Xerox TREC-5 Site Report: Routing, Filtering, NLP, and Spanish Tracks

David A. Hull* Gregory Grefenstette* B. Maximilian Schulze*
Eric Gaussier* Hinrich Schütze† Jan O. Pedersen‡

*Rank Xerox Research Center
{hull,grefen,schulze,gaussier}@grenoble.rxrc.xerox.com

†Xerox PARC
schuetze@parc.xerox.com

‡Verity Corporation
jpederse@verity.com

1 Introduction

Xerox participated in TREC 5 through experiments carried out separately and conjointly at the Rank Xerox Research Centre (RXRC) in Grenoble and the Xerox Palo Alto Research Center[1]. The remainder of this report is divided into three sections. The first section describes our work on routing and filtering (Hull, Schütze, and Pedersen), the second section covers the NLP track (Grefenstette, Schulze, and Gaussier), and the final section addresses the Spanish track (Grefenstette, Schulze, and Hull).

2 Routing and Filtering

For the routing and filtering tasks, Xerox continues its work on using statistical learning algorithms to develop optimal query profiles. The goal is to take advantage of the large number of evaluated documents available in the TREC context to build a more accurate model of each topic. There are a number of challenges in applying traditional learning algorithms to the routing/filtering problem, including a large and poorly defined feature set and a very low density of positive training examples. In this section, we describe the Xerox approach to document routing and filtering, concentrating on the differences from our experiments last year. Readers are encouraged to refer to the Xerox TREC-4 site report for more details on our work last year [11].

For the moment, Xerox does not have a specially designed text routing/filtering system. Instead, we rely on a traditional information retrieval system to simulate the text filtering problem. While this approach does not allow us to measure the time/space efficiency of our algorithms in a real-time system, it does provide accurate estimates of system performance according to precision-recall and utility measures. Xerox uses the Text Database System (TDB) developed at Xerox PARC [4] for parsing, tokenizing, stemming, and stop word removal. Indexed *terms* include single words as well as adjacent word pairs (not including stop words). Unlike last year, no document segmentation was performed for TREC-5.

[1] Jan Pedersen was formerly at the Xerox Palo Alto Research Center.

2.1 Preliminary Data Reduction

Xerox uses a three step filtering process. First, an expanded query is created by taking the vector sum of the original query and all relevant documents in the training set. Non-relevant documents are ignored. Next, all available documents are ranked according to their similarity to the expanded query and the top 2000 documents are selected to define the *local region* for that query. All unevaluated documents in the local region are assumed to be not relevant. This preliminary filtering step serves several functions; the local region has a higher density of relevant documents and the non-relevant documents selected for training are those which are most difficult to distinguish from relevant documents, which should increase the discriminatory power of the system. Note that while we use the full document collection in these experiments, there is no reason why the local region could not be defined only in terms of the evaluated documents for a particular query. The only advantage of our approach is that it makes it easier to determine a threshold which defines when test documents should belong to the local region.

In the second step of the filtering process, an optimal feature set is determined for each query by selecting among the large number of potential terms available in the local region. We begin by ordering all terms according to their score based on the binomial likelihood ratio test statistic [5], with relevant and non-relevant documents representing the two classes. This statistic assigns a high score to documents whose probability of occurence differs significantly among the two classes. We then filter out all terms with scores less than 5.0 (which corresponds to a p-value of 0.025 according to the chi-square test statistic), those which occur in fewer than five relevant documents, and those which assign a higher probability of occurrence to the non-relevant class (negative predictors). In preliminary experiments, we found no advantage to using negative predictors in our classifiers. This may be because the domain of non-relevant documents is so broad that it is dangerous to choose predictors based only on the local region. Of the remaining terms, the top 200 are selected to serve as the feature set for the statistical classifiers, and a reduced profile is created for each document in the local region. The binomial likelihood ratio test (LRT) is different from the chi-squared test that we used at TREC-4, even though it has the same reference distribution (chi-square). It seems that the LRT makes more realistic estimates of the importance of low frequency terms.

2.2 Statistical Learning of the Logistic Model

In the third and final step, the reduced profiles are sent to a statistical learning algorithm. In previous experiments, we have examined a number of different classification techniques for the routing problem, including logistic regression, nearest neighbors, linear discriminant analysis, and neural networks [14, 11]. At TREC-4, we found that a combination of techniques achieved the highest level of performance, while a single-layer neural network based on the logistic model was the best individual classifier. For TREC-5, we decided to concentrate on the logistic model and compare several different strategies for reducing overfitting,[2] which is probably the biggest barrier to good performance using this model. The logistic model produces explicit estimates of probability of relevance by applying the logistic transformation $g(\pi) = \log(\pi/(1-\pi))$ to the output of a linear classifier. More details on logistic regression and the logistic model can be found in [8, 3].

