Reliable Information Access Final Workshop Report

Chris Buckley
Sabir Research Inc.
Gaithersburg, MD 20878
chrisb@sabir.com

Donna Harman
National Institute of Standards and Technology
Gaithersburg, Maryland 20899
donna.harman@nist.gov

January 30, 2004

1 Introduction

For many years the standard approach to question answering, or searching for information, has involved information retrieval systems. These systems were initially Boolean systems, requiring major effort from the users, but new systems allow natural language input for questions and return ranked lists of documents.

Whereas the question answering (QA) task has evolved beyond just the ranked list approach to answers, the QA systems depend on the information retrieval (IR) technology at two different stages. First, the IR technology is needed to make the initial cut at finding the information from many gigabytes of text. Most of the QA systems use heavy amounts of NLP technology and therefore use IR technology to narrow the pool of potential information to 100 or 200 documents before doing more intensive processing. But a second, and equally important, need of IR systems is to provide a fall back position for questions that are beyond the current abilities of the QA systems. Having a ranked list of documents is clearly better than having nothing!

Current IR systems generally provide a set of reasonable results for the QA systems to work with. However results from these systems are unpredictable in that there are occasional failures. Even more important for the AQUAINT program, these systems are unpredictable to the system researchers; that is, the systems cannot reliably customize output on a per question basis. This leads to lower precision in the top set of documents for some questions, and radical variance in performance using different retrieval technologies for the same input question.

Even so, the current statistical approaches to IR have shown themselves to be effective and reliable in both research and commercial settings. Almost by definition, statistical IR focuses on the matching of word occurrences in topics and documents rather than on semantically understanding the texts. Given that, improvements in statistical IR must come from either re-weighting the importance of existing word matches, or from expanding the texts and adding new words that can be matched. Thus query expansion has been a central focus of statistical IR throughout its research history.

Experimental environments such as TREC show that retrieval results vary widely according to both topic and system. This is true for both the basic IR systems and for any of the more advanced implementations using, for example, query expansion. Some retrieval approaches work well on one topic but poorly on a second, while other approaches may work poorly on the first topic, but succeed on the second. If we could determine in advance which approach would work well, then a dual approach could strongly improve performance. Unfortunately, despite many efforts no one knows how to choose good approaches on a per topic basis.

The major problem in understanding retrieval variability is that the variability is due to a number of factors. There are topic factors due to the topic statement itself and to the relationship of the topic to the document collection as a whole, and then there are system dependent factors including the approach algorithm and implementation details. In general, any particular researcher is working with only one system and thus finds it very difficult to separate out the topic variability factors from the system variability.

Thus the causes of retrieval variability are poorly understood. For most IR algorithms, we do not sufficiently understand the reasons for retrieval variability of that algorithm well enough to be able to predict whether the algorithm will succeed or fail on a topic. This in turn means we don't understand the basic characteristics of our algorithms. For example, query expansion works well on average, but is it working because the expansion adds

1. one or two good alternative words to original query terms (synonyms)?

2. one or two good related words?

3. a large number of related words that establish that some aspect of the topic is present (context)?

4. specific examples of general query terms?

5. better weighting to original query terms?

Different systems are in effect tuned to emphasize a different choice of the five expansion goals. Which system is better may depend on the topics, but until the tradeoffs between the choices are understood, it won't be possible to make rational system design decisions.

The goal of the RIA Workshop was to understand the contributions of both system variability factors and topic variability factors to overall retrieval variability. The workshop brought together seven different top research IR systems and set them to common tasks. Comparative analysis of these different systems enables system variability factors to be isolated in a way that never before has been possible.

There were two main tracks to the RIA investigation of system and topic variability, one high-level and one low-level. In the high-level track, the seven systems performed a large number of variations of one single query expansion task. In some sets of experiments, the systems changed their own tuning parameter settings. In other experiments, each system used as the source of expansion terms those from each of the other systems. This idea was expanded to experiments in which each system used the actual expansion terms determined by other systems. The overall goal of the analysis was to isolate the system effect and discover why each system is succeeding in its query expansion efforts on each topic.

The bottom up track was a massive comparative failure analysis. Each of six systems contributed one representative run. Then for each designated topic, a detailed manual analysis of each run with its retrieved documents was done. The analysis goal was to discover why systems fail on each topic. Are failures due to system dependent problems such as query expansion weaknesses or system algorithm problems, or are the problems more inherent to the topic? For each topic, what would be needed to improve performance for each system? How can this be predicted by the system?

For each of the two tracks the workshop participants collected enormous amounts of data. Only a small portion of the analysis of the data could be completed during the workshop. The preliminary analysis that has been done has already produced a number of surprising results. The entire collection of data will be released to the community during the spring of 2004, and will enable useful research for years to come.

2 Participants

By its very nature, the RIA workshop required participation from a large number of groups and experts. Bringing together seven of the top research systems in one location with both high-level theoretical expertise and also practical system expertise was difficult, especially given the 6-week duration of the workshop. There were two groups of participants; the senior experts who generally were present for 1 to 2 weeks of the workshop spread out over several trips, and the graduate students who for the most part were at the workshop for the full 6 weeks. Altogether, there were 28 people from 12 organizations participating.

The following list gives the organizations, people, and software contributing to the workshop. Appendix A of the report gives a fuller description of each system as written by the participants. The amount of time the people were able to be at the workshop is listed (some of the senior people were heavily involved remotely as well) along with some of the special contributions. This was an open workshop environment where everybody was constantly contributing ideas and efforts. As well as working with their own research systems, most participants were also in charge of several of the daily failure analysis and one or two of the system experiments.

- NIST
  - Donna Harman. 2 1/2 weeks. Workshop founder and co-leader
  - Ian Soboroff. 2 weeks.
  - Ellen Voorhees. 2 weeks.

  NIST was the organizer of the workshop. Donna Harman was the high-level leader of the workshop, and Ian Soboroff and Ellen Voorhees provided senior guidance during the periods they were present. NIST also contributed the Beadplot software used in failure analysis [11].

- Sabir Research
  - Chris Buckley. 6 weeks. Workshop co-leader

  Sabir furnished the SMART IR system, Version 14.2; a vector space, weighted retrieval with expansion approach [2, 3]. Sabir also supplied a version of SMART performing retrospective,upper bound runs. Chris Buckley supplied and modified trec_eval, a program to evaluate run results in TREC results format [1]. Buckley designed most of the infrastructure to support the workshop, and was the day-to-day leader of the workshop.

- Waterloo
  - Charlie Clarke. Faculty. 1 1/2 weeks
  - Gord Cormack. Faculty. 1 1/2 weeks
  - Tom Lynam. Grad student. 6 weeks
  - Egidio Terra. Grad student. 2 weeks

  Waterloo brought the MultiText retrieval system, a statistical passage-based system with retrieval and/or expansion based on finding regions of document text with sufficiently dense occurrences of query terms [4, 19]. They also made available two versions of Question Answering MultiText that had been used for TREC QA. In addition, Waterloo supplied WUI, a flexible browser based user interface for examining retrieved documents.

- CMU
  - Jamie Callan. Faculty, 2 weeks
  - Paul Ogilvie. Grad student, 2 weeks
  - Yi Zhang Grad student, 2 weeks
  - Luo Si. Grad student, 2 weeks
  - Kevyn Collins-Thompson. Grad student, 2 weeks

  CMU used the Lemur system, a freely available very flexible research statistical IR engine [9, 20]. They used a language modeling approach with massive expansion.

- UMASS
  - Andres Corrada. Research Faculty, 3 weeks

  UMASS also used Lemur, with a different language modeling approach [9, 12, 8].

- Clairvoyance
  - David Evans. Company President, 2 1/2 weeks
  - David Hull. Senior Research Staff. 1 1/2 weeks
  - Jesse Montgomery. Research Staff. 6 weeks

  Clairvoyance furnished 2 versions of their commercially available system, a statistical system with strong NLP components oriented to interactive users [6, 7, 10]. They also furnished AWB, an interactive tool useful for analyzing retrieval results.

- City
  - Andy MacFarlane. Professor, working remotely

  City furnished the Okapi system, a probabilistic retrieval system [14, 15]. Because of visa problems, MacFarlane was not able to attend and worked remotely, aided by Zhenmei Gu (below).

- Albany

- Tomek Strzalkowski. Faculty 1/2 week
- Sharon Small. Faculty 1 week
- Sean Ryan. Grad student, 6 weeks
- Ting Liu. Grad student, 5 weeks

Albany used a SMART/HITIQA hybrid system. SMART is an old version (Version 11.0, 1991) of the SMART retrieval engine and HITIQA is an extensive NLP based system that here post-processed retrieval results to cluster and analyze result passages and suggest expansion phrases [16, 17].

- MITRE
  - Warren Greiff. Research staff, 6 weeks

Warren Greiff contributed statistical analysis of data during the workshop, and was responsible for the R statistical database of run results.

- Rutgers
  - Paul Kantor. Faculty. 2 weeks

Paul Kantor provided the foundations and preparations for both short term (within workshop) and long term statistical analysis of the experimental results.

- MITRE supplied students
  - Robert Warren. Grad student from Waterloo, 6 weeks
  - Zhenmei Gu. Grad student from Waterloo, 6 weeks
  - Luo Ming. Grad student from Virginia Tech, 6 weeks
  - Jeff Terrace. High school student, 6 weeks

MITRE held a competitive search for 3 graduate students and a high school student for the workshop. Robert Warren was extremely active in developing the infrastructure to support the workshop. Of particular note, he and Jeff Terrace constructed an elaborate web-based system that allowed dynamic browser access to the entire database of research results, notes, data, and reports. Zhenmei Gu was drafted to be the local representative for City, and aided in the arduous process of getting the City system to work at all, and then performing the needed experimental runs. Zhenmei and Luo Ming were responsible for replicating experimental runs of all systems on additional experimental databases.

3 Topic Failure Analysis Investigation

The topic failure analysis investigation was an attempt to discover why current research information retrieval (IR) systems fail and to propose concrete areas of concentrated research to improve effectiveness. IR systems improved tremendously at the beginning of the 1990s as the TREC corpora offered new large collections upon which to base statistical information retrieval. However, there has been very little improvement in the core IR retrieval algorithms in the past few years. Different groups have tried many different attacks on the IR problem, but without general success as measured by TREC ad hoc task evaluation.

