given collection. The method used in TREC-4 was based on document length, and the frequency of the term in the collection. Either of these factors makes some sense on its own. However, the combination was discovered by accident, and is difficult to justify.

The "good" part of this combination is that d_t depends upon the average document length in the collection. In other words, the average document length controls the importance of a ± 1 variation in tf.

The "bad" part of this combination is that d_t depends partly upon the frequency of the term in the collection. It is unclear why such an "idf-style" statistic should be part of the d_t.

These adjustments to the estimation formula were tested against more than just the TREC document collection. In experiments prior to TREC-4, they were found to yield improvements at all levels of recall on the CACM (2 query sets), West FSupp (2 query sets), NPL, and TREC (2 query sets) document collections. These collections vary widely in number of documents and average document length, suggesting that the new formula might be relatively robust.

2 Description of the Routing Experiments

The routing experiments for TREC-4 were an extension of past routing efforts and an incorporation of new ideas inspired by other TREC research groups. Queries were expanded by adding terms, adjacent word pairs, and nearby word pairs. The selected concepts were chosen from a large candidate set by comparing their occurrences in relevant and non-relevant training documents. Weights were assigned using the Rocchio formula applied to INQUERY's version 2.1 weighting scheme. Finally, the weights were adjusted by fitting them more closely to the training data using a technique very similar to one described by Buckley and Salton [2].

2.1 Term selection

The training data for the routing queries consisted of all known relevant documents in TREC disks 1–3, and the same number of top-ranked non-relevant documents retrieved by the original query on that database. (Non-relevant documents are those not judged relevant; they may not have been specifically judged non-relevant. The "original query" refers to the result of creating an INQUERY structured query from the original TREC topic.)

For each query, all terms occurring in the relevant documents were identified, and were then ranked by their relative occurrences in the relevant and non-relevant documents. That is, by:

$$\frac{df_{rel}}{n_r} - \frac{df_{nonrel}}{n_{nr}}$$

where df_{rel} is the total number of relevant documents containing the term, df_{nonrel} is that count in non-relevant documents, n_r is the number of relevant documents, and n_{nr} is the number of non-relevant. The top 50 *non-query* terms in that order were chosen and weighted using a Rocchio formula:

$$\beta \cdot \frac{1}{n_r} \sum_{rel} belief - \gamma \cdot \frac{1}{n_{nr}} \sum_{nonrel} belief$$

where $\beta = 2$, $\gamma = \frac{1}{2}$, and the belief for term t in doc d was calculated by the formula:

$$0.4 + 0.6 \cdot (0.4 \cdot \min(1, \frac{200}{maxtf_d}) + 0.6 \frac{\log(tf_{t,d} + 0.5)}{\log(maxtf_d + 1)}) \cdot \frac{\log((n_t + 0.5)/N)}{\log(N + 1)}$$

where $tf_{t,d}$ is the number of occurrences of term t in document d, $maxtf_d$ is the largest number of times any term occurs in documents d, n_t is the number of documents in the collection containing term t, and N is the total number of documents in the collection. This equation is the belief function used by INQUERY version 2.1.

2.2 Additional concepts

The same process described above was applied to find concepts based upon pairs of terms also. In this case, candidate pairs were found considering *only* the 200-word passage of the training document which best matched the original query. From those passages, 50 adjacent term pairs (ordering significant) were chosen. In addition, 50 each of word pairs within 5, 20, or 50 words (order *in*significant) were added. Selection and weighting were done exactly as described above.

In all, each query was augmented with 250 new concepts, though there was some overlap. In query 3, for example, "joint venture" appeared in every category.

The original query and additional concepts were combined in two ways. For official run INQ204, they were put together into a new query of the form:

```
#wsum( 1.0   1.0 original-query
             1.0 50-terms
             1.0 50-#1
             1.0 50-#uw5
             1.0 50-#uw20
             1.0 50-#uw50 )
```

Note that the original query, and each set of 50 new concepts all received the same significance in the query. (The next section mentions how that is changed.) The Rocchio weights for the concepts were incorporated within each group, so the weights balanced term against term, but not term against pair.

For run INQ203 all 250 additional concepts were added at the same level, meaning that their Rocchio weights were deciding the significance of terms and pairs relative to each other. In addition, the original query structure was flattened so that its components were balanced directly against the new concepts.

