CLARIT TREC-8 Manual Ad-Hoc Experiments

David A. Evans, Jeffrey Bennett, Xiang Tong, Alison Huettner, Chengxiang Zhai, Emilia Stoica

CLARITECH Corporation

Abstract. CLARITECH's submission in TREC-7 demonstrated the utility of document clustering in retrieval. We continued this work in TREC-8, using a clustered document presentation exclusively. We also added significant new functionality to the manual ad hoc user interface, integrating it with an entity extraction subsystem (upgraded and customized for TREC). Extracted entities represent an alternate set of document features. Our experiments suggest that in many cases users might construct more effective queries by moving beyond surface terms and drawing from this more abstract pool of semantic types. Despite the interface enhancements, our focus this year was on system rather than human subject performance, and we simplified the experiment design accordingly. From the users' perspective, there was only one run; the five separate submissions represent variations in post-processing. We spent minimal time preparing the initial queries. Users had 20 (instead of last year's 30) minutes for relevance judgments, and were allowed to modify the query from the start. This year, as well, we reintroduced "vector-length optimization" in the post-processing of feedback. Recent CLARITECH systems have augmented the manually generated queries with a fixed, arbitrary number of selected terms from top-ranked documents. This year, we experimented with a principled truncation of the candidate term list, and found this had a positive effect on the performance of both of our TREC-7 and TREC-8 final queries. We feel that further performance improvements are likely to be achieved only by developing several complementary techniques and applying them selectively to fine-tune individual queries. User-directed feature selection and vector-length optimization are two such promising techniques.

1 Introduction

CLARITECH's approach to manual ad hoc retrieval in TREC-7 involved the use of clustering to facilitate users' identification of relevant documents for subsequent feedback and automatic processing. Our results demonstrated the positive effect of clustering retrieved documents (vs. ordinary ranked list presentation). In particular, at all sampled time points for subjects giving relevance judgments, subjects who used clustered sets of documents out-performed those who used ranked lists. Our overall system results were quite good.

This year, we revisited the problem of clustering by adding the ability to cluster documents using a variety of document features, including entities and semantic abstractions, as well as terms. Our hypothesis was that, depending on the type of query, different document features would afford the most natural basis for organizing results. For example, questions about a specific topic might best be addressed by having retrieved results clustered primarily by entities such as person, place, organization, etc., and only secondarily by terms. In our TREC-8 experiments, we offered users the opportunity to cluster results by several such user-selected features. In addition, we shortened the amount of time that users were given to complete their reviews of documents, from a full 30 minutes per query in TREC-7 to 20 minutes per query this year. We sampled results at five-minute intervals during the 20-minute task and can also report on the relative trade-off in time on task (efficiency) vs. performance.

Subsequent to obtaining users' judgments, the CLARIT system processes the judged documents fully automatically to expand the original query and select the final set of results. Such processing depends on identifying terms in judged documents to be added to the source query vector. In the recent past, we have had good results when using a fairly large, but arbitrarily truncated set of discovered terms. This year, we returned to an approach that we used in early TREC experiments and used a principled truncation of candidate supplementary terms—a process we call "vector-length optimization". In pre-TREC-8 experiments on the TREC-7 data, using our submitted final queries from last year, we achieved more than 10% improvement over our TREC-7 results by truncating the query vector at that point where the expected contribution of an additional new term drops below a threshold of utility. Using such an approach, we achieved a higher performance on TREC-7 data with queries that averaged 50 terms vs. the 250 terms in our submitted final results. In our automatic processing of judged documents this year, we completed runs that used both our TREC-7 approach (fixed-length vectors) and our new technique (length-optimized vectors).

The two new techniques that we introduced in our work this year—(a) active use of a variety of document features, including entities, and (b) vector-length optimization—are important, general techniques for

information management, not only for information retrieval. We have used these techniques in our other track work this year; and we see in them great potential for helping to solve that very challenging problem in information processing—the fine-tuning of several complementary approaches to the individual requirements of a query or task.