While the two preliminary steps have substantially reduced the document and term set, previous experiments have shown that logistic regression still leads to overfitting [14]. One possible approach is to use an early stopping rule. Fitting the logistic model is an iterative process where parameter estimates gradually converge to the optimal solution. Unfortunately, the optimal solution for the training set is generally not an optimal model of the underlying population. We can come up with a better solution by estimating performance during convergence using cross-validation and stopping

[2] Overfitting is defined as impoving performance on the training data while reducing performance on the test data. It is the result of techniques with high descriptive power which fail to generalize sufficiently from training data that is only an approximate model of the true underlying distribution.

when performance on the validation set begins to decrease. We have had good success in the past using an early stopping rule in conjunction with a neural network to attack the overfitting problem, but there are other possible solutions. For TREC-5, we examine three other techniques to reduce overfitting: regularization, variable selection, and non-negative logistic regression.

We can attack one obvious source of overfitting by taking advantage of how the feature set for the training data was selected. Since all terms in the reduced set are chosen to have a large positive association with relevance, it is fair to assume that any negative parameter estimates for these terms are not accurate. In an ideal world of term independence, this would never happen, but in practice, strong term correlation patterns may lead to negative estimates for certain parameters. Therefore, simply forcing all parameter estimates to be non-negative should lead to a more accurate model. We define this technique as non-negative logistic regression, and implement it in a simple fashion. After every five iterations of logistic regression, all terms with negative parameter estimates are removed from the model (which is equivalent to setting the estimate to zero). This process is continued until there are no negative parameter estimates. There are more principled ways to solve this problem based on the use of constrained optimization, but our heuristic approach seems to work well enough in practice. This simple step alone significantly improves the performance of the logistic model.

Regularization is a well-known statistical technique for improving the estimation of an overspecified model by biasing the estimated parameters towards a fixed value. In our case, we choose to use a simple probabilistic independence model as the fixed model. This approach can also be viewed as method combination, for it essentially boils down to taking a weighted combination of the fixed and fitted models. At TREC-5, we used this technique in conjunction with non-negative logistic regression.

We already applied variable selection in stage two of our algorithm to reduce the full feature set to roughly 200 selected terms that have a strong positive association with relevance. However, the number of selected terms is still large compared to the number of positive training examples, so further selection could be valuable to produce a more general model. One possible approach would be to use the same technique again but select an even smaller number of terms. However, the original algorithm has the key flaw that terms are selected independently of one another. A better strategy would select new terms that are the most helpful given the terms that are already in the model. This can be accomplished by using a backward stepwise variable selection technique in conjunction with logistic regression. Backward means that the model starts with all terms and terms are gradually removed in inverse order of their contribution to the model. Stepwise means the variables are removed one (or a few) at a time and the value of each remaining term is reassessed at each step in the process. We use a standard approach to model selection which evaluates each variable subset according to the AIC (Akaike Information Criterion) to decide which variables to remove and when to terminate the process. This approach bears strong similarity to the one adopted by City University at TREC-3 and TREC-4 [6], except that they use a forward selection process and optimize performance with respect to average precision directly.

Unfortunately, we could not fully test all the presented techniques within the context of TREC-5. Our three submitted runs all use the same preliminary data reduction steps described in the previous section, differing only in the learning algorithm. The first run applies a simple probabilistic independence model to the selected terms and is designed to serve as a baseline. No learning, other than the probabilistic term weights, is incorporated into this run. This model is not optimal because the terms are far from independent. Our second run is a linear combination of 0.7 times the probabistic run plus 0.3 times the estimate from non-negative logistic regression. The parameter 0.7 was selected because it optimized performance over a previously analyzed query set and was left constant for all queries. The final run used the variable selection technique described in the previous paragraph.

2.3 Summary of the Xerox Algorithm

We provide a simple summary below of three stage training and testing process for our routing/filtering algorithm. The training process:

- Determine 2000 nearest documents to expanded query (local region).
- Select up to 200 important terms based on LRT statistic.
- Apply variants of logistic regression to determine linear classification rule for the local region.

The testing process:

- Select documents that exceed threshold for local region.
- Build reduced profile based on important terms.
- Score document according to classification rule.
- Routing: add document to priority queue of test documents.
- Filtering: accept or reject document based on filtering threshold.

The reader will note that we have made no distinction up to this point between the routing and filtering task. This is because our algorithms work identically in both cases, since logistic regression allows us to estimate the probability of relevance for each incoming document, and a simple transformation of the utility function allows us to describe the filtering threshold as a probability. The only difference comes after the document is scored. In routing, the document is added to a priority queue which ranks it relative to the other test documents, while in filtering, it is accepted or rejected by comparing its score to the probability threshold(s).

