Filters and Answers: The University of Iowa TREC-9 Results

Elena Catona[1], David Eichmann[2,3] and Padmini Srinivasan[1,2]

[1]Department of Management Sciences
[2]School of Library and Information Science
[3]Computer Science Department
The University of Iowa
{elena-catona, david-eichmann, padmini-srinivasan}@uiowa.edu

The University of Iowa participated in the adaptive filtering and question answering tracks of TREC9. The filtering system used was an extension of the one used in TREC-7 [1] and TREC-8 [2]. Question answering was done using a rule-based system that employed a combination of public domain technologies and the SMART retrieval system.

1 – Adaptive Filtering:

Our approach to filtering involves a two-level dynamic clustering technique. Each filtering topic is used to create a primary cluster that forms a general profile for the topic. Documents that are attracted into a primary cluster participate in a topic-specific second level clustering process yielding what we refer to as secondary clusters. These secondary clusters, depending upon their status, are responsible for declaring, i.e., retrieving, documents for the topic.

As documents are temporally processed they are attracted to a primary cluster if their similarity with the cluster vector is above a primary threshold. These documents enter the secondary clustering stage where again, based on similarity to cluster vectors and a secondary threshold, they either join an existing secondary cluster or start a new one. If at some point the similarity between a secondary cluster and the primary cluster exceeds a third declaration threshold then the document most recently added to the secondary cluster is retrieved for the user.

When deriving representations we use TF*IDF weights after stemming the terms using Porter's stemmer. We also limit document vectors and cluster vectors to the best 100 and 200 stems respectively.

In TREC-8 adaptation was explored at several different levels [2]. First a secondary cluster's future behavior would depend upon past performance. If a secondary cluster declares a document that turns out to be relevant then it is colored green. This means that it declares all documents that join it in the future. If instead the declared document is non relevant then the cluster is colored red and all future documents are not declared. A non relevant document that joins a green cluster spawns an independent red cluster allowing the original cluster to remain green. Another adaptive dimension was to have the primary cluster vector adapt as relevant judgements were obtained. A version of Rochio's feedback method is built into the system for this purpose. A differential adaptation scheme is also built in for this purpose. The key distinction is that in the differential scheme positive and negative term vectors are comprised only of terms not found in the other vector or in the original query vector.

Recent experiments conducted with TREC-8 data explored additional dimensions of adaptation. For example, we experimented with adapting the primary threshold as the performance measure varied. For this, the performance measure such as the utility score was computed at regular intervals when a "snapshot" of the system is taken. We also explored adaptation of secondary and declaration thresholds. In all these the most profitable approach appears to be adaptation of the break threshold using a step function that responds to changes in performance across snapshots. Our OHSU runs use the system as described above with adaptation of the break threshold.

Other key extensions to the system for TREC-9 include the ability to specify the type of index vectors to utilize. A phrase recognizer loads dictionaries of phrases derived from sources such as the WordNet thesaurus and matched phrases are included into the document vectors. More recently a rule-based entity recognizer has been developed that allows the indexing of documents by person names, organizations, locations and events. Our MESH run includes this technology as well special support for medical terminology. The MeSH hierarchy, an associated lexicon of synonyms and a supplementary list of concepts such as drug names were used. The MESH run involved index vectors that were populated using only the entities extracted from the source text.

OHSU Runs:

For TREC-9 we submitted two OHSU runs. These runs employed word based indexing, Rochio feedback for the profile adaptation and adaptation of the declaration threshold. Both OHSU runs used the controlled vocabulary field (MeSH terms). The two runs differ only in their starting threshold values. The primary, secondary and declaration thresholds were 0.3, 0.32, and 0.3 respectively for OHSU1 and 0.25, 0.27, and 0.25 respectively for OHSU2. The declaration threshold was adapted in each case using a step-wise strategy. Figure 1 shows the performance in terms of utility for our OHSU1 run. The dashed bars represent median performance across systems for each topic. There are 24 topics for which OHSU1 was better than the median and another 24 for which it was below the median.

