TREC-4 Experiments at Dublin City University: Thresholding Posting Lists, Query Expansion with WordNet and POS Tagging of Spanish

Alan F. Smeaton[1], Fergus Kelledy and Ruairi O'Donnell

School of Computer Applications, Dublin City University, Glasnevin, Dublin 9, IRELAND (asmeaton@CompApp.DCU.ie)

Abstract

In this paper we describe work done as part of the TREC-4 benchmarking exercise by a team from Dublin City University. In TREC-4 we had 3 activities as follows:

- In work on improving the efficiency of standard SMART-like query processing we have applied various thresholding processes to the postings list of an inverted file and we have limited the number of document score accumulators available during query processing. The first run we submitted for evaluation in TREC-4 (DCU951) used our best set of thresholding and accumulator set parameters.
- The second run we submitted is based upon a query expansion using terms from WordNet. Essentially, for each original query term we determine its level of specificity or abstraction; for broad terms we add more specific terms, for specific original terms we add broader ones; for ones in-between we add both broader and narrower terms. When the query is expanded we then delete all the original query terms in order to add to the judged pool, documents that our expansion would find that would not have been found by other retrieval. This is run DCU952.
- The third run we submitted was for Spanish data. We ran the entire document corpus through a POS tagger and indexed documents (and queries) by a combination of base form of non-stopwords plus their POS class. Retrieval is performed using SMART with extra weights for query and document terms depending on their POS class.

The performance figures we obtained in terms of precision and recall are given at the end of the paper.

---

[1]The work reported here was carried out while the first author was visiting the CLIPS-IMAG laboratory in Grenoble during 1995 funded by the Ministère de l'Enseignement Supérieur et de la Recherche and by Dublin City University. Thanks are due to François Paradis (Grenoble) for help with Perl scripts. Thanks are also due to Jim Cowie and staff at CRL in NMSU for use of their SPOST POS tagger for Spanish, at their site and for their help with tagging the Spanish corpus

1. Introduction.

The work reported in this overview is a description of the research we carried out for our efforts in TREC-4. Essentially, we have submitted 3 official runs, the first two for Category A ad hoc, and the third for Spanish ad hoc. Each of the 3 runs are based on 3 separate streams of research and are described independently in the following 3 sections of the report.

For all our work, the platform we used for our experiments was a SUN SparcStation 20 with 96 Mbytes RAM and 12.5 Gbytes local disk attached directly to the SCSI port and a further supply of disk space available via NFS and an ETHERNET LAN.

2. Run Based on Thresholding the Postings Lists

2.1 Thresholding

Under normal processing conditions, the query space in an inverted file is made up a concatenation of the posting lists of query terms. As a retrieval algorithm processes entries in the posting lists, a set of accumulators are used to keep scores for individual documents and at the end of the processing the top-scoring documents, identified by the highest accumulator values, are retrieved.

In our approach to query processing, the query space is composed of query term posting lists sorted in order of increasing posting list length with each posting list also sorted in order of decreasing value to its respective index term [Kelledy and Smeaton, 1995].

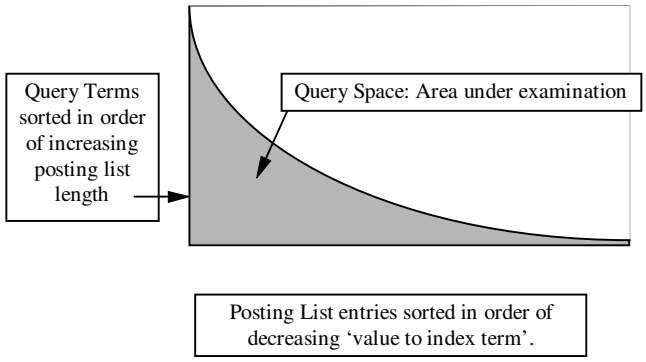

Figure 2.1 Abstract view of Query Space

Figure 2.1 gives us an abstract view of the Query Space (*QS*). It is this *QS* which must be processed in the most efficient and effective manner as possible in order to respond to user queries with meaningful answers, in this case a list of documents ranked in order of possible relevance to the user's query. Due to the structure of the *QS* in which the posting lists are arranged vertically and horizontally in order of probable value to the query, one can assume that the concentration of relevant information with respect to the query is higher in the top left part of Figure 2.1 than in the bottom right part. This *QS* structure facilitates the notion of thresholding, in which not all of the *QS* need be processed. This results in four main options available for investigation, namely no thresholding, query term thresholding, where only a subset of the terms within the query are processed, posting list thresholding, where not all of the posting information in each posting list is

processed and finally, the combination of query term thresholding and posting list thresholding, where only a portion of the posting list information for a subset of query terms are processed.

2.2 No Thresholding

Figure 2.2 illustrates the basic approach to processing the query space. In this variation no thresholding whatsoever is applied. The result of this approach is that all posting entries of all query term posting lists are processed. In a situation such as ours where we could have large queries (on average ~50 terms per query) and a very large document collection the cost of processing the query space associated with a given query is a non-trivial one. This coupled with the fact that a lot of noise[2] would also be generated from processing non-discriminating terms with low within document frequencies (the lower right hand side of Fig. 2.2) leads to relatively poor results. This lower right hand size of the query space is more likely to contain postings which, firstly, are associated with terms that occur in a large proportion of the documents within the collection therefore reducing there discriminating power and secondly, their within document frequency is likely to be very low (due to the sorting process). These two facts make it more likely that postings in this region of the query space for non-discriminating terms with low within document frequency are more prone to generating erroneous information i.e. noise and therefore slowing down overall system performance.