2.4 TREC-5 results

Measure	Prob	RNN-Logit	BStep-Logit
avg.prc	0.171	0.177	0.175

Table 1: Average uninterpolated precision for the 3 Xerox routing runs: probabilistic term weighting (Prob), regularized non-negative logistic regression (RNN-Logit), and backward stepwise logistic regression (BStep-Logit).

The Xerox TREC-5 routing results are presented in Table 1. Obviously, these results do not represent a triumph for our system, given that they are less than half the scores we received with similar runs last year. After analysis of the results, we found that the preliminary filtering step (selecting the local region via query expansion) failed in a dramatic fashion. To give you some idea of the magnitude of the problem, consider that only 49.6% of the relevant documents appeared in the top 2000 test documents (compared to 88.8% for TREC-2 and 94.2% for TREC-3) when computing similarity with respect to the expanded query.

Our query expansion technique is very simple, merely take the vector mean of the query and all relevant documents, so it is surprising to see such a difference from previous years. Further analysis reveals that the stop list (!) is responsible for the problem. We reduced the stop list from hundreds down to only a few entries in order to catch some important pairs that we had been losing in previous years. This added a large number of spurious terms and pairs which turned out to have a dramatic impact on expansion performance. Conventional wisdom in IR is that the selection of terms for removal has only a small impact on system performance, and this has always been the case in our previous experiments, so we thought we could manipulate the stop list with impunity. Obviously, when indexing term pairs in the context of the routing problem, one has to be a lot more careful. After running the query expansion with a more substantial stop list, we found the coverage on the test set jumped to about 78% (and could move higher with more careful selection). We were not able to run the rest of the routing experiments in time for this paper.

Given the low coverage of relevant documents in the test set, we are reluctant to make any conclusions about the relative performance of the learning algorithms from the scores in Table 1. While average uninterpolated precision depends heavily on good recall, the same is not true for the utility measures used in filtering this year. Therefore, our filtering performance is much more competitive with other TREC-5 groups, although the system still probably suffers to some extent from the low recall problem.

3 Natural Language Processing and IR

For the Natural Language Processing (NLP) task, Xerox attempted to test whether linguistically motivated light parsing would improve retrieval results over the classic IR approximation to noun phrase recognition. This can be seen as a large-scale reproduction of Joel Fagan's experiments with CACM and CISI [7], in which he compared using positionally derived phrases to syntactically derived noun phrases as additional indexing terms.

3.1 Approximations to NLP in IR

In order to capture linguistically similar documents, Information Retrieval has always used approximations to some levels of natural language processing: tokenisation, for example, involves general knowledge of how words are formed; stopwords involves syntactic knowledge of what words act as function words; and stemming involves morphological knowledge about word formation. Recognition of one of the most common syntactic structures in English, the compound noun phrase, has long been approximated by considering all pairs of contiguous words between stopwords as additional indexing terms[12].

These approximations have the advantage of being relatively simple to implement, requiring little data, and resulting in fast treatment. Though tokenisation involves keeping state information while reading non-space characters, stopwords and stemming is essentially table lookup, and binary phrases involve only contiguous word pairs.

Natural Language Processing, on the other hand, involves larger structures than those considered in IR, and additional time is required to access those structure during text processing. In addition to more simple word tokenisation, the input text must also tokenized into sentences for most steps in NLP. The context of the entire sentence is used in part-of-speech disambiguation, a necessary step for proper lemmatisation[3] and for subsequent parsing. Within each sentence, more complicated structures such as complex noun phrases and verb phrases are recognized by NLP and relations within and between these phrases are extracted as additional index terms for information retrieval. This structuring of the sentence involves more work than in the classic Information Retrieval approach, but it allows the recognition of syntactic relations between non-contiguous words, such as between verbs and nouns separated by intervening words. We wanted to test whether proper recognition of syntactic word dependence would eliminate noise caused by chance co-occurrence of words and reduce silence by recognising non-contiguous syntactically related word pairs.

3.2 Non-NLP runs

In order to see if NLP provides any substantial gain over non-NLP methods, Xerox created classic Smart runs with and without phrases as baselines. These are the non-NLP runs described below.

[3]Lemmatisation is the reduction of inflected words to their root forms, like *thought* as a verb to *think*. Though rarely the case in English, in most other languages, verbal forms and nominal forms lemmatize differently.

Original Text (**non-stopwords in boldface**):
    Where and for what **purpose** is **scuba diving** done **professionally**?
Stemmed indexed terms:
    purpos scub dive profess
Additional phrasal indexing terms (found between stopwords):
    dive_scub

Figure 1: Classic IR approximation to phrasal indexing. The new term is composed of the stemmed versions of the contiguous now-stop words. In addition, the order of the elements is normalized into alphabetical order [12].

The NLP runs use a finite-state light parser[10] to extract syntactic word dependencies. The NLP runs using these extracted pairs as additional index terms are compared against the same baseline.

