Filters, Webs and Answers:

The University of Iowa TREC-8 Results

David Eichmann and Padmini Srinivasan

School of Library and Information Science
University of Iowa
{david-eichmann,padmini-srinivasan}@uiowa.edu

1 – Introduction

The University of Iowa attempted three tracks this year: filtering, question answering, and Web, the latter two new for us this year. All work was based upon that done for TREC-7 [2], with our system adapted for the specifics of the QA and Web tracks.

2 – Adaptive Filtering Track

Our existing approach to search/filtering involves a dynamic clustering technique where the threshold for formation of new clusters and the threshold for visibility of 'sufficiently important' clusters can be specified by the user when the topic is presented to the system. The TREC requirements for multi-query support and simulation of user judgment responses led us to modify the single set-of-clusters model, creating a two-level scheme. Note that we did not use the controlled-language field in the FT database.

The primary level corresponds to the internal representation of a topic definition. The original threshold specifications were retained here to allow specification of the first-order recall of the system. We experimented with a variety of means of generating a primary similarity measure, but settled on one based upon the text of the topic's concept definitions for the submitted runs.

The secondary level is where the adaptive portion of the system functions and where we found the most benefit in parameter tuning. Each primary cluster (and hence, each topic) has a private set of zero or more secondary clusters. When a document clears the threshold for a primary cluster, it either joins an existing secondary cluster or forms a new one, based upon a membership threshold. The shift from a single membership threshold to a primary/secondary pair allowed us to achieve a tunable level of recall (by using a lower primary threshold, as mentioned above) while teasing out distinctions between candidate document clusters through use of a higher secondary threshold.

Introduction of a declaration threshold for secondary cluster similarity to the primary then gave us a means for adaptation. When a secondary cluster's similarity first exceeds the visibility threshold, its most recently added document is declared to the user and a relevance judgments is obtained. The secondary cluster is then colored appropriately. Secondary clusters containing relevant (and unjudged, if any) documents are colored green and have all subsequent members declared as relevant. Secondaries containing non-relevant (and potentially, unjudged) documents are colored red and declare no further members. A non-relevant document joining a green cluster spawns an independent and new red cluster. Adaptation then occurs over time as secondary clusters

exceed the visibility threshold and are colored, with red secondary clusters mitigating the lack of precision provided by the recall-centric primary threshold.

Secondary clusters exceeding the declaration threshold potentially contain a mix of different document types (relevant, non-relevant and unjudged). We currently address this in the following, conservative manner: if a secondary cluster contains

- the most recent document is relevant, color it green;
- the most recent document is non-relevant documents, color it red;
- fewer than a specific number (currently 10) of unjudged documents and no relevant or non-relevant documents, leave it uncolored until the first relevant or non-relevant document is added, then color it appropriately (note that this optimistic stance has a distinct effect w.r.t. false positives); and finally,
- more than a specific number of unjudged documents and no relevant or non-relevant documents, color it red (we do this pessimistically due to the low density of judged documents in the corpus).

Refinements for this year involved implementation of two primary cluster term adaptation schemes and a phrase recognizer. The first adaptation scheme supported a Rocchio-based weighting of positively and negatively judged documents in calculating the primary similarity. Due to the relatively high density of negative and unjudged documents in the document stream, negative judgments are used in the weighting only in the presence of positive judgements. Positive judgments are always used. The second, 'differential' adaptation scheme is similar to the first, except that the positive and negative term vectors are comprised only of terms not found in the other vector or in the original query vector.

The phrase recognizer loads a dictionary of phrases derived from the WordNet thesaurus and injects matched phrases into the term vectors for queries and documents as they are lexed. The original terms are retained to accommodate partial terminology matches.