2.3 Weight adjustments

Inspired by the Dynamic Feedback Optimization approach of Buckley and Salton[2] (which was in turn inspired by the term selection method of City University[9]), we adjusted the chosen weights to achieve higher effectiveness in the training data, predicting that this effort will result in better effectiveness in the test documents.

The approach starts by evaluating the query on the training data. Then some concept weight is adjusted and the slightly different query is evaluated. If the effectiveness has improved, the change is retained; if the new weight hurts effectiveness, the original is restored. In both cases, the next concept weight is tried. This process repeats until no improvement is made.

The reweighting algorithm operated at the top-level of the query network only. For INQ203 that meant that each concept or query element could have its weight adjusted; for INQ204, the balance between the query and the various sets of concepts could change.

For efficiency reasons, the evaluations were done using only the 5000 documents retrieved in response to the new query (prior to reweighting). Weights were adjusted in 5 passes, with factors of 2.0, 1.5, 1.25, 1.125, and 1.0625. In each pass, each concept or query element was potentially reweighted by $w_{new} = w_{prev} \cdot \mathit{pass\_factor}$. Unlike Buckley and Salton's technique, a pass was continued until no concept's reweighting improved results. The purpose of stopping sooner than that is to avoid overfitting the training data; however, the better fitting was preferable during tests on other databases.

3 Description of the Spanish Experiments

The Spanish retrieval experiments built upon the Spanish work done in TREC-3 [1]. This year's effort incorporated InFinder [8] for query expansion, and focused on comparing INQUERY 2.0 (used in TREC-3) with the modified version of INQUERY 3.0 used for the English adhoc experiments.

3.1 Query Processing

Query processing for the Spanish topics is similar to that of the English topics and was used to generate base queries for retrieval. We do not have a Spanish part-of-speech tagger, but the text in the Spanish topics was analyzed with a simple noun phrase recognizer. Sequences of nouns and noun-adjective pairs were chosen for the #PHRASE operator.

The English stop phrase heuristics have not been translated into Spanish, but a few simple stop phrases were removed automatically. A list of the discarded stop phrases is given in Table 1.

Spanish	English translation
evidencia de	evidence of
hay	are/is there
indicaciones de	indications of
cuáles son	which are
cómo van	how is
tendrá	will it be/have
información sobre	information about

Table 1: Stop phrases removed from Spanish topics.

3.2 Noun Phrase Recognizer

Our noun phrase recognizer uses morphological rules to identify words that are likely to be nouns. Sequences of capitalized words are suggestive of proper nouns and some word endings are indicative that a word is likely to be a noun. For example, nearly all Spanish words ending in "d" are nouns [10]. Eleven types of endings were used to identify possible nouns and are given in Table 2.

Ending	Example
-dor	matador (bull fighter)
-d	verdad (truth)
-ata	corbata (tie)
-z	arroz (rice)
-[sz]mo	capitalismo (capitalism)
-miento	conocimiento (knowledge)
-[cs][ií]a	democracia (democracy)
-[cgnstx]i[oó]n	lección (lesson)
-az[oó]n	corazon (heart)
-cida	conocida (acquaintance)
-i[ae]nte	pariente (relative)

Table 2: Spanish Noun Word Endings

There is some ambiguity in the use of word endings as a heuristic to identify nouns. The word "denuncia" can be the noun "report" or can mean "he/she/you are reporting". Therefore we employ syntactic rules to reduce the ambiguity. For example, we require that a definite article precede a noun. In the case of the -cia ending, it is possible for the rule to fail to correctly classify a word. *La denuncia* can mean *the report*, but *No la denuncia* translates to *he/she/you is/are not reporting her.* The syntactic rules are heuristics and may not determine the part of speech to which a word belongs, but they increase the probability that a word is a noun.

There are several ways to modify a noun in Spanish. Qualitative adjectives generally follow nouns while quantitative adjectives precede them. Prepositions may also be used in a noun adjective phrase. For example, *trabajos de repavimentación* can be said to mean "repavement work" where "de" is a preposition meaning "of". The recognizer contains rules to recognize these phrasal constructs, in addition to recognizing nouns and proper nouns.

3.3 Query Formation

The nouns and noun phrases selected by the recognizer were used in lieu of tagged text, to identify phrasal concepts for InFinder. For the Spanish retrieval experiments, an InFinder database was created from the entire 208 MB INFOSEL collection. Table 3 shows a sample query and the top 20 phrases returned for it. Spanish queries were expanded, using the Spanish InFinder database, with the same techniques described in Section 1.2 for expanding English queries.