We conducted several experiments after the official submission deadline to better understand the different aspects of our filtering system and its weak performance on the OHSU task. The first question asked was whether the primary filter was effective. In other words how good was it at filtering out non relevant documents while allowing through the relevant documents? Figure 2 shows the percentages filtered through over time, with snapshots taken every 1000 documents. The figure shows that if we divide the snapshots into three groups then the primary filter allows about 50%, 60% and then 59% of the relevant documents that arrive over the first, second and third sequence of snapshots respectively. At the same time the percentage of non relevant documents allowed through stays less than 1% of the number seen. We then examined the effectiveness of the secondary filter. Note that this analysis of the secondary filter was limited to those documents allowed through by the primary filter. Figure 3 shows that the secondary filter was successful in reducing the percentage of non relevant documents allowed through (dashed bars). However, at the same time it also restricts the passage of relevant documents although not as severely. Next we took a different track in our analysis and examined the effectiveness in adapting the declaration threshold. Figure 4 displays these results. We can observe that if we eliminate break threshold adaptation performance degrades significantly over time (dashed bars). In contrast, the adaptive mode is able to

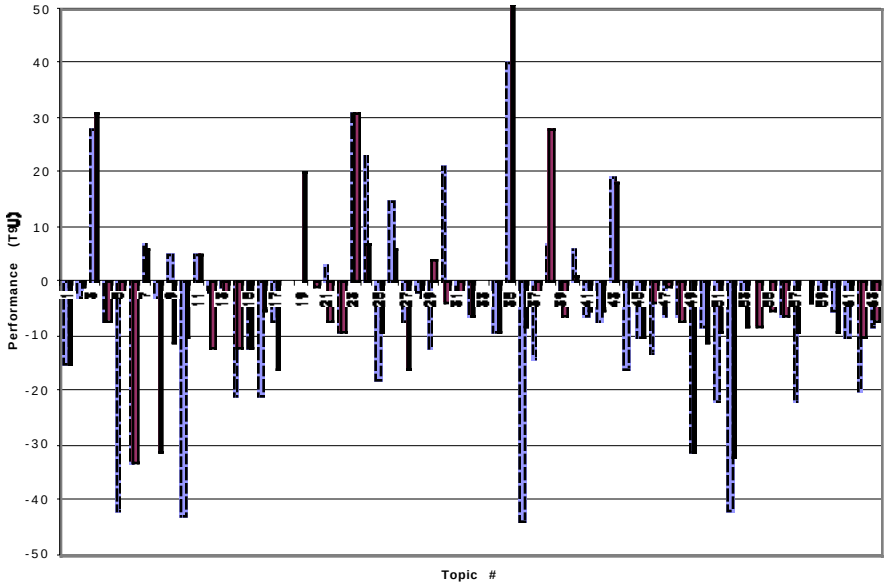

Figure 1: Performance of OHSU1. Dashed bar: OHSU1, Solid bar: median performance

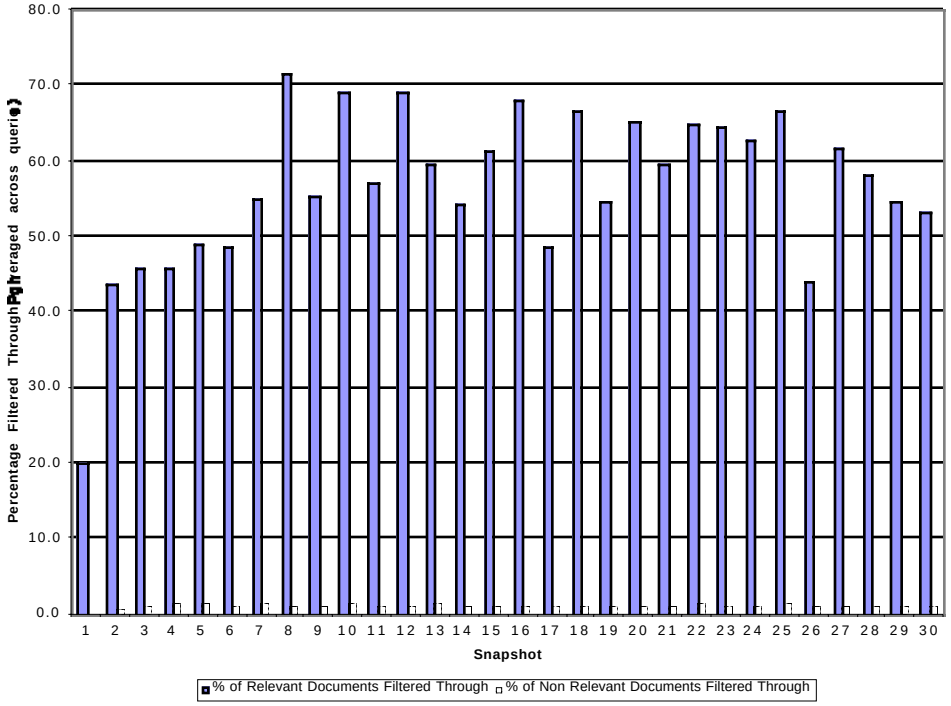

Figure 2: Assessment of Primary Filter

stay somewhat steady - although on the negative side of the performance axis. At this point we suspected that our break threshold may not be restrictive enough. Figure 5 shows the effect of testing this by contrasting a run where the break threshold was increased from the original 0.25 (dashed bars) to 0.3 (solid bars).