We submitted two runs optimized for LF1, IOWAF992 using no phrase recognition and the Rocchio-based weighting scheme (scores shown in Figure 1) and IOWAF991 using phrase recognition and the differential-based weighting scheme (scores shown in Figure 2) The performance of the Rocchio-based approach proved to be surprisingly conservative, adapting well to negative information, but not substantially acquiring relevant documents. (Note that we do not 'turn off' queries – all topics were active for the entire run.) The performance of the differential approach is substantially the same as the Rocchio-based scheme with the exception of a small number of distinctly poorly performing topics. Our current suspicion is that this is due to the appearance in the phrase dictionary of a number of 'stop' phrases (e.g., 'and so on,' 'in point of fact,' etc.) that occur in the text of the topic, similarity was skewed higher for a number of non-relevant documents. We will be experimented with more limited phrase dictionaries as a means of controlling this, as well as with more domain-specific phrase dictionaries.

Hull, in his summary of the TREC-8 systems, computed scaled utility [3]. The scaled utility $u_s^*(S,T)$ accounts for the fact that the topics differ in the number of relevant documents that exist

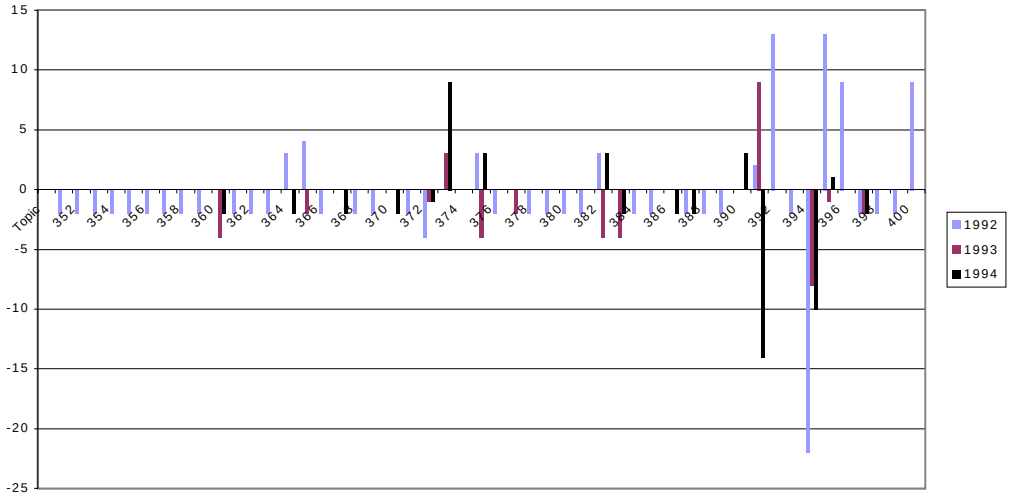

Figure 1: Rocchio Filtering - LF1

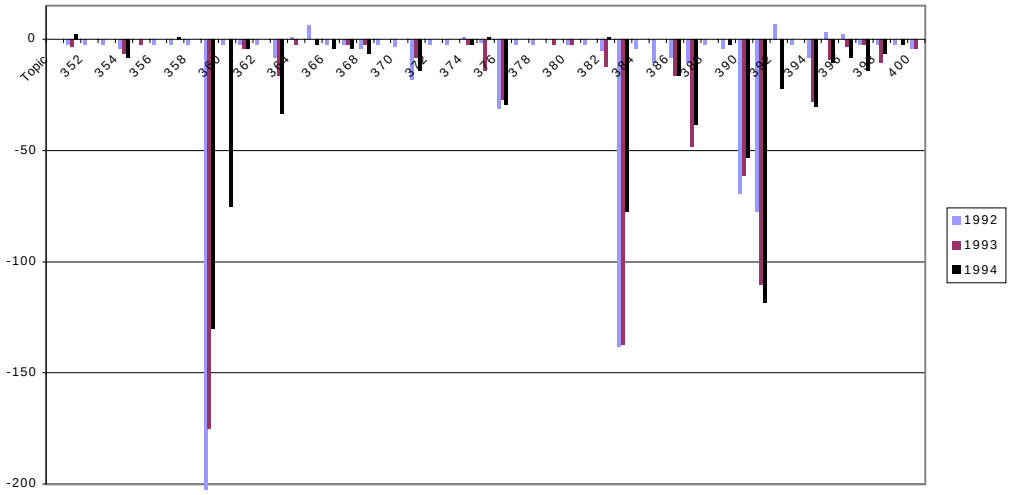

Figure 2: Differential Filtering - LF1

in the database. (The smallest number of positive judgements is 0 and the largest is 747 while the average across the queries is 114 with a standard deviation of 155).

$$u_s^*(S,T) = \{ \max(u(S,T), U(s)) - U(s) \} / \{\max U(T) - U(s) \}$$

where u(S,T) is the utility for the system S and topic T pair; U(s) is the utility of retrieving s non-relevant documents (and 0 relevant ones); MaxU(T) is the maximum possible utility for topic T which is dependent upon the number of relevant documents present for the query.