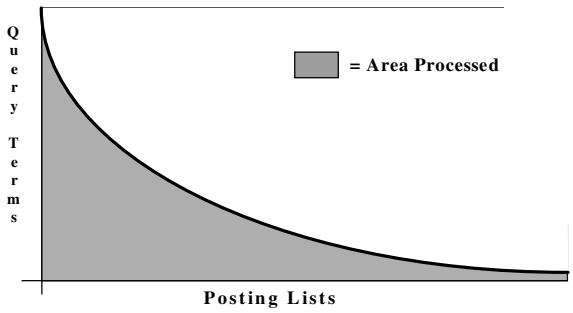

Figure 2.2 No Thresholding on the Query Space.

Figure 2.2 illustrates the overall procedure involved in processing the *QS* without any thresholding techniques in operation. The enture shaded area in Fig. 2.2 is processed and every accumulator that acquires a weight greater than 0 is passed on to the document scoring and ranking procedures.

2.3 Query Term Thresholding

Improvements to efficiency and effectiveness of the retrieval process can be achieved by incorporating Query Term Thresholding (*QTT*) into the processing of the query space when the queries are long. This involves computing the percentage of documents within the collection in which each of the query terms occurs. There is very little additional overhead involved in computing this and these percentages are used to exclude the processing of the most non-discriminating query terms (those that occur in a large percentage of documents within the collection). At the present stage in our experiments the thresholding percentage is fixed for all queries, we hope to modify this procedure to allow the threshold value to be automatically computed based on criteria associated with the query. Figure 2.3 illustrates the net effect of *QTT* on the query space, query terms up to a cut-off point are processed, after which the longer, more

[2] Noise: Erroneous data which might interfere and degrade overall system performance.

time consuming and 'noisy' posting lists are discarded therefore reducing the time taken to process the query space.

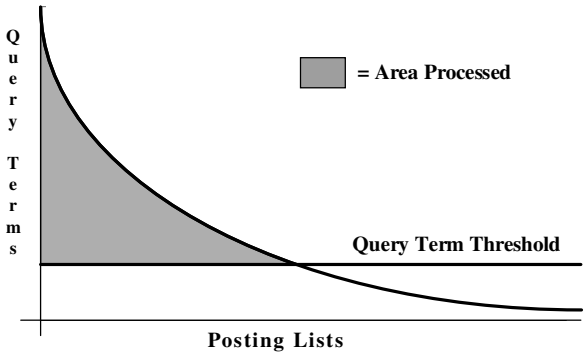

Figure 2.3 Query Term Thresholding the Query Space

2.4 Posting List Thresholding

Further improvements to efficiency can be achieved by incorporating Posting List Thresholding (*PLT*) into processing of the query space. This involves computing percentage of the posting list for each query term that is to be processed either in advance as static values or dynamically depending, for example, on the ranked position of the term within the query. There is little overhead involved in computing these values which in turn are used to exclude the processing of the most non-discriminating posting list entries (those that occur a small number of times within a given document). Figure 2.4 illustrates the effect of *PLT* on the query space where query term posting lists up to a cut-off point are processed, after which the least useful postings for that term are not considered to be of great importance to the query relative to the postings already processed. This reduces the time taken to process the query space by eliminating the need to process all postings entries for a given posting list, it will also reduce the number of document accumulators activated by noisy posting list entries i.e. a document accumulator being activated by a once off occurrence of a term within a document which could possibly have a negative affect of the overall systems performance. This document accumulator once activated must be included in the sorting procedure therefore also having a negative impact of the system's efficiency.

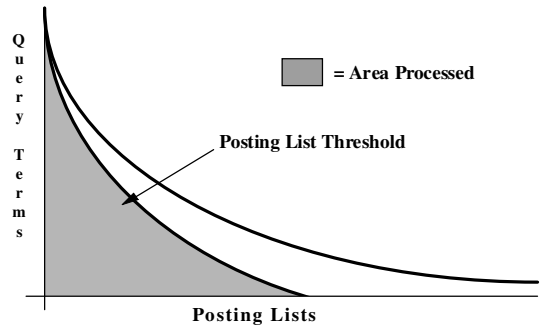

Figure 2.4 Posting List Thresholding the Query Space

2.5 Query Term and Posting List Thresholding

The above two approaches, namely *QTT* and *PLT,* can be combined in order to further reduce the amount of the query space to be processed. The result of combining these two thresholding approaches can be seen in Fig. 2.5. Depending on the parameter values chosen for *nqt'* and pqt_i quite a large reduction in the area of the query space to be processed can be achieved.

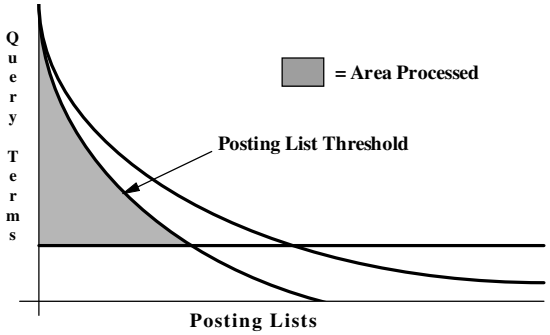

Figure 2.5 Posting List and Query Term Thresholding the Query Space.