The final query form combined document-level and passage-level evidence, as was done for the English experiments (Section 1.3). The Spanish experiments were conducted with two different term weighting algorithms, apparently changing the relative worth of document-level and passage-level evidence. Query set SIN010, which was run against INQUERY 2.1, gave passage-level evidence twice the weight of document-level evidence. Query set SIN011, which was run against INQUERY 3.0 using the modified term weighting formulae described above, weighted them evenly.

Query:	indicaciones de las relaciones económicas y comerciales de México con los paises europeos	indications of the economic and comercial relations between Mexico and the European countries

belief	Phrase	Translation
0.486925	relaciones comerciales	commercial relations
0.485157	relaciones economicas	economic relations
0.483628	naciones europeas	european nations
0.479479	cuenca del pacífico	pacific basin
0.478936	comunidad economica europea	european economic community
0.478386	comunidad europea	european community
0.478051	jacques delors	president of the EEC
0.476194	ronda del uruguay	round of talks (GATT) in Uruguay
0.475229	comunidad europea	european community
0.474429	países europeos	european countries
0.474275	asuntos mundiales	world affairs
0.474199	barreras comerciales	commercial barrier
0.474028	grupo de río	river group
0.473967	barros valero	subsecretary of exterior relations
0.473196	lazos comerciales	commercial ties
0.472589	economico comerciales	incomplete phrase, probably was “relaciones económico comerciales”
0.472376	cancilleres	chancellor
0.472358	gary hufbauer	investigator for the Institute for International Economies
0.472293	acuerdos comerciales	comercial agreements
0.471928	viceministros	vice-ministers

Table 3: Sample Query and Top 20 InFinder Phrases

4 Description of the Collection-Merging Experiments

In the collection merging track, the document collection was divided by source and/or date into 10 smaller document collections. The goal of the collection merging track was to select one or more collections to search for a given query, to search, and then to merge the document rankings returned into a single consistent set of rankings.

Our experiments were all based on testing variations of the techniques described in [6]. Five experiments were conducted, labeled INQ206 – INQ209. Each experiment was conducted with the INQ201 query set created for the adhoc track. The collection ranking and results merging algorithms were varied, as described below.

INQ207: A previously published method [6]. The “traditional” INQUERY term weighting formulae were used.

INQ208: Similar to INQ207, except that prior to merging document rankings, a document's score was normalized based on the minimum and maximum possible scores a document could obtain in that collection for that query.

INQ205: Similar to INQ207, except that the “modified” term weighting formulae (Section 1.5) were used, which required minor modifications to the collection ranking algorithm.

INQ206: Similar to INQ208, except that the "modified" term weighting formulae (Section 1.5) were used, which required minor modifications to the collection ranking algorithm.

INQ209: Similar to INQ206, except that the merging algorithm used only the collection's score and the document rank with respect to its collection, i.e. the document's score was *not* used.

Briefly, the goal for INQ208 was to produce a more normalized document score from each collection. INQ205 and INQ206 replicated INQ207 and INQ208, but with the "modified" term weighting function discussed in Section 1.5. INQ209 investigated what could be accomplished if less information were available for merging rankings.

5 Ad-Hoc Results and Discussion

Two sets of results, INQ201 and INQ202, were evaluated in the ad-hoc document retrieval evaluation. The INQ201 results were based on completely automatic processing of the TREC topic statement into a query, automatic query expansion, use of passage-level and document-level evidence, and adjustments to INQUERY's estimation formula. INQ202 was a semi-automatic experiment in which a user was allowed to edit the INQ201 query prior to submitting it to NIST.

The official results for INQ201 and INQ202 are summarized below.

Query Type	**Average Precision**			
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ201	.51	.35	.25	.24
INQ202	.60 (+16.8%)	.44 (+25.4%)	.31 (+22.1%)	.29 (+21.0%)

Limited user modification of INQ201 produced a very significant 21.0% improvement in average precision. Most of this improvement appears to be due to 1) deleting useless query terms introduced by query processing or InFinder, and 2) grouping query terms with proximity operators. Clearly there remains room for improvement in automatic query processing techniques.

Query expansion with InFinder was much less effective than in the past. The inclusion of InFinder terms in the query yielded a 3.5% improvement in average precision (Table 4), compared to a 9.6% improvement last year.