2 Experiment design

For this year's TREC experiment, the 50 queries (401–450) from NIST were entered into the CLARIT system with minimal editing. We started with the text of the title, description, and narrative fields as the query, with editing by a single researcher. The researcher spent very little time on each query (well under 5 minutes), and was not permitted to retrieve any documents. Editing was limited to:

- punctuation changes (e.g., replacing commas with semicolons)
- the omission of query sentences describing non-relevant documents
- the removal of "empty" words such as *documents that discuss* or *a relevant document should include*
- repetition of nouns modified by conjoined adjectives (e.g., *genetic and environmental factors* became *genetic factors and environmental factors*)
- very occasional addition of an obviously relevant word or phrase (for example, *quilt show* in query 418)
- occasional addition of a query constraint, possibly involving extraction entities

The user's task was as always to submit the initial queries to a database consisting of the target corpora and judge the results. Results were presented as clustered groups of the top 150 documents. Users' relevance judgments were automatically collected at 5, 10, 15, and 20 minutes. Users were allowed to reformulate the initial query and retrieve potentially new results at any time during the 20-minute task. They could also terminate the task before the 20-minute time limit if they felt they had found all the relevant documents.

All the documents in the database used by the subjects were indexed with extracted entities. Extraction entities include cities, provinces, nations, personal names, employee/appointee titles, business names, and names of other organizations, such as government bodies and universities. Such entities are identified automatically using the CLARIT extraction engine, which utilizes both standard patterns (e.g., *honorific + known first name + initial + unknown word* is a standard pattern for personal names, like *Mr. Hubert M. Narmalee*) and large or exhaustive lists (e.g., the names of the 322 nations in the world today). Once identified, these items can be indexed as terms, but with their entity type remaining available. The set of entities in a retrieved document could be viewed by the subjects, either highlighted in context or in a separate list; the subjects could also use entities and/or entity types in constraint formulation. For example, for query 401—*What language and cultural differences impede the integration of foreign minorities in Germany?*—the user could require that all documents retrieved include the nation entity *germany*. For query 428—*What other countries besides the United States are considering or have approved women as clergy persons?*—the user could require that all documents retrieved include one or more nation entities, or that the specific nation entity *united states* be excluded. It was also possible to include constraints requiring or prohibiting a particular term or verbatim string.

The system presents the initial results as term-based document clusters, but supports clustering (and cluster summarization) by extracted entities in subsequent clustering (or reclustering) operations. All subjects had had some prior searching experience, though some were new to the CLARIT system. All subject actions, with time stamps at one-minute intervals, were written to a relational database.

The users generated a set of relevance judgments; these were further processed fully automatically to produce the submissions. Due in part to the new user interface, which simplifies the use of advanced features such as constraints, the subjects made heavy use of the constraint mechanism. Nearly three quarters of the queries contained constraints; 50% of the total had "term" constraints, and 22% had constraints involving extraction entities. Four percent were negative constraints (i.e., they excluded documents containing certain specific terms or entities).

We used the same set of user relevance judgments in four experimental runs:

- CL99SD, the "empty" run, which used neither of our new techniques
- CL99XT, in which we took advantage of extraction entities
- CL99SDopt, using vector-length optimization[1]
- CL99XTopt, using both extraction entities and vector-length optimization

[1] We are omitting from this discussion our second optimized run, CL99SDopt2, in which we lowered the weights of the user-generated query to match the feedback term weights. This uniformly hurt performance. (For simplicity, we refer to CL99SDopt1 as CL99SDopt throughout this paper.)