One major part of the failure to improve is that any individual group analyzing retrieval performance on a topic wll find it very difficult to separate out the system-dependent failures from the topic-dependent failures. Some topics, or even aspects of topics, are hard for all systems. Other topics may be easier for some systems to do well on than other systems, but nobody currently understands why.

This investigation looked at IR system performance across a set of different research IR engines on a topic by topic basis. By comparing how each system succeeds or fails on any individual topic, strategies for successfully dealing with that topic can be devised. By considering enough individual topics, overall strategies for retrieval can be devised, along with identification of areas of research that need to be investigated before improvement can be expected.

Before the start of the workshop, the following “facts” were known

- Good research systems all have about the same average performance as measured by Mean Average Precision (MAP), or any other standard TREC evaluation measure.
- Systems perform differently across topics
- On any given topic, performance varies across systems
- Systems retrieve quite different documents even if performance on a topic is about the same.

These facts suggested the following general hypotheses

1. Systems are failing on individual topics in different ways
2. Systems are emphasizing different aspects of each topic
3. Systems are looking at different relationships between aspects in a topic.
4. Systems will need to look at relationships between aspects to improve, and these relationships can be categorized.

3.1 Failure Analysis Standard Runs

During the March pre-workshop meeting it was decided that all seven groups would submit a *standard* retrieval run that in some sense was representative of their group's approach to IR. There were no restrictions on what was done in the run as long as it was completely automatic. These standard runs, one per group, would be used for topic failure analysis throughout the workshop. The runs were due three weeks before the workshop started, to allow for a lead time for problems and to allow for an initial analysis of the standard runs to determine the topics to be analyzed.

The collection used for the standard runs was the base collection for the workshop: the 150 topics of TREC 6, TREC 7, and TREC 8, run on disks 4 and 5 of the TREC document distribution, not including the Congressional Record subcollection since that was used only for TREC 6. There were a total of 528,155 documents, averaging a bit over 3600 characters each.

Several groups had the expected problems of getting their system properly running on a new machine; by the time the workshop started, 5 groups had representative runs, 1 group had a run that got reasonable results, and 1 group was still working at getting their system running.

- CMU: Lemur software implementing the KL-divergence based unigram language modeling approach [9, 20]. Used blind feedback query expansion, expanding each query by several hundred terms (much more than other systems).
- UMASS: Lemur software implementing query-likelihood unigram language modeling approach with Dirichlet smoothing of document probabilities [9, 12]. There was no query expansion or blind feedback done.
- Waterloo: MultiText system with blind feedback using passage retrieval and hotspot term-extraction, followed by an implementation of the Okapi BM25 algorithm for the final documents [4, 19].
- Sabir: SMART Version 14 vector space system with Lnu-ltu weighting. Used blind feedback and statistical phrases, implementing the base run from TREC 4 [2, 3].
- Clairvoyance: CLARIT Java (CLJ) system based on full CLARIT indexing of sub-documents. Used blind feedback based on comparatively few documents adding precise (low frequency) terms [6, 7, 10].
- City: Okapi probabilistic system using the Okapi BM25 algorithm. Because of problems getting the system running, their standard run did not use blind feedback and was discovered to use the title field of the topics instead of the required description field [14, 15].
- Albany: Used SMART Version 11 with HITIQA NLP system. Despite valiant efforts of the Albany students, their old (from 1991) SMART system was not working properly for several weeks, and afterward the HITIQA system did not work well given the SMART retrieval output. The Albany system was not included in the topic failure analysis at all, though the Albany participants were active throughout doing failure analysis on other systems.

During week 1 of the workshop, it was discovered that the *standard* runs done before the workshop were not quite as standard as desired. Some groups had potentially indexed all fields of the documents while others had abided by the TREC restrictions of the appropriate years and not indexed the manually added fields (the "SUBJECT" section of the LA TIMES). In addition, groups threw out different portions of some of the highly stylized TREC description topics ("*A relevant document must identify ...*"). To make results more comparable, it was decided to standardize a set of patterns in topics that all systems would discard, and to not include the manual field of the LA TIMES documents. By Friday of week 1, all 5 representative run systems had re-run their standard runs (the City run remained not directly comparable because it indexed titles). The first 3 failure analysis topics were done with the original pre-workshop runs; they were later checked to make sure the conclusions were still valid on the re-run retrievals. All subsequent failure analysis sessions were done with the re-run retrievals. All the experimental runs were made with these same standardized topics and documents.

3.2 Topic Failure Analysis Process

Topic failure analysis was a major activity of the workshop; with 90 minutes to 2 hours per day allocated for the individual and group analysis. This was the first time this sort of group comparative failure analysis had ever been done, so substantial effort the first week was spent developing the process of failure analysis: what we wanted to do given our goals and the available tools. The details of the process changed as the participants gained experience throughout the workshop but the overall process remained the same after the first week.

1. The topic (or pair of topics) for the day was determined, with a leader being assigned the topic, on a rotating basis among all participants

2. Each participant was assigned one of the six standard runs (or systems) to examine, either individually or as a team

3. Each participant or team spent from 60 to 90 minutes investigating how their assigned system did on the assigned topic, examining how the system did absolutely, how it did compared to the other systems, and how performance could be improved for it. A template (see Figure 1) was generally filled out to guide both the investigation and subsequent discussions

4. All participants assigned to a topic discussed the topic for 20 to 30 minutes, in separate rooms if there were two topics. The failures of each system were discussed, along with any conclusions about the difficulty of the topic itself.

5. The topic leader summarized the results of the discussion in a short report (a template was developed for this by week 3 of the workshop). If there were 2 topics assigned for the day, each leader would give a short presentation on the results to the workshop as a whole.

Initially one topic per day was analyzed. People worked as teams on pairs of systems, with one person of the team being familiar with each system. The two systems were compared against each other as well as individually. People were rotated each day with the goal of each participant having a chance to work with somebody from each other system. This worked well though it couldn't be done fully; for example, there was nobody from City physically present. In this way, each participant learned the important details of the other systems from an expert, as well as learning how the various failure analysis tools could be used.

Given the available tools, it was just as easy to compare one system against all the other systems instead of just one other paired system, so that was done by week 2 of the workshop. By the end of week 2, people were assigned systems as individuals instead of teams, and two topics could be done in parallel. As new people arrived during the course of the workshop, they would be assigned to work as a team with an experienced participant for 2 to 3 days, and then were assigned to work individually.

There were several conflicting goals to consider when coming up with a template for failure analysis. The whole idea of failure analysis is to apply human intellect and experience to the question of what is happening with a given system's retrieval of documents. Having a detailed template that would be required to be filled out would defeat this purpose, especially given the very wide range of experience of both systems and tools among the workshop participants. For example, the Analyst Workbench (AWB) of Clairvoyance was an excellent tool for grouping documents

- Behavior on top relevant documents [*How many of the top documents for this system were relevant and could they be categorized and distinguished from others?*]
- Behavior on top non-relevant documents [*Why were the top non-relevant documents retrieved?*]
- Behavior on unretrieved relevant documents [*Why weren't these relevant documents retrieved within the top 1000?*]
- Beadplot observations [*How does the ranking (especially among the top 50 documents) of this system compare to all other systems?*]
- Base Query observations [*What did the system think were the important terms of the original query, and were they good?*]
- Expanded Query observations [*If the system expanded the query (4 out of 6 systems did), what were the important terms of the expansion, and were they helpful?*]
- Blunders of system [*What obvious mistakes did the system make that it could have easily avoided? Examples might be bad stemming of words or bad handling of hyphenation*]
- Other features of note [*Anything else.*]
- What should system to do improve performance? [*The individual's conclusion as to why the system did not retrieve well, and recommendations as to what would have made a better retrieval.*]
- What added information would help performance? How can system get that information? [*Is there implicit information in the query, that a human would understand but the system didn't? Examples might be world knowledge (like Germany is part of Europe).*]
- Assessing agreement (were there major issues? was relevance determined by "Desc"?) [*The NIST assessor who originally judged relevance of documents might have a different idea of what was relevant than the text of the description indicates or than the workshop participant thinks should be relevant. It also may be unclear where and why the NIST assessor drew the line between marginally relevant and non-relevant documents.*]

Figure 1: Topic Failure Analysis Template for Individual System

and investigating what was happening with those groups. But it was difficult to learn well enough to be helpful, and was comparatively time consuming, and so it was only used by three or four participants. The template was designed to accommodate the reporting of results from various types of failure analysis investigations rather than prescribing the failure analysis was to be done. Figure 1 shows this template, along with explanatory text and comments (not part of the template or explicitly given to the participants) in italics.

The individual templates appeared to work well. They focused the less experienced participants on problems they could address while allowing the more experienced participants freedom to report what they found. A major weakness of the template was that people tended to keep track of the raw data of their observations as they went along, which was good, but didn't make explicit their conclusions from their observations, which was not good. It might have been nice if people had added more to their conclusions after the group discussions of the topic, but in general they didn't.

There was no template for the report by the topic leader on the topic as a whole initially. It was felt that the topics varied so much that a template would be overly constraining. However, the variability of the topic reports in the first few weeks prompted the development of a second template (see Figure 2) by the end of week 3. For this template, the explanatory text was included in the template.

Again, there was the conflict between demanding enough detail in the template to provide good information common across topics and demanding so much detail that the template questions no longer fit the important parts of the topic analysis. Like the individual template, we opted for mostly general questions; though having a top-level very general template item would have been helpful.

- *Failures common to several systems* If all or most of the systems fail on this topic in some common ways, describe them here. For example, common stemming failure, or a critical word that no one found during expansion.
- *Notable failures unique to one system* Idiosyncratic failures, notable ones only please.
- *Winning strategies* If one or two systems performed spectacularly on this topic, can you see why? For example, an unusual expansion term might enable one system to find a lot more relevant documents.
- *Classes of missed and false alarm documents* Are there identifiable clusters among missed documents (relevant, not retrieved)? What about false alarms (not relevant, retrieved)? Do most of the misses and/or false alarms fit into these classes, or are there a lot of special cases?
- *Notes on topic statement and relevance judgments* Everyone's got a gripe, but sometimes systematic idiosyncrasies in judging or critical words missing from a description are good to know.
- *Testable Hypotheses* The above observations may lead to a few hypotheses one how this topic's performance can be stabilized. List them here. Hypotheses should be testable, that is, it should be possible in principle to implement the solution in one or more of the systems and re-run the experiment.