Passage retrieval was actually detrimental. Combining passage-level and document-level evidence produced a 1.6% drop in average precision, as compared with last year's 15.7% improvement.

Combining InFinder query expansion and passage retrieval had little effect this year. There are many possible causes. The poor performance of passages alone is a likely cause. However, the new approach to using InFinder terms (i.e., not including them in the #passage operator) may also have been a factor.

The change to the estimation formula, described above, appears not to be the cause for the lower precision in this year's results. The new estimation formula provided a 4.7% increase in average precision for the basic query processing (QP, above) when compared to the old formula. Passages, InFinder, and a combination of the two all yielded slightly larger improvements with the old formula than with the new, but the improvements (2.1%, 4.5%, and 9.8%, respectively) were unimpressive. Further testing is required to isolate the cause.

6 Routing Results and Discussion

Two sets of results, INQ203 and INQ204, were evaluated in the document routing evaluation. Both sets of results were based on completely automatic processing of the TREC topic statement and relevance judgements into a query. The official evaluations are summarized below.

Recall	Precision (50 queries)						
	Query Processing (QP)	QP With InFinder (IF)		QP With Passage (PS)		QP and IF and PS (All)	
0	0.7248	0.7130	(−1.6)	0.6789	(−6.3)	0.7029	(−3.0)
10	0.5006	0.4863	(−2.9)	0.4778	(−4.6)	0.4929	(−1.5)
20	0.4074	0.4023	(−1.3)	0.3903	(−4.2)	0.3993	(−2.0)
30	0.3423	0.3419	(−0.1)	0.3316	(−3.1)	0.3484	(+1.8)
40	0.2775	0.2833	(+2.1)	0.2858	(+3.0)	0.2936	(+5.8)
50	0.2213	0.2380	(+7.5)	0.2320	(+4.8)	0.2475	(+11.8)
60	0.1608	0.1871	(+16.4)	0.1699	(+5.7)	0.1846	(+14.8)
70	0.0890	0.1126	(+26.5)	0.0853	(−4.2)	0.0933	(+4.8)
80	0.0484	0.0622	(+28.5)	0.0456	(−5.8)	0.0606	(+25.2)
90	0.0171	0.0246	(+43.9)	0.0232	(+35.7)	0.0276	(+61.4)
100	0.0012	0.0015	(+25.0)	0.0008	(−33.3)	0.0007	(−41.7)
avg	0.2330	0.2412	(+3.5)	0.2293	(−1.6)	0.2407	(+3.3)

Table 4: The effects of InFinder and passages on retrieval effectiveness.

Query Type	**Average Precision**			
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ203	.65	.59	.48	.41
INQ204	.69 (+6.2%)	.58 (−1.8%)	.46 (−3.5%)	.40 (−2.9%)

The two methods of forming queries delivered very similar results. INQ204 was significantly better (6.2%) at the 5 document cutoff, but slightly worse (1–3%) at all other cutoffs and levels of recall. These differences, while consistent, are not considered noticeable or significant.

Recall that the difference between INQ203 and INQ204 is primarily in how the dynamic reweighting was performed. In INQ203, each concept was reweighted independently. In INQ204, they were reweighted in groups, depending upon their types. The small difference in effectiveness between the two experiments, suggests that either our dynamic reweighting had little effect, or that the "right" weight for a concept depends upon its type, i.e. whether it is a #1, #uw5, etc.

The concepts added to the query includes terms, pairs of adjacent terms, and pairs of terms occurring near each other. The following table demonstrates the relative value of each class of concepts. There are a few interesting results highlighted by that table. The large percentage improvements reflect the (by now unsurprising) value of relevance feedback. Note, though, that adding pairs of words which occur "nearby"—even as far as 50 words apart—has a more pronounced impact on effectiveness than adding single terms or adjacent pairs of terms.

The effect of adding pairs and terms is not cumulative, however, although using all of them does improve effectiveness more than any of them alone. Earlier experiments (not reported here) have suggested that the `#uw20` and `#uw50` pairs are each of roughly equal value, but that combining them provides virtually no increase in effectiveness over one of them alone. We are investigating methods for choosing the "best" proximity for a pair of terms found in relevant documents.

The last row of the table is precisely INQ204, and shows the value of Dynamic Feedback Optimization, which provided roughly a 3% gain over the original weights. There is some evidence to suggest that our method overfit, but the difference appears to be a matter of 3-4%.

