Ad hoc Retrieval Using Thresholds, WSTs for French Mono-lingual Retrieval, Document-at-a-Glance for High Precision and Triphone Windows for Spoken Documents

Alan F. Smeaton[1], Fergus Kelledy[2] and Gerard Quinn[1]

[1]School of Computer Applications
Dublin City University
Glasnevin, Dublin 9, IRELAND.

[2]Broadcom Éireann Research
Dublin, IRELAND

Abstract:

This paper describes work done by a team from Dublin City University as part of TREC-6. In this TREC exercise we completed series of runs in 4 categories. The first was the mainline ad hoc retrieval task in which we repeated our entry for TREC-5, without modification. This is based on applying various thresholds to processing a query including query term and posting list thresholds, in order to improve retrieval efficiency. As our previous work has shown, this can be done without any loss in retrieval effectiveness. Our second set of submitted runs were as part of the cross-lingual retrieval track where we ran French topics against French texts, effectively mono-lingual retrieval. What is novel about our approach is that it is based upon matching word shape tokens derived from character shape codes, rather than matching word stems or base forms. This technique is useful for retrieving from scanned document images rather than full texts and is something we are currently refining for English texts (and English queries). With those other experiments we have obtained surprisingly effective retrieval and this venture in TREC-6 was to see how effective WST-based retrieval could be for French. The third series of experiments we submitted were based on the high precision track in which we used a graphical representation of a ranked list of documents and the positional occurrences of search terms within those top-ranked documents, relative to each other. Our final experiments were as part of the spoken document retrieval track in which we removed the tags used for story bounds, turned transcripts and topics into a phonetic representation using a phoneme dictionary and we then retrieved story identifiers based on a triphone match between topic and fixed-width windows of triphones in the transcripts. We also applied a weighting function to triphones as they occurred in story "windows" based on their offset within those windows.

1. Introduction

TREC-6 is Dublin City University's fourth consecutive year for involvement in TREC and our largest to date. Our work is neatly divided into four distinct areas representing the mainline ad hoc retrieval task and three of the specialist tracks. Each is described here in turn and conclusions are drawn about each of our work areas.

2. Mainline Ad Hoc Retrieval Task

Our submissions to the mainline ad hoc retrieval task in TREC-6 were the same as our submissions for TREC-5 and in fact we used the exact same parameter settings as in TREC-5. This work is based on reducing the search space for processing a query, and hence the execution time, by applying a combination of thresholds to the processing. These include reducing the number of search terms looked up in the inverted file, the proportion of entries in the inverted file for a given search term, and the total number of document accumulators or registers used to hold document scores, i.e. the total number of documents assigned a score of any kind. While the motivation for this work is to reduce the processing time we have repeatedly shown in TREC-5 and in experiments on the data from TREC-4 and TREC-3 that this can be accomplished without loss of retrieval effectiveness, and indeed with a marginal improvement in precision and recall figures. We refer the reader to other publications [Smeaton & Kelledy, 1997a and 1997b] for further details of the methods we use.

The results we obtained for our TREC-6 submissions are summarised below in Table 1. We present the results of two runs, both based on using the full topic description but the first run (**DCU97lnt**) used no thresholding while the second run (**DCU97lt**) used the query term, postings list and accumulator threshold settings used in TREC-5. These TREC-5 settings were used blind in TREC-5 and were based on runs on the TREC-3 and TREC-4 data set. The settings we used in TREC-6 are similarly blind, i.e. they remain untuned to the data set. Comparing these two results we see that the thresholding has an enormous improvement on retrieval effectiveness, probably because of the number of noise words from the full topic description that are discarded by the query term thresholding.

The results for the other two runs we submitted in the ad hoc retrieval task are also shown in Table 1 and are based on using the short version of the topics (description only) in run **DCU97snt** and based on using the topic titles only in **DCU97vs**. Both of these latter runs are done with no thresholding used. In later work to be reported elsewhere we shall explore how applying our best thresholding settings impacts retrieval effectiveness and execution speed on these runs.