It is obvious that the above approaches either taken alone or combined will have a positive effect on the efficiency of a retrieval process. However it remains to be seen what impact the above approaches will have on the effectiveness and the above approaches can only be justified if they do not noticeably degrade overall system effectiveness.

2.6 Restricting Document Accumulators

Processing the *QS* is only one aspect of the retrieval process. The output from the processing of the *QS* is passed to the Accumulator Space (*AS*) which can be visualised as a set of counters, one for each retrieval unit. The unit of retrieval can be a document, page, paragraph, sentence or a word. The more fine-grained the retrieval unit the larger the *AS* becomes. A given query will only activate a subset of the *AS*, by activate we mean store a non-zero value in an accumulator. This Active Accumulator Space (*AAS*) must then be processed. This processing might entail a conversion from one unit of recovery to another, for example, page to document resolution. This means that the individual query page similarity scores are combined in some fashion to form query-document similarity scores. These query unit similarity scores, be they page or document similarities, must then be ranked in order of estimated relevance to the query and presented to the user.

The extraction of the *AAS* from the *AS* and its subsequent processing (unit resolution and sorting) contributes much of the overhead involved is query processing. This coupled with large *QSs* and no thresholding optimisations results in poor system performance in terms of efficiency. In such a situation even if the system is providing high quality results the user may not be prepared to wait the necessary amount of time to receive the response from the IR system.

It is therefore of great importance that attention be given not only reducing the amount of the *QS* that is processed but also the storage structures used to manipulate the *AS*. A simple assumption that can be made at this point is that the more information that ends up in the *AAS* the longer it will take to process the query. Following on from this assumption it is logical to try and control the amount of information that goes into the *AAS*. This control of the *AAS* information can be implemented during the processing of the *QS* by introducing the notion of restricting the number of accumulators that can become active with respect to a given query.

2.7 Experimental Results

The variations on the basic query processing and retrieval algorithm outlined so far require a number of parameter settings in order to operate efficiently and effectively. We turned TREC-3 topic descriptions, generally accepted as being long, into a shortened form and used the shortened form as queries in order t determine the best set of parameters used in *QTT*, *PLT* and the reduced accumulator set.

Figuures 2.6, 2.7 and 2.8 show how setting the maximum number of document accumulators effects the average precision, the interpolated precision at 0.0 and the precision at recall level 0.10 respectively.

