FXPAL Interactive Search Experiments for TRECVID 2007

John Adcock, Jeremy Pickens, Matthew Cooper, Lisa Anthony,
Francine Chen, and Pernilla Qvarfordt
FX Palo Alto Laboratory
Palo Alto, CA 94304
{last name}@fxpal.com

In 2007 FXPAL submitted results for two tasks: rushes summarization and interactive search. The rushes summarization task has been described at the ACM Multimedia workshop [1]. Interested readers are referred to that publication for details. We describe our interactive search experiments in this notebook paper.

1 Summary of submitted runs

We submitted six interactive search runs for TRECVID2007, including 2 single user and 4 collaborative runs. In one single-user run (FXPAL_MMA) the searchers had access to all resources including text search, text similarity, image similarity, and concept similarity search. In the other single-user submission (FXPAL_MMV) the transcripts were not available and only features derived directly from the audio-visual content were available. This is a typical baseline for the manual and automatic search systems, but has been a less popular variation among the interactive search submissions. The other submitted runs (FXPAL_CO*) employed a novel real-time, multi-user, collaborative search system. The complete set of submitted runs in priority order with system names and brief descriptions:

Submission	**MAP score**	**Description**
FXPAL_CO	0.2376	15 minute collaborative search
FXPAL_CO15	0.2377	15 minute collaborative search with post-processing bug fix
FXPAL_MMA	0.2076	Single-user search with text available
FXPAL_CO11	0.2035	Collaborative search with simulated stop at 11.25 minutes
FXPAL_MMV	0.2031	Single-user search with no text available
FXPAL_CO07	0.1563	Collaborative search with simulated stop at 7.5 minutes

Table 1: MAP scores for the search submissions in performance order. MAP scores are shown to 4 digits to expose slight differences which may or may not be statistically significant.

We used 4 searchers to complete our 2 single user runs. Each searcher performed 6 topics for each run. 2 of our searchers had experience in previous years of TRECVID search, and two did not. In addition to the traditional single-user runs we also submitted 4 multi-user, collaborative runs. In these submissions two users worked simultaneously to complete the topics. The same 4 searchers who performed the single-user searches were joined into teams and each of the four teams performed 6 of the topics (each user performed 12 total topics). One 15 minute interactive session was performed for each topic, but the submitted runs simulate the termination of the session at shorter times. All runs were fully interactive, type A. Table 1 lists the runs in performance order. We have yet to perform statistical significance analysis on these results but similarly small differences in 2005 proved to have a very high level of statistical significance.

It may be instructive to note that the collaborative runs were performed first, followed by the single-user runs with text, and finally the single-user run without text. The non-text run may therefore benefit from learning effects.

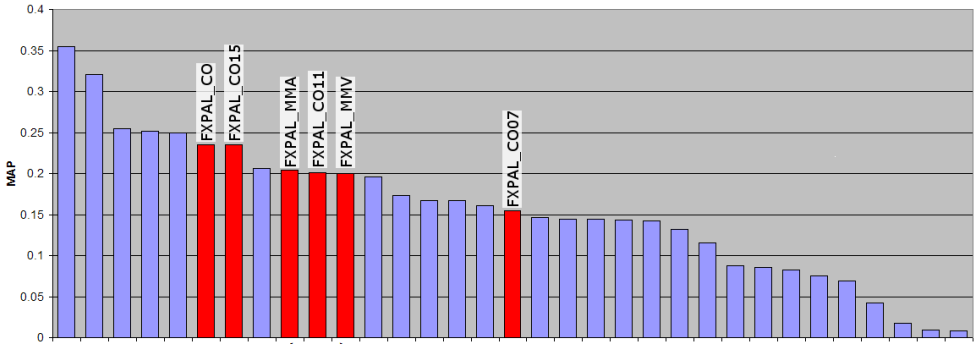

Figure 1: MAP performance of all interactive search submissions with FXPAL submissions labeled.

2 Single User Interactive Search

The MediaMagic interface was designed for efficient browsing and rich visualization of search results, and is largely unchanged from previous years [3, 4, 5]. It serves as both the sole interface for the single user runs and a component of our two user collaborative search system.