Figure 2: Topic Failure Analysis Template for Overall Topic

There was debate in the workshop on several occasions as to whether the topics should be categorized as to major failure causes as they were being analyzed. It was decided that for future investigations, coming up with a set of failure categories would be useful, but during this workshop we didn't know enough about the types of expected failures until the end. The danger of prematurely defining categories is that people may be tempted to force a topic into a pre-defined category that it may not really fit. One of the results of this workshop is a list of failure categories.

Paul Kantor and David Evans performed a one-day assessment mid-workshop of the failure analysis process. Their general conclusion was that the process was serving the needs of the workshop well. However, some of the separate discussion groups on the topics were less focused than others, and as a result the final written conclusions about those topics were less focused. They recommended a more systematic discussion of the individual templates to make sure all desired points and systems were fully discussed. This recommendation was followed for the remainder of the workshop.

3.3 Failure Analysis Topic Choice

Given the large time requirements for failure analysis (from 11 to 40 person-hours per topic), it was obvious that not all 150 topics could be examined. We had hoped we could do 50 topics and we actually finished 45 topics. Ideally, we should randomly choose topics that fit our desired criteria. In practice, we did not know enough about what criteria we wanted. Our first attempt failed badly. We ranked topics by variability of the ranked document results; topic 368 was the most variable. While retrieved documents and scores varied widely, all systems did fairly well on it: Mean Average Precision (MAP) varied from .38 to .76. During the failure analysis, for this topic participants ended up spending their time judging how well each system matched the TREC assessor's line between marginally relevant documents and barely non-relevant documents; a task impossible to do well, and not intellectually interesting. It was then decided to focus on topics where the systems in general scored below the overall MAP average. The overall criteria used were

- Average MAP of systems at or below the overall MAP average of about .21
- Large variance between system scores
- Variance in general not due to the City Okapi run (different vocabulary since it used topic title instead of description field)

In addition, we analyzed several topics (about 4 each) that

- Performed differently than others in some experiment

- Had the basic form of a TREC Question Answering question

Note that the analyzed topics can not be said to be random selection of all the topics. We did not analyze most of the easy topics (only 2 out of the top 56, ranked by average MAP), since they had high MAP for most systems. We also did not analyze many of the topics that all systems did extremely poorly on (only 4 out of the bottom 23), since there was generally very little variability between system scores on those topics. Since each topic took about 12 person-hours to analyze, we concentrated on those topics for which there was evidence of some system-dependent effect and some evidence of system failure, and analyzed most of those topics.

3.4 Tools for Topic Failure Analysis

There were a wide variety of tools used in failure analysis. They include

- WUI: The Waterloo User Interface was the major tool used for examining retrieved documents of a system. It offered a GUI controlled front-end that constructed database queries to be sent to a Waterloo back-end database engine. For example, four mouse clicks and typing in the topic id might start showing all non-relevant documents retrieved by Sabir's standard run with rank less than 40 for topic 301. Another three mouse clicks might give all relevant non-retrieved documents for the same run. Gord Cormack spent most waking hours of week 1 adapting Waterloo's existing judgment system to become a very useful tool for our specific task. Each individual document in the initial display of documents would give whether the document was relevant, what rank the document was retrieved at for each of the 6 standard runs, the system's best guess at the most relevant excerpt from the document, and several user-defined buttons and text boxes with which a user could make notes. Clicking on the excerpt would give the full text of the document, with any user designated words or patterns highlighted.

  Most participants used WUI for about 75% of the time they spent on individual failure analysis.

  WUI itself is freely available, however the back-end database engine is controlled by the University of Waterloo. Please contact Gord Cormack or Charlie Clarke for details.

- Beadplot: Freely available from NIST. Beadplot represents each retrieved document of a target retrieval ranking by a color bead on a rank axis. Each other retrieval ranking can be visually compared to the target ranking by seeing whether the color patterns of the two rankings are similar. Beadplot was used by some participants on all topics, but it was not as useful as anticipated. For most topics, the ordering of retrieved documents was just not similar enough between systems, with the occasional exception of the two Lemur systems (CMU and UMASS). The workshop version of Beadplot was adapted by Sean Ryan of Albany to automatically work with the 6 standard runs.

- AWB: The Analysis Work Bench from Clairvoyance is a very nice package allowing grouping of documents by arbitrary pipe-lined filters, with easy analysis or clustering of the groups. For example, it is easy to tell that a certain term occurs in 80% of the top relevant retrieved documents, but only 5% of the relevant documents that were not retrieved. Unfortunately, despite tutorials and substantial efforts by Jesse Montgomery of Clairvoyance at setting up installations of AWB working in the failure analysis environment, only a couple of non-Clairvoyance participants were able to use it. The learning curve was too high for the very limited amount of workshop time available. The AWB is part of commercial products available through Clairvoyance

- smart_std: The SMART system itself offers the ability to look at documents and retrieval results of the 6 standard runs. Most of the capabilities were much more nicely available through WUI. One feature that was occasionally used by participants were the ability to get a table giving the ranks at which each relevant document for a topic was retrieved for each of the 6 runs, sorted by collection. This enabled easy determination of whether systems had a collection dependent bias to their retrieval for a topic (such biases existed for several topics and systems.) It is hoped that SMART will shortly be freely available for research purposes. However, that hope has existed for several years at this point, though now Cornell has verbally agreed to it.

- smart_retro: An adaptation of the SMART DFO approach performed on the retrospective collection where relevance information is known for all documents. For a given topic, smart_retro attempts to construct the optimal simple vector query given the relevant documents. Over all topics, smart_retro averages a MAP score of .83. Participants were told not to look at the smart_retro optimized query until after they performed their

individual failure analysis, since it could very easily bias the analysis to simply presence or absence of key terms. Three noteworthy points of the optimized queries:

1. Often the manual failure analysis would indicate that poor performance was due to a certain term being missed or weighted lightly. The optimized query served to verify or reject that hypothesis. It was surprising how often the hypothesis was rejected.
2. An indication of the overall difficulty of the topic. If the optimized topic had a low MAP score (for example, below .5), that was evidence that a bag of words approach was not ideal for the topic and that relationships between terms would be needed.
3. An indication of importance of sub-collections. It was surprising how highly weighted terms were that only indicated the source of the document. For example, all Financial Times documents contained the terms "FT" in the headline. That term was very often among the top 5 terms in the optimized query.

smart_retro will hopefully be made freely available once SMART becomes releasable. The optimized queries themselves constitute a run and will be made available with the release of the run database in the spring.

- Web interface: The web interface to the failure analysis topics was an enormously helpful tool by the end of the workshop. Rob Warren and Jeff Terrace built the web interface to the entire workshop from scratch; by week 1 of the workshop we still had not even gotten permission from MITRE Security to run a web server. By the end of the workshop, the failure analysis page for a topic would bring up the topic itself, all evaluation scores of the six standard runs for the topic, the full text of the topic including narrative, lists of words occurring in the description field suitable for cutting and pasting into the WUI highlighting tool, similar lists for the narrative field, standard measures of textual difficulty of the topic text, any categorization of the topic that had been performed in the workshop, count of relevant documents that occurred in each sub-collection, and links to any failure analysis that had already been done on the topic. Having all that information collected in one place made the intellectual task of failure analysis that much easier. The web interface is freely available; attempts were made to make it portable, but there is no experience yet at re-installing it elsewhere.
- Individual Systems: Many participants used their own system as a tool for looking at results and trying alternative query formulations. There was general agreement that their own systems could have been much more helpful at failure analysis if there had been enough time and expertise to adapt the systems before the workshop started,

3.5 Topic Failure Analysis Categorization

As discussed previously, there was not any attempt to construct explicit categories of topic failures as the topics were being analyzed. However, toward the end of the last week of the workshop, after topic failure analysis had finished, Chris Buckley constructed categories which in his opinion represented the different sorts of failures seen as important during the summer.

1. General success - present systems worked well
   - 1 topic, primary category
   - Sample Topic *368 Identify documents that discuss in vitro fertilization.* This was the only topic examined in failure analysis on which systems worked well and is included just to be complete. The blind feedback expansion systems had noticeable improved performance, but the participants had no other suggestions.
2. General technical failure (stemming, tokenization)
   - 2 topics, primary category
   - Sample Topic *353 Identify systematic explorations and scientific investigations of Antarctica, current or planned.* Almost all systems did not stem "Antarctica" and "Antarctic" to the same stem
   - Sample Topic *378 Identify documents that discuss opposition to the introduction of the euro, the European currency.* All systems preferred matches to all the various hyphenated forms of "euro-something" instead of preferring the currency "euro"

3. All systems emphasize one aspect; missing another required term
    - 7 topics
    - Sample Topic *422: What incidents have there been of stolen or forged art?* All systems would do much better if the term "art" was emphasized
4. All systems emphasize one aspect; missing another aspect
    - 14 topics
    - Sample Topic *355 Identify documents discussing the development and application of spaceborne ocean remote sensing.* All systems needed to emphasize the aspect of "ocean". Much like Category 3, except some collection of expansion terms related to "ocean" would likely be needed to improve performance
5. Some systems emphasize one aspect; some another; need both
    - 5 topics
    - Sample Topic *363 What disasters have occurred in tunnels used for transportation?* Some systems emphasized disasters and others tunnels. Both were needed
6. All systems emphasize one irrelevant aspect; missing point of topic
    - 2 topics
    - Sample Topic *347 The spotted owl episode in America highlighted U.S. efforts to prevent the extinction of wildlife species. What is not well known is the effort of other countries to prevent the demise of species native to their countries. What other countries have begun efforts to prevent such declines?* All systems emphasized spotted owl and US efforts.
7. Need outside expansion of "general" term (*Europe* for example)
    - 4 topics, primary category
    - 4 topics, secondary category (topic was assigned a different primary category)
    - Sample Topic *398 Identify documents that discuss the European Conventional Arms Cut as it relates to the dismantling of Europe's arsenal.*
    - Sample Topic *448 Identify instances in which weather was a main or contributing factor in the loss of a ship at sea.* Systems needed to expand the concept of weather.
8. Need QA query analysis and relationships
    - 2 topics, primary category
    - 1 topic, secondary category (topic was assigned a different primary category)
    - Sample Topic *414 How much sugar does Cuba export and which countries import it?* Need notions of quantity and relationships between query terms
9. Systems missed difficult aspect that would need human help
    - 7 topics, primary category
    - 1 topic, secondary category (topic was assigned a different primary category)
    - Sample Topic *413 What are new methods of producing steel?* New methods very difficult
    - Sample Topic *393 Identify documents that discuss mercy killings.* mercy killings difficult (for example, need to match "right-to-die")
10. Need proximity relationship between two aspects
    - 1 topic, primary category
    - 5 topics, secondary category (topic was assigned a different primary category)

- Sample Topic *438 What countries are experiencing an increase in tourism?* As a primary focus, need aspect of increase. As a secondary focus, want aspects of increase and tourism to be close together.