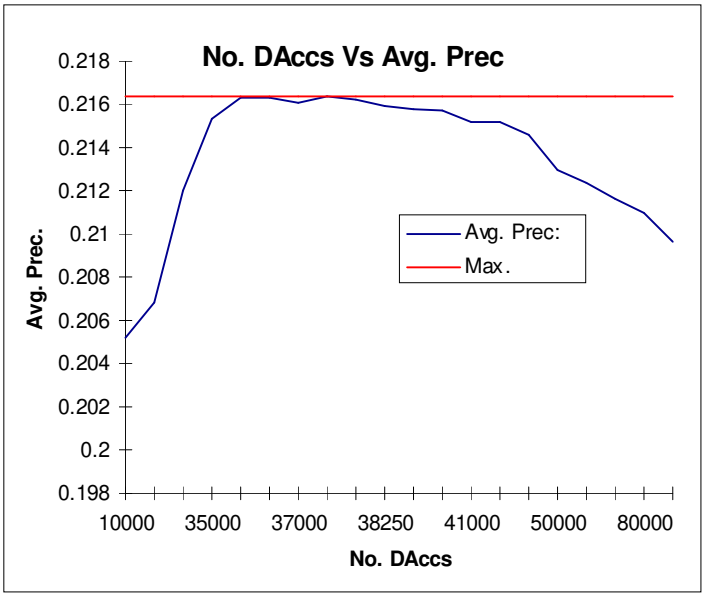

Figure 2.6 No. of Document Accumulators Vs Average Precision

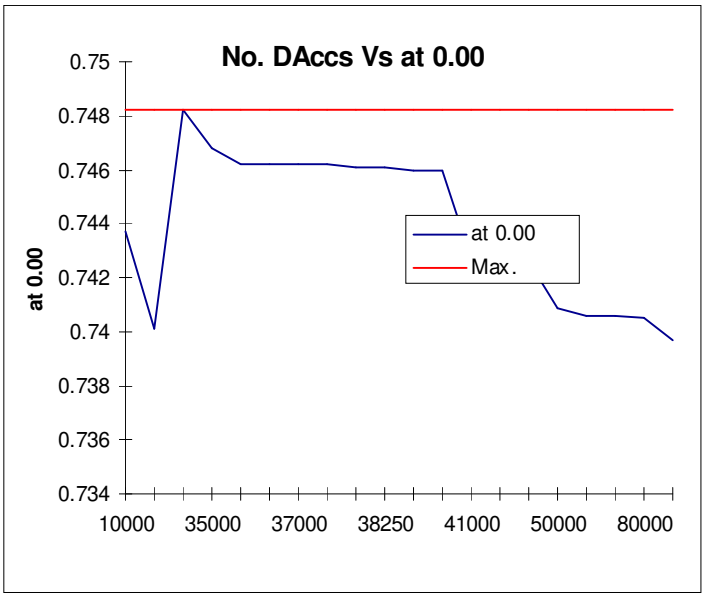

Figure 2.7 No. of Document Accumulators Vs Precision at 0.00

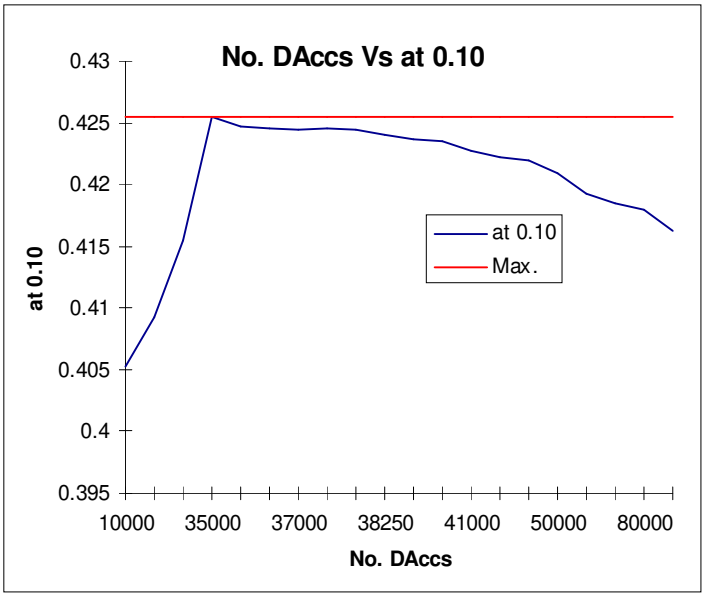

Figure 2.8 No. of Document Accumulators Vs Precision at 0.10

These three evaluation aspects show the peak performance in terms of effctiveness is where the maximum number of accumulators is set to between 35,000 and 36,000 registers. Given that there are a much larger set of documents than this in the collection this is making a considerable impact on efficiency. The big surprise is the effect these restrictions have on effectiveness. It is a positive **improvement** in retrieval effectiveness to introduce such efficiency considerations. This is shown even more dramatically in the graph in Fig. 2.9 illustrating retrieval effectiveness when no thresholding at all is used vs. our best parameter setting. These experiments were run on TREC-3 topics as training for the parameter settings for our TREC-4 runs.

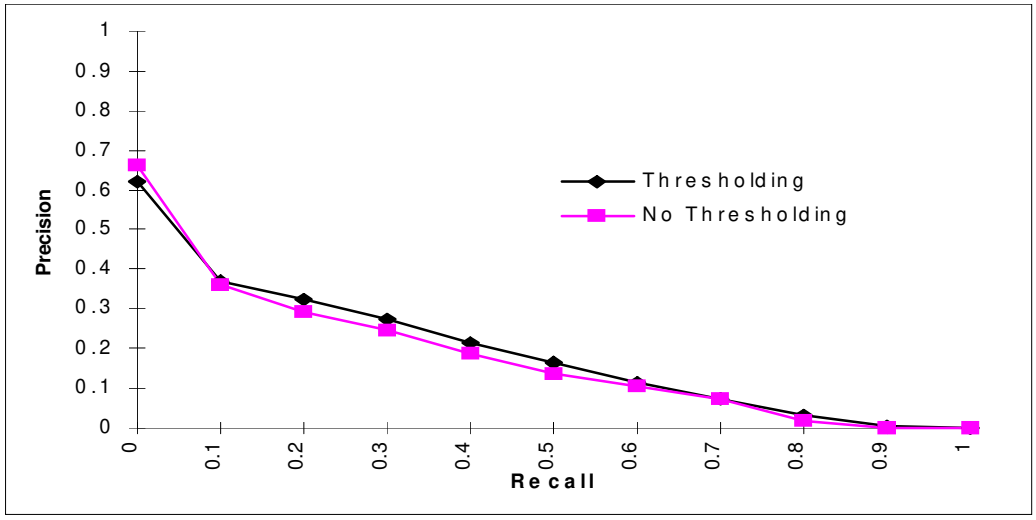

Figure 2.9 Recall-Precision for Thresholding Approaches. (TREC-3)

Here we can see that the thresholding is causing the overall effectiveness to be improved, as well as making improvements in retrieval efficiency. Given these results we settled on the best number of accumulatrors, *QTT* and *PLT* thresholding as determined by training on TREC-3 topics and used these in our TREC-4 run. This run was coded DCU951 and the precision-recall figures are shown in Fig.2.10. Post-official-submission, we re-ran these queries with no thresholding or accumulator restrictions and the results we obtained in that subsequent run is also shown in Fig. 2.10