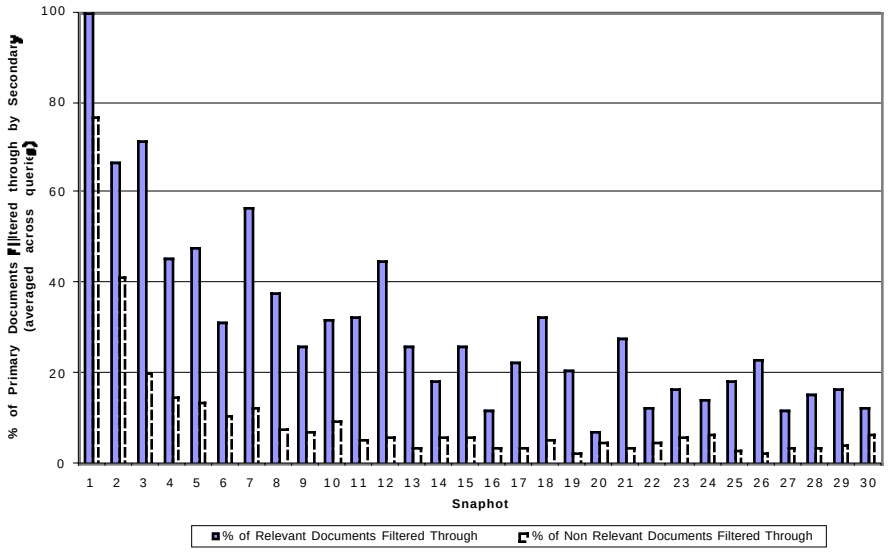

Figure 3: Assessment of Secondary Filter

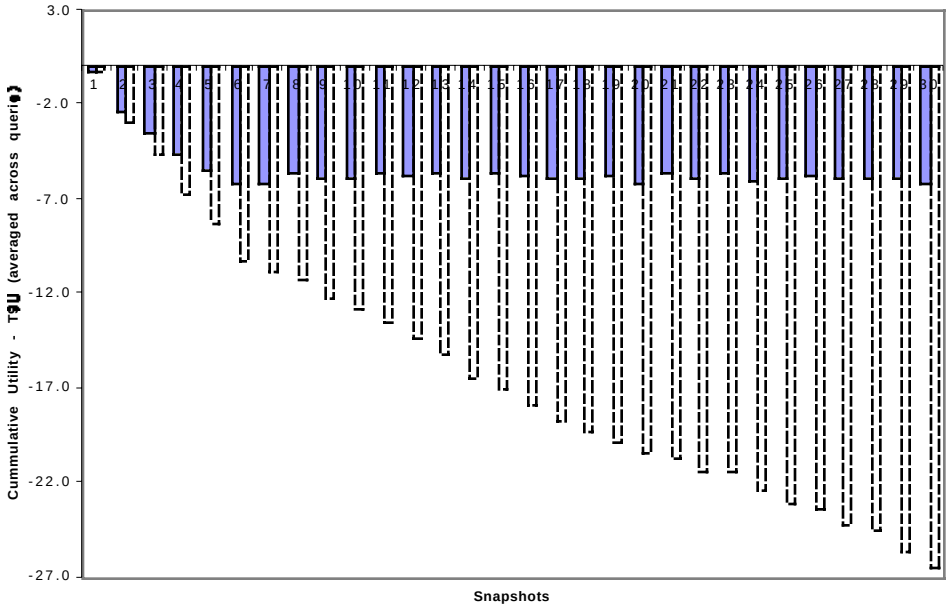

Figure 4: Assessment of Declaration Threshold. Solid bar: adaptive; dashed bar: non adaptive

MeSh Run:

Figure 6 shows the performance in terms of utility score for our MESH run. As mentioned before for this task we employed a rule-based entity recognizer which uses the MeSH hierarchy, an associated lexicon of synonyms and a supplementary list of concepts such as drug names. This run involved index vectors that were populated using only the entities extracted from the source text.

Entity-based performance on the MeSH subset proved to be quite intriguing. In 92 of the topics our system yielded the highest score - in some cases substantially higher than median performance.