3.2.1 Phrases in Classic Information Retrieval

The classic phrase[4] constructing method used in information retrieval systems like SMART is to create a new indexing term from contiguous pairs of non-stopwords[1]. See Figure 1.

SMART allows these new terms to be added into a distinct indexed field, the weighting of whose elements depends on the original document length. In this way, adding in extra terms does not make the document seem longer to the weighting procedure[5], and thus the original uniterms (like `purpos, scub, dive` and `profess` in Figure 1) have the same weights as the word without the additional phrase.

The addition of these simply-derived phrases has consistently shown to improve retrieval[2]. This improvement has been reproduced in our implementation of this classical strategy.

As an efficiency issue with regards to space, the inclusion of phrases in the indexes greatly increases the number of index terms. Though much rarer, there are many more phrases than individual stems. Cornell [2] reports that they use as index items only those phrases that appeared at least 25 times in the initial TREC1 document set. Here for the Xerox nonNLP and NLP experiments, we retained any phrase appearing in the query as an index term. Since the weights of the phrases do not depend on the existence of any other phrases[6], this allows us to limit the number of terms indexed without altering the theoretical results.

We will now describe the runs that we have performed for the NLP track of TREC 5. A recapitulative table is given in Figure 2. The table indicates results for TREC topics 251–300, run over Disk 2 data, 250 MB of Wall Street Journal text. Seventeen topics had 4 or fewer relevant documents in this set. A change in the relevance position of one or two documents in such queries can significantly alter the average precision of the entire run. For example, if a query has only one relevant document, if that document moves from second place in the ranked list of returned documents to the first place, then there is a 50% gain in precision, which even when averaged over 50 queries can still modify the total precision of a run by 1%. In order to eliminate the effect of these queries, Figure 2 also indicates results excluding these topics under the heading "33 topics."

[4] Here *phrase* corresponds to a an index term derived from more than one word.
[5] In the normal case, the weight of each term in a document decreases as the number of index terms increases.
[6] Their weights only depend on their frequency throughout the database, and the length in words of the original documents.

		All 45 Topics				**33 Topics (> 4 rel docs)**			
RUNS		Avg Prec		R Prec		Avg Prec		R Prec	
NonNLP									
stemming	xerox_nlp1	0.200	—	0.206	—	0.189	—	0.228	—
+contig pair	xerox_nlp2	0.215	7%	0.217	5%	0.196	4%	0.228	0%
+wind pair	window	0.187	-6%	0.219	6%	0.158	-16%	0.216	-5%
NLP									
nlp	mwe	0.231	15%	0.248	20%	0.202	7%	0.240	5%
contig&nlp	xerox_nlp4	0.228	14%	0.246	19%	0.198	5%	0.237	4%
Manual Correction									
nlp	xerox_nlp5	0.232	16%	0.249	21%	0.204	8%	0.241	6%
contig&nlp	xerox_nlp6	0.230	15%	0.247	20%	0.200	6%	0.239	5%
Short Queries									
contig&nlp	xerox_nlp3	0.140	-30%	0.154	-25%	0.147	-22%	0.172	-25%

Figure 2: Comparative table of results Xerox's TREC 5 NLP track runs. **Stemming** is the baseline, a classic SMART run. The phrasal baseline is given by **+contig pairs** which, in addition to stems, includes additional index terms derived from sorted stemmed pairs of contiguous non stopwords. **+wind pair** includes pairs derived from a window of 10 non stopwords around each stem; **nlp** includes stems and sorted stemmed pairs of syntactically related words found by a light parser; **contig&nlp** includes the union of pairs derived from the light parser and from SMART contiguous pairs. The two runs under the heading *Manual Correction* underwent manual elimination of unwanted pairs form the queries. *Short Queries* is an automatic run using pairs derived from the parser and the SMART contiguous non stopword method, but only using the description field of the query. All other runs also use the "Narrative" field. The results given under the column labeled **33 Topics** give a truer picture, since topics with less than 5 relevant documents, whose results skew performance, are not considered.

3.2.2 Baselines: xerox_nlp1 and xerox_nlp2

Our first non-NLP baseline is a classic SMART run, with stemming turned on. Documents were indexed using the "lnc" and queries using "ltc" weighting schemes[7].

The run xerox_nlp2 is an implementation of SMART phrases.

Since adding in new phrases means that words in the phrases are accounted for twice, once in the term and once as individual words, a common practice is to downweight the extra phrases. After a period of testing on former TREC queries, we found that multiplying the additional terms by 0.5 gave the best results.