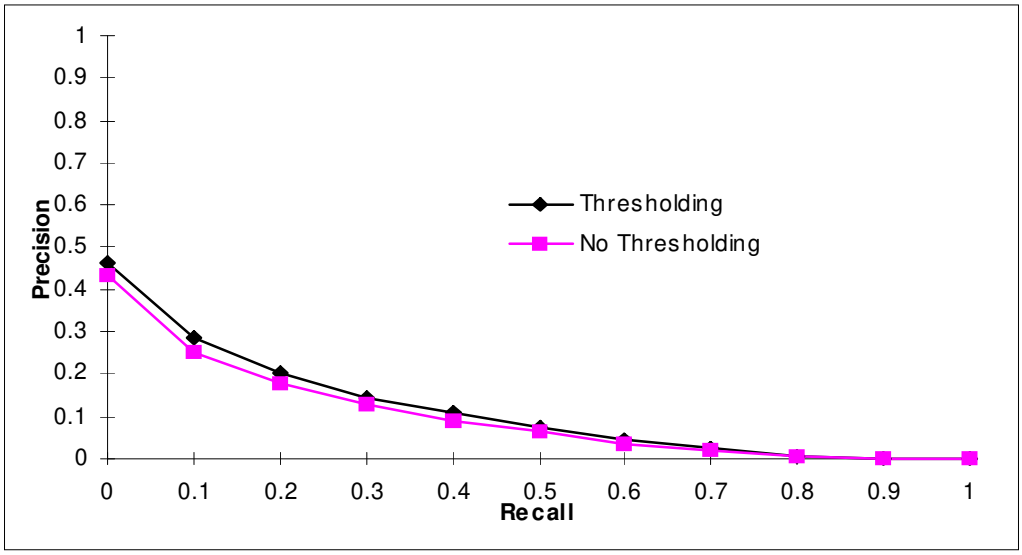

Figure 2.10 Recall Precision for Thresholding Approaches (TREC-4)

	No Thresholding	Thresholding	% Improvement
Postings Processed	58,284,390	29,678,636	49.08%
Elapsed Time (Sec)	5732	853	85.12%
Accumulators Used	25,537,453	1,499,865	94.13%

Figure 2.11 Efficiency improvements (TREC-3)

	No Thresholding	Thresholding	% Improvement
Postings Processed	9,606,861	7,796,370	18.85%
Elapsed Time (Sec)	1,353	452	66.59%
Accumulators Used	7,057,794	1,440,602	79.59%

Figure 2.12 Efficiency improvements (TREC-4)

Figure 2.11 and 2.12 illustrate the improvements in efficiency gained by employing the thresholding techniques outlined above. It can be clearly seen that these thresholding approaches perform better in terms of improving efficiency when operating on larger queries as is the case in TREC-3. While the results achieved using our thresholding approaches on the TREC-4 data are

not as impressive as TREC-3 results they still represent a significant saving in the overall time taken to process the queries. The reason for the poorer results in TREC-4 is due to the much smaller queries which leave the thresholding techniques much less room in which to operate. We believe these thresholding approaches will offer maximum benefit when applied to large or automatically expanded user queries.

The overall performance of our system in terms of effectiveness with and without thresholding was fairly poor compared to other. This we attribute to an over simplistic tf*idf weighting scheme coupled with the restriction of only indexing documents and queries by WordNet nouns and verbs (necessary in order to link with our other submitted run). We intend carrying our further work in this area by removing the WordNet noun / verb restriction and replacing the existing weighting scheme with other well known, high performing weighting schemes (OKAPI,Cornell) and determining whether or not our thresholding approaches attain similar improvements in efficiency without adversely affecting effectiveness.

3. Run based on WordNet Query Expansion.

The approach we describe here is based on expanding an original query by adding terms from a structure derived from a lexical database, WordNet. We perform a thesaurus-like expansion of original terms and in doing this we hope to collect relevant documents not found by the more traditional keyword IR approaches. For the purposes of fine-tuning, we trained the system using queries from TREC-3 which are considerably longer than the queries for TREC-4. To account for this we produced shortened versions of TREC-3 queries and used these for experimentation.

Although the query expansion work reported here was carried out in Grenoble, France, the experiments were run on computers in Dublin. Documents have been partitioned into fixed-size pages based on word counts and the "score" for each document is the "score" for the highest-scored partition it contains. This addressed the issue of normalisation for document length.

Each document has had stopwords removed and has been indexed by WordNet nouns and verbs, both single word and collocational, that occur in the text. Adjectives which can only appear as adjectives (*big, long,* etc) are discarded as are adverbs where these do not have verb or nominal forms. Terms indexing documents (or document partitions) are weighted by their IDF weights and a document (partition) score is the sum of the tf*IDF weights of query terms occurring in that document (or partition). The system used to implement this retrieval was written in C by Fergus Kelledy and uses an inverted file for efficiency. It was developed to facilitate research into improving search time by sorting and applying thresholds to postings lists in inverted files as in our first official TREC-4 run. A version of the system which searches the TREC collection and runs in batch mode was developed and the input to runs of this system was a query file which contain, for each query, the query terms to be used. This meant that when we wanted to evaluate a query expansion strategy we generate a new query file containing the query terms we wanted and used this as an input to a run. The query files were developed in Grenoble and sent to Dublin by ftp and then used as input to runs on the machines in Dublin.

The format of the query file was such that it allowed us to assign weights to query terms depending on the query expansion strategy being evaluated. This was done by specifying, in the query file, each term to be used plus a weight for that term to be used as an envelope factor in computing

document scores. An individual document's score would be the sum of the tf*IDF weights of original query terms plus 0.5 times the sum of the tf*IDF weights of synonyms. A good deal of experimentation took place varying the contribution of these envelope weights. The time taken to perform a run of the system depended on the number of terms added to queries; the more query terms, the longer the processing. Because of the way in which the implementation worked, the increase in time taken was almost linear. For a set of 50 queries searching c.2 Gbytes of document text with an average of 8 terms per query, a run took between 10 and 15 minutes of elapsed time. Most of this was spent on disk I/O.

3.1 WordNet and HCGs

WordNet is the product of a research project at Princeton University which has attempted to model the lexical knowledge of a native speaker of English [Miller, 1990]. The system has the power of both an on-line thesaurus and an on-line dictionary, and much more. Information in WordNet is organised around logical groupings called synsets. Each synset consists of a list of synonymous word forms and semantic pointers that describe relationships between the current synset and other synsets. A word form can be a single word or two or more words connected by underscores, (referred to as collocations).