The scaled utility normalizes performance against a given number of retrieved non-relevant documents. The scaled utility computed with s = 25, 50, 100 and 200 are presented for the top four TREC-8 systems in the second through fourth rows of Table 1. The table provides the number of relevant documents retrieved (RR), number of non-relevant documents retrieved (NR), number of

Group	Run Name	RR	NR	UN	LF1	u_{25}*-ubl	u_{50}*-ubl	u_{100}*-ubl	u_{200}*-ubl
	Baseline	0	a	a	0.0	0.0	0.0	0.0	0.0
U. of Iowa	Iowaf992	57	120	0	-69	-0.022	-0.012	-0.007	-0.003
Claritech	CL99afL1d	180	a	a	-100	-0.056	-0.029	-0.014	-0.007
U. of Twente	Uttnolf1f	b	a	a	-60	-0.029	-0.016	-0.008	-0.004
U. of Mass.	INQ610	266	a	a	-406	-0.113	-0.065	-0.035	-0.019

Table 1: TREC-8 Performance Scores for Top Four Systems (LF1 and Scaled Utility

a. Data is not available to us.

b. This data is not available to us. Please see the footnote on page 3. These calculations were made using the data distributed by David Hull to TREC-8 filtering track participants. The slight differences in scaled utility figures may be due to differences in rounding up strategies.

unjudged documents retrieved (UN), the LF1 scores and the scaled utility scores. (The raw data used for these calculations for the other systems are from David Hull's preliminary analyses presented at TREC-8.) *.

3 – Question Answering Track

Our work in this track involved two distinct implementations employing different sources of relevance judgements.

3.1 – Run 1: Lexical Clues and Singhal's documents

Our aim was to determine how an approach based on surface analysis using lexical clues could be successful in the question answering task. We used the 200 top documents for each question distributed by Singhal. A question grammar was designed by starting with the training questions and expanding upon it using our own experience regarding the nature and structure of questions. The question grammar was used to insert appropriate tags into the free-text questions. For example, questions that began with "When" or containing the word "date(s)" or "year(s)" had a DATE tag inserted. Similarly, questions with the word "dollar(s)" or "cost(s)" had a MONEY tag inserted. In addition to DATE and MONEY, clues to tag the questions and sentences with NUMBER, and NAME were developed. These lexical clues were embedded into rules in a lex program used to preprocess the test questions. The 200 document sets were processed using a parallel method. First each document was segmented into a set of sentences. Next, each sentence was processed through an equivalent sentence grammar that also inserted the same set of tags. Finally SMART was used to retrieve the top ranking five sentences which formed the basis for the submission.

* It should be noted that when David Hull presented filtering results at the TREC-8 conference he had mentioned that the IOWAF992 run was the best. However he later provided corrected data for the filtering runs due to an error in his data for theTwente group. Our analysis of the corrected data indicates that the Twente run is slightly better than ours in that it yields -60 LF1 across all topics. However, our scaled utility scores are slightly better than theirs. All Twente data reported here are derived from the corrected data distributed by Hull.

We explored retrieval strategies based on free-text alone against retrieval based on free-text augmented with tags on the training set. The latter strategy seemed most effective. Similar explorations indicated that weighting the tags higher than the free-text terms yielded better results.

Error analysis indicate weaknesses in our sentence segmentation algorithm. Many output sentences were far from being informative. Also, there were errors in the tagging programs. For example, identifying names turned out to be very challenging. Interestingly, this simple approach yielded a mean reciprocal rank of 0.267 over the 198 questions. Answers were found in the top 5 ranks for 81 questions. When considering the difficulty of each question, this method provided the best rank for 37 questions and the second best rank for 18 questions. This covers 68% of the 81 questions answered.