The categories above are roughly sorted in order of increasing Natural Language Understanding (NLU) being needed to improve performance once it is understood a topic belongs in that category. Topics were placed in the least restrictive category (toward the top of the list) that would give substantial improvement if the problem could be addressed.

The assignment of topics to these categories does not address the problem of how difficult it is to automatically discover what category the topic belongs to. Thus it may be extremely difficult and require full NLU and world knowledge to distinguish those topics in, for example, category 9 from those in category 4. But if the system can distinguish those categories, possibly by using more information than is available in just the topic, it should be able to attack the missing aspect problem of category 4 in a straightforward fashion, while the missing aspect of category 9 topics will be very difficult to attack.

It should be noted that these categories and assignments of topics to categories are the results of one person's efforts. If 100 experts were asked to do the same exercise and given the same information (the filled in topic templates and the resulting discussions), there would undoubtedly be 100 different results. Nonetheless, some conclusions can be reached.

The first conclusion is that the root cause of poor performance on any one topic is likely to be the same for all systems. Except for the six topics of categories 1 and 5, all systems fail for the same reasons. Beadplot and other tools show that the systems are retrieving different documents from each other in general, but all systems were missing the same aspect in the top documents.

The other major conclusion to be reached from these category assignments is that if a system can realize the problem associated with a given topic, then for well over half the topics studied (at least categories 1 through 5), current technology should be able to improve results significantly. This suggests it may be more important for research to discover what current techniques should be applied to which topics, than to come up with new techniques.

Overall, most of the incoming hypotheses stated in the introduction turned out not to be true. Despite the fact that systems retrieve different documents, all systems tended to fail in the same way. In addition, the type of semantic relationship between aspects is not yet the primary cause of failure. There should be a lot of improvement possible without understanding relationships, though in the long-term, relationships will be necessary. Finally, understanding the semantics of the topic well enough to just identify the important aspects would seem to be crucial for many topics.

4 RIA Retrieval Experiments

The retrieval experiments of the RIA workshop were a massive investigation of how different systems vary while performing a single query expansion task, that of blind feedback. Blind feedback was chosen as the target task since

- It is known to have a high degree of topic variance. Within any one system, it works very well on some topics but hurts performance on other topics. Most systems find a mild average benefit to the use of blind feedback.
- Most systems have used it at some point in their past; thus the implementation effort required for experimentation is minimized
- It has a number of important parameter settings that systems in practice set to different values, and that can be changed easily

In the blind feedback task, systems automatically expanded the original query by adding terms that occur in documents (or passages) that the system thought were closely related to the query. On each topic, a system

1. Performed an initial retrieval with terms from the text of the original topic.
2. Without any user looking at them (thus "blind"), the system assumed that the top X documents were responsive to the topic and would be useful for expansion
3. The system chose N terms from the top X documents and added them to the original query terms.
4. All terms were reweighted

5. The new expanded query was re-run against the entire document collection again, and a ranking of the top documents was produced.

6. In a live system, these documents would be given the user. In the experimental setting, the ranking was evaluated based on the ranks of known relevant documents.

4.1 Overall Statistical Analysis

In the retrieval experiments, variations of each of the possible parameter choices were studied. These included the number of documents to draw expansion terms from (X), the number of expansion terms to add (N), the choice of the expansion documents, and the choice of the expansion terms. There is an inherent system performance of each system due to their weighting, indexing, and matching algorithms. The major goal of the analysis was to see if the variability due to topics can be separated from that inherent system-dependent variability. Different expansion approaches work well on different topics. If we can isolate out the topic-dependent effect, then we can start to learn what factors of each topic determine the success of an expansion approach. Then each system can adjust its approach and parameters based upon those topic dependent factors.

Somewhat more formally, evaluation scores can be explained in terms of the topic, the inherent system, and the run (system parameter settings).

$$p(t,s,r) \sim et + es + er + esr + etr + est + estr$$

where

$p(t,s,r)$	is the score,
t	is the topic,
s	is the system,
r	is the run,
et	is the topic effect,
es	is the system effect,
er	is the run effect,
esr	is the effect of the interaction between system and run,
etr	is the effect of the interaction between topic and run,
est	is the effect of the interaction between system and topic, and
$estr$	is the interaction of all three parameters, which is ignored here.

In the basic sets of experiments, we altogether had 150 topics, 7 systems, and about 100 different runs for a total of 105,000 data points. Our focus was to look at *etr*, the interaction of the topic and run. This can be used to classify topics according to what sort of approach and parameters should be used. Ideally, this classification can be matched to a classification based on topic information alone. In that case, we have an effective decision procedure for how to choose the approach and parameters on a per topic basis.

4.2 Understanding System Performance

The other major goal of the retrieval experiments was simply to increase human understanding of what is happening with query expansion and blind feedback. Most research groups have experimented extensively with blind feedback at some point or another, but because blind feedback is so topic and system dependent, it has been very hard to analyze why it works or doesn't work on particular topics. Most groups have been content to just optimize for maximum average performance.

As was said earlier, when query expansion improves performance, it tends to be because one or more of the following is added:

- synonyms
- one or two good related words
- a large number of related words that establish that some aspect of the topic is present (context)

	bf.0.0	bf	bf.20.20	bf.20.100
Albany	.126	.154	.139	.154
City	.186	.216	.213	.193
CLJ	.185	.210	.209	.192
CMU	.201	.225	.217	.218
FullCL	.169	.188	.196	.196
Sabir	.204	.226	.226	.225
UMass	.196	.235	.220	.234
Waterloo	.198	.228	.215	.211

Table 1: MAP scores for bf_base runs

- specific examples of general original query terms
- better weighting to original query terms

It is very likely that each of the five effects is of primary importance to some set of topics but not to other sets. Until we know how important each of these effects is, we can't adjust systems to improve expansion performance.

The goal here is to understand for a system what worked for individual topics as compared to all other approaches that this system or other systems tried. Given the problems caused by topic variability, it is much easier to compare against other system results than to attempt to judge whether an approach succeeded or failed on some absolute basis.

4.3 Retrieval Experiments

Each of the retrieval experiments done during the workshop is briefly described below. There was very little time for analysis of the experiments during the workshop; with the exception of the first introductory experiment, analysis is continuing now and will be presented in research publications in the future. Preliminary results are stated where they can be, but the detailed analysis and supporting evidence is deferred for the later publications.

For each experiment, a brief description, the goal, leader, participants, basic methodology, any preliminary results, and some of what the continuing analysis is studying are stated.

4.3.1 bf_base

- Description: Basic investigation of blind feedback
- Goal: Establish whether blind feedback works for the participating systems
- Leader: Andres Corrada-Emmanuel, UMASS
- Participants: All 8 IR systems (2 from Clarvoyance)
- Methodology: Perform 4 runs per group:
  1. No feedback at all; initial retrieval. bf.0.0
  2. Standard blind feedback run of system with whatever parameters the system normally uses. bf
  3. Set the number of documents used for feedback to 20, and the number of expansion terms to 20. bf.20.20
  4. Set the number of documents used for feedback to 20, and the number of expansion terms to 100. bf.20.100
- Results and Comments:

  All groups got reasonable average performance increases of between 10 and 20% using expansion as opposed to not (see Table 1). Some groups got mildly better performance expanding by a lot of terms as opposed to a few; other groups got mildly worse scores.

  The parameters used for the standard bf run, where each system could choose its own parameters, varied widely as can be seen in Table 2. Systems such as CLJ, which tended to add very specific terms, used comparatively few

documents and terms, while systems such as UMass, which added more general terms, used more documents and added more terms. CMU added a different number of terms for each topic, averaging an additional 412 terms per topic.

	num documents	num added terms
Albany	20	100
City	10	20
CLJ	6	30
CMU	10	hundreds
FullCL	6	30
Sabir	20	60
UMass	30	100
Waterloo	25	25

Table 2: Parameter choices for bf run

- Future Analysis: None

4.3.2 bf_numterms

- Description: Vary the number of terms added to original by blind feedback expansion
- Goal: Along with bf_numdocs, one of the two workhorse experiments investigating blind feedback parameters and variability
- Leader: Paul Ogilvie
- Participants: All 8 systems (two from Clarvoyance)
- Methodology: Perform 37 blind feedback runs with expansion based on the top 20 documents. Start by adding 0 terms (just reweight original topic) then add 1 term, 2 terms, ... , 20 terms, 25 terms 30 terms, ... , 100 terms
- Results: Average behavior different for the systems. All systems kept on improving on average as the number of terms increased from 0 to 15. As the number of terms continued to increase, some systems mildly improved further, other systems got worse. An oracle that chooses the best number of query terms to add based upon the results can improve results as much as 30%. There is significant possibility of system improvements if we can understand how to choose the best number of terms to add for a particular topic.

  On a per topic basis, the systems with continuous improvement as number of terms increased tended to have a bi-modal distribution, either near 0 terms should be added or near 100 terms should be added.

  Topics can be categorized by counting the number of added terms in the top 20 which actually improved performance as opposed to without the term. Strong improvements overall in expansion were strongly correlated with 5 or more helpful terms being added. For no topic for which most systems agreed that only 1 to 4 terms were helpful, did expansion help strongly overall.

  The above two points suggest that improvements across systems are coming from ensuring the context of the topic is represented in the documents, rather than adding a small number of good synonyms, examples, or related terms. But this needs to be analyzed much more thoroughly.
- Future Analysis: This is the major experiment which needs to be understood on a per topic basis in order to understand blind feedback expansion. The topic categorization based on number of helpful terms needs to examined carefully, and compared against all the other topic categorizations.