2.1 Data pre-processing

We perform a completely automatic pre-processing step to identify topic or story units to augment the reference shot boundaries [6]. These story segments provide the basic unit of retrieval during queries. To accomplish this segmentation we use the reference shot boundaries and the ASR transcripts [2] as described in [7]. This year we performed a run without any transcripts. For this run we use our semantic concept vector in place of the text for story segmentation. In preparation for interactive operation text indices are built for both the shot-level and story-level segmentations using Lucene [8] (for keyword search) and our latent semantic indexing system (for fuzzy text search) Color correlograms [9] are pre-computed for each shot thumbnail image. For each shot thumbnail image, surf descriptors with their y location are computed [10] and then quantized into 200 bins using online k-means [11]. The quantized descriptors are used together with the color correlograms as the features for semantic concept detection.

2.2 Search Engine

Queries are specified by a combination of text and images. The searcher can choose an exact keyword text search, a latent semantic analysis (LSA) based text search, or a combination of the two whereby the keyword and LSA-based retrieval scores are averaged together to form a combined score. We use only the provided ASR transcript to provide text for story and shot segments. The exact text search is based on a Lucene [8] back end and ranks each story based on the tf-idf values of the specified keywords. In this mode the story relevance, used for results sorting and thumbnail scaling and color coding as described in following sections, is determined by the Lucene retrieval score. When the LSA based search is used [14], the query terms are projected into a latent semantic space (LSS) of dimension 100 and scored in the reduced dimension space against the text for each story and each shot using cosine similarity. In this mode, the cosine similarity determines the query relevance score. In our application the LSS was built treating the text from each story segment as a single document. When determining text-query relevance for shots, each shot gets the average of the retrieval score based on the actual shot text and the retrieval score for its parent story. That is, the shots inherit text relevance from their stories. Image similarity is provided based on color correlograms [9]. Any shot thumbnail in the interface can be dragged into the query bar (Figure 2 B)and used as part of the query. For each shot thumbnail the color correlogram is compared to the correlogram for every shot thumbnail in the corpus. The maximum image-similarity score from the component shots is propagated to the story level. The document scores from the text search and image similarity are combined to form a final overall score by which the query results are sorted. A query returns a ranked list of stories.

2.3 Concept Similarity

We use SVM-based concept detectors for the 35 lscom-lite concepts to provide concept-based similarity measurements. Each shot has an associated 35 element vector describing the posterior probability of each

of the high-level features. For the concept distance between two shots we use the mean absolute distance (normalized L1) between their concept vectors.

2.3.1 Detector construction

We construct single concept detectors for the lcsom-lite concept set using support vector machines (SVMs). First we extract keyframes from each shot in the reference segmentation by minimizing the chi-squared distance between each frame histogram and the centroid for the the shot. We compute YUV color histograms and image descriptors for each keyframe as follows. We compute 32-bin global frame histograms, and 8-bin block histograms using a 4×4 uniform spatial grid for each channel. We also use the SURF features described in Section 2.1 and [10].

We use reduced training sets for parameter tuning and classifier training. For each concept we generate a separate training set by randomly downsampling the set of training examples. Denote the positive and negative training examples used for classifier construction by $\mathcal{P}$ and $\mathcal{N}$, respectively. Then

$$\begin{aligned} |\mathcal{P}| &= \min(990, |\{\text{all positive samples}\}|) \\ |\mathcal{N}| &= \min(1800, 9 \times |\mathcal{P}|) \quad . \end{aligned}$$

The choices for these training set sizes were not systematically optimized. Given the full training set $\mathcal{T} = \mathcal{P} \cup \mathcal{N}$, we perform a basic parameter optimization via grid search using the Python routine provided with the distribution of LibSVM [15]. Specifically, we learn C,which is the penalty for misclassifications, and γ, which scales the radial basis kernel function used by the SVM. We then train three separate SVMs using the learned paramters. For each we use different training sets by resampling the development data using the proportions of positive and negative examples described above. After training the SVMs we combine their probabilistic output predictions by averaging.

2.3.2 Deployment

During interactive operation the user can choose to perform a "find similar" operation on a set of selected shots. This action uses the same components that are used at the end of the interactive session (described in section 2.5. Two similarity measures are combined; one between the text of the selected segment(s) and those of candidate stories, and one between the concept vectors the selected segments and those of candidate stories. The text-similarity is the cosine distance between the text of the selected segment(s) and the text of each candidate segment. The concept distance is the minimum distance between the concept vectors of the example shots and the concept vectors of each candidate segment. The two similarity scores are averaged together to create a similarity score for each candidate segment.

New in 2007, the user can alternatively choose to perform a "find similar looking" operation on a set of selected shots. In this operation the selected shots are used to perform an image-based search using color correlograms. It is equivalent to putting all the selected shots in the image-query area and clearing the text search box, and thus isn't extending the capabilities of the system but rather provides a significant shortcut.