	DCU97lnt	DC97lt	DCU97snt	DCU97vs
	Full topic, no thresholding	Full topic, thresholding	Description only, no thresholding	Title only, no thresholding
No. relevant documents found	1488	1637	1796	2180
P @ 0.0	0.2384	0.3388	0.5179	0.6016
0.1	0.1143	0.1895	0.2917	0.4086
0.2	0.0646	0.1362	0.2374	0.3343
0.3	0.0513	0.0930	0.1542	0.2570
0.4	0.0391	0.0770	0.1334	0.2232
0.5	0.0222	0.0448	0.1099	0.1830
0.6	0.0163	0.0325	0.0843	0.1421
0.7	0.0095	0.0185	0.0655	0.0988
0.8	0.0053	0.0085	0.0483	0.0554
0.9	0.0016	0.0015	0.0144	0.0341
1.0	0.0002	0.0009	0.0121	0.0325
Avg. Precision	**0.0372**	**0.0696**	**0.1296**	**0.1941**
P @ 5 docs	0.0960	0.1600	0.2800	0.3800
P @ 10 docs	0.0820	0.1460	0.2000	0.3280
P @ 30 docs	0.0687	0.1127	0.1500	0.2553

Table 1: Results for mainline ad hoc retrieval task

3. Character Shape Coding for Mono-Lingual French Retrieval

Character shape codes (CSCs) are a reduced alphabet into which the full range of case-sensitive alphanumeric characters that can occur in printed form in documents, can be mapped. This mapping is based on image processing considerations relative to the task of identifying characters using a kind of OCR process. Essentially, characters with similar "shapes" are mapped into the one CSC and words in the original document are mapped into word shape tokens (WSTs).

Character shape codes were initially developed as a means to identify the language being used in a document. Since then they have been used as a pre-process for full-scale optical character recognition [Spitz, 1997], and for word-spotting from images [Spitz, 1995]. In TREC-5 we attempted to use character shape codes and WSTs for information retrieval, simulating the situation where printed documents would have been scanned and CSC recognition applied to the images instead of full-scale OCR. This work has proceeded with further experiments reported in [Smeaton & Spitz, 1997] showing exciting promise for the technique. The work reported here is an attempt to apply WST-based indexing to French documents. Document texts are turned into their WST equivalents and indexed by these WSTs. Topics are also encoded as WSTs and retrieval is based on matching WSTs instead of word stems.

Several scenarios could motivate CSC recognition including poor quality original documents (Faxes, photocopies of photocopies, etc.) and the need to reduce computing overheads as CSC recognition is an order of magnitude faster than OCR and much more accurate. A number of CSC mappings have been defined but in this work we use the simplest, and thus the fastest and most accurate, known as V_0. To illustrate the mapping, in figure 1 below we have drawn the baseline and the x-line (top of the lowercase character x) on a sample of text.

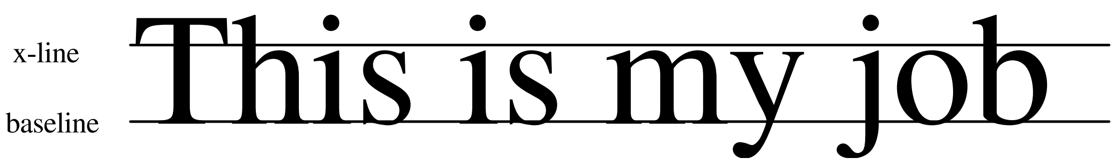


Figure 1: Sample of Word Shape Tokens

Using a few simple rules we can devise a mapping based on whether each character rises above the x-line [A-Zbdfhklt], dips below the baseline [gpqy], does neither [acemnorsuvwxz], does neither but has a dot above it [i] or dips below and has a dot above it [j]. Using the representative letters **A, g, x, i, j** we can represent the text in Figure 1 as Aaix ix xg jxA. To adapt this mapping to the character set used in French, any lowercase accented characters such as é, ì or ô are mapped to the CSC "i" while ç is mapped to a "g". Uppercase accented characters are mapped to "A". Punctuation characters are discarded.

While aggregating similar shaped letters into one does yield a huge loss of information, for English we have found the uniqueness of some WSTs to be surprising, enough on which to develop an information retrieval system based on WST matching and that is what we try here for French.

One of the major difficulties with using WSTs for information retrieval is that each document is represented by the surface form of the word occurrence and variations in surface form (case mixing) and in word morphology (plurals, word endings) must be taken into account at the query processing stage. Thus when a user inputs a search term, all legitimate variations of that term due to case changes and word morphology must be generated at query time and used as search terms. In TREC-5, working on English texts, we found that approach to generate many search terms whose surface form is shared by many other surface forms and this is effectively adding many noise terms to the query. As an extreme example, the word forms *Pommes, terres, Mesure* all generate the WST Axxxxx as do 1,285 **other** surface forms we came across while the surface forms *terre* and *luxes* generate Axxxx which is shared by 1412 surface forms in total that we know of. To compensate however, sometimes the mapping can be surprisingly unique; the words *religion, autrichian, recyclage* and *automobiles* all have unique WSTs in the texts we processed