4.3.3 bf_numdocs

- Description: Vary the number of documents from which added terms are extracted in a blind feedback expansion
- Goal: Along with bf_numterms, one of the two major experiments in blind feedback parameterization
- Leader: Jesse Montgomery, Clairvoyance
- Participants: All 8 IR systems
- Methodology: Perform 36 blind feedback runs, expanding by 20 terms taken from a variable number of top documents. Start by considering 1 top document, then 2, 3, ... , 20, 25, 30, ... , 100
- Results: All systems had a single peak performance as the number of documents increased, but the peak location varied, and the effect of using more documents varied - some systems dropped off rapidly and others were more robust. There was no explicit correlation found between the optimal number of documents and any of query length, number of relevant documents, or how the top documents clustered. Analysis is continuing.
- Future Analysis: Again topics can be categorized by how often using more documents helped performance. That categorization is possibly correlated with categorization by how many terms helped performance. In addition, it should be interesting to categorize topics by what percentage of the top documents should be relevant in order for feedback to help. The bf_numdocs_relonly experiment described later shows that if all documents are relevant, then performance will increase as documents are added. Is there a percentage threshold above which adding documents is expected to help?
- Future Analysis: As well as number of documents, are there particular documents that in general helped blind feedback across all systems? Are there documents that hurt blind feedback across systems even though they are relevant? Can these documents that either help or hurt be characterized?

4.3.4 bf_swap_doc

- Description: Each system uses top documents found by initial runs of other systems instead of using its own initial run.
- Goal: Determine how much the initial retrieval strategy of each system affects whether blind feedback works.
- Leader: Tom Lynam, Waterloo
- Participants: All 8 IR systems participated
- Methodology: All 8 groups prepare a list of their initial retrieved documents in TREC results format. Then each group does 8 blind feedback runs, using each other's list of initial retrieved documents as the source of expansion terms, but using their own methods and default parameters to choose and weight terms. At the end, each group will have done a retrieval run based on Albany's top documents, a run based on City's documents, and so on for all 8 groups.
- Results and Comments:

  Some systems are much more sensitive to the initial set of documents than others as shown in Table 3. For example, reading the scores across the row for Waterloo or UMass, scores vary by at most 10% as the source of top documents change. But other systems, like FullClarit, vary by 50%. These differences in sensitivity are somewhat surprising, given the uniform improvement from just using blind feedback. One explanation is that some groups may get their feedback improvement from the reweighting of the original query terms, rather than the effect of term addition. This is still being analyzed.

  Another surprising feature is how often systems prefer to use documents from other systems rather than their own documents. For instance, all systems prefer to use Sabir's top documents rather than their own. Evaluation scores of Sabir's top documents are less than some other systems on average, though insignificantly. CMU prefers Sabir's documents by almost 20% over UMass's documents, while other systems consider them about the same. This preference is still being analyzed; despite efforts we do not have an explanation. In general,

	Albany	City	CLJ	CMU	FullCL	Sabir	UMass	Waterloo
Albany	.154	.184	.189	.188	.179	.191	.187	.173
City	.173	.217	.216	.218	.215	.224	.220	.201
CLJ	.204	.222	.207	.213	.203	.215	.216	.197
CMU	.171	.235	.227	.225	.231	.244	.205	.223
FullCL	.145	.218	.201	.211	.192	.214	.212	.183
Sabir	.207	.216	.203	.217	.209	.226	.216	.199
UMass	.220	.234	.235	.235	.245	.241	.235	.233
Waterloo	.221	.226	.238	.238	.237	.234	.238	.219

Table 3: MAP scores for bf_swap_doc runs

though, perhaps the use of other system's documents makes the retrieval more robust, for much the same reason that fusion of runs improves performance. The areas of weakness of one system are counterbalanced by strengths of another system. This is being investigated further.

As a side note, this experiment is a strong argument for doing this kind of work in a workshop environment. People found all kinds of minor bugs or incorrect assumptions in other system's runs. Some groups ran the experiment as many as 10 to 15 times as the initial runs of their and other groups changed.

- Future Analysis: So far we do not understand the effects of swapping documents. We need to look much more closely at the characteristics of the topics for which swapping top documents makes a large difference; it is not a question of just number of relevant documents being considered.

4.3.5 bf_swap_doc_term

- Description: Each system uses both top documents and expansion terms found by other systems instead of using their own documents and terms.
- Goal: Determine how much term selection algorithms of each system affect whether blind feedback works.
- Leader: Tom Lynam, Waterloo. Ting Liu, Albany
- Participants: 7 IR systems participated (CLJ did not)
- Methodology: This is a challenging experiment to perform. The root of the problem is the definition of expansion term. For many systems, this is available only in some internal form that needs to be translated into a form other systems can use. The problem is that different systems
  - Tokenize text differently
  - Stem text differently
  - may treat phrases as a single term

  After much debate and consideration of what was possible to do internally within the various systems, the following scheme was adopted. Each system creates the following three files:

  1. Results file containing the top initial 20 documents for each topic
  2. File qid_wt_word.added containing, for each topic, the 5 words added, in unstemmed form, where word might be a phrase
  3. File qid_wt_word.base containing, for each topic, the words of the base query, in unstemmed form, where word might be a phrase

  System A, when running the the expansion terms of system B, will create the query to be run from the set union of the words in its own base query (A's qid_wt_word.base) plus the words from B's expansion set (B's qid_wt_word.added). System A then stems the words and weights the stemmed terms as it normally does, based

upon the original query and occurrences in the top 20 documents of system B. Note that B's top documents must be used, since system B's expansion terms may not occur in system A's top documents, and therefore may not be able to be weighted from system A's documents.

The intersection of A's qid_wt_word.base and A's qid_wt_word.added is required to be empty, but that the intersection of A's qid_wt_word.base and B's qid_wt_word.added may not be empty. For example, B may consider "Antarctica" and "Antarctic" as two separate words, one of which may be an original query term and the other an expansion term, while A may stem the two words to the same stem, and thus will consider them only as an original query term. In this case, A will be expanding the query by only 4 terms instead of 5.

Each system A performs 2 runs for each other system B, both using B's top 20 documents and expanding by 5 terms. The first run expands by A's choice of 5 terms; just as in the bf_swap_doc experiment (except here 20 documents and 5 terms are specified.) The second run uses B's top 5 terms. Any phrases in B's list are handled in any fashion that A chooses; it was impossible to require anything else.

- Results and Comments:

	Albany	City	CLJ	CMU	FullCL	Sabir	UMass	Waterloo
Albany	.127	.140	.145	.139	.139	.147	.140	.137
City	208	.211	.215	.215	.215	.219	.220	.220
CLJ	.215	.221	.211	.222	.208	.228	.218	.215
CMU	.224	.221	—-	.214	.215	.227	.218	.211
FullCL	.172	.201	.190	.196	.181	.162	.200	.189
Sabir	.187	.208	.208	.216	.208	.221	.213	.202
UMass	.191	.192	.191	.192	.195	.204	.191	.182
Waterloo	.216	.217	.226	.224	.224	.222	.224	.210

Table 4: MAP scores for bf_swap_doc.20.5 runs

	Albany	City	CLJ	CMU	FullCL	Sabir	UMass	Waterloo
Albany	.125	.141	.140	.138	.131	.144	.141	.137
City	.191	.202	.210	.165	.202	.212	.199	.202
CLJ	—-	—-	—-	—-	—-	—-	—-	—-
CMU	.221	.219	.216	.214	.215	.229	.208	.212
FullCL	.182	.194	.207	.191	.181	.201	.190	.186
Sabir	.182	.201	.203	.200	.183	.220	.203	.195
UMass	.155	.176	.190	.155	.174	.179	.191	.163
Waterloo	.202	.198	.222	.215	.208	.224	.216	.211

Table 5: MAP scores for bf_swap_doc_term.20.5 runs

Table 4 gives the results for each system running the other systems' documents but using its own choice of terms. Table 5 gives each system running the other systems' documents and using the other systems' choice of terms. Comparing the two tables, there is very little difference except for a couple of FullClarit runs. The various systems chose quite different term lists even though they were dealing with the same document sources; only 15% to 25% of terms overlapped in general, and it was much less for FullClarit and Albany which added phrases in this experiment.

We conjecture that the exact choice of terms is not critical on average as long as related terms are added. The evidence here is not yet strong enough to support that conjecture, though.

- Future Analysis: There has been no topic analysis or categorization done for these runs. It will be interesting to examine those topics for which choice of terms does make a difference.

4.3.6 bf_numdocs_relonly

- Description: Vary the number of potential documents from which added terms are extracted in a blind feedback expansion, but actually add only relevant documents
- Goal: This is a paired experiment with bf_numdocs. The goal is to determine how much the non-relevant top documents hurt the expanded query.
- Leaders: Rob Warren, Waterloo. Ting Liu, Albany. David Evans, Clairvoyance
- Participants: All 8 IR systems
- Methodology: Perform 36 blind feedback runs, expanding by 20 terms taken from a variable number of top documents. Start by considering 1 top document, then 2, 3, ... , 20, 25, 30, ... , 100. For each run, delete all non-relevant documents from the top documents before query expansion. Thus, if the initial retrieval for a topic contains no relevant documents between ranks 11 and 20, then the 10 retrieval runs for sets 11 through 20 will be identical for that topic.
- Results: This is an upper-bound experiment. Among other things, it simulates having an actual user making relevance judgments from a set of top documents of size N, and using only those relevant documents for feedback As would be expected, all systems have a slow, monotonic growth in MAP as the size of the candidate set of documents increases. The upper limit of MAP differs substantially among systems. For example, CMU is .292, Waterloo is .316, and Sabir is .370. That gap is enormous; and should shed some light on differences between systems once it is fully understood.

  Extensive analysis of this experiment is continuing.
- Future Analysis: One interesting fact that was noticed during analysis is that there are certain relevant documents that hurt performance for all systems when they are included as a source for query expansion terms. It would be interesting if these documents could be automatically determined.

4.3.7 bf_pass_numterms

- Description: Vary the number of expansion terms added to the original query in a blind feedback expansion. The initial retrieval is of passages rather than entire documents, thus considerably less but presumably more focused text to serve as the source of expansion terms.
- Goal: Understand how passage retrieval differs from document retrieval in the expansion process. This was an initial experiment dealing with passages in preparation for more extensive Question Answering experiments.
- Leader: Zhenmei Gu, Waterloo/MITRE. Ming Luo, Virgina Tech/MITRE
- Participants: 4 IR systems - City, CMU, Waterloo, FullClarit
- Methodology: The same methodology as the bf_numterms experiment, except each system returns a passage instead of the entire document. Each system defines its own definition of passage; the only enforced requirement is that a set of passages be non-overlapping.