In our work on defining relationships between words we only use the nouns and IS-A links from WordNet, ignoring the verbs, adjectives and adverbs. Where a TREC query contains words in other classes these will be nominalised as discussed earlier. For non-noun word occurrences in documents, these will be ignored. This leaves us with c.60,000 word forms of which more than half are single words and the remainder collocations. This is adequate for our task.

The initial form of our KB consisted of a number hierarchical concept graphs, (HCGs), constructed directly from WordNet. The root concepts of the HCGs were chosen as result of a set of experiments to determine what root concepts would, as a group, provide maximum coverage of the nouns in WordNet whilst minimising the degree of overlap between HCGs. The motivation for creating HCGs was purely efficiency in order to make the computation more manageable and as we will see later we have not sacrificed effectiveness in any way. The resulting HCGs ranged in size from 43950 unique concepts (Entity) to 688 concepts (Shape). While we would have preferred a more even balance across HCG size, we believe the disparity in the HCGs we use does not cause any major problem.

The information based approach to measure semantic similarity which we used in these experiments is motivated from work carried out by Resnick [Resnick, 1993] and was done by Ray Richardson [Richardson & Smeaton, 1995]. The semantic similarity of concepts is estimated using the distributional behaviour of classes of words in a large text corpus. Resnick viewed noun synsets as a class of words where the class is made up of all words in a synset as well as words in all directly or indirectly subordinate synsets. Conceptual similarity is considered in terms of class similarity. The similarity then between two classes is approximated by the information content of the first class in the noun hierarchy that subsumes both classes. The information content of a class is approximated by estimating the probability of occurrence of the class in a large text corpus. As such, the similarity of two classes of words, $c1$ and $c2$, can be expressed as:

$$Sim(c_1, c_2) = \max_{C_i}[\log \frac{1}{P(C_i)}]$$

where {Ci} is the set of classes dominating or superordinate to both c_1 and c_2, P(Ci) is the class probability of class c_i, and $\log 1/P(C_i)$ is the information content of class Ci. In our work we took 278 Mbytes of text from the *Wall Street Journal,* tagged this with a parts-of-speech tagger to detect occurrences of nouns, or word occurrences which were potentially nouns, and used the frequency of occurrence of WordNet nouns and collocations to compute the information content of synset classes in our HCGs.

The information based semantic similarity estimator measure is not without weaknesses. Perhaps foremost is the fact that this technique ignores information in WordNet that may be useful namely that only the synonym and IS-A relations are used while the other relation types, which are used effectively by the adapted Rada-like conceptual distance approach [Rada, 1989], are unused. A second weakness is apparent in the method we used to calculate the information content of classes whereby many polysemous words will have an exaggerated information content value. If one takes for instance the polysemous word *'bank'*, then the information content for this word will include all occurrences of the string *'bank'* occurring as a noun in the corpus, regardless of meaning. This is due to the fact that we did not, and indeed cannot yet automatically distinguish WordNet word senses from free text with any degree of accuracy above 65% [O'Donnell & Smeaton, 1995]. This net result of being unable to distinguish word senses gives the same exaggerated information content value to a 'commercial bank' as to a 'river bank'. To further complicate things, given the fact that the information content of a class is defined in terms of the information content of its subordinate classes, superclasses of classes containing polysemous words are similarly over-valued. This disregard of ambiguous words is a particular problem given the fact that classes in WordNet refer to particular senses.

3.2 Disambiguating into Multiple Word Senses

In doing the sense disambiguation of TREC-3 and then of TREC-4 queries, we found many instances of simply not being able to distinguish between WordNet's fine senses. The original query term *prison* for example, has (at least) the following WordNet senses which could both be legitimate in the context of query 151 (Coping with overcrowded prisons)

{prison}
a_kind_of {penal_institution}
a_kind_of {institution}
a_kind_of {establishment}
...
{prison, detention, hold, custody}
a_kind_of {confinement}
a_kind_of {state}
...

We sense-disambiguated queries into any number of reasonable WordNet senses and found in our training runs that this did not degrade performance and as it was a natural thing to do, we did this for our official run. The figures for this were that in the original one-sense-per-term version for

TREC-3 queries there were 310 query terms while disambiguation into any number of senses yielded a total of 448. A similar ration was found for TREC-4 topics. For our official TREC-4 runs we manually sense disambiguated query terms into as many senses as made sense.

3.3 Expanding Terms

In the expanded form of queries, each query term was expanded independently and equally (in terms of terms added) and all synonyms from the expanded generation synsets were added and all of those weighted equally in turn. For the expansions, parents of query terms (P) were given weights, as were grandparents, children (C) and grand-children (GC). We did not do any expansions beyond these generations.

For example, in (this hypothetical) query the original term is "*prison*" which can be expanded to

> { prison 1.0, penal_institution 0.5, institution 0.25, camp 0.5, work_camp 0.25, prison_camp 0.25, prison_farm 0.25, prison_camp 0.25, internment_camp 0.25, prisoner_of_war_camp 0.25, POW_camp 0.25, college 0.5, house_of_correction 0.5, hold 0.5, keep 0.5, jail 0.5, jailhouse 0.5, gaol 0.5, lockup 0.5, lock_up 0.5, cooler 0.5, pokey 0.5 }

... where prison has weight 1.0, penal_institution is a parent with weight 0.5, institution is its grand parent with weight 0.25, etc., and where *P(), GP(), C(), and GC()* refer to parent, grand-parent, etc. The children of prison who have a weight of 0.5 are camp, college, house_of_correction, hold, keep, etc.