In order to generate the morphological variants of words in the topics, the topics were linguistically analysed by the Xerox processor at the Rank Xerox Research Laboratory in Grenoble and for the base form of each words in the topic descriptions, morphological variations were generated. These were then post-processed to generate surface variations such as starting capital letters if they could have occurred as the first word in a sentence, and then these terms were turned into their WST form thus creating our topics. As a sample text corpus from which to generate frequency statistics, approximately two-thirds of the document texts were processed to record each unique surface form occurrence and the number of times that form occurred in the sample texts. This yielded 128,380 unique surface forms, a good deal short of the c.300,000 we have for English texts but sufficient to work with. Many of these surface form occurrences would be spelling errors but we did not clean this list in any way. Content-bearing terms were manually identified in each of the topics and all the surface forms associated with those content-bearing terms were concatenated to constitute our starting point for our topics.

Rather than simply turn these word lists into WSTs and run them against the documents, which we realised from TREC-5 experiments would not be effective, we submitted two runs for assessment. In the first run (DCU97Fv1, which was not judged) we used only the WSTs derived from topic tokens where the WST was shared by no more than 10 other surface forms in the list of 128,380 forms we had recorded. In some post-TREC-5 experiments on English texts we found this to be crude but surprisingly effective [Smeaton & Spitz, 1997]. In our second run (DCU97Fv2 which was judged) we manually selected WSTs for each topic based not only on the number of surface forms sharing a WST but on the importance of that WST to the topic. For example, topic CL9 (not our best-performing topic) is entitled “Les effets de déforestation” and the tokens (and the total number of other word tokens sharing that WST) are:

effet (3) effets (1)	Effet (18)	Effets (10)	déforestation (1)
déforestations (1)	désertification (1)	désertifications (0)	frêner (177)
changement (4)	changements (3)	climat (5)	Climat (8)
épuisement (1)	épuisements (0)	terre (1412)	terres (1288)
inondation (3)	inondations (1)	ouragan (41)	ouragans (25)

Clearly words like *terre* and *frêner* should be eliminated and pruning such terms based on WST frequency would not hurt this query too badly. This would have corresponded to the tokens used in run DCU97Fv1. For our second run, DCU97Fv2, we would have judiciously omitted some terms but left others in, using the number of shared WSTs per search term as an **indicator** rather than the deciding factor on whether to include a search WST or not. For example, the terms *changement* and *changements* would have been excluded even though the uniqueness of their WSTs suggest they be left in. In fact, for this query, such adjustments did improve the performance of the topic. Tokens with zero above obviously did not appear in the sample of document texts we used to record possible WSTs.

To illustrate a case where simple pruning based on frequency does harm a query, if we look at topic CL11 entitled “Le coton écologique” we see the following terms and their frequencies:

coton (135)	Coton (616)	écologique (2)	production (10)	usage (34)
usages (64)	bénéfice (19)	bénéfices (10)	positif (2)	positifs (1)
positive (5)	positives (6)	terre (1412)		

Eliminating terms with frequency >10 leaves no content-bearing terms at all and indeed this topic retrieves no relevant documents at all for us in the top-1000 out of the 8 known relevant topics in the corpus.

For the record, the number of WST terms included in the topics for the v1 run (automatic pruning based on frequency) was 10.0 on average while the manually chosen WSTs in the v2 run yielded an average of 7.8 terms per topic. The original set of surface form occurrences chosen manually as content-bearing yielded 17.8 surface forms per topic. In terms of performance, the precision-recall figures for our runs using the 21 judged topics, are given below in Table 2.

	DCU97Fv1 (automatic terms)	**DCU97Fv2 (manual choice)**
R @ 0.00	0.3833	0.3580
0.10	0.2374	0.2471
0.20	0.1982	0.1940
0.30	0.1497	0.1657
0.40	0.1238	0.1322
0.50	0.1029	0.1067
0.60	0.0673	0.0702
0.70	0.0607	0.0600
0.80	0.0413	0.0388
0.90	0.0366	0.0377
1.0	0.0085	0.0087
Avg P	**0.1073**	**0.1102**
P @ 5 docs	0.1905	0.1905
P @ 10 docs	0.2143	0.2095
P @ 30 docs	0.1698	0.1730

Table 2: Performance of WST-based French mono-lingual retrieval

These results are poor in terms of precision-recall, especially averaged over the whole range, though higher precision is what we would have hoped for in terms of overall performance. The reader is reminded that we are simulating a retrieval situation in which scanned images of documents are retrieved based on the shapes of words occurring within them, albeit by a 100% accurate CSC recognition process.