  The FullClarit and Waterloo systems already expand queries by considering passages; There runs are unchanged from the bf_numterms experiment.
- Results: Both CMU and City got very mild average improvement (1% to 2%) over the corresponding bf_numterms when averaged over all 36 runs. CMU improved uniformly over most runs; City was considerably more variable. In general, City improved for low number of terms added, but deteriorated for larger number of terms added. We conjecture that as City tried to draw more and more terms from a constant 20 small passages, it can no longer find good terms.

  One general observation is that the per topic performance with passages is more variable as the number of terms increases; possibly because rarer terms are being added from the passages as opposed to the documents

4.3.8 bf_swap_doc_cluster

- Description: This is the first of 3 small experiments in which the source of documents from which expansion terms are drawn is chosen using some outside criteria. This experiment was an upper bound experiment, clustering the retrieved set and choosing the cluster with the most relevant documents
- Goal: Investigate the effect criteria other than initial retrieval have on expansion performance.
- Leader: Jesse Montgomery, Clairvoyance
- Participants: 5 IR systems - Albany, City, FullClarit, Sabir, Waterloo
- Methodology: Perform blind feedback runs using all documents provided by the outside source. For this experiment, the documents are from a FullClarit initial run where the top N documents are clustered by the FullClarit system, and the best cluster is chosen. Two runs with values for N being 50 and 100. For $N = 50$, the number of documents per topic ranged from 2 to 45. For $N = 100$, the number of documents per topic ranged from 2 to 73.
- Results:

	N = 50	N = 100
Albany	.204	.221
City	.236	.236
FullCL	.222	.240
Sabir	.224	.249
Waterloo	.255	.271

Table 6: MAP scores for bf_swap_doc_cluster

The results are shown in Table 6. Analysis is still being done, but the most interesting point is that Waterloo was able to take advantage of the good clusters of documents, much more than other systems. The current conjecture is that the Waterloo expansion by passages within each document was able to pick out a common good text piece that was responsible for both relevance and the clustering.

4.3.9 bf_swap_doc_hitiqa

- Description: This is the second of 3 small experiments in which the source of documents from which expansion terms are drawn is chosen using some outside criteria. This experiment uses a set of documents determined as good by the Albany HITIQA system.
- Goal: Investigate the effect criteria other than initial retrieval has on expansion performance.
- Leader: Sean Ryan, Albany
- Participants: 5 IR systems - Albany, City, FullClarit, Sabir, Waterloo
- Methodology: Perform a blind feedback run using the documents provided by the outside source. For this experiment, the base initial set of documents was obtained by using the HITIQA NLP system to index and cluster a given initial set of documents. The HITIQA system matches passages against the query in a frame-based manner. The passages are then clustered with the documents being provided to other systems being the documents containing those clusters. Systems use all of the documents returned by HITIQA. The number of documents ranged from 3 to 72 per topic.
- Results:

  The results are shown in Table 7. Overall, the results were below standard blind feedback runs. One factor affecting performance is that HITIQA did a good job at finding good passages in long documents. This passage

	bf.hitiqa
Albany	.166
City	.197
FullCL	.189
Sabir	.179
Waterloo	.220

Table 7: MAP scores for bf_swap_doc_hitiqa

information was then thrown away and systems were only given the long documents themselves. With the exception of the passage-based Waterloo system, the long documents proved less useful. There was not enough time to repeat the experiment in a passage environment.

4.3.10 bf_swap_doc_fuse

- Description: This is the third of 3 small experiments in which the source of documents from which expansion terms are drawn is chosen using some outside criteria. This experiment fuses the retrieval sets of all systems together into a single retrieved set; thus emphasizing those documents that were retrieved highly by multiple systems.
- Goal: Investigate the effect criteria other than initial retrieval has on expansion performance.
- Leader: Tom Lynam, Waterloo
- Participants: 6 IR systems - Albany, City, CMU, FullClarit, Sabir, Waterloo
- Methodology: Perform 2 blind feedback runs using the documents provided by the outside source. For this experiment, one set of initial documents was obtained by fusing the bf.0.0 runs of all systems (the initial run without expansion). The second set of initial documents was the result of fusing the bf runs of all systems (the standard blind feedback run of each system). Each system used its standard blind feedback approach, choosing a system dependent number of documents to use as a source, and number of terms to add to the query.
- Results:

	bf.0.0	bf
Albany	.172	.183
City	.236	.228
CMU	.235	.235
FullCL	.237	.230
Sabir	.225	.210
Waterloo	.236	.243

Table 8: MAP scores for bf_swap_doc_fuse

The results are shown in Table 8. For the most part, the fused runs are noticeably better than each system's base bf run. The one exception is Sabir. As the swap_doc experiments showed, Sabir was the best source of documents for other systems to use; perhaps the fusion hurt the overall quality of Sabir's initial documents. This experiment reaffirms the conjecture that the use of documents from other systems can improve performance.

4.4 Retrieval Experiments Conclusion

The workshop completed a large number of retrieval experiments investigating the effectiveness of blind feedback and query expansion. From 100 to 200 runs per system were done on the standard test collection, investigating the system options of

- Number of documents to use for expansion
- Number of terms to add
- Source of the best documents to use
  - Own initial retrieval
  - Other system's initial retrieval
  - Fusion of other systems
  - Chosen by analysis and clustering
  - Use of passages instead of full documents

Each of these experiments is interesting in itself, but the overall purpose is not to optimize parameters or retrieval performance, but to understand the interaction of systems and topics. That analysis is only beginning.

5 Run Database

One of the major resources for future research produced by the workshop is the database of runs. Each group produced well over a hundred evaluated retrieval runs on the standard collection of 150 topics used in TRECs 6, 7, and 8, as described in the Experiments section. Then the major experiments were all rerun (replicated) for each group on the TREC 5 ad hoc task, about 95 runs. In addition, 2 key experiments (bf_numdocs and bf_numterms, about 73 runs) for each group were replicated on each of the TREC ad hoc tasks from TREC 1, 2, 3, 4. Finally, one run was made for each group on the merged document collection formed from the news articles in TRECs 1-8, using all available topics (1-450). Altogether, there are 4,088 run results in the database, taking up over 22 gigabytes of disk space. Zhenmei Gu and Luo Ming were responsible for the run replications, performing amazing feats as they were able to get all the different systems indexing and retrieving on all the different collections.

The replicated runs have not yet been examined in any detail; that lies in the future. The main purpose of the replicated runs is to validate the experimental analysis done on the results from the standard collection. We need to verify that our experimental conclusions are not dependent on the particular topics and documents of the standard collection, but hold true on other collections as well. In addition to the validation purpose, the replicated runs are themselves useful for research as described below.

The primary difficulty in studying topic and collection variability has been the fact that evaluated retrieval runs from a single version of a system on large numbers of topics have not been available. The 50 topics in a typical TREC experiment run on a single collection have not been sufficient. The results from the 400 topics run here will provide the first good test bed to look at topic variability of TREC style topics. 400 topics is still not enough to represent the entire universe of topics, especially given the rather stylized nature of TREC topics, but it is enough to investigate how topics group together, both in their characteristics and in their resulting search behavior.

The runs done for the merged document collection (TRECs 1-8 news articles) should be a useful resource for research in themselves, even though there are only 6 runs total. The standard blind feedback approach (bf) for each group was used to retrieve the top 5000 documents for each of the 450 topics. Only partial relevance judgments are available for each topic; only the documents from the two (out of five total) volumes of the TREC disks used during the year the topic was introduced were ever judged. Research that can be done using these runs includes

- Does retrieval improve when documents from outside the target collection are used for blind feedback? The results from the bf run here can be restricted to a particular TREC ad hoc task, and then compared against the results of the bf run only on that task.
- Does the ranking of systems for ad hoc retrieval on the same document collection agree with the rankings of systems for Question Answering? The document set used here is exactly the document set used for TREC 9 and 10 Question Answering. For several groups we have both the ranking results of IR topics 1-405 and the ranking results of QA questions 201-1393.
- Can we devise a valid evaluation methodology for comparing runs when we know we only have only very partial relevance judgments? This is an increasingly important topic as we develop new, much larger test collections with much more incomplete relevance information.

5.1 Individual Run Documentation

The results from each run are stored in a UNIX-filesystem database, keyed by the system, collection, and runname of the run.

- *System*: Each of the 7 groups participating in the workshop provided at least one system. Claritech provided two systems, *fullclarit* and *clarit*, the latter being a flexible stripped down java-based implementation of the full version (and denoted as CLJ above). Sabir and Waterloo had one or two special purpose systems in addition to their standard systems.
- *Collection*: Each collection name encodes the document set, the topic set, and the topic length. For example, the standard collection is named *v45.301-450.d* which indicates
  - Volumes 4 and 5 of the TREC disks (though not including the Congressional Record)
  - Topics 301-450
  - Use of only the description field of the topic

  All runs in the workshop used the topic description field only; basically one sentence describing the user's information need.
- *Runname*: The experiment dependent name of the run. For example, *bf_swap_doc_cmu.20.5* is a blind feedback run, using CMU's documents in place of the system doing the run (swapping documents), and basing the blind feedback on the top 20 documents, expanding the original query by adding 5 terms.

Every run has several mandatory files associated with it, possibly along with some optional files. Mandatory files:

- *results*: Standard TREC retrieval results format, 1000 top docs per topic
- *run*: The needed commands and actions for running this run
- *desc*: Two part description of what the run is. The first part is a free form text description; the second part is a set of parameter name/value pairs.
- *eval*: Evaluation of run. Automatically computed from *results* using a workshop-modified version of trec_eval. For each topic, 25 evaluation measures in a format of
  measure_name topic_id score
  The averages over all topics are included with a topic_id of "all".
- *run.html*: An html page automatically constructed from the other files

The standard optional files for each run:

1. *query*: The system's query, after expansion, for each topic in a system dependent form
2. *qid_wt_stem*: if appropriate and possible, a standard form of the system's query after expansion. The file is in relational format
   qid weight stem
   where *stem* is the system-dependent stemmed form of the term as it occurs in the expanded query with id *qid*, and *weight* is the system dependent notion of importance of the term.