3.4 Selectively Expanding Terms Based On Abstraction Level

In the final run we submitted, we examined the information content value (ICV) [Richardson & Smeaton, 1995] of query term nodes in HCGs and used those values as indicators of whether a query term is abstract or specific. We define a broad term as one which would be high up in the hierarchy ... query terms such as

cause 0.967
human 0.973
individual 0.937
person 0.973

with their associated ICVs are abstract. Terms such as ...

minority 6.085
ineffectiveness 6.039
fishing 6.562
asbestosis 6.085
killing 6.340

with their ICVs are more specific. ICVs vary from zero to 7 (in theory, but our lowest ICV was 0.937 and highest was 7.039 (*food_fish*)). Given we can estimate the level of abstraction of a

query term from its ICV, we computed the ICVs for all 448 TREC-3 query term word senses and found the mean to be 3.428 (for the terms *legislation* and *school* respectively). Once again a similar distribution of levels of abstraction was found for TREC-4 queries. Roughly, therefore we thus define broad as ICV<3.0 and narrow as ICV>4.0 with intermediate terms having an ICV between 3.0 and 4.0.

For our official TREC-4 run we submitted a run in which we add in children (1.0) and grandchildren (0.5) of broad terms (ICV<3.0) and parents (1.0) and grandparents (0.5) of narrow terms (ICV>4.0) and for the intermediate terms we add in both parents (1.0), grandparents (0.5), children (1.0) and grandchildren (0.5). In TREC-3 queries there were 135 such QTs expanded with that definition of broadness, and 106 QTs terms expanded with that definition of narrowness, with 207 of the 448 query terms expanded by both parents and children

The formal definition of out TREC-4 strategy is based on an idea adopted from that used in the IOTA system [Chiaramella *et al.,* 1987] and is defined as:

$$Q = \left\{ \begin{array}{l} ((syn(P(QT_i")),1.0) \cup (syn(GP(QT_i")),0.25)) \;\; \ldots \; iff \; ICV(QT_i") > 4.0 \\ \cup \;\; ((syn(C(QT_i")),1.0) \cup (syn(GC(QT_i")),0.25)) \;\; \ldots \; iff \; ICV(QT_i") < 3.0 \\ \quad \cup ((syn(P(QT_i")),1.0) \cup (syn(GP(QT_i")),0.25) \\ \quad \cup .(syn(C(QT_i")),1.0) \cup (syn(GC(QT_i")),0.25)) \;\; \ldots \;\; otherwise \end{array} \right\}$$

An important factor to note is that our approach to participation for this run in TREC is **NOT** to obtain the best possible results in terms of precision and recall; there are enough other groups taking part in TREC striving to do this. Instead of mixing together all techniques known to yield improve retrieval effectiveness, our interest in this run is in isolating and evaluating the contribution to be made by query expansion using our WordNet-derived structure. To emphasise this we removed all the original query terms from the queries used in the official runs and thus evaluated based on the expanded queries only.

3.5 Results of Query Expansion Run

As would be expected from any retrieval strategy based on *throwing* away the user's original query terms, we performed quite badly in terms of precision and recall figures as shown in Table 1, reproduced from the official results. The interesting thing to note is that for the 49 topics, we retrieved a total of 347 of the 6501 relevant documents, meaning that there are 347 relevant documents who have no query terms, as determined by our indexing into WordNet nouns, at all. It will be interesting to examine the files submitted by other TREC participants to see if we retrieved any relevant documents which were not found by other strategies at all. This is the real evaluation of the strategy we adopted in DCU952. Meantime, we will use the relevant document set determined for TREC-4 topics, to evaluate the query expansion strategy in more realistic experiments based on keeping rather than discarding user's original query terms.

4. Ad hoc Spanish Runs.

TREC-4 was the second time that Dublin City University submitted runs for ad hoc Spanish retrieval.. In TREC-4 we had decided to do something which involved POS tagging of the document and query texts.

About 3 years ago we ran some experiments on the (English) CACM test collection in which we tagged all of the tokens in the documents and queries [Smeaton, 1992]. We then performed a standard SMART-like retrieval based on term weighting but increased the weight for a query and for a document term depending on whether the term occurred as a headnoun, modifying noun, verb, adjective, adverb or stopword. These POS categories are quite rough, for example we did not distinguish between different forms of verbs, but were adequate for our needs. In subsequent experimental runs, which we never published except as an internal report, we were surprised to find that when we doubled the *tf*IDF* weight for adjectives we obtained the best of our results in terms of precision and recall. Our Spanish efforts in TREC-4 were based on repeating those experiments.