In terms of comparison to submitted runs by other groups we have one topic which is the median, two topics (three in the case of V1) which are above median and the remainder of the 21 are below the median (as measured by average precision. However, WST-based retrieval works reasonably well for some topics, but terribly badly for others, even with the manual selection of WST search terms. The graph in Figure 2 shows the average Precision for each of the judged topics for each of our runs and also the average Precision per topic for all the submitted runs by all participating groups. This average Precision across all submitted runs gives an indication of the degree of difficulty of each topic. Figure 2 clearly shows that for some topics we perform reasonably well compared to others (topics 1, 6, 9, 12, 16 and 20) but there are others where we perform badly, even with manually selected WST search terms, and where the topic is not so difficult (topics 5, 7, 8, 10, 13, 14, 17, 18, 19, 21, 23 and 24) and it is the performance of these latter topics that brings down our overall performance figures.

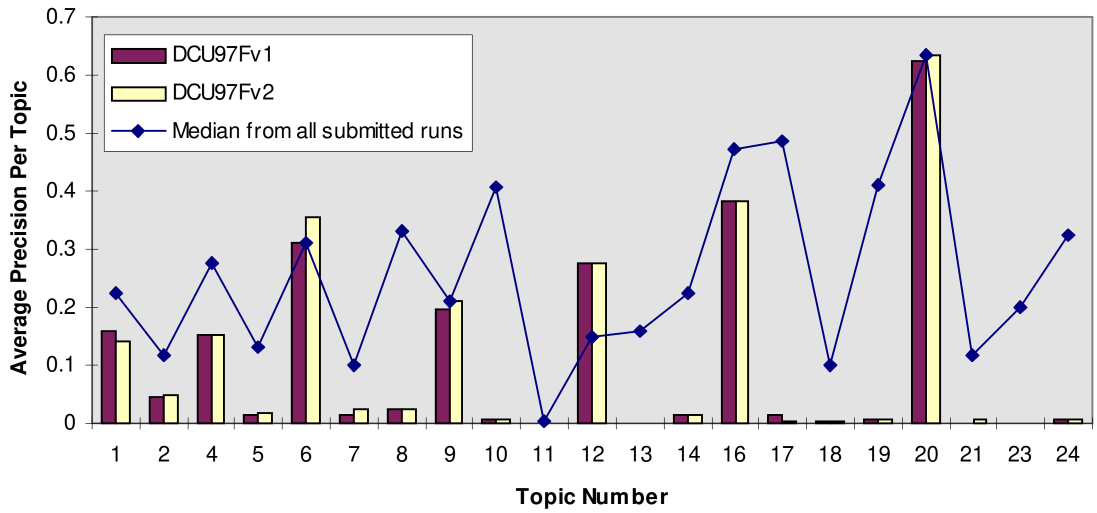


Figure 2: Avg Precision per topic for DCU runs and Median from all submitted runs

What is somewhat surprising in our results is the fact that v2 (manual pruning of search terms) is not considerably better than v1 (automatic pruning). We would have expected that manually choosing search terms based on WST frequencies and on importance to the query would have been much more effective than doing so by using an automatic frequency threshold, though perhaps this may be due to the small sample size (of topics evaluated).

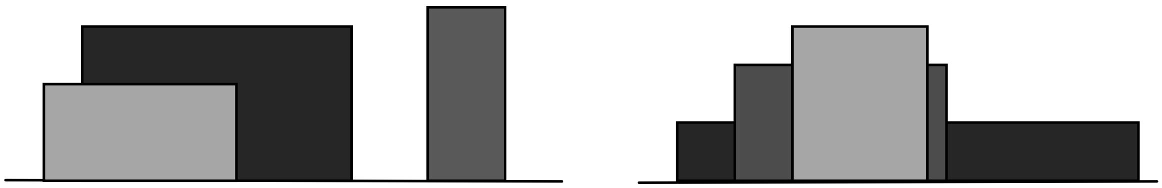
We are encouraged to continue with this work, though for English rather than for French and the progression of this work will be reported elsewhere.

4. "Document-at-a-Glance" for High Precision Retrieval

Several approaches have been developed within the last few years for presenting the results of a ranked retrieval of documents graphically. These have included techniques based on sophisticated visualisation such as LyberWorld [Hemmje et al, 1994] and Cat-a-Cones [Hearst & Pedersen, 1997] as well as those based on more simple graphical icons like Tilebars [Hearst, 1995] and the work by Veerasamy [Veerasamy & Heikes, 1997]. At Dublin City University we have developed a graphical iconic representation which has similar features to Tilebars and the work by Veerasamy. We call our representation the "Document-at-a-Glance" (DaaG).