	5 docs	30 docs	100 docs	11-pt
Original query (Q)	0.46	0.38	0.30	0.23
Q plus 50 terms	0.61	0.50	0.39	0.33 (+46%)
Q plus 50 #1's	0.53	0.47	0.36	0.29 (+26%)
Q plus 50 #uw5's	0.59	0.50	0.38	0.31 (+34%)
Q plus 50 #uw20's	0.66	0.55	0.43	0.35 (+54%)
Q plus 50 #uw50's	0.66	0.56	0.44	0.37 (+61%)
Q plus all	0.67	0.56	0.45	0.39 (+70%)
Q plus all, reweighted	0.69	0.58	0.46	0.40 (+75%)

Table 5: Effects of expanding by different concepts

7 Spanish Results and Discussion

Two sets of results, SIN010 and SIN011, were evaluated in the Spanish track. Both sets were based on automatic processing of TREC topics SP25-SP50 into queries, automatic query expansion, and use of document-level and passage-level evidence. SIN010 was evaluated with the normal INQUERY estimation formula and SIN011 was evaluated with a modified version of that formula. The official results for both query sets are summarized below.

Query Type		**Average Precision**		
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
SIN010	0.5040	0.4000	0.2804	0.2523
SIN011	0.5040 (+0.0%)	0.3880 (−3.0%)	0.2760 (−1.6%)	0.2458 (−2.6%)

Modification of the estimation formula did not improve performance. In fact it led to a slight decrease in performance overall (1-3%), but the drop is not significant.

Experiments were run after TREC-4 to investigate the effects of each phase of query processing. Results are given in Table 6. The average precision improves with query processing by 15.8% and 4.2% over raw words alone for SIN010 and SIN011, respectively. Although performance on raw words is 5.3% higher for SIN011 than for SIN010, the modified evaluation function used for SIN011 leads to a drop in average precision (2%-5%) with respect to SIN010 for each stage of query processing.

Passage retrieval led to the best performance. It yielded a 5% improvement in average precision for both SIN010 and SIN011. The modified evaluation function of the latter yielded lower average precision (5-7%) with passage-level and document-level query modification than did the original INQUERY evaluation function.

Query expansion with InFinder resulted in a drop in performance for both query sets (6-8%) and was worst at low levels of recall. This drop in performance is probably be due to low recall of noun phrases. Noun phrases were identified using a simple noun recognizer that only identifies roughly 55% of the nouns in any document (Section 3.2). As a result, InFinder may fail to consider many good noun phrases during indexing, and may also overestimate the importance of those noun phrases it does consider. Either would cause performance to drop. Tables 7 and 8 show the effects of query modification on performance for SIN010 and SIN011 respectively.

We are currently working on building an InFinder database using a POS tagger to identify nouns. The tagger should have much higher noun recall which is expected to improve results for InFinder query modification.

Recall	Precision (25 queries)			Precision (25 queries)		
	SIN010			SIN011		
	Raw Words (RW)	RW W/O Stop Phrases (NS)	NS WITH #PHRASE op (QP)	Raw Words (RW)	RW W/O Stop Phrases (NS)	NS WITH #PHRASE op (QP)
0	77.8	80.5 (+3.5%)	86.1 (+10.7%)	79.1	81.3 (+2.9%)	80.6 (+1.9%)
10	42.0	47.0 (+11.9%)	50.2 (+19.4%)	49.4	49.0 (−0.8%)	50.7 (+2.6%)
20	35.6	39.7 (+11.5%)	41.2 (+15.6%)	39.9	40.3 (+0.9%)	40.3 (+0.9%)
30	29.6	33.0 (+11.5%)	34.0 (+14.8%)	31.1	32.2 (+3.5%)	32.3 (+3.7%)
40	24.8	28.3 (+14.0%)	28.9 (+16.3%)	26.1	26.7 (+2.4%)	27.5 (+5.2%)
50	21.3	23.4 (+9.6%)	24.7 (+15.8%)	21.1	21.9 (+3.8%)	22.3 (+5.4%)
60	16.3	18.2 (+11.5%)	19.2 (+17.2%)	16.1	16.6 (+3.0%)	17.0 (+6.1%)
70	12.4	14.1 (+14.1%)	14.3 (+15.6%)	11.9	12.5 (+4.8%)	13.2 (+10.6%)
80	8.7	9.8 (+12.9%)	10.4 (+19.6%)	8.5	8.9 (+4.9%)	10.0 (+18.3%)
90	4.8	5.3 (+11.0%)	6.4 (+34.2%)	4.9	5.4 (+9.6%)	6.5 (+33.0%)
100	0.7	0.7 (+8.0%)	2.0 (+201.9%)	0.7	0.7 (−0.9%)	0.7 (+3.5%)
avg	24.9	27.3 (+9.5%)	28.8 (+15.8%)	26.3	26.9 (+2.3%)	27.4 (+4.2%)

Table 6: The effect of query processing on the retrieval effectiveness of SIN010 and SIN011.