Our first task was to locate a POS tagger for Spanish and we had two offers. The first came from UMass who were developing one and the second came from NMSU who had already developed SPOST. As neither tagger had been evaluated in terms of accuracy, after consideration we opted for SPOST on the grounds that it would be different to the UMass tagger and one of the overall objectives in TREC is to build a test collection so with a different tagger to UMass it would be more likely we would retrieve differing document sets that would be better. Problems arose in using SPOST in that it is tied to a Spanish dictionary and in order to move the code to Dublin we would have had to buy a licence for the dictionary which we did not have funds for. In the end we opted to tag the corpus using machines at NMSU, during their night-time period (close to our day given the time difference). We chopped the corpus into manageable sized chunks and tagged each chunk on a different workstation in *nice* mode. At one stage we were running on 8 SUN workstations. The output of the tagger was post-processed at NMSU using code we developed in Dublin, compressed and ftp-ed back to Dublin. There it was used as the input to SMART which had been tailored to allow a document to be indexed by a baseform of a word and the associated grammatical category. Retrieval is based on the usual SMART *tf*IDF* weighting of terms and document ranking.

For retrieval we tagged the Spanish topics in the same was as the documents and input them to SMART as terms and their grammatical categories. As there is no large reliable set of topics and relevance assessments for the Spanish data we submitted runs where the base forms of words which are true adjectives are given double their normal *tf*IDF* weights. When the Spanish relevance assessments are made available we will perform a series of runs varying the categories that get weighted and the amount of those weights and we expect to have an updated result set, in addition to the results of the official run, by the time of TREC-4.

We were initially rather surprised by the poor performance of our approach to ad hoc retrieval in the context of performances from other groups. Our performance figures are given in Table 1 as DCUSP0. What surprised us was the rate at which our precision tailed off. In subsequent, post-official runs we have not been able to move out of this ballpark in terms of effectiveness. This leaves us a long way behind approaches to Spanish TREC ad hoc based on simple stemming and

term weighting, an approach taken by the group from Cornell, for example. The following explanations are all possible:

- baseforms derived from POS tagging, while being true baseforms, are not as good as word stems, even using a crude stemming algorithm
- the accuracy of the POS tagger has not been evaluated
- we discovered, after submission of results,[3] that verb and nominal forms of words may not have been reduced to the same baseform in the way we used SPOST. For example, in English, one of the the verb forms of *"to run"* would be *"runs",* as in "*he runs for the bus every day"* and a nominal form would be "*runs"* as in "*he likes to have his runs every day"*. Using stemming, these occurrences would be reduced to the same term but the way we using the POS tagging meant that these would not have been identified as the same base form (which they are not) or even the same concept (which they are).
- the variant of the term weighting scheme we used, *tf*IDF* weighting from SMART, was not experimented with at all in our experiments

Recall	DCU951	DCU952	DCUSP0
0.00	0.4640	0.0577	0.4066
0.10	0.2907	0.0136	0.2120
0.20	0.2040	0.0061	0.1356
0.30	0.1471	0.0042	0.0967
0.40	0.1093	0.0006	0.0775
0.50	0.0760	0.0001	0.0427
0.60	0.0453	0.0001	0.0202
0.70	0.0246	0.0001	0.0129
0.80	0.0050	0.0001	0.0093
0.90	0.0000	0.0000	0.0000
1.00	0.0000	0.0000	0.0000
Av. Precision over all reldocs	**0.1066**	**0.0037**	**0.0735**

Table 1: Official Results for DCU runs

5 Conclusions

Participation in TREC-4 has been a rewarding experience for the team from Dublin City University. In TREC-3 we participate as category B using the reduced data set and managed to "complete the course", albeit with poor results. In TREC-4 our participation was managed comfortably within deadlines and we succeeded to pursue three differing lines of research. Instead of mixing as many ingredients to improve retrieval effectiveness, however, in each of our differing lines we concentrated on one aspect only.

[3] This pointed was highlighted for us by Ted Dunning.

For the future we will continue to participate in TREC, concentrating on efficiency issues and perhaps also exploring the corrupted data track.

References

[Chiaramella *et al.,* 1987] "A Prototype of an Intelligent System for Information Retrieval; IOTA", Y. Chiaramella *et al., Information Processing and Management,* 23(4), 285-303, 1987.

[Kelledy & Smeaton, 1995] "Thresholding the Postings Lists in Information Retrieval", F. Kelledy and A.F. Smeaton, Proc BCS-IRSG Colloquium, Crewe, April 1995 (also available as http://www.compapp.dcu.ie/CA_Working_Papers/wplist95.html)

[Miller, 1990] "Nouns in WordNet: A Lexical Inheritance System", G.A. Miller, *International Journal of Lexicography,* 3(4), 245-264, 1990.

[Rada *et al.*, 1989] "Development and Application os a Metric on Semantic Nets", R. Rada *et al., IEEE Transactions on Systems, Man and Cybernetics,* 19(1), 17-30, 1989.

[Resnick, 1993] "Selection and Information: A Class-based Approach to Lexical Relationships", P. Resnick, *PhD dissertation, University of Pennsylvania,* 1993.

[Richardson & Smeaton, 1995] "Using WordNet in a Knowlede-Based Approach to Information Retrieval", R. Richardson and A.F. Smeaton, Proc BCS-IRSG Colloquium, Crewe, April 1995 1995 (also available as http://www.compapp.dcu.ie/CA_Working_Papers/wplist95.html)

[Smeaton, 1992] "An Evaluation of Retrieval Performance Using Simple Statistics and SIMPR Linguistic Processing on a Standard Collection of Texts", A.F. Smeaton, *SIMPR Project Report 50.2I, Dublin City University,* March 1992.

[Smeaton & O'Donnell, 1995] "A Word Sense Disambiguation Algorithm Using Word Distances; Evaluation on SemCor", *forthcoming, 1995.*