A DaaG is an icon where the horizontal axis corresponds to the length of the document. Marked on this horizontal axis are the location of the first and the location of the last occurrence of each of the search terms which occur within that document. Using these "first" and "last" offsets, a rectangle can be drawn whose "height" is some function of the number of times that search term has occurred in the document, and the IDF weight of that search term. Clearly, the higher this "box", te more important in terms of weight and/or number of occurrences that search term is to that document. Such rectangles are drawn for each search term that occurs in each of the top-ranked documents and these are laid out in such a way that all are visible. The boxes are coloured and the colours match the colours assigned by the system to each of the search terms in the topic.

For example, consider the query "computer science education" and the hypothetical DaaGs generated for two documents as shown below.



where the "colour" codes[1] correspond to computer science education

If we assume that the RSVs for the two documents are about the same then clearly the document on the left is about computer education but not about computer science whereas the document on the right is more likely to be about computer science education because those three terms co-occur in the same parts of the document. With the Document-at-a-Glance icon, a user can have such a summary of search term occurrences generated visually. In our implementation we generate DaaGs for all top-ranked documents and let the user see these before deciding whether a document text is worth the effort of looking at. Double-clicking on a DaaG opens that document as text in a separate WWW browser for viewing. Users are not presented with document titles but are allowed to view the DaaG representation, or the full document.

We indexed the document collection by word stems or statistical phrases as documented in our TREC-5 mainline submission and provided a ranked retrieval as described in section 2 above. For the high precision track runs we used no query term or posting list thresholding. The user entered a query into a text box and for these experiments this consisted of whatever search terms the user thought appropriate. The search term weighting was tf*IDF. A WWW interface to our retrieval engine was developed for this track and the code to create a DaaG as the result of running a query, was developed. Encoding a DaaG was not as much of an overhead as might initially seem since our inverted file entries store, for each set of index term occurrences in a document, the location of the first occurrence, the location of the last occurrence, and the number of occurrences, as measured using non-stopword term offsets. These 3 values are compressed to fit into 4 bytes of storage and the file of index term occurrence offsets is used as a "shadow" file on the main inverted file [Kelledy, 1997]. This has the advantage of not doubling the size of the raw inverted file unnecessarily, and this positional information can be used or ignored without effecting retrieval performance.

The code to generate a DaaG was written in Java and uses widgets from the Java Development Kit (JDK) 1.1 in order to present the results. At the time of running these experiments, neither the then current versions of Netscape not Internet Explorer supported this so we used the HotJava browser. When loading the URL into HotJava, an applet is launched which invites the users query into a text box. When the SUBMIT button is clicked, this query text is sent back to a server program running at DCU which calls the search engine to run the query. The resulting ranked list of document identifiers and the positional information of search terms in each of the top ranked documents, is

---

[1] Naturally, for the version we used in experiments these were real colours, not greyscales.

then sent back to the applet for display. A screendump of a mid-search is shown below in Figure 3.