2.4 Interface Elements

The interactive search system is pictured in Figure 2. The TRECVID test question and supporting images are shown in section C. Text and image search elements are entered by the searcher in section B. Search results are presented as a list of story visualizations in section A. A selected story is shown in the context of the video from which it comes in section E and expanded into shot thumbnails in section F. When a story or shot icon is moused-over an enlarged image is shown in section D. When a video clip is played it is also shown in section D. User selected shot thumbnails are displayed in section G.

Shots are visualized with thumbnails made from the keyframes computed using the reference shot segmentation and histogram features as in Section 2.3. Story thumbnails are built in a query-dependent way. The 4 shot thumbnails that score highest against the current query are combined in a grid. The size allotted to each portion in this 4- image montage is determined by the shots score relative to the query.

Semi-transparent overlays are used to provide three cues. A gray overlay on a story icon indicates that it has been previously visited (see Figure 2 A and E). A red overlay on a shot icon indicates that it has been explicitly excluded from the relevant shot set (see Figure 2 F). A green overlay on a shot icon indicates that it has been included in the results set (see Figure 2 F). A horizontal colored bar is used along the top of stories and shots to indicate the degree of query-relevance, varying from black to bright green. The same color scheme is used in the timeline depicted in Figure 2 D.

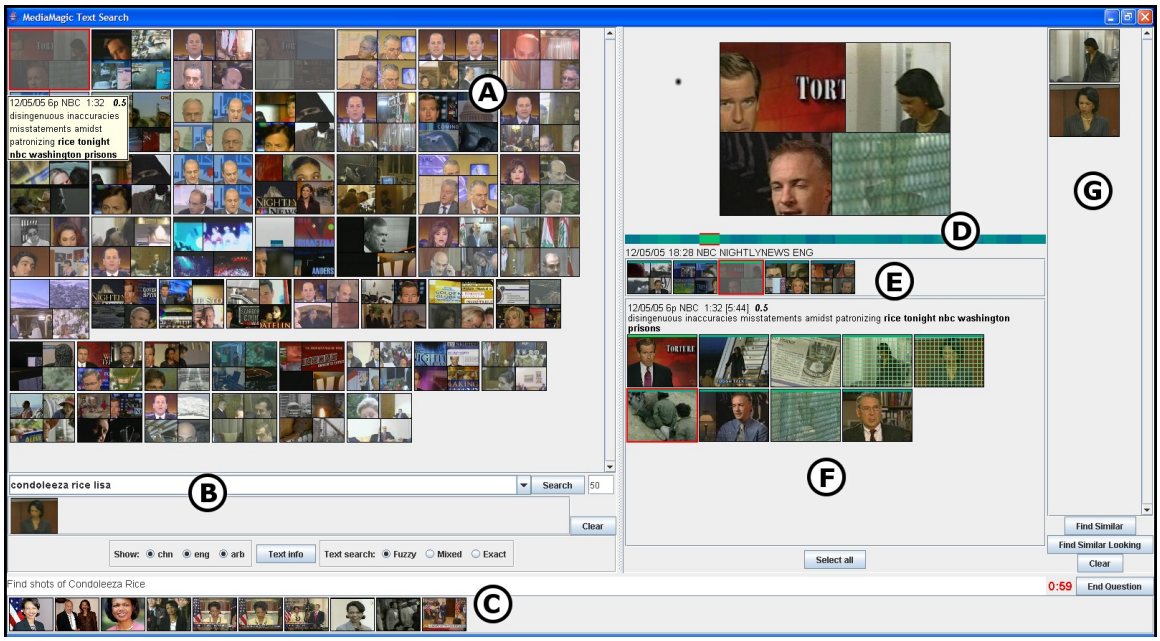

Figure 2: Interactive system interface. (A) Story keyframe summaries in the search results (B) Search text and image entry (C) TRECVID topic display (D) Media player and keyframe zoom (E) Story timeline (F) Shot keyframes (G) Relevant shot list

An optionally displayed dialog provides information about the underlying transcript and text query operation. The dialog shows the transcript from the selected shot or story along with terms related to the query (determined from the latent semantic space) and query terms that are not contained in the dictionary. Also the dictionary is displayed in a scrolling window allowing the user to browse the available terms.

2.5 Post-Interactive Processing

When the searcher decides to end the task or when the 15 minute allotted time expires, the search system performs an automated search process to fill out the remaining slots in the 1000 shot result list.