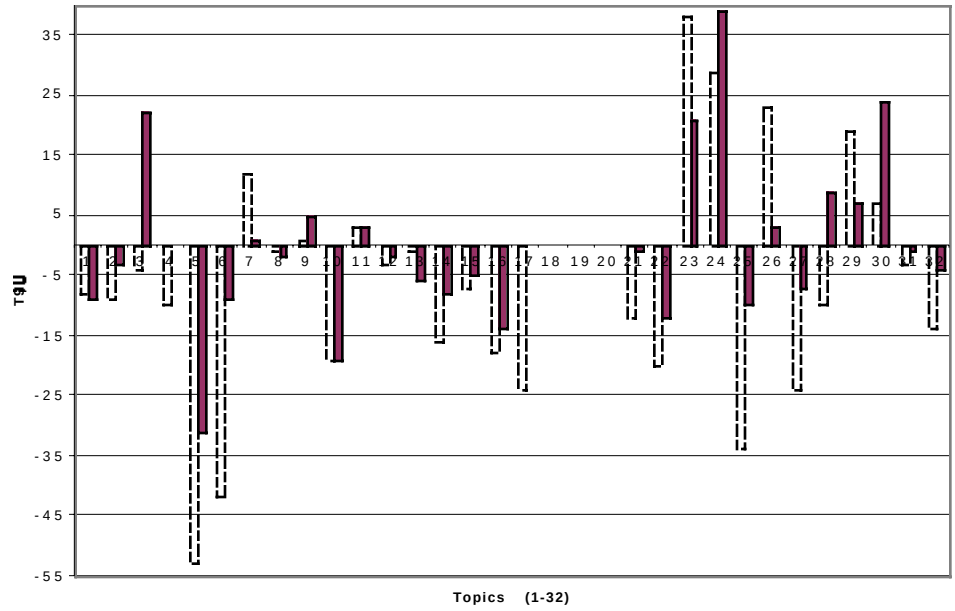

Figure 5: Assessment of Higher Break Threshold.
Dashed bar: 0.25 Solid bar: 0.3

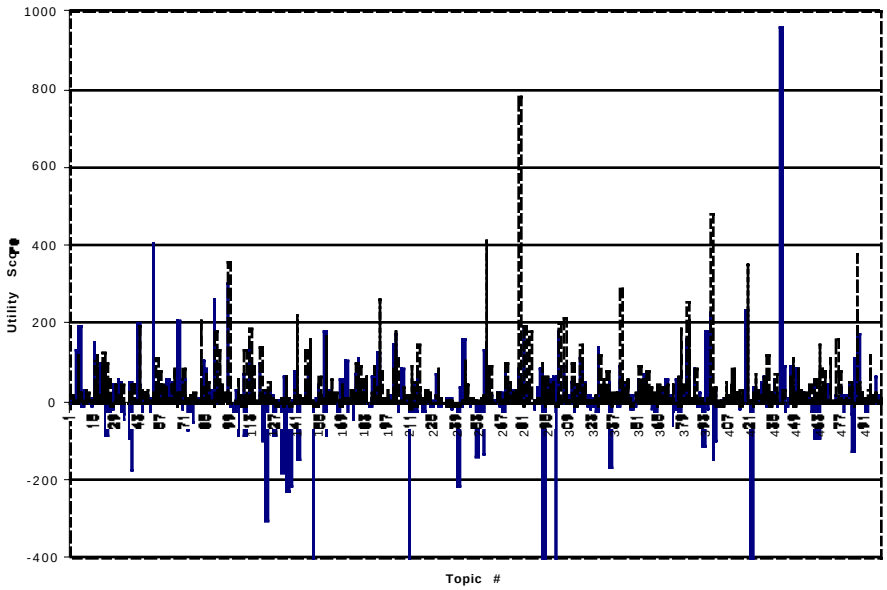

Figure 6: Performance of MESH Run. Solid line: median performance, Dashed line: MESH run.

At the same time, in 147 of the topics our system yielded the lowest score - again in some cases substantially lower than median performance. We conjecture that the pure entity scoring yields high quality results - but for some topics our secondary cluster scheme is generating too many high-relevance clusters that prove to be off-topic. This may be due in part to the score being generated by ancestor/descendant MeSH term tree matches. Figure 7 presents performance (utility score) as