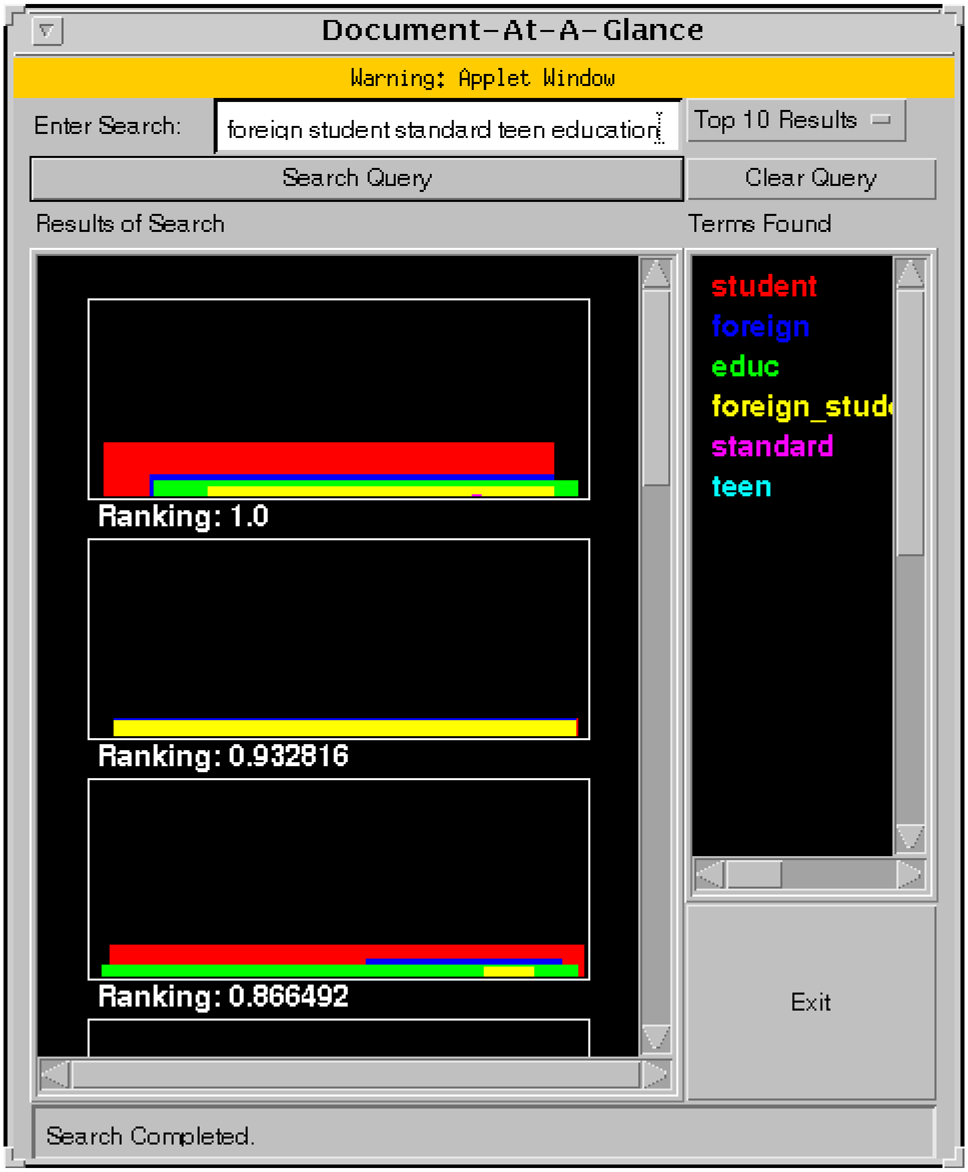


Figure 3: Sample DaaG

Here we see the user's query, which is for topic 346 has generated a ranking and we are looking at the DaaGs for the first 3 documents in the ranking. The user's search terms are "foreign student standard teen education" and the phrase recogniser has picked up "foreign_students" as a phrase. The top-ranked document (or more correctly, a fixed-length window in this top-ranked document) has occurrences of the term student spanning the start of the document through to almost the end and the height of its rectangle (the one at the back) in the DaaG shows it contributes most to the score of this document. For this document the terms contributing to the RSV are, from the back of the Daag to the front, student, foreign, education, foreign_student and there is a tiny occurrence, probably just once and with a low IDF weight, of the word standard about three quarters through the document. For the document which is third in the ranking, the scoring is again dominated by the contribution of the term student and then education with the terms foreign_student and foreign occurring towards the end of the document. A user would click on this DaaG at some point towards the end of the document and it would be loaded at that position in the document viewer.

We used one subject for our high-precision runs and she ran all of the 50 topics over a period of about 4 days, using only 5 minutes but spacing out the sessions for sanity. The subject was an undergraduate female student (economics major) in her early 20s, with no experience of information retrieval or of using, let alone searching, the WWW, a real naïve user ! She was given a 15 minute overview of the system and its interface during which she ran some dummy searches to become acquainted with the system.

The retrieval results we obtained in our single submitted run in this track were as follows, averaged over all 50 topics.

Avg. Precision @ 10 docs	0.3820
Relative Precision @ 10 docs	0.4031
Unranked-Avg-Precision @ 10 docs	0.0633

We analysed how many relevant documents were found in each of the 50 topics and found the following distribution:

No. Reldocs found	10	9	8	7	6	5	4	3	2	1	0
No. topics	1	3	2	5	4	6	6	4	4	5	10

This table showed that there were 10 topics (of 50) where we retrieved no relevant documents in our top-10 selected. In looking at the output of the search in order to select documents for retrieval, the user examined an average of 30.7 DaaGs. We did not record how many documents were actually viewed but it was somewhat less than this.

In general, these results look poor but like the other 4 participants in this track, we were there to learn. Some of the things we learned were that we need to use more than one subject as she became disenchanted (or brain-dead !) with the task, and we need to use subjects who are more familiar with the technology. While it is a nice idea to use a complete novice, in today's world of information retrieval, web searching has made the user more IR-savvy and the more sophisticated the user, the more we feel our DaaG is appropriate as a tool. Another thing we learned were changes to make to the software to improve it. While it did not crash, because it has to be run from the HotJava browser it is slow and unwieldy to move windows about and to scroll. We created DaaG icons for the top-10 documents at a time and had these concatenated together as a palette which was scrollable, but given the computing environment, slow to refresh. For the next version we will load a DaaG icon onto a fixed place in a window as a document is highlighted. Another feature we will replace is the display of the title of a document, which was missing from the experiments reported here. We were surprised how information-rich the title of a document can be. The final change we will make to the software is to improve the document loading speed, which was loaded as an HTML file in a browser, and was slow.