We were not interested in extensively modifying SMART code, so we decided to build additional fields for each query and document that would contain phrasal indexes derived from other indexed fields. SMART provides an indexing method in which terms from such a field do not lengthen the document for weighting purposes, so the number of extra unused terms added does not affect retrieval performance of the individual words. Still, adding in new phrases means that words in the phrases are accounted for twice, once in the term and once as individual words. The common practice is to downweight the extra phrases. After a period of testing on former TREC queries, we found that multiplying the additional terms by 0.5 gave the best results. To reduce the size of these fields, we only retained, as document phrases, phrases that were also derivable from some query among 251 to 300. Although, this is not the approach that would be taken in a commercial system, it provides the same theoretical results as if all possible contiguous word pairs were indexed for each document.

3.2.3 Non-contiguous non-NLP pairs: window

Another run (labeled "+wind pair" in Figure 2, but not submitted to TREC evaluators) extends the idea of creating additional indexing terms from pairs of non stopwords. Instead of just indexing contiguous pairs, any pair within a window of 10 non-stop words was created as an indexing term. This run was performed in order to test the idea that any gain from syntactic recognition of non-contiguous pairs could be approximated by simply widening the window from which pairs were drawn. In order to be complete, we should have run this test with window lengths of 2 to ten.

3.2.4 Automatic NLP runs: mwe and xerox_nlp4

The "mwe" run[8] includes index terms made from pairs of words sharing some syntactic dependency in indexed text. Using a light parser[9, 10], a shallow syntactic analysis was made of the queries and documents, and any pair of words found to be in one of the following relations were extracted: subject–verb, verb–direct object, verb–adjunct, noun modifying noun, adjective modifying noun, adverb modifying verb. Pairs involving a SMART stopword were eliminated. The words in remaining pairs were stemmed using SMART's stemming algorithm, and alphabetically sorted pairs were constructed, as shown in Figure 3.

The "mwe" preprocessing[9] that produces an extra field in each document before classical SMART indexing consists of the following steps: (i) mapping SGML-coded characters to IS0, (ii) tokenisation (iii) morphological analysis (iv) part-of-speech disambiguation (v) verb group recognition (vi) noun group recognition (vii) marking verb group aspect and head (viii) marking noun group heads

[7] Actually, instead of using log of document frequency, we prefer using square root of document frequency. This requires the modification of one line of SMART code.

[8] This run was not submitted for evaluation, since the strategy used in xerox_nlp4 had performed better on our test corpus.

[9] This entire process (a sequence of binaries, shells, and *gawk* programs) performs at about 360 words/second or 2.4 Kbytes/sec on a SPARC Ultra 1 with 64 Mbytes of memory. Preparing the whole NLP track (250MB), before SMART indexing, takes about 36 hours. 2 additional hours were needed by another *perl* program to restem the extracted pairs according the SMART stemmer, and to eliminate pairs not appearing in the queries. Indexing the preprocessed data on a SPARC Ultra 2 with 512 Mbytes of memory takes about 15 minutes.

Original Text:

Where and for what purpose is scuba diving done professionally?

Stemmed indexed terms:

purpos scub dive profess

Parser-derived relations:

do professionally
dive professionally
scuba do
scuba dive

Additional phrasal indexing terms (after stemming and stopword removal):

dive_scub dive_profess

Figure 3: Light parsing version of indexing pairs. In addition to stemmed indexed single-word terms, the same as SMART classically produces, additional index terms are derived from the parser output. The parser produces binary relations between lemmas. If the pair contains a stop word, it is eliminated. Otherwise, the lemmas are stemmed according to SMART's stemmer, sort in alphabetic order and joined by an underscore to produce the new index term.

(ix) extraction of syntactic dependency relations (x) elimination of syntactic dependency relations involving the word 'document' (xi) reformatting of indexing pairs (xii) stopword elimination.

The "xerox_nlp4" run combined the pairs extracted by the NLP method outlined in the last paragraph with those extracted by the classical SMART method. For example, from the phrase "to combat alien smuggling", the SMART phrase builder will generate "alien_combat" and "alien_smuggl" while the light parser will only return "combat_smuggl" and "alien_smuggl". The union provided by both methods produces all three index-terms. This run was also performed to see if errors in the parses, notably silence, could be stopgapped by including the simple contiguous pairs of classical SMART phrases[10]

3.2.5 Manual NLP runs: xerox_nlp5 and xerox_nlp6

The additional indexing phrases produced for the topics from runs "mwe" and "xerox_nlp4" were manually edited to remove pairs derived from incorrect syntactic analyses and containing query metalanguage to produce the topics used in runs "xerox_nlp5" and "xerox_nlp6." Only removal was performed, there were no additions or corrections.

For example, in topic 251, the description reads: "Documents will report the exportation of some part of U.S. Industry to another country." The light parser extracted "countr_industr" as a pair. The human filterer judged this to be an incorrect parse and eliminated the phrase.

3.2.6 Discussion of Results

The results given in Figure 2 show an improvement in Average Precision and in R-Precision using all the multiword generating techniques, except the "window" which includes all possible non stopword combination generatable over 10 word windows.