Recall	Precision (25 queries)						
	Query Processing (QP)	QP With InFinder (IF)		QP With Passage (PS)		QP and IF and PS (All)	
0	86.1	69.3	(−19.5%)	87.1	(+1.1%)	78.9	(−8.3%)
10	50.2	47.3	(−5.7%)	54.6	(+8.8%)	49.2	(−1.9%)
20	41.2	39.4	(−4.3%)	42.7	(+3.8%)	42.2	(+2.6%)
30	34.0	34.6	(+1.8%)	35.3	(+3.8%)	35.1	(+3.2%)
40	28.9	28.8	(−0.4%)	31.5	(+9.1%)	30.6	(+6.0%)
50	24.7	23.4	(−5.3%)	26.2	(+5.9%)	25.4	(+3.0%)
60	19.2	18.8	(−1.7%)	19.4	(+1.5%)	21.0	(+9.5%)
70	14.3	15.6	(+9.4%)	14.9	(+4.3%)	15.7	(+9.8%)
80	10.4	11.2	(+7.9%)	11.1	(+7.4%)	11.6	(+11.8%)
90	6.4	7.9	(+22.8%)	6.9	(+8.2%)	7.6	(+19.5%)
100	2.0	2.4	(+20.7%)	2.1	(+2.5%)	2.3	(+12.3%)
avg	28.8	27.2	(−5.9%)	30.2	(+4.6%)	29.1	(+0.8%)

Table 7: The effects of passages and InFinder on the retrieval effectiveness of SIN010.

Recall	Precision (25 queries)						
	Query Processing (QP)	QP With InFinder (IF)		QP With Passage (PS)		QP and IF and PS (All)	
0	80.6	66.3	(−17.7%)	77.0	(−4.5%)	71.5	(−11.3%)
10	50.7	46.3	(−8.7%)	52.8	(+4.0%)	49.7	(−2.0%)
20	40.3	37.9	(−6.0%)	41.9	(+4.0%)	41.0	(+1.7%)
30	32.3	30.6	(−5.1%)	34.2	(+5.9%)	32.5	(+0.6%)
40	27.5	25.9	(−5.8%)	29.6	(+7.8%)	28.3	(+2.8%)
50	22.3	21.4	(−4.1%)	25.4	(+14.2%)	24.1	(+8.0%)
60	17.0	16.4	(−3.6%)	19.8	(+16.1%)	18.9	(+10.7%)
70	13.2	13.5	(+2.5%)	15.2	(+15.4%)	15.1	(+14.6%)
80	10.0	10.4	(+4.1%)	11.4	(+13.4%)	11.9	(+19.0%)
90	6.5	7.7	(+18.2%)	7.2	(+11.1%)	7.6	(+17.0%)
100	0.7	1.4	(+107.7%)	1.4	(+99.5%)	1.5	(+119.3%)
avg	27.4	25.3	(−7.7%)	28.7	(+4.9%)	27.5	(+0.3%)

Table 8: The effects of passages and InFinder on the retrieval effectiveness of SIN011.

8 Collection Merging Results and Discussion

Five sets of results, INQ205 – INQ209, were evaluated in the collection-merging document retrieval evaluation. After the results were submitted, we discovered that we inadvertently submitted the INQ206 results twice, as INQ206, and as INQ209. In the tables below, INQ209c is an unofficial run showing what we intended to submit (the "corrected" run). The official INQ209 results are not shown, because they are identical to INQ206.

The INQ201 adhoc run was treated as the baseline run. It gives the results of treating the subcollections as a single collection, the traditional IR paradigm. The official evaluations, and one unofficial run, are summarized below.