Our future work with the Document-at-a-Glance will continue once an updated version has been completed. For evaluation however, we will remember that we will be evaluating the DaaG, not the whole system. Thus our future evaluations will compare performance against the top 10 ranked documents and against a retrieval of the best 10 based on previewing documents and their titles only.

5. Triphone Retrieval of Spoken Documents using Fixed Windows

The objective of our participation in this track was to develop the best input parameters to use in a related project we are working on with Ireland's national radio broadcasters - RTÉ. In this related project we are developing a system to search an archive of radio news broadcasts and we use the opportunity of this track to determine parameters related to our search implementation. Our initial searching technique on the RTÉ archive of audio will be on a stream of audio with no story boundaries available. For this reason we decided to do treat the QSDR transcripts as a live stream of text and we removed all story boundaries from the TREC data thus making our task more difficult but closer to reality.

All the story boundaries and tags from the baseline (.srt) and reference (.ltt) transcripts were removed. These raw text files "streams" were then broken into overlapping windows or "documents" of fixed sizes and fixed amounts of overlap. A weight was assigned to each word in the documents based on location within these windows. The identifier of each window was a combination of the file name and percentage of window offset into the transcript eg the document b960610a_23.4. is 23.4% into the file b960610a. The individual windows were indexed by the terms occurring within them. Topics were matched against these fixed-sized windows to find the highest-scored windows using the retrieval engine from our mainline ad hoc retrieval submission which is based on tf*Idf weighting of terms. The document ID of these windows in our ranking were then used to search back into the original baseline or reference transcripts, where a seek of x% was made into the file (percentage in the document identifier). A search was then made backwards to find the first occurrence of a story identifier which was then returned in the ranking as the result.

Our retrieval of RTÉ news broadcasts is based on indexing the audio stream by phoneme units aggregated into triphones[2]. To simulate this in the TREC QSDR track we transcribed the words (from reference and baseline texts) into triphones using a pronunciation dictionary of 160,000 terms in the format of 'computer' represented as *`k ax m p y uu t axr'*. This pronunciation dictionary had been adapted for the RTÉ project by replacing North American pronunciations by Hiberno-English equivalents (colleagues from North America who have participated in previous TRECs will recall the different pronunciations of the term "routing"). Again we broke the "stream" (without story boundaries) into overlapping windows of triphones and indexed these windows by weighted triphones based on their offsets within windows. Topics were also turned into triphones and matched against windows and the stories in which these occurred were returned in the same way as described above.

[2] Our definition of a triphone is of a concatenation of three phones where phones are taken from an alphabet of 41 possible phones

Table 3 shows the combinations of window sizes in words and in triphones, as well as the overlap between these windows, in our submitted runs, on the reference and on the baseline texts.

Text	**Run**	**Representation**	**Window Size**	**Overlap**	**Weighting**
Ref	DCU97QSDR-R1	Words	90 words	30 words	Yes
Baseline	DCU97QSDR-B1	Words	90 words	30 words	Yes
Ref	DCU97QSDR-R2	Triphones	120 triphones	40 triphones	Yes
Baseline	DCU97QSDR-B2	Triphones	120 triphones	40 triphones	Yes

Table 3: QSDR Track Runs

Run:	DCUSDR-R1	DCUSDR-B1	DCUSDR-R2	DCUSDR-B2
Mean rank:	9.91	11.80	18.04	14.97
Mean recip:	0.5196	0.5480	0.5022	0.4287
<= 5	34	37	32	28
<= 10	37	39	35	28
<= 20	40	40	39	30
<= 100	46	44	43	35
Not found:	3	4	4	12

Table 4: QSDR Track Results

Overall the runs seemed to produce fairly respectable results. For the reference transcript our average rank on words was about mid-table but on triphones we were almost worst (we were 18.04, max was 18.12). For the baseline recognised text our average rank was 11.8 for word-based and 14.97 for triphone based while the best was 10.11, the median was 17.96 and the maximum was 36.06. It seems therefore that our approach performed better, relatively speaking, on the baseline recognised text with all its inherent noise. Our system failed when there were a lot of short stories close to each other in the transcript as in these cases the wrong story identifier may have been returned instead of the story above or below the correct story or even two stories above. This happened because the actual file space taken up by the tags weren't taken into consideration when the percentages offsets into files were being calculated, i.e. % of how far windows were into a file, and hence when we searched the transcript files we weren't seeking to the exact location of the window and this sometimes caused the wrong story to be returned. This happened in a few cases.