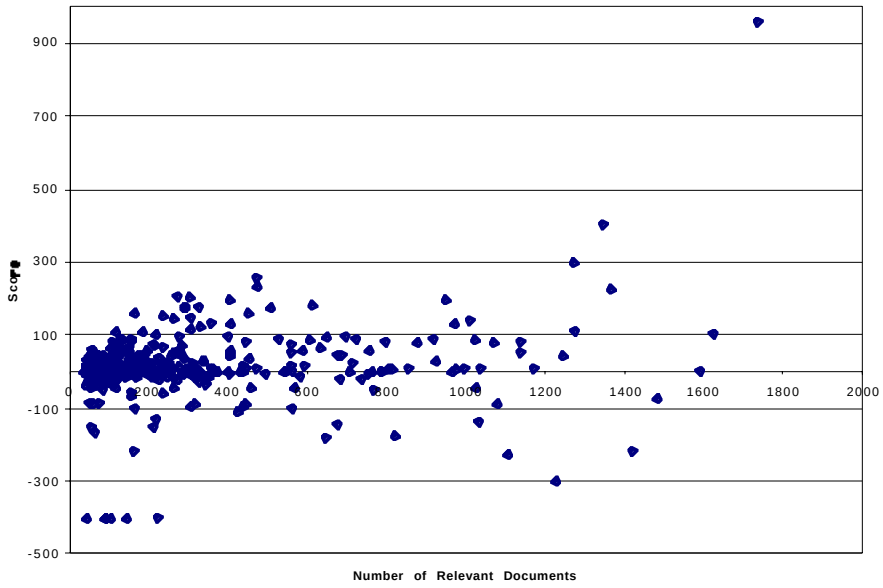

Figure 7: Performance versus Number of Relevant Documents for Topic

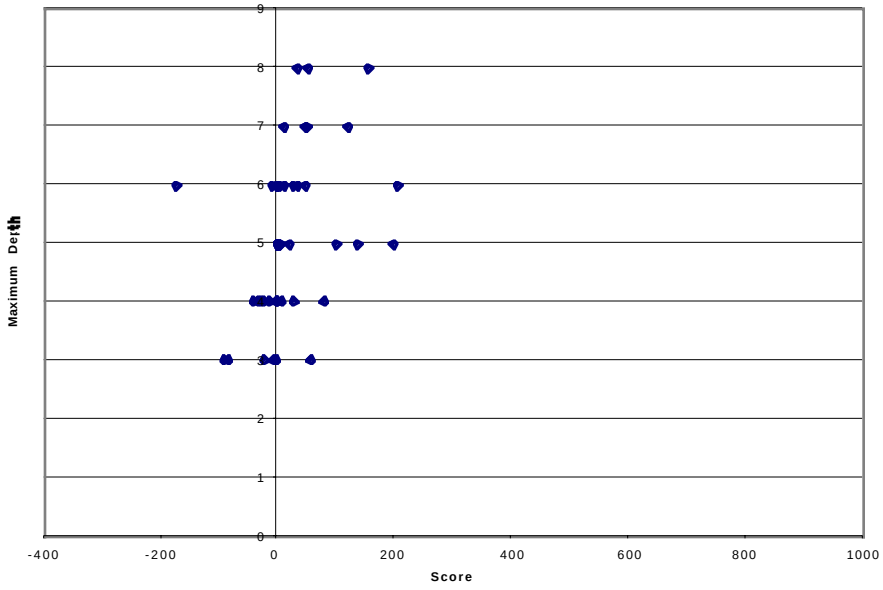

Figure 8: Maximum Depth of Topic's MeSH Phrases versus Performance

a function of the number of relevant documents present for a given topic. The figure shows 500 data points one for each topic. One may observe a general trend that as the availability of relevant documents improves, performance increases. Moreover, most of the scores are on the positive side of the Y axis. Figure 8 explores a different aspect. Our process extracts MeSH descriptors for each topic description from the MeSH hierarchy. In the figure we plot the maximum depth expressed by the group of extracted MeSH phrases for a topic and plot this against utility score. There are 50 data points corresponding to the first 50 MeSH topics. One may observe that except for a few outliers there is a slight trend for scores to improve with the ability to identify deeper i.e., more specific