The automatic nlp run "mwe" performs twice as well as the classic SMART contiguous pair technique[11]. One might argue that the improvement comes from the fact that the parser, by creating

[10] For example, the query title "Combating Alien Smuggling" yielded no dependencies from the version of the parser used in TREC5. "Alien_smuggl" was returned from the Description section of the topic which contained "to stop the smuggling of aliens" but the index term "combat_smuggl" was not found for the run "mwe" for query 252.

[11] Albeit at a cost of 36 hours processing time.

new terms from syntactically dependent words, creates additional local non-contiguous terms. The "window" run, while not invalidating this hypothesis, does weaken it somewhat since the window includes all the pairs found by the contiguous pairs and the parsing technique, and then some, yet it performs worse than simple stems. This shows that adding in all possible local pairs is not good, and suggests that light parser making some of the right choices about what should be combined.

The Figure 2 presents results averaged over all 45 topics which had 1 or more relevant documents, showing a significant improvement, and then over the 33 topics for which at least 5 relevant documents exist in the collection. The improvements found throughout the runs in this latter set are more modest. We decided to present these results after examining the result of Topic 253 "Cryonic suspension services." According to assessors, this document only has one relevant document in the WSJ set used for the NLP track. This relevant document is ranked in second position by the runs using classic SMART and classic SMART with phrases, but becomes the first document in the NLP runs. This gives a 100% improvement for this query, which, averaged over the 45 topics, adds a 0.011 absolute improvement to the Average Precision, which would account in itself for 5% improvement in the whole run (given a 0.200 baseline as in Figure 2). Since the NLP runs show a 15% improvement in Average Precision for all 50 topics, this means that 33% of the improvement comes from the simple permutation of one document in one query. This is clearly unreasonable to have such a small change play such a large part in evaluation. To negate such wild effects, then, we removed the 17 topics that had 4 or fewer relevant documents from consideration, as shown in the column labeled "33 Topics (>4 relevant)" in Figure 2.

Analysing the reasons for this improvement means going into the documents and extracting what matched between query terms and competing runs. We developed an analysis tool that allows us to compare the index matches between a given query and a given document over a number of runs. For topic number 253 and the relevant document numbered 19058 by SMART, comparing classic SMART phrase indexing (run "xerox_nlp2" in Figure 2) and light parsing (run "mwe") gives us the following chart, in which the runs are labeled 0 and 1, and in which the column "word" gives the index terms that matched in query and document with their weights (a dash in the weights means that the term was not present in that run):

```
        Term Table for Query 253 and Document 19058
0 ... xerox_nlp2
1 ... mwe
                word        0       1
--------------------------------------
                body    0.00717 0.00717
        body_preserv       -    0.01992
                cure    0.01004 0.01004
               freez    0.04717 0.04717
             industr    0.00361 0.00361
                long    0.00413 0.00413
            nitrogen    0.01867 0.01867
             preserv    0.01168 0.01168
              report    0.00262 0.00262
              servic    0.00295 0.00295
                stor    0.00503 0.00503
```

From this table we see that the improvement in the runs come from the addition of the index term "body_preserv" found by the light parser in both topic 253 and in document 19058 (WSJ900917-0124). In that document, which is really about freezing CD's, we find the sentence:

> Cryogenics is best known as the method used to preserve the bodies of dead people until cures for their illnesses are discovered.

The light parser correctly extracts the syntactic relation between "preserve" and "bodies" which stems to "body_preserv." But the topic 253 contains only the line:

> Cryonics suspension is the practice of quick-freezing a human body immediately upon death and its preservation in a nitrogen environment for possible resuscitation in the event that a cure is found for the cause of death.

The parser here has extracted "body_preserv" because it has incorrectly recognized the scope of the preposition "upon" and had analyzed the sentence as if it read "a human body ... upon preservation" which although semantically correct here is only recognized by chance since the parser does no analysis of pronouns and can not reattach "its" to "body."

Topic 253 is not included in the "33 Topic" column of Figure 2 since it only has one relevant document. A better explanation of the NLP runs modest improvements in this column comes from Topic 283 "China trade policy." For this query, document 36340(WSJ910617-0166) jumps from 21st position to 11th because it contains two terms that the simple continguous word phrase-finding method finds neither "chin_polic" nor "chin_trad":

```
       Term Table for Query 33 and Document 36340
0 ... xerox_nlp2
1 ... mwe
               word         0       1
---------------------------------------
               chin    0.07687 0.07687
         chin_polic       -    0.01337
          chin_trad       -    0.02414
            comp_us    0.01112 0.00988
             compan    0.00230 0.00230
             consum    0.01175 0.01175
               cost    0.00285 0.00285
             effect    0.00502 0.00502
            foreign    0.01202 0.01202
              polic    0.00848 0.00848
             produc    0.00788 0.00788
               trad    0.00991 0.00991
                u.s    0.01893 0.01893
              union    0.00430 0.00430
               work    0.00445 0.00445
```

The light parser can extract "china_trade" from the document's "...China's main trade" while the contiguous pairs method will not.