6 Topic Categorization

One of the major goals for the workshop was to understand how topics differ from each other, and how this affects system performance. An initial approach analyzing this, done during the final week of the workshop, was automatically assigning topics to categories based upon performance and other measures. Some of the categories examined include:

1. Non-relevance-dependent:

- syntactic analysis of topic text (readability, idf)
- ranking of retrieved documents (Clarity, clusterability)
- comparison of ranks before and after feedback within system
- comparison of ranks from different systems or approaches

2. Relevance-dependent:
    - MAP (evaluation) score of the topic
    - How much blind feedback improves MAP score
    - How often individual added terms improve MAP score

Most of the categories are measure-based. Some measure gave a score to each topic, and these scores were used to roughly categorize topics according to the procedure described in the next section. After that, the various intersection of the categories was examined. Of particular interest was whether the non-relevance-dependent categories were related to, and could therefore potentially be used to predict, the relevance-related categories. If a system can use factual (non-relevance based) information to decide whether a topic is difficult or whether blind feedback would be helpful, then it can tailor its approach to the topic to improve performance.

6.1 Measures to Categories

We needed to categorize a topic given some measure on that topic (or possibly a measure on the topic given each of the systems). For the purposes of this initial investigation, we were interested in the extremes of each measure. Was the behavior of the topic different for those topics which were given a high score for the measure, as opposed to those topics given a low score for the measure? So given a measure on the topics, we divided the topics into three categories:

- Positive: The top 30 topics according to the measure score.
- Negative: The bottom 30 topics according to the measure score.
- Neutral: The remaining (90) topics

If the behavior (typically intersection with another measure categorization) was not different between the Positive category and the Negative category, then the measure was uninteresting for our present investigation. If the behavior was different, then the relationship between the measures bears further investigation.

Some of the more natural measures, such as MAP scores, were system dependent as well as topic dependent. It could be handled by averaging the measure across systems, but outliers and system blunders can strongly affect the average. Instead, the system dependence was handled by a voting mechanism in a two step process.

1. Step One: For each system, divide the topics into three categories.
    - PositiveScore: The topic has a measure score greater than the top X% (typically 20-30%) of the observations across all topics.
    - NegativeScore: The topic has a measure score greater than the top X% (typically 20-30%) of the observations across all topics.
    - NeutralScore: The remaining topics
2. Step Two: Vote on the above categorization among the systems (normally there were 7 or 8 systems).
    - Positive: Y% (Y $>$ 50%, typically 70%) of the systems called the topic PositiveScore in Step One.
    - Negative: Y% (Y $>$ 50%, typically 70%) of the systems called the topic NegativeScore in Step One.
    - Neutral: Y% (Y $>$ 50%, typically 70%) of the systems called the topic NeutralScore in Step One.
    - Mixed: None of the above (no agreement between systems on this topic)

The parameters X and Y were chosen by hand on a per measure basis to give roughly 30 topics in each of the PositiveVote and NegativeVote categories.

6.2 Categorization Results

There were a total of 20 categorizations done with 14 investigated in some detail, including one based upon the manual topic failure analysis. Much more work needs to be done, but several interesting results have already been discovered. The following result discussions look at the intersection of two categorizations and concentrates on correlation between the Positive (or PositiveVote) categories defined by the two measures.

Similar rankings among all systems vs blind feedback MAP

The rankings produced by the 8 standard runs were compared against each other by averaging the anchormap measure which compares similarity of a pair of retrieval rankings: The top Xxi (here 30) documents of Ranking A were used as the only relevant documents to calculate MAP scores of Ranking B. If those top X documents of A were near the top of B, then the rankings were similar.

The topic categories produced by the anchormap measure were compared against the categories produced by the top MAP scores. The Pearson correlation between the topics in the Positive groups was an extremely high .557. The topics for which the systems found the same top documents were indeed the topics that the systems got the best scores on. Out of the 30 topics with the most similar rankings, 19 of them were in the top 26 highest scoring topics and 0 topics were in bottom 24 scoring topics. Conversely, of the 30 topics with least similar rankings, 0 were among the top MAP scores and 9 were in the bottom 24 scores. Thus if different systems or approaches get similar top documents, then the topic is easy and standard techniques should work well.

Similar rankings among all systems vs blind feedback improvement

This categorization comparison was the same as above except instead of comparing anchormap similarities against the top scoring topics, they were compared against the topics for which blind feedback improved the most. Here the correlation among Positive categories was a very high .327. If systems or approaches get similar documents, then blind feedback is likely to help.

Anchormap similarity and like approaches can conceivably be used to detect and correct the problem of a system missing aspects of a topic. For instance, instead of anchoring the map score in the top documents of a base run and an expansion run, anchor it in only the top documents that have some threshold similarity to a topic aspect. The absolute value of the map score of a base run counting only the documents with high similarity to a topic aspect will indicate whether the aspect is being retrieved, and the anchormap similarity, given those documents with the aspect, of the base run and and expanded run will indicate whether the expansion is moving toward or away from an aspect.

Similar rankings between base and feedback vs blind feedback MAP

To explore the blind feedback improvement more, instead of comparing the similarity among the rankings of 8 different systems, the ranking similarity between the initial run and the blind feedback run of the same system was used. Topics were categorized by the voting procedure described above which chooses topics for which most systems agree have the same sort of ranking similarity. The correlation among positive groups was again a very high .371. If after blind feedback expansion, the top documents of the expanded search are still the top documents of the initial search, then the topic is likely to be successful. This makes sense, since the top documents of the initial search were used for expansion terms and weighting in the expanded search. If different documents were retrieved then it's very possible that the new search got off-topic by over-emphasizing one aspect of the top initial documents.

Similar rankings between base and feedback vs blind feedback improvement

This comparison was the same as above except directly comparing whether blind feedback improves performance. The Positive groups had a high correlation of .287, again suggesting that blind feedback should be used when the initial top documents remain stable in their rankings.

Clarity vs blind feedback MAP

The Clarity measure, developed at UMass [5], uses the topic and ranking obtained from a language model system to predict how easy a topic is. We ran the Clarity measure on the CMU base run, categorized topics, and compared

against MAP score categories. The correlation among Positive groups was .167. Since Clarity can predict hardness of a topic, this strongly suggests that the anchormap approaches, with a much higher correlation, should also be able to predict hardness. That remains for future work.

Note that it may be fairer to compare Clarity against MAP score of baseline systems instead of blind feedback systems. Doing so gives a correlation of .177, a mild improvement but in the same ballpark.

Clarity vs blind feedback improvement

It has never been claimed that Clarity can predict blind feedback improvement without modification of the Clarity measure. Indeed, our investigations showed a correlation among Positive categories of only .038. The correlation between the Positive Clarity category and the Negative improvement category was .098, substantially higher.

Topic rare term vs blind feedback MAP

If the topic contained a rare term, as measured by the maximum idf of all original topic terms, then it was more likely to be easy. The correlation between Positive categories was .299.

Topic rare term vs blind feedback improvement

If the topic contained a rare term, as measured by the maximum idf of all original topic terms, then it was not particularly likely that blind feedback will help. The correlation between Positive categories was .038, or roughly neutral. What was quite interesting was that the correlation between the Positive idf category and the Negative improvement category was .294 (like Clarity, higher than between Positive categories). For a very substantial number of topics with rare terms, blind feedback hurts.

6.3 Categorization Conclusions

Overall, the results of our our initial categorization efforts surpassed our expectations. We showed high correlations between a number of categories, including several described above that should be able to be transformed into a predictive process, that gives insight as to what sort of retrieval approaches are likely to be successful on a particular topic.

As yet, there are no real results comparing the categories determined by the manual failure analysis with the categories described above. There were too few topics in each failure analysis category to use the same procedure. A different approach needs to be developed.

7 Summary of Research Results and Suggested Future Work

There are many detailed results and suggested further work given in the sections above; these will not be repeated here. However, there are several broad areas that should be emphasized. These are drawn from the work above, and from the half-day review discussions that each two-week workshop session ended with.

1. Current research IR systems are failing for the same reason on individual topics. They are retrieving different individual documents, but have the same general classes of failure documents (whether non-relevant retrieved or relevant not retrieved).

2. Current system failures are dominated by presence or absence of topic aspects in the retrieved documents. The relationship between aspects, needed for factoid Question Answering, is not an important failure mode yet. This suggests that IR systems must do a better job of simply recognizing aspects of a topic, or of recognizing that the retrieved documents do not include an aspect of the topic.

3. The data is now available for understanding why blind feedback improves results. The five possibilities listed in this report's introduction can be looked at. Our preliminary work here indicates that when blind feedback works well across systems, it works because large numbers of terms (five or more) are helpful, possibly ensuring the context of a retrieved document is correct.

4. Automatic (non-relevance-based) categorization of topics is needed. Different topics have to be treated differently in the retrieval process. We've introduced some categories that need to investigated further, and others need to be added. We also have introduced a methodology for looking at whether those categories can be useful.

5. Categorizing topics by measuring the similarity of retrieval rankings of different approaches is both possible and informative. The anchormap similarity between rankings of several different approaches both predicts the hardness of the topic and identifies topics for which feedback should work. Topics that have retrieved sets that are comparatively stable using different approaches are more likely to be successful and more likely to improve using blind feedback.

   Other anchormap-like similarities of retrieval rankings should also be investigated. For example, comparing a full topic ranking against a ranking based on only one aspect of the topic will give a measure of the importance of that aspect to the retrieved set.

6. There is now massive data across several collections to support statistically differentiating the effect of the topic and the system upon results. Incorporating this with the automatic categorization of topics, and with the manual categorization due to failure analysis, should give insights as to how we can use different approaches with different topics.

7. At a lower level of analysis, the massive data should support finding the expansion source documents and expansion terms that most aid retrieval. The next question is determining the properties of these terms and documents that can be used to select the best candidate terms and documents in the future.

8 Conclusion

The RIA workshop presented a very special opportunity to the IR community to *start* work on understanding how and why systems vary in performance across questions (topics). Once there is a better understanding of this, then we will have more robust IR systems, which will in turn lead to better QA systems. The initial work has been done, what remains is further analysis of the results by the entire IR community.

Appendix A - System Descriptions

The participants were asked to write a paragraph or two describing the systems or algorithms used during the workshop.