Three methods are used to identify and rank candidate shots for the post-interactive portion of the system operation.

Bracketing The shots neighboring (or bracketing) the user-identified relevant shots are added to the result list even if they were marked as not-relevant by the user.

LSA-based Text Similarity In this method the text from the shots that have been judged by the searcher to be relevant is combined to form a single LSA-based text query. This query is applied to the unjudged shots and the highest scoring ones retained for the result list.

Concept Similarity In this method the concept vector of a shot is compared against the concept vectors of the marked relevant and not-relevant shots. For each group (relevant, not-relevant) the minimum distance is computed, yielding a positive and negative similarity measure for each candidate shot.

First bracketing is performed, and then the remaining unjudged shots are ranked by an equal weighting of concept similarity and text similarity to form an ordering from which to select likely shots. Shots judged non-relevant by the user are excluded from the results (except for the bracketing step which may include a shot in the results despite a user judgement to the contrary).

2.6 Text-free run

We submitted a run that made no use of text in the system. This was accomplished by substituting the shot-indexed vectors of concept detector outputs in place of latent space text vectors in the story segmentation step. As described in Section 2.3 the concept detectors use only visual features.

3 Collaborative Search

The FXPAL system for interactive search opens a new direction for real-time retrieval: Synchronous, Explicit, Algorithmically-Mediated Collaboration.

The terms "collaboration" or "collaborative search" are overloaded with many meanings, ranging from from multiple searchers working separately in parallel, but with shared interface awareness [12] to multiple users sharing a single interface [13]. It has also been used to refer to collaborative filtering and personalization, the "Web 2.0", asynchronous approach to collaboration in which aggregate crowd behavior is used to steer the individual searcher toward the most relevant pieces of information by helping them find information that previous users have already discovered. An individual, working alone, is implicitly boosted by ("collaborated with") prior search behaviors of the crowd or community.

The problem with the aforementioned crowd-based approaches is not only that there will be large numbers of documents in a system with no prior user attention, thus rendering such approaches ineffective, but the intentionality or information need of the crowd might not match the need of the current searcher. Additionally, the problem with the aforementioned interface-only approaches to collaboration is that manual effort is still required for the searcher to correlate the information displayed with his or her own search efforts. While awareness of one's co-searcher is an important first step, interface-only solutions still require too much cognitive load to reconcile and integrate one's own activities with the opinions and actions of teammates.

Therefore, our vision is for a collaboration mechanism wherein searchers, rather than collaborating implicitly with crowds, collaborate explicitly with each other in small, focused search teams. Furthermore, the activities of each searcher are mediated algorithmically. The information that one member finds is not just seamlessly presented to other team members. It is used by the underlying system to automatically influence and subtly alter (for the better) the otherwise independent search behaviors of one's team members.

This is a radical perspective shift for search collaboration that opens up new realms of possibility for both information retrieval algorithmics and HCI design to support such scenarios. Explicit collaboration allows a fundamental conceptual shift in system design, away from algorithms and interfaces that support re-finding and re-discovering (crowd-based collaboration) to algorithms and interfaces that support exploration and the discovery of information that no other member has yet found, but that is relevant to the overall information needs and activities of the team.

Toward this end we describe our 2007 TRECvid system for algorithmically mediated, collaborative, synchronous, exploratory search: "Cerchiamo". It comprises a set of interfaces and displays, enabling rapid query iteration and collection exploration, along with a middleware layer for handling traffic and an algorithmic engine optimized for collaborative exploratory search. This system allows a small group of focused information seekers to search through a collection of information in concert.

The collaborative exploratory search system provides tools and visualization support to focus, enhance, and augment searcher activities. The system provides exploratory feedback not only based on the individuals search behavior, but on the current, active search behavior of ones fellow searchers. Searchers can, by interacting with each other through system-mediated information displays, help each other find relevant information more efficiently and effectively. In addition, each searcher on a team may fill a unique role, with interface and display components optimized for that role: query origination, results evaluation, and results partitioning are a few examples of types of roles that a search team might comprise. We discuss team patterns and role types in more detail below.

3.1 System Architecture

The architecture for our collaborative system is generic and can be implemented for other types of collections than video collections. It consists of three parts: the User Layer, the Regulator Layer, and the Algorithmic Layer (see Figure 3).

3.1.1 User Layer

The user layer contains all the input and output devices for human-computer interaction. Within the user layer is the MediaMagic video search interface for issuing image, text and concept queries and browsing results. An RSVP interface is used for rapid visual display and relevance assessment of video shots. Finally, a shared display contains displays the collaborated activities of the two users, as well as relevant information such as system suggested queries, based on the activities of both users.