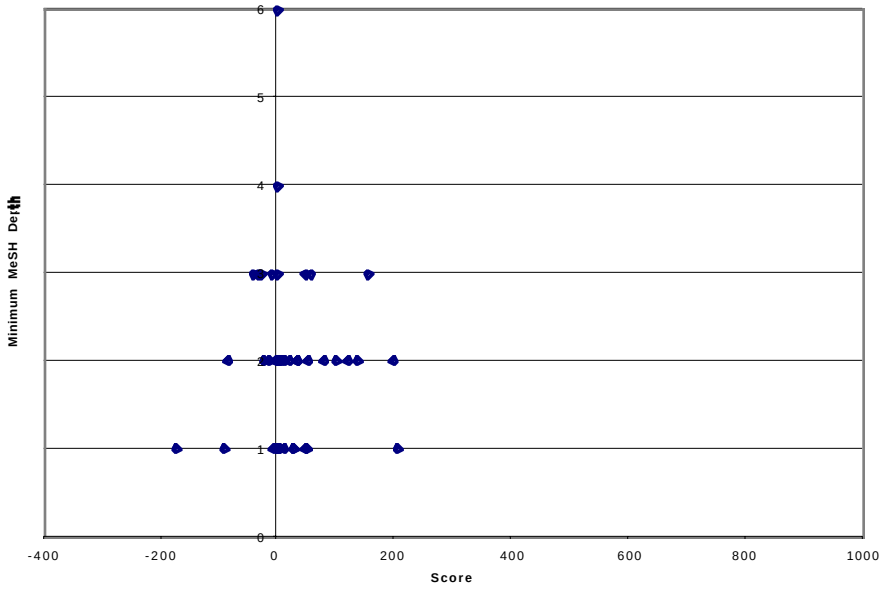

Figure 9: Minimum MeSH Depth for Topic versus Score

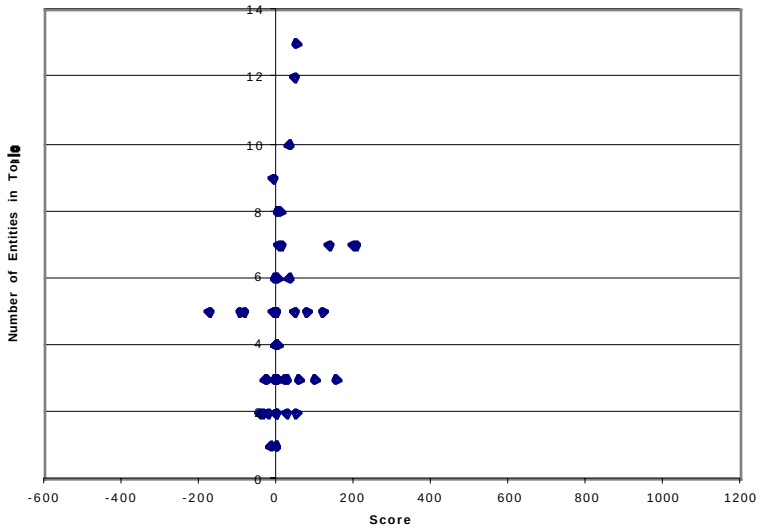

Figure 10: Number of Entities Recognized from Topic versus Performance

MeSH phrases. Interestingly the same sort of analysis using minimal MeSH depth for the topic, as shown in Figure 9, does not yield a recognizable trend. We also explored the effect of entity recognition on performance. Figure 10 represents the number of entities recognized on the Y axis and performance on the X axis. The graph shows that barring a few exceptions there appears to be a slight trend for performance to improve as the number of entities recognized increases.

In summary, the switch in domain from the newswire domain to MEDLINE proved to be challenging. The thresholds used in our submitted run were essentially our best guesses. For the future

we also plan to explore different term weighting strategies as well as query expansion strategies prior to starting the filtering run.

2 – Question Answering:

We submitted two runs for this track, UIQA001 and UIQA002. Both utilized only the top 50 documents that were retrieved and distributed by Singhal. UIQA001 gave the better performance score with mean reciprocal rank of 0.227(strict) and 0.245 (lenient). The other run gave almost identical scores. Our overall QA approach is shown in below.

Document processing:

1. Extract only the textual parts of each document
2. Apply a sentence detection program to identify distinct sentences. We use the publicly available nxterminator program for this.
3. Apply part of speech tagging on each sentence
4. Apply our rule-based entity tagger on each sentence
5. Create a database of sentences formatted for the SMART retrieval system. Here each record corresponds to a single sentence with 3 different fields. The first holds the original untagged sentence, the second field holds the tagged sentence while the last field of the record holds only the particular entities extracted from each sentence.