3.3 Conclusion of NLP track

The principal approximation to linguistic knowledge used by classic English information retrieval is stemming. The stemming algorithms developed for English over the past thirty years have become very close to capturing much of inflectional and derivational morphology of English. Using stemmers provides a significant increase in information retrieval performance. A minor improvement is added by the approximation of phrases, by joining together contiguous non-stop words. From our results, it would seem that light parsing can slightly improve this phrasal indexing, but at a cost, of course, of a much heavier linguistic apparatus, and lengthened time of preprocessing. Nonetheless, we are optimistic that this approach will prove useful in the long run for a number of reasons: (1) For non-English languages more work is being done on linguistic analysis than on information retrieval. This implies that morphological analysers for these languages may largely outperform simple stemming routines that have undergone the same maturation time as English stemming routines. (2) As machine become more powerful, the preprocessing times will continue to fall, making more complicated text analysis more economically feasible. (3) Robust parsing is progressing in the variety and accuracy of structures recognized. If contiguous structures improve retrieval, the recurring but separated structures should also, as Figure 2 supports.

4 Spanish Track

For the Spanish track, Xerox continues to test its linguistic analysis tools in the context of information retrieval. At TREC-4, we examined the performance of a linguistically motivated lemmatiser for Spanish stemming and found that it consistently improved performance (and is slightly more effective than the English version). For TREC-5, we use a Spanish part-of-speech tagger and pattern matcher to identify and index noun pairs as a linguistic alternative to simple adjacent pairs that are often used in information retrieval. We also conduct our first experiments on English/Spanish cross-language retrieval using English versions of the query and an English-Spanish bilingual dictionary.

4.1 Xerox Spanish runs

Our baseline run (xerox-spS) uses Xerox Spanish morphological tools as a linguistic alternative to classical stemming algorithms. In this process, the text is tokenized, tagged, and then the tagged version is passed to the lemmatizer [13], which means that terms are unambiguously reduced to their root forms. In general, this approach is robust and accurate, with a few minor exceptions: tagger errors and terms which are missing from the lexicon. The former may result in terms being reduced to the wrong root form while the latter means that some terms remain unreduced. Accents are removed to help recognize and conflate accented and unaccented version of the same terms. There are very few words in Spanish that have different meanings depending on their accentuation, so this step does not introduce much ambiguity. However, it is not a general solution for all languages. Since these experiments were conducted, we have developed versions of our morphological tools which will conflate accented and unaccented versions of a term if there is no ambiguity. The resulting text is indexed and documents are retrieved in the traditional fashion (using a classic SMART system).

The two noun-phrase runs (xerox-spP and xerox-spD) use indexed phrases which are built by combining nouns with adjacent non-stop words, ignoring the prepositions: 'de', 'para', and 'a,' and their composed forms. Xerox-spP is run on the long version of the query while xerox-spD uses the description field only. For example the sentence: "Senalo que en ese estudio tecnico tambien debera establecerse la factibilidad de realizar una nueva convocatoria para otorgar mas concesiones para ese servicio." would generate "estudio_tecnico factibilidad_realizar convocatoria_nuevo convocatoria_otorgar" in addition to the lemmatised word forms. As with the English phrases described in the section on the NLP track, phrases are ignored for document length determination, and their term weight is halved to reflect the fact that phrases also match on the component terms.

For our cross-language experiment (xerox-spT), we use the English versions of the full Spanish topics that were supplied by NIST this year. Our system applies tagging and lemmatization [13] to reduce all English terms to their root forms. The terms are then automatically looked up in a general-language bilingual dictionary and a translated query is generated by replacing each English term with all of its Spanish translations. The newly constructed Spanish query consists of the union of all of these translation equivalents and is evaluated in the traditional fashion. We have been recently experimenting with a weighted boolean model for cross-language information retrieval that tends to work much better than the vector space model, but this research was not completed in time for our submission. We hope to explore this model further at TREC-6.

4.2 TREC-5 Spanish results

Unfortunately, our FTP session was interrupted while we were in the middle of downloading the AFP data for testing, and we were unaware that this had happened. As a result, we ended up running our Spanish experiments on only three-quarters of the data, so the runs submitted to TREC-5 are not accurate indicators of the true performance of our system. The biggest impact of this problem is to reduce our recall, but precision may also have been affected. The results of running over the full data is given in Figure 6.