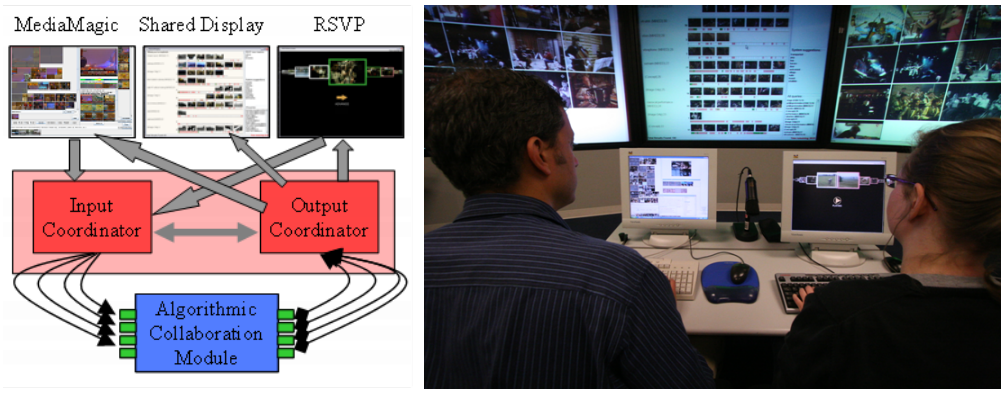

Figure 3: Collaborative System Architecture (left); Two collaborating searchers in action (right)

3.1.2 Regulator Layer

Within the regulator layer there is an input regulator and an output regulator. The input regulator is responsible for intercepting searcher activities, such as queries and relevance judgments, and contains coordination rules that then call the appropriate subset of algorithmic collaboration functions, using the appropriate data at the appropriate time. In effect, the input regulator enforces a policy that allows the users to act in certain predetermined collaborative roles. The output regulator is similar, accepting information from the algorithmic layer and routing the correct information to the appropriate user or information display at the appropriate time.

3.1.3 Algorithmic Layer

The algorithmic layer consists of a number of functions for combining the activities of two or more searchers, to produce documents, rankings, query suggestions, or other pieces of information relevant to the search. The key function of this entire algorithmically mediated collaborative search architecture is to ensure that the best information flows seamlessly to the right searcher at the right time, so that they can be the most effective in completing their search task. This should happen with little to no extra effort from the other collaborators.

Figure 4: Shared Display interface (left) and RSVP interface (right)

3.2 Collaborative Search Roles (Specializations)

The synchronous and explicit nature of the collaboration enables searcher specialization. In our system, collaborating users adopt the specialized, complementary roles of Prospector and Miner, supported respectively by our existing MediaMagic and newly developed RSVP interfaces and the underlying algorithms connecting the two interfaces. The role of prospector is designed to allow a user to open up new avenues for exploration

into the collection, while the role of miner insures that richer veins of information are more quickly and effectively explored.

The system functions as such: whether it is a text query, an image query, a concept query or some combination of the above, when a MediaMagic user enters a query, the input regulator runs that query against a search engine, stores the list of results, and routes the list back to both the MediaMagic client and the shared display via the output regulator. When the MediaMagic client enters more queries the same thing happens. Any shots that either the MediaMagic client or the RSVP client subsequently mark relevant are passed to the input coordinator as well and stored.

When the RSVP client makes a request for the upcoming best set of shots (30 at a time) the input regulator kicks into action. First, it has kept track of which shots the MediaMagic user has already examined and makes sure not to send those shots to the RSVP user to avoid duplication of effort. But more important than the obvious and necessary de-duplication is the call it makes to the collaborative algorithmic layer. For all the shots that have so far been retrieved by the multiple MediaMagic queries, but have not yet received any search team attention, the collaborative algorithm decides which shots to feed to the RSVP client. In the next section we will describe this algorithm, but the point is that the RSVP client does not have to manually decide which shots to comb through, reducing the cognitive load. Moreover, the shots that get fed to the RSVP user change constantly, depending on what the MediaMagic user is doing; searcher activities are algorithmically coordinated.

The third major part of the interface, the shared display, consists of a large screen in the front of the room. It shows continually updating data about the current search run: the queries that have been issued in the course of the run, the relative ranks of the shots that were retrieved by those queries, and the associated relevance, non-relevance, or unseen state of each of those queries. It also scrolls through visual thumbnails of the relevant shots that have been retrieved by a particular query. Most important to the overall collaboration is the area showing collaborative system-suggested query terms. It is through this interface that the activities of the RSVP user are fed back to the MediaMagic user. This will be described in the next section.