In future work, we will extend our explorations with these ideas after first refining the present approach which will be followed by suitable extensions.

3.2 – Runs 2, 3 and 4: Part-of-Speech Tagging and TRECcer's documents

The remaining three runs used the top 10 documents matched out of the primary similarity scoring for TRECcer, our adaptive filtering system. Each question was tagged and matched against a coarse taxonomy of question types (basically, who/what/when/where/...) to establish the document features necessary for a match. Each matched document was segmented into distinct sentences and these sentences were then tagged using an implementation of Brill's rule-based algorithm [1]. Separate vectors of verb phrases and noun phrases were generated for each sentence and these were scored against the feature set extracted for the question. Three outputs were then generated, each at a different level of granularity. The first (*sentence*) comprised the complete sentence, truncated if necessary to fit within the 250 byte limit. Most sentences were significantly smaller than this. The second (*50 byte*) comprised the first 50 bytes of a matching sentence. The third, and most aggressive, (*noun phrase*) attempted to narrow the response down to a single clause, typically a noun phrase for 'who' or 'where' questions. A combined plot of results from the four runs appears in Figure 3. One interesting result of our two-pronged approach is that of the 23 matches at any level for the 250-byte POS approach, only 7 overlap with the matches at any level for the Lexical Clue approach.

4 – Small Web Track

Our work in the small Web track involved adapting our Web search engine to accommodate TREC-style source document specification and generation of a vector similarity score for each document against each of the queries. This is rather distinct from our normal mode of operation, where all document vectors are stored in an underlying database layer and a lexicon of term frequency is generated. Instead, due to time constraints relating to database commit overhead, we opted for an approach more akin to adaptive filtering, where the term statistics were accumulated as the run progressed. This allowed us to process the data quickly, but at the price of less-than-optimal weights for terms early in the run. Figure 4 shows the results of the content-only output. There is a definite trend as the number of relevant documents increases to fail to match a proportionally increased number of documents. This effect is due, we believe to the term frequency issue.