Original Spanish Query SP75:

```
<title> Heroin in Latin America
<desc> Description: Hay un mercado para la heroina producida en Amrica Latina
<narr> Narrative: Con las grandes incantaciones de la coca enviada a los EEUU,
ciertos de los productores latinos han vueltos a la produccin de heroina.
```

TREC-supplied English version of SP75:

```
<title> Heroin in Latin America
<E-desc> Description: Is there a market for the heroin produced in Latin America?
<E-narr> Narrative: With the massive seizures of cocaine shipped to the U.S.,
certain Latin American producers have turned to the heroin trade.
```

Re-translated Spanish version of SP75:

```
<desc>  valor en el mercado, precio de mercado, plaza del mercado, mercado, los puestos del
mercado, lonja, informe de mercado, fuerzas, economia de mercado, cuota de
mercado,creador de mercado, comercializar, bolsa de valores, analista de mercado heroina,
heroinomano tener, surtir, sacar, realizar, prolongar, productos alimenticios, producir,
presentar, poner en escena, montar, fabricar, encargarse de la puesta en escena,
encargarse de la produccion, encargarse de la direccion, dirigir, devengar, dar, causar,
...(126 other words)
agriar, adelantar, acostarse, acortar,abrir, (hacer) girar heroina, heroinomano traspaso,
traspasar, ruta comercial, publicar las estadisticas de la balanza comercial, publicacion
especializada, oficio, jugador traspasado, intercambiar,industria, hacer un cambio, guia
de fabricantes y comerciantes, feria comercial, explotar, entregar como parte del pago,
el desequilibrio de la balanza comercial, deficit en la balanza comercial, descripcion
comercial, comercio, comerciar, capitalizar, barrera arancelaria, acuerdo comercial
</desc>
```

Figure 4: Cross-Linguistic Baseline: Each English word is replaced by all translations found in general language bilingual dictionary.

Spanish Runs		Avg Prec	R-Precision
inflectional stemming	xerox-spS	0.295	0.338
noun phrase pairs	xerox-spP	0.301	0.350
" (short Queries)	xerox-spD	0.275	0.326
translated Eng-Span	xerox-spT	0.116	0.155

Figure 5: Spanish results over three-quarters of the data.

Spanish Runs		Avg Prec	R-Precision
inflectional stemming	xerox-spS	0.410	0.415
noun phrase pairs	xerox-spP	0.417	0.425
" (short Queries)	xerox-spD	0.361	0.384
translated Eng-Span	xerox-spT	0.160	0.188

Figure 6: Spanish results over full data.

References

[1] Chris Buckley, Gerard Salton, James Allen, and Amit Singhal. Automatic query expansion using smart: TREC-3. In D.K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*. U.S. Department of Commerce, 1995.

[2] Chris Buckley, Amit Singhal, and Mindhar Mitra. New retrieval apporaches using smart: Trec4. In D.K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*. U.S. Department of Commerce, 1996. to appear.

[3] Wm. S. Cooper, Aitao Chen, and Fredric C. Gey. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. In D.K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 57–66. U.S. Department of Commerce, 1994.

[4] Douglass R. Cutting, Jan O. Pedersen, and Per-Kristian Halvorsen. An object-oriented architecture for text retrieval. In *Conference Proceedings of RIAO'91, Intelligent Text and Image Handling, Barcelona, Spain*, pages 285–298, April 1991. Also available as Xerox PARC technical report SSL-90-83.

[5] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[6] S.E. Robertson et al. Okapi at TREC-3. In D.K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, pages 57–66. U.S. Department of Commerce, 1995.

[7] Joel L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods*. PhD thesis, Cornell University, September 1987.

[8] Norbert Fuhr and Ulrich Pfeifer. Probabilistic information retrieval as a combination of abstraction, inductive learning, and probabilistic assumptions. *ACM TOIS*, 12(1), Jan 1994.

[9] G. Grefenstette. Use of syntactic context to produce term association lists for text retrieval. In *Proceedings of SIGIR'92*, Copenhagen, Denmark, June 21-24 1992. ACM.

[10] G. Grefenstette. Light parsing as finite state filtering. In *Workshop on Extended finite state models of language*, Budapest, Hungary, Aug 11–12 1996. ECAI'96.

[11] Marti Hearst, Jan Pedersen, Peter Pirolli, Hinrich Schütze, Gregory Grefenstette, and David Hull. Xerox TREC-4 site report. In D.K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*. U.S. Department of Commerce, 1996. to appear.

[12] Gerard Salton, C. S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. *JASIS*, 26(1):33–44, 1975.

[13] Anne Schiller. Multilingual part-of-speech tagging and noun phrase mark-up. In *15th European Conference on Grammar and Lexicon of Romance Languages*, University of Munich, Sept 19–21 1996.

[14] Hinrich Schütze, David Hull, and Jan Pedersen. A comparison of document representations and classifiers for the routing problem. In *Proc. 18th Int'l Conference on R&D in IR (SIGIR)*, pages 229–237, 1995.