3.3 Collaborative Algorithm

The main purpose of the collaborative algorithm is to alter, in real time, the information presented to each search team member, based on the activities of all members. There are two parts to this algorithm. The first part is how the shots fed to the RSVP user are chosen. The second part is how the system-suggested query terms are fed, via the Shared Display, to the MediaMagic user. (Note that the algorithm we describe here is only one of many possibilities.)

3.3.1 RSVP Shot Priority

This algorithm determines the order in which unseen shots are fed to the RSVP user. The foundation of the algorithm is weighted Borda count fusion. When a query is issued, the higher in that ranked list an unseen shot appears (Borda count), the greater its position in the priority queue of shots to send to the RSVP user. However, this rank information is tempered by the overall quality of the query to which a shot belongs. Two weighting factors are used, query freshness (w_f) and query relevance (w_r).

Query freshness is given by the ratio of unseen to seen results that have been retrieved by that query: $w_f = \frac{unseen}{seen}$. Query relevance is given by the ratio of relevant to non-relevant shots that have been found in the seen results for a query: $w_r = \frac{rel}{nonrel}$. These two factors counterbalance each other. If a query has been successful in retrieving a lot of relevant shots, you want the RSVP user to continue examining those shots (relevance). However, if most of the shots from that query have already been examined, you want to start to give other queries priority (freshness). Similarly, a query with only a few examined shots receives high priority (freshness). However, after a few sets of shots have been examined and turn out not to be relevant, remaining shots from that query are downplayed (relevance). Underscoring both the relevance and freshness weights is the original Borda count (rank) given to the shot. If a shot is found in more than one query queue, its weighted value is summed.

3.3.2 Collaborative System-Suggested Term Selection

While the actions of the MediaMagic user (queries performed, shots examined, shots marked relevant) have an effect on the ordering of the shots fed to the RSVP user, the actions of the RSVP user (shots examined, shots marked relevant) have an effect on the system-suggested query terms fed to the MediaMagic user.

The basic idea is similar to above, with relevance and freshness weights. However, instead of a Borda count, a "term frequency in the query" (tf_q) count is used, instead. When a query is issued, all the term counts associated with all the separate shots retrieved by that query are summed. The higher the frequency of that term in the retrieved set, the higher its priority for appearing as a system-suggested term. However, this tf_q count is tempered by the same two factors: relevance and freshness. The more relevant documents are found in a query, the higher the weight on that count. However, as more shots in that query are examined, that query loses freshness, and tf_q counts are downweighted.

In this manner, the RSVP user constantly and automatically updates the system-suggested term list. The more the RSVP user explores fresh, relevant pathways, the more the associated terms related to those pathways appear. As the RSVP user switches paths, once a particular avenue loses freshness or does not exhibit relevance, the automatically-suggsted terms switch with him. No cognitive load is required for the RSVP user to suggest terms, just as no cognitive load is required for the MediaMagic user to tweak the order of shots fed to the RSVP user.

3.4 Analysis

In this section we present a number of metrics comparing collaborative systems with not only similarly-instrumented single user runs, but also with pooled single user runs.

3.4.1 Mean average precision

From the results presented in Table 1 it should be clear that the collaborative runs outperform the standalone MediaMagic runs. There are two ways of looking at these improvements. First, the CO15 run shows a 14.6% MAP improvement over MMA, and a 16.9% improvement over MMV. Another way of measuring improvement is to look at the amount of time it takes to obtain an equivalent MAP score. Our CO11 run was simulated by submitting all the results obtained by the collaborative team at 11 minutes and 15 seconds. Table 1 shows that the same MAP can be obtained in 75% of the time, a 25% improvement if time is of the essence.

While both of the above metrics show decent performance improvements, they do not help us better understand what is happening, under the hood. We wish to examine these results on a per-topic basis. We feel that statistics based on actions performed during the actual runs (how many relevant shots were found, how many non-relevant shots were examined, etc.), rather than post hoc padded results, are more enlightening. Finally, we will examine some of these statistics as functions of topic size.