Query Type	**Average Precision**			
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ201	.51	.35	.25	.24
INQ207	.49 (−4.8%)	.35 (−2.1%)	.24 (−3.8%)	.21 (−13.4%)
INQ208	.48 (−6.4%)	.35 (−0.8%)	.24 (−3.5%)	.21 (−13.4%)
INQ205	.50 (−2.4%)	.35 (−2.1%)	.24 (−6.2%)	.21 (−14.7%)
INQ206	.51 (−0.8%)	.35 (−1.4%)	.24 (−6.1%)	.21 (−15.0%)
INQ209c	.41 (−19.2%)	.33 (−6.9%)	.24 (−5.7%)	.18 (−26.8%)

The data support several conclusions.

- The "modified" term weighting functions were more effective than the "traditional" term weighting functions at very low recall (1-15 documents). After that, they were generally worse.

- Normalizing the document score based on the maximum and minimum scores that the query could generate in the collection was marginally useful with "traditional" term weighting, and more consistently useful with "modified" term weighting. However, the effects were small in both cases.

- Merging rankings based on document rank was significantly worse than merging based on document score. This is not a surprising result; indeed, the result is consistent with what others have found [13].

In general, we were pleased with the results. The best methods (INQ205 and INQ206) are not significantly worse than the single collection results until nearly 100 documents are retrieved. Even the worst method (INQ209) is probably adequate for interactive use.

9 Summary and Conclusions

We continue to believe in highly structured queries, sophisticated query processing, and in combining multiple sources of evidence. However, the adhoc experiments showed that we still have much to learn about these subjects. Query processing and structured queries were generally useful, but query expansion and passage-level evidence were not.

The routing track occupied more of our attention this year, which appears to have paid off. However, the relative similarity of our routing runs raises questions about the new dynamic reweighting algorithm. Much of the routing effectiveness may be due more to traditional factors, e.g., better term selection or using a wide range of proximity operators, than to the reweighting algorithm.

The Spanish track is a useful part of our research on foreign languages and IR. Last year we were happy to have Spanish results. This year, we wanted the same high level of effectiveness that we see in English. We expected the Spanish InFinder to help significantly, but it did not, suggesting that more research is required.

The collection merging experiments were a first step down the path to effective networked retrieval systems. The first step was surprisingly good, given the brevity and difficulty of the adhoc queries. However, there were so few collections that no strong conclusions can be drawn. We would like to see at least 100 (presumably smaller) collections in future evaluations.

Acknowledgements

We thank John Broglio, Stephen Harding, and Dan Nachbar for their assistance in the work described here. This paper is based on research supported by NRaD Contract Number N66001-94-D-6054 and the NSF Center for Intelligent Information Retrieval at the University of Massachusetts, Amherst.

References

[1] J. Broglio, W. B. Croft, J. Callan, and D. Nachbar. Document retrieval and routing using the INQUERY system. In D. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 29–38. National Institute of Standards and Technology Special Publication 500-225, 1995.

[2] C. Buckley and G. Salton. Optimization of relevance feedback weights. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–357, Seattle, 1995. Association for Computing Machinery.

[3] J. P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, Ireland, 1994. Association for Computing Machinery.

[4] J. P. Callan, W. B. Croft, and J. Broglio. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31(3):327–343, 1995.

[5] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.

[6] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, Seattle, 1995. Association for Computing Machinery.

[7] W. B. Croft, J. Callan, and J. Broglio. TREC-2 routing and ad-hoc retrieval evaluation using the INQUERY system. In D. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*. National Institute of Standards and Technology Special Publication 500-215, 1994.

[8] Y. Jing and W. B. Croft. An association thesaurus for information retrieval. In *RIAO 94 Conference Proceedings*, pages 146–160, New York, October 1994.

[9] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, (in press). National Institute of Standards and Technology, Special Publication 500-225.

[10] Richard V. Teschler. *Spanish Orthography, Morphology and Syntax for Bilingual Educators.* University Press of America, Inc., 1985.

[11] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In Jean-Luc Vidick, editor, *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 1–24. ACM, September 1990.

[12] Howard R. Turtle and W. Bruce Croft. Efficient probabilistic inference for text retrieval. In *RIAO 3 Conference Proceedings*, pages 644–661, Barcelona, Spain, April 1991.

[13] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The collection fusion problem. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, (in press). National Institute of Standards and Technology, Special Publication 500-225.

[14] J. Xu, J. Broglio, and W. B. Croft. The design and implementation of a part of speech tagger for english. Technical Report IR-52, Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, 1994.