CMU[9, 20]

Carnegie Mellon's contribution to the NRRC-RIA workshop used the KL-divergence based language modeling approach to text retrieval implemented in the Lemur Toolkit (Lemur). In this approach, queries and documents are modeled as unigram language models, or probability distributions over a vocabulary. Documents are ranked so that the documents whose probability distributions diverge the least from the query are higher in the list. The query expansion used was the divergence minimization approach (Zhai and Lafferty 2001). The divergence minimization approach estimates a language model that minimizes the divergence between a new query expansion language model and the feedback documents while using the collection language model as a controlling factor.

UMass[12, 8, 9]

The Center for Intelligent Information Retrieval at UMASS (CIIR) contributed three systems to the NRRC-RIA Workshop. They were all based on statistical language modelling techniques. The system used for the 'standard' run in the IR portion of the Workshop used the query-likelihood algorithm [Ponte and Croft, 1998] using unigram scoring and Dirichlet smoothing of document probabilities.

For the runs investigating the behaviour of feedback algorithms, we utilized a hybrid of query-likelihood and Relevance Models [Lavrenko, 2001] designed to fit the feedback strategies utilized by the other systems at the workshop. We performed a query-likelihood initial ranking of documents. This ranking was then used to build a Relevance Model for a query. The Relevance Model, which can be thought of as an expanded query, was then combined with the initial query to create a hybrid query. The weights used to combine the two queries were designed to never give the original query less than half the probability mass and approach the initial query as the number of feedback terms went to zero. That is, a run with one feedback term allowed, had most of its mass assigned to the initial query and gave some small probability mass to the top feedback term.

During the QA portion of the workshop, we utilized a dynamic passaging system that calculated the query-likelihood of fixed-byte-size passages within documents. This is a system that is designed to identify answer passages and cannot extract 'exact' answers as defined in the current TREC QA main task. For the initial passage extraction run that was used as input to the QA systems, we ranked 250-byte passages.

Waterloo [4, 19]

The MultiText Project, University of Waterloo

For participation in the NRRC workshop, the MultiText Project adapted passage-retrieval and term-extraction methods from their QA system to the task of blind-feedback query expansion. The MultiText passage-retrieval algorithm locates "hotspots" within the corpus where many query terms cluster in close proximity (Clarke et al., 2001). After stopword elimination and stemming, the terms from the description field are used by the algorithm to locate the top ranked hotspots. Feedback terms are extracted from these hotspots, a score is computed for each extracted term, and the highest scoring feedback terms are added to the original query set. This expanded query is then executed using the MulitText implementation of Okapi BM25 to return the top 1000 documents. Details may be found in the MuliText TREC 2003 paper, where the technique was used for their Robust track runs (Yeung et al., 2003).

Clairvoyance [6, 10, 7]

Clairvoyance Corporation used two systems: CLARIT, a commercial information management toolkit written in C++, and CLJ (CLARIT Java), a recently developed IR research toolkit built on top of CLARIT. The CLARIT system provides both indexing and retrieval functionality, as well as a wide range of other information management tools, including text classification and filtering, extraction, and summarization. CLJ consists of a set of retrieval functions (only) that run on top of a CLARIT index. CLJ was included in the experiments primarily because it is a more

suitable environment to make the rapid modifications required for the NRRC workshop. In practice, we found the CLARIT toolkit was also flexible enough to complete the tasks required for the workshop within the time constraints. In addition, Clairvoyance contributed its Analyst Workbench, a graphical interface for text mining highly suitable for detailed failure analysis on individual topics and exploratory data analysis.

There are two special distinguishing characteristics of the CLARIT indexing process. CLARIT uses NLP for tokenization, storing individual words, noun phrases, and sub-phrases as index terms. Terms can be filtered by part-of-speech categories at indexing time; all the major content-bearing categories were used in the NRRC experiments. CLARIT indexes on paragraph-sized "sub-docs" (passages) instead of full documents, typically varying in size between 8 and 20 sentences and averaging about 12 sentences in length. Document score/rank in retrieval is determined by the highest scoring sub-doc in a document. Passages of different (often smaller) sizes can be re-created on the fly for query expansion.

CLARIT retrieval in the NRRC-RIA experiments used a traditional frequency-normalized TF-IDF algorithm. For query expansion based on the top k documents, CLARIT used the method of Evans and Lefferts (1994,1995): feedback is based on all sub-docs ranked equal to or higher than the highest scoring sub-doc found in the k-th retrieved document. The term selection (expansion) algorithm is based on either Rocchio or a customized method called "Prob2" (or "CLProb"), a variant of the traditional probabilistic term weighting algorithm (Milic-Frayling et al. 1998). CLJ runs with a slightly modified version of Okapi BM25 for standard retrieval and Prob2 for query expansion. CLJ operated with an index based on individual words only (no noun phrases) for the NRRC experiments. In both CLARIT and CLJ, expansion terms added to the source query are given differential weights based on the score/rank they receive under the selection algorithm.

Albany[16, 17]

SUNY Albany's contribution to the NRRC-RIA workshop was the HITIQA system, an analytic question answering system developed under ARDA AQUAINT program. HITIQA uses a document retrieval engine to fetch initial set of documents from a database. At RIA workshop, we used an old version of Cornell's SMART information retrieval system. It was modified during the workshop in order to participate in the document swapping retrieval experiments.

The HITIQA QA capabilities were utilized during the last session of the workshop to test the effects of the different retrieval approaches on the effectiveness of question answering. HITIQA is an interactive open-domain question answering technology designed to accept complex analytic questions in natural language. Many of the TREC topics used in RIA experiments could be considered as synopses of reasonably complex analytic questions. The interactive features were not used during RIA.

Typically, top 50 documents retrieved from the database are passed for answer search within HITIQA. In addition to using SMART output, HITIQA also accepted external document sets provided by all the other retrieval systems participating in RIA. HITIQA answer search includes segmenting the documents into passages, and then clustering these passages into a small number of tight topics. Representative passages from each topical cluster are subsequently mapped onto templates (called frames) which identify key topical relations and their attributes. Frame-level comparison with the input question determines the degree of fit for each frame. Frames with more than 1 attribute mismatch with the question are not considered part of the answer (Small et al. 2004). In the interactive mode of HITIQA, conflict frames are negotiated with the user through a clarification dialogue, which may result in changes to the answer space. For example, frames may be added to the answer if the user decides to accept or override their matching conflicts with the question (Strzalkowski et al., 2004). Since the clarification dialogue was not used in RIA experiments, the initial answer space produced by HITIQA was also the final answer.

The effectiveness of the question answering process was measured by the number of frames comprising the answer obtained from a given set of retrieved documents. The QA process was at its most effective when the size of the answer space was maximum. Separate statistics were collected for exact-match frames (zero-conflicts) and for one-conflict near miss frames, as well as for the combined set.

City [14, 13]

City University's contribution to the NRRC-RIA workshop used the Robertson/Sparck Jones Probabilistic model (1976), implemented in the Okapi System. In this model indexed terms are weighted independently on the basis of their estimated (or probable) relevance. The BM25 weighting function was used for all experiments (Robertson

et al, 1995). The Term Selection Value devised by Robertson (1990) was used for term selection in pseudo relevance feedback experiments. The full range of the Okapi BSS was used to support the experiments including passage processing and term extraction.

Sabir [18, 2, 3]

SMART is a flexible IR research engine based on the vector space model as developed by Gerard Salton[1,2]. Documents and topics are broken into their component words and phrases, and are then statistically weighted for importance and matched. The version and parameters choices used in RIA were kept as simple as possible to allow full understanding of the effects as algorithms changed within RIA. In fact, the settings and algorithms were those used for the Cornell TREC 4 base run nine years ago[3]. The only slightly non-standard setting used in RIA was the use of SMART statistical phrases.

References

[1] C. Buckley. trec_eval IR evaluation package. `ftp://ftp.cs.cornell.edu/pub/smart`.

[2] C. Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Computer Science Department, Cornell University, Ithaca, New York, May 1985.

[3] C. Buckley, A. Singhal, M. Mitra, and G. Salton. New retrieval approaches using SMART: TREC-4. In D. K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 25–48, October 1996. NIST Special Publication 500-236.

[4] C. L. A. Clarke, G. V. Cormack, and T. R. Lynam. Exploiting redundancy in question answering. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 358–365, 2001.

[5] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the Twenty-Fifth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, 2002.

[6] D. A. Evans and R. G. Lefferts. Design and evaluation of the CLARIT-TREC-2 system. In D. K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 137–150, 1994. NIST Special Publication 500-215.

[7] D. A. Evans and R. G. Lefferts. CLARIT-TREC experiments. *Information Processing and Management*, 31(3):385–395, 1995.

[8] V. Lavrenko and W. B. Croft. Relevance-based language models. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, 2001.

[9] Lemur. The lemur toolkit for language modeling and information retrieval. `http://www-2.cs.cmu.edu/~lemur/`.

[10] N. Milic-Frayling, C. Zhai, X. Tong, P. Jansen, and D. A. Evans. Experiments in query optimization: The CLARIT system TREC-6 report. In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, pages 415–454, 1998. NIST Special Publication 500-240.

[11] P. Over. Beadplot. `http://www.itl.nist.gov/iaui/894.02/projects/beadplot/`.

[12] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In W. B. Croft, A. Moffat, C. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[13] S. Robertson. On term selection for query expansion. *Journal of Documentation*, 46:359–364, 1990.

[14] S. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, May-June 1976.

[15] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3) [Proceedings of TREC-3]*, pages 109–126, 1995. NIST Special Publication 500-225.

[16] S. Small et al. A data driven approach to interactive question answering. In M. T. Maybury, editor, *New Directions in Questions Answering*. AAAI/MIT Press, 2004. To appear.

[17] T. Strzalkowski et al. Question answering as dialogue with data. In T. Strzalkowski and S. Harabagiu, editors, *Advances in Open-Domain Question Answering*. Kluwer, 2004. To appear.

[18] D. Williamson, R. Williamson, and M. Lesk. The Cornell implementation of the SMART system. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 2, pages 43–44. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1971.

[19] D. L. Yeung, C. L. A. Clarke, G. V. Cormack, T. R. Lynam, and E. L. Terra. Task-specific query expansion. In E. M. Voorhees, editor, *The Twelfth Text REtrieval Conference (TREC-12)*, 2004. forthcoming.

[20] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, 2001.