In the following subsections we will compare CO15 with MMA. Additionally, we have created an artificial collaborative run: MMAV, or "merged". Given that the MMA and MMV systems had essentially the same MAP, we wondered how well the post hoc combination these two runs would perform. Duplicate relevant shots are removed, and duplicate non-relevant shots are removed as well, simulating the effect of an interface-only collaborative system in which users are simply made aware of the previous search activities of their partners, but no algorithmic support is made available. Can the collaborative CO15 system outperform not only a single-user system, but the merged system as well? If not, then there is not much point to designing algorithmically-mediated collaborative systems; two users might as well use standalone systems, independently.

3.4.2 Recall and Precision

Because we are now looking at shots actually examined by users during the 15 minute runs, rather than padded ranked lists from the end of a run, the metrics we will use are precision and recall from this manually-examined set.

For precision, CO15 shows a per-topic average 1.5% improvement over MMA and a 15.4% improvement over MMAV. These differences are not statistically significant. We frankly did not expect them to be, because precision alone, in the manually-retrieved set, essentially measures user agreement with NIST judgements. We would not expect any system in which users are making an honest effort to find relevant shots to differ significantly in precision alone.

For recall, CO15 shows a per-topic average 101.1% improvement over MMA and a -10.7% improvement over MMAV. Collaborative retrieval significantly outperforms the standalone system, but as we will see in Section 3.4.4 some of those improvements are only slight, while some are large. The merged system actually does slightly better than the collaborative, but this different is not statistically significant.

It appears that these metrics again confirm what we learned in Section 3.4.1: Collaborative search outperforms single user search. Unfortunately, it does not appear that, on average, it outperforms the combined standalone runs, MMAV. For the collaborative system, precision is slightly better and recall is slightly worse, for an average minimal difference.

What these average values obscure is interesting. We need to take two additional factors into account. The first is the number of shots that the user(s) of a system manually churn through. The second is the size of the topic. The next two subsections will address this. If we wish to get a true measure of the relative effectiveness of the collaborative system, we need to take these factors into account.

3.4.3 Normalized Recall and Precision

To normalize recall and precision, we take the various, manually obtained counts and divide them by the the total number of manually examined shots, during the course of that topic's run. This gives us a per shot, effort normalized value of that count, essentially trying to account for brute force approaches.

The metric for precision does not change. $\frac{TP}{TP+FP}$ is the same as $\frac{\frac{TP}{SS}}{\frac{TP}{SS}+\frac{FP}{SS}}$, where SS is all seen shots for that topic by that system. Recall, on the other hand, changes from $\frac{TP}{TR}$, where TR is the total relevant (pooled across all NIST participants) for that topic, to $\frac{TP}{TR*SS}$.

We wish to understand whether one system is performing better than another simply because it has been able to examine more shots or whether there is and underlying systemic, algorithmic effectiveness improvement. We feel that per-shot statistics are one way of exploring this.

The average unique (non-duplicated) examined shot counts of the various system are as follows: MMA = 2123, MMV = 2601, MMAV (merged) = 4184, CO15 = 2614. When normalizing by these values (using the actual values for each topic, rather than the above averages), CO15 still shows the same 1.5% and 15.4% precision improvement over MMA and MMAV. However, the recall numbers tell a different story. CO15 does 73.9% better than MMA and 44.1% better than MMAV.

This shows is that the collaborative system is actually more effective than not only the standalone system, but the merged system as well. MMA gets through fewer shots than CO15, so there is a lower normalization factor for MMA, which does increase its overall score. (In the unnormalized version, CO15 had 101.1% higher recall; normalization lowers this slightly to 73.9%). Even still, C015 outperforms MMA by a significant margin. Similarly, MMAV (merged) goes through more shots than CO15, but there is also more thrashing: More shots are required to obtain similar recall numbers. Normalized CO15 outperforms MMAV.

It was a disadvantage of our system that users did not get through more shots; this was due mainly to an RSVP interface that was not as optimal as we had hoped. Nevertheless, these results show that the collaborative system was able to work "smarter" than either of the other systems. These results speak well for collaborative search; this is evidence in favor of the idea that algorithmic collaboration is an improvement over both single and multi-user non- or interface-only collaboration.

3.4.4 Per Topic Normalized Recall and Precision

As mentioned above, one more factor is important in determining how well our system is performing. We need to examine recall and precision values by topic size. Figure 5 contains two graphs, one for recall and one for precision. Along the x-axis are the various topics, represented by the total number of relevant documents available in the collection (pooled NIST). Along the y-axis are the percentage differences between CO15 and either MMA or MMAV. In order to get a better sense of the patterns inherent in this data, values have been smoothed using kernel regression with simple exponentially decaying windows.