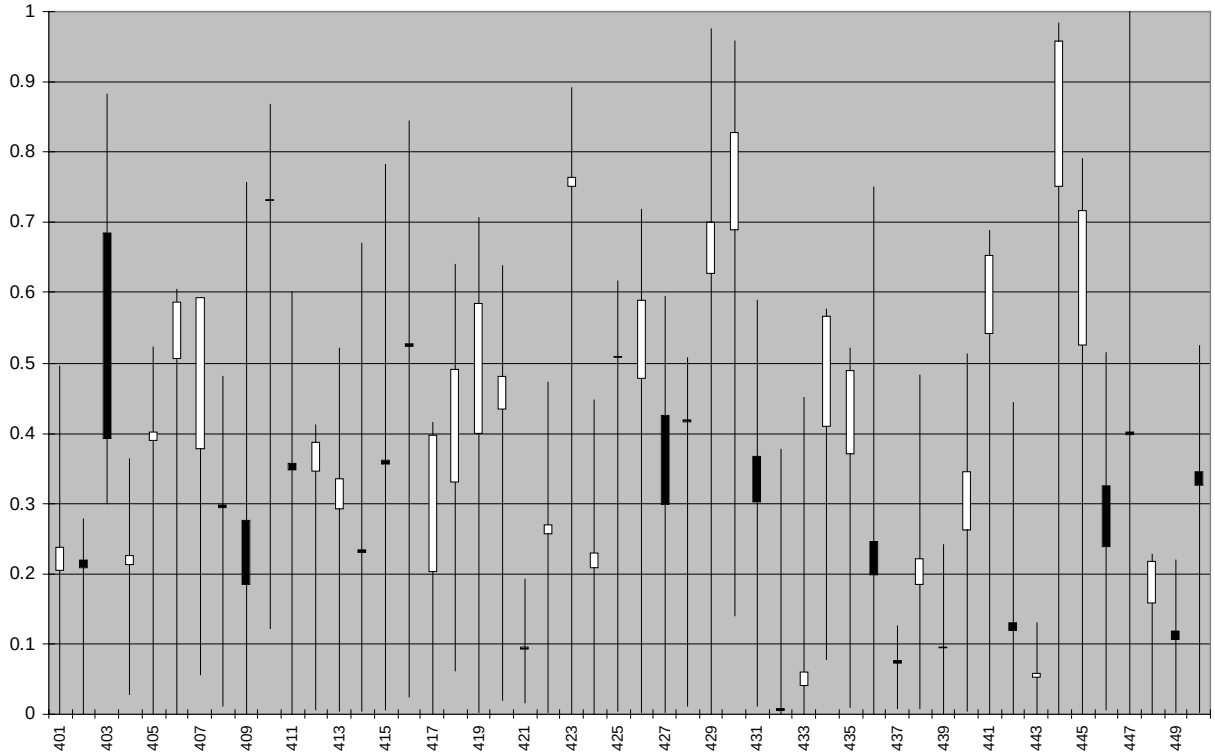

Figure 1. Comparative performance analysis: CL99XTopt vs. the group

All runs used pseudo-relevance feedback (an adaptation of the Rocchio method); the system assigned a coefficient of 0.5 to all new query terms. The system excluded documents explicitly marked non-relevant by users, and promoted marked relevant documents to the top of the ranked list. The XT runs (the two using the entity database) used constrained queries. To ensure a complete submission, these runs required two retrievals—one with and one without the constraints. All documents satisfying the constraints were returned first; if necessary, the system rounded out the top 1000 with documents from the unconstrained retrieval.

For the baseline run, CL99SD, the system removed all query constraints, and added a standard fixed-length vector of feedback terms (250). (This duplicates the approach we took in TREC-7.) CL99XT, the baseline entity run, included constraints (using the merging algorithm described above). CL99SDopt used vector-length optimization; CL99XTopt used both constraints and optimization.

The purpose of vector-length optimization is to avoid the "over-fitting" that can occur when adding too many feedback terms to a query.

Though we have observed good results in the past using a fixed-length vector of 250 terms, we often find that reducing this number yields even better performance. In fact, reducing the vector to a mere 20 terms increases the average precision of our TREC-8 baseline run (CL99SD) from 0.3537 to 0.3638—nearly a 3% improvement.