A more significant problem with our approach was that our system didn't do any query expansion on the topics, and there are some topics where the relevant story and the topic texts have no words in common, e.g. topic number 3 Clearly this is something to be considered for future work.

We are encouraged with the results obtained and will proceed to evaluate other window and overlap sizes as well as other weighting functions which consider word/triphone offsets within windows.

6. Overall Conclusions

Dublin City University was delighted to participate in TREC-6 following 4 different lines of action. While somewhat stressful and a bit manic coming up to submission time, we believe that participation in TREC has focused and pushed our work. As a result of this year's participation we are encouraged to continue with the development and evaluation of our document-at-a-glance, the performance figures we obtained in our submissions to the spoken document retrieval track will help us in our own work on retrieval from radio news, and we will return to WST-based retrieval on English texts.

References:

[Hemmje et al, 1994] "LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval", M Hemmje, C Kunkel and A Willett, *in Proceedings of SIGIR'94, Dublin*, July 1994, 249-

[Kelledy, 1997] "Query Space Reduction in Information Retrieval", F. Kelledy, PhD thesis, Dublin City University, February 1997.

[Hearst, 1995] "Tilebars: Visualization of Term Distribution Information in Full Text Information Access", M.A. Hearst, *in Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, CO, May 1995.

[Hearst & Karadi, 1997] "Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy", M A Hearst and C Karadi, in*: Proceedings of ACM SIGIR Conference, Philadelphia*, July 1997, pp.246-

[Smeaton & Kelledy, 1997a] "TREC-5 Experiments at Dublin City University: Query Space Reduction, Spanish & Character Shape Encoding.", A.F. Smeaton & F. Kelledy, *in Proceedings of TREC-5,* NIST (in press), 1997.

[Smeaton & Kelledy, 1997b] "Improving Information Retrieval Efficiency by Search Space Reduction", A.F. Smeaton & F. Kelledy (in preparation).

[Smeaton & Spitz, 1997] "Using Character Shape Coding for Information Retrieval", A.F. Smeaton and A.L. Spitz, *in Proceedings of the 4th International Conference on Document Analysis and Recognition, ICDAR'97, Ulm, Germany*, August 1997, 974-978.

[Spitz, 1995] "Using Character Shape Codes for Word Spotting in Document Images", A L Spitz, *in: Shape, Structure and Pattern Recognition,* D.Dori and A Bruckstein (Eds)., World Scientific, Singapore, 1995

[Spitz, 1997] "Moby Dick meets GEOCR: Lexical Considerations in Word Recognition", A.L. Spitz *in: Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR'97),* Ulm, Germany, August 1997, 221-226.

[Veerasamy & Heikes, 1997] "Effectiveness of a Graphical Display of Retrieval Results", A Veerasamy and R Heikes *in: Proceedings of ACM SIGIR Conference*, Philadelphia, July 1997, pp.236-

Acknowledgement: Part of the work reported here was carried out while GQ was in receipt of funding from FORBAIRT under grant ST/96/707. We also acknowledge support from Greg Grefenstette and the Rank Xerox Research Centre (Meylan, France) for linguistic processing of French topics. Larry Spitz continues to provide advice to us on the subtleties associated with character shape coding.

Appendix I: CLIR TRACK QUESTIONNAIRE:

1. OVERALL APPROACH:
2. MANUAL QUERY FORMULATION:
3. USE OF MANUALLY GENERATED DATA RESOURCES:

None of these questions are applicable to our submission.

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES:

4.1 Form of the automatically constructed data resources?
[x] Lexicon … <u>simple wordlist of word form occurrences</u>

4.2 What sort of training data was used to construct them?
[x] Same data as used for searches, <u>subset of data used for searches, about 180 Mbytes.</u>

4.3 Size
[ ] 128,380 entries
[ ] 7 MBytes

4.4 Was there any manual clean-up involved in the construction process?
[x] No <u>clearly benefit could be obtained from cleaning up the wordlist as we have shown for English</u>

4.5 Rough resource estimates for building the data resources (ie. an indicator of the computational complexity of the process).
[ ] __1 hour______________________ hours
[ ] __50 Mbytes disk _______________ MBytes of memory used
[ ] __50 Mbytes ___________________ temporary disk space

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[x] Easily replaceable, ________________________

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?

[x] Yes, somewhat, a larger lexicon created from a larger sample of French texts

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?

[x] Yes, somewhat,

5.4 Are similar resources available for other languages than those used?

[ ] Yes, we have a similar wordlist for English but much more extensive (300k entries)