6. Retrieve the top N sentences for each query. Maintaining the three distinct fields for each sentence record allows us to explore the relative merits of using different types of information for retrieving the sentence most likely to contain the answer. Using SMART allows us to explore different weighting schemes during retrieval.
7. Post process each of the N sentences to extract the top five 250-byte segments.

Query processing:

1. Apply part of speech tagging on each query
2. Apply our rule-based entity tagger on each query. Notice in the case of the query where possible its focus (a specific entity type) is identified in addition to all the entities contained in the query. This focus is utilized during the post processing step in 7 above.

The two runs differ very slightly in the post processing stage. Generally, this step includes cleanup of the sentences to remove any non informative strings, reduction of each sentence to a 250 byte string around the query focus (if known), removal of duplicate answer strings, and selection of the top 5 phrases. The difference between the two is in the extent to which cleanup of the sentences was done. As our results show, this did not influence performance in any way since the two runs yield almost identical results.

Error analysis indicates much room for improvement. Due to insufficient time, we were able to implement only very simplistic 250-byte segment selection strategies that proved to be a significant problem for our system. Secondly, our performance was limited by the availability of the answer within the top 50 document sets distributed. Again with less time pressures we should be able to explore the 1K datasets and also conduct our own retrieval runs for the top 1K or so documents. The results indicate that our approach managed to extract the answers for about 38 to 40% of the questions.

References

[1] Eichmann, D., M. E. Ruiz and P. Srinivasan, "Cluster-Based Filtering for Adaptive and Batch Tasks," *Seventh Conference on Text Retrieval*, NIST, Washington, D.C., November 11 - 13, 1998.

[2] Eichmann, D. and P. Srinivasan, "Filters, Webs and Answers: The University of Iowa TREC-8 Results," *Eighth Conference on Text Retrieval*, NIST, Washington, D.C., November, 1999.