We observe that, in general, longer documents require more feedback terms, while document sets containing many rare terms need fewer feedback terms. More specifically, there seems to be a relation between the distribution of term weights and the number of feedback terms required to maximize average precision. Sorting the candidate feedback terms by decreasing weight, the point of diminishing (and eventually negative) value occurs as the curve begins to "flatten," as the difference in weight between successive terms approaches zero. We use a simpl0e heuristic to estimate this point: determine the range of term weights and include all terms with weight greater than or equal to $min + p * (max - min)$, where p is a parameter. We also imposed an upper limit of 250 terms on the feedback. The CL99SDopt and CL99XTopt runs described here used $p = 0.05$. (Our official CL99SDopt submission used $p = 0.1$.)[2]

[2] The two non-entity runs that we actually submitted used an older version of the CLARIT system—a version without the entity-indexing option—and $p = 0.1$. In this discussion , we ensure comparability with our XT results by substituting a new set of SD runs, using the new system and $p = 0.05$.

3 Retrieval performance

Our official results show the strong positive effect of extraction entities and the weaker positive effect of vector-length optimization.

Figure 1 details our results relative to the median. The whiskers show the entire performance range (Worst to Best); the boxes show the median and CLARITECH's (XTopt) average precision score. If the box is white, the CLARIT score is given by the top edge, the median by the bottom edge; in these cases, we outperformed the median. If the box is black, the CLARIT score is given by the bottom edge, the median by the top edge; in these cases our performance was below median.

Table 1 shows the results for the re-run versions of the SD runs (SD and SDopt) using the same code base as for the XT runs. (Our actual submission used an older code base for the baseline runs.).

The table shows a slight improvement in average precision due to vector-length optimization (2% for SD → SDopt and 1% for XT → XTopt). There is a much stronger effect for use of extraction entities and constraints (about 9% for SD → XT and 8% for SDopt → XTopt). Note that the XT run used a fixed 250-term vector, while post-TREC experiments determined that a 20-term vector significantly improves the baseline performance.

The distinct clumping of values in the other columns is intriguing and suggestive of the effect of each technique (constraints and vector optimization) on the retrieval process. Initial precision and total recall seem indifferent to entities and constraints, but respond strongly to optimization. Sustained precision seems to be aided by constraints, but actually harmed by vector optimization.

It is instructive to examine the specific types of constraints that were used, to see whether performance is sensitive to the particular form of the constraint. We have divided all constraints into four (slightly overlapping) types. General Entity constraints specify entity types—not specific entities. For instance, one such constraint might require all documents to contain a person entity. A Specific Entity constraint might require the person name *Abraham Lincoln*. A Term constraint requires the presence of a specific term that is not an entity recognized by the extraction system (e.g., *ship* or *storm*). Finally, a Negative constraint requires the *absence* of a term, entity, or entity type. Tables 2–4 reflect this analysis for four General Entity, seven Specific Entity, 25 Term, and four negative constraints. For comparison, the tables also include results for the 14 completely unconstrained queries, and the overall averages or totals.

Note that constraints were not used at all for the baseline (SD) runs, yet the average precision values nevertheless vary widely. This indicates that queries the users thought needed general entity constraints were "easy," while those requiring negative constraints were the most difficult. Queries requiring specific entities were similarly difficult. It is also striking that actually using the constraints helps in nearly all cases.

Run	Avg. Precision	Initial Precision	Precision @ 100	Recall
CL99XTopt	0.3765	0.9245	0.3030	3366
CL99XT	0.3730	0.9060	0.3078	3367
CL99SDopt	0.3489	0.9285	0.2732	3300
CL99SD	0.3425	0.9081	0.2766	3282

Table 1. Comparison of four CLARIT runs on standard metrics.

Average Precision	SD	XT	SDopt	XTopt
General Entity	0.4621	0.5235	0.4659	0.5215
Specific Entity	0.2221	0.2782	0.2203	0.2723
Term	0.3186	0.3576	0.3303	0.3655
Negative	0.1908	0.2201	0.1904	0.2171
Unconstrained	0.3459	0.3457	0.3463	0.3463
All	0.3425	0.3730	0.3489	0.3765

Table 2. Average precision by constraint type, for four CLARIT runs.

Note again the striking performance variation, and the dominance of the entity runs.

Despite the general improvement due to constraints, we fear that in some cases a general constraint might dredge up large numbers of documents from the bottom of the ranked list. Since documents that satisfy the constraint are always ranked above those that do not, some relevant documents might be excluded. We will examine the effect of limiting the number of such documents in order to avoid this problem. We can imagine several more sophisticated techniques for merging constrained and unconstrained retrieved document sets.