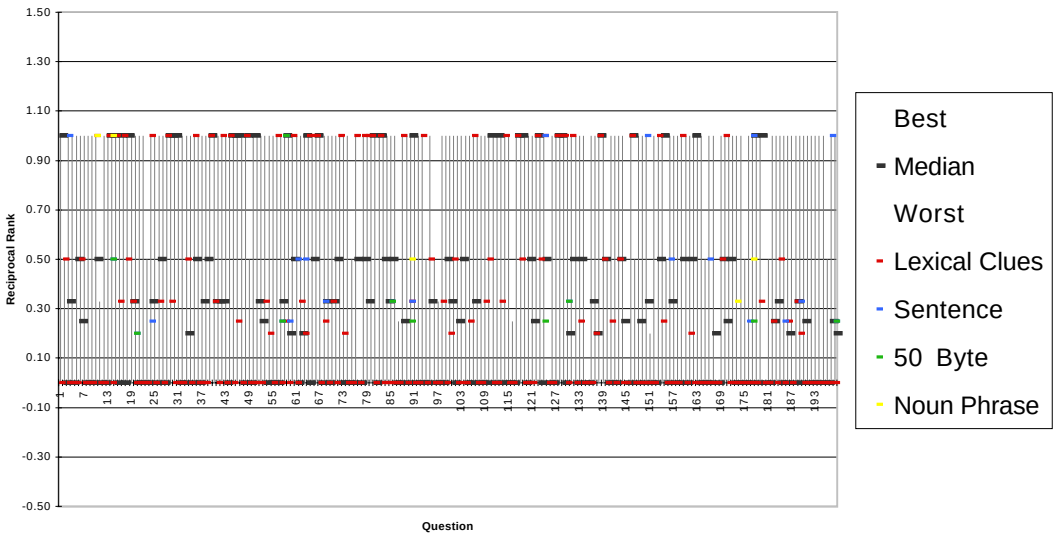

Figure 3: Question Answering Performance

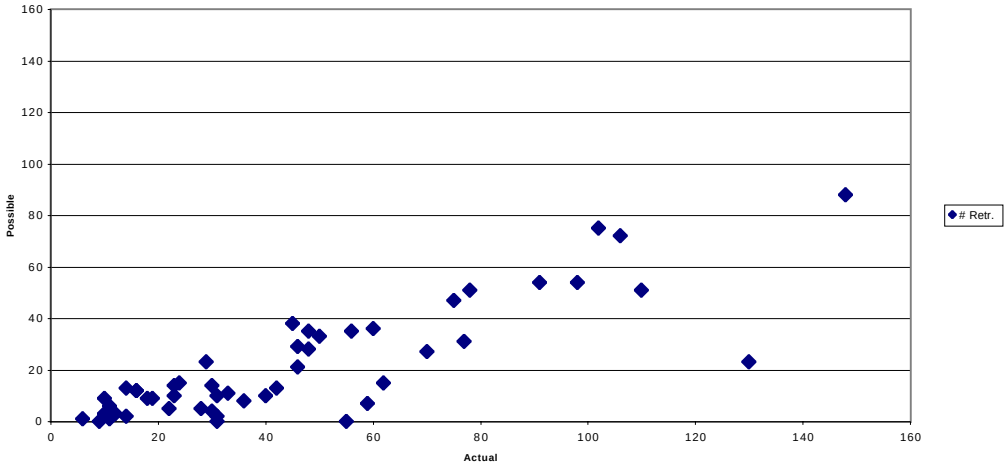

Figure 4: Web Track, Content Only

One of the hazards in the small Web track is that the sampled document are not guaranteed to comprise a connected Web subgraph. Our previous means of computing content+link scoring hence did not fare well compared to a simple content-only approach. Contrasting the exact precision against percentage retrieved of relevant documents, as shown in Figure 5, demonstrates that weighting a document's similarity with its link connectivity with few exceptions degraded performance. Because of this we feel that the small Web task, if it is to remain in the Web track, should employ documents that comprise a connected subgraph. This is much more typical of the data that would be acquired by a spider.

References

[1] Brill, E., "A Simple Rule-Based Part-of-Speech Tagger," *Proc. of the Third Conference on Applied Natural Language Processing*, Trento, Italy, pp. 152-155.

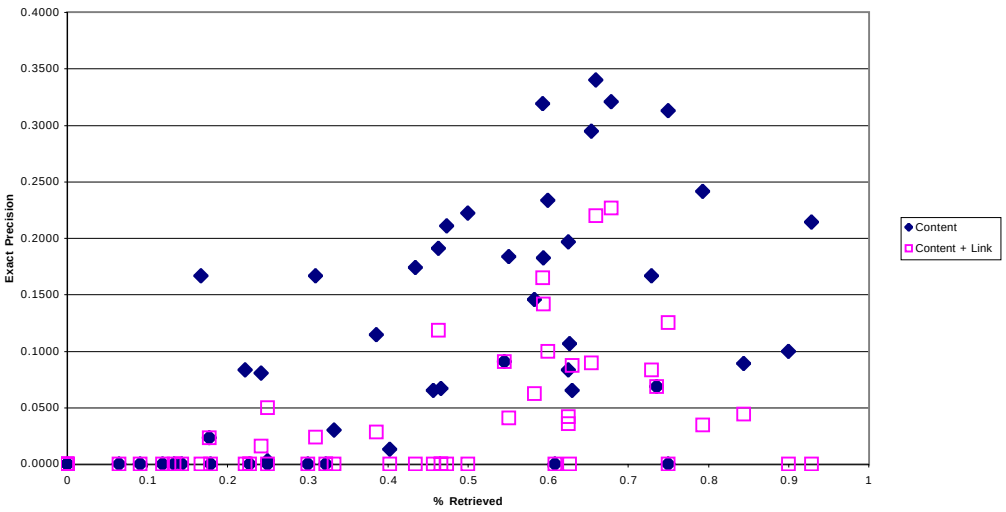

Figure 5: Web Track, Content + Link

[2] 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.

[3] Hull, David. Introduction to Filtering. Text Retrieval Conference (TREC-8). November 1999.