Another parameter that merits further examination is the coefficient of feedback terms. We have traditionally assigned feedback terms lower weights than user-generated terms, yet follow-up experiments on both TREC-7 and TREC-8 have indicated that improved performance often results from assigning weights to feedback terms that are closer to the average manual term weight.

Tables 5 and 6 show the TREC-reported statistics for the runs.

Precision @ 100	SD	XT	SDopt	XTopt
General Entity	0.4100	0.4550	0.4075	0.4425
Specific Entity	0.2214	0.2871	0.2271	0.2871
Term	0.2448	0.2828	0.2400	0.2776
Negative	0.2060	0.2340	0.2020	0.2420
Unconstrained	0.2969	0.2969	0.2900	0.2900
All	0.2766	0.3078	0.2732	0.3030

Table 3. Precision at 100 documents by constraint type, for four CLARIT runs.

Average Recall	SD	XT	SDopt	XTopt
General Entity	88.5	90.3	89.0	90.0
Specific Entity	81.7	92.0	82.3	93.1
Term	57.8	59.2	57.6	58.4
Negative	65.6	69.0	65.6	69.6
Unconstrained	60.5	60.5	61.4	61.4
All	3282 (Total)	3367 (Total)	3300 (Total)	3366 (Total)

Table 4. Recall by constraint type, for four CLARIT runs.

Precision	SD	Sdopt	XT	XTopt
0	0.9047	0.9163	0.9061	0.9245
0.1	0.7438	0.7509	0.7491	0.7703
0.2	0.5875	0.6030	0.5984	0.6148
0.3	0.4601	0.4704	0.4794	0.4826
0.4	0.3794	0.3927	0.4061	0.4021
0.5	0.3148	0.3320	0.3423	0.3407
0.6	0.2555	0.2665	0.2836	0.2862
0.7	0.1984	0.2165	0.2275	0.2335
0.8	0.1549	0.1699	0.1797	0.1842
0.8	0.0852	0.0924	0.1160	0.1135
1.0	0.0392	0.0439	0.0467	0.0443
Avg.Prec	**0.3537**	**0.3682**	**0.3730**	**0.3766**

Table 5. Recall level precision averages.

Docs	SD	SDopt	XT	XTopt
5	0.7600	0.8000	0.7680	0.7680
10	0.7020	0.7080	0.6920	0.6920
15	0.6453	0.6440	0.6280	0.6280
20	0.5840	0.5870	0.5730	0.5730
30	0.4860	0.4887	0.4907	0.4940
100	0.2816	0.2862	0.3078	0.3030
200	0.1969	0.2046	0.2148	0.2144
500	0.1099	0.1132	0.1173	0.1172
1000	0.0661	0.0675	0.0673	0.0673
R-Prec.	**0.3709**	**0.3837**	**0.3829**	**0.3788**
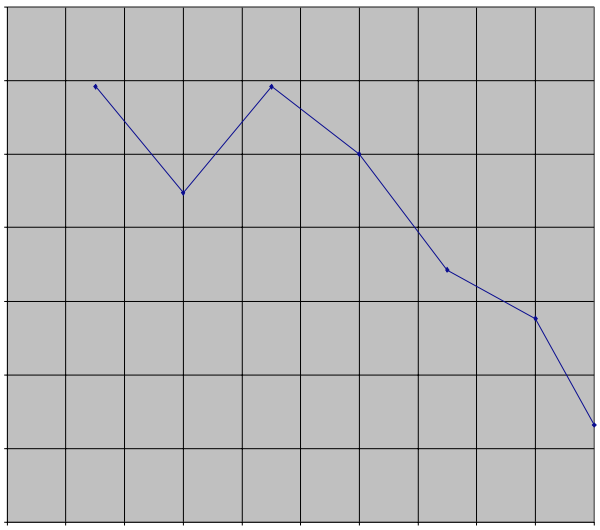
Table 6. Document level precision averages.