The interesting thing about these plots is that the improvements are not randomly distributed. Where CO15 most outperforms MMA and MMAV is on topics with fewer total available relevant shots, haystacks with fewer needles. And the CO15 performance on these more "difficult" topics is not just slightly better; it is much better.

Analyses are still at an early stage, but we feel this pattern demonstrates an interesting tradeoff for collaborative search. When there are a lot of relevant shots to be found, it appears that searchers should be freer to work on their own, without algorithmic collaboration. There are going to be enough available relevant shots that each independent searcher can spend all 15 minutes working separately. However, when relevant shots are more difficult to come by, two searchers working independently are not quite able to find them. On the one hand, this seems counter-intuitive: With more people searching for something, the chances are greater

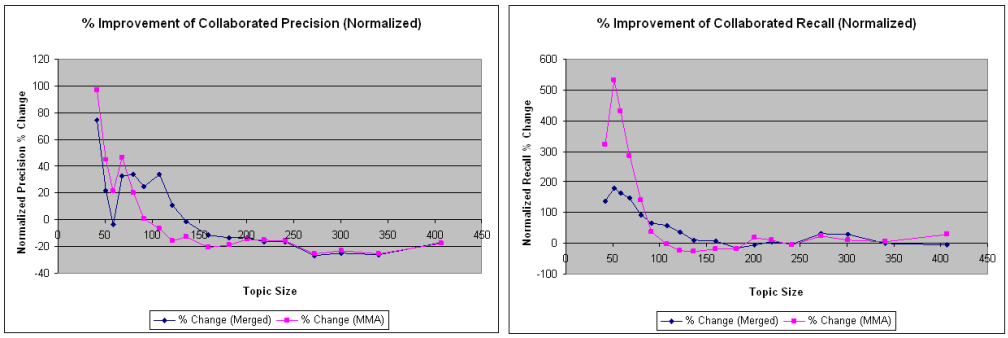

Figure 5: Precision (left) and Recall (right) performance metrics plotted against topic size.

that any one person will find it. On the other hand, this demonstrates the advantages of algorithmically collaborated search: It is only when two searchers' activities are coordinated by the underlying system that they are more able to push further into the collection, and find those nuggets, than had they been working separately.

References

[1] F. Chen, M. Cooper, and J. Adcock. Video Summarization Preserving Dynamic Content. *TRECVID BBC Rushes Summarization Workshop at ACM Multimedia'07*, Augsberg, Germany, 2007.

[2] Marijn Huijbregts, Roeland Ordelman and Franciska de Jong. Annotation of Heterogeneous Multimedia Content Using Automatic Speech Recognition. *To appear in proceedings of SAMT*, December 5–7 2007, Genova, Italy

[3] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, E. Rieffel, and L. Wilcox. FXPAL Experiments for TRECVID 2004. *Proceedings of TRECVID 2004*, 2004.

[4] M. Cooper, J. Adcock, H. Zhou, and R. Chen. FXPAL Experiments for TRECVID 2005. *Proceedings of TRECVID 2005*, 2005.

[5] M. Cooper, J. Adcock, and F. Chen. FXPAL at TRECVID 2006. *Proceedings of TRECVID 2006*, 2006.

[6] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System *Proceedings of TRECVID 2004*, 2004.

[7] J. Adcock, M. Cooper, A. Girgensohn, and L. Wilcox. Interactive Video Search Using Multilevel Indexing. *Proc. International Conference on Image and Video Retrieval*, 2005.

[8] Jakarta Lucene. http://jakarta.apache.org/lucene/docs/index.html.

[9] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms *In Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.*, pages 762–768, 1997.

[10] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features, *Proceedings of the ninth European Conference on Computer Vision*, May 2006.

[11] L. Bottou and Y. Bengio Convergence properties of the K-means algorithm. *In Advances in Neural Information Processing Systems*, volume 7. MIT Press. 1995.

[12] Morris, Meredith R. Interfaces for Collaborative Exploratory Web Search: Motivations and Directions for Multi-User Designs. *CHI 2007 Workshop on Exploratory Search and HCI*, 2007

[13] Smeaton A.F, Lee H, Foley C, Mc Givney S, and Gurrin C. Fischlar-DiamondTouch: Collaborative VideoSearching on a Table. *Multimedia Content Analysis, Management, and Retrieval*, January 15–19, San Jose, CA, 2006

[14] Michael W. Berry, Susan T. Dumais and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval *SIAM Review, v.37 n.4, p.573-595*, Dec. 1995

[15] LibSVM. http://www.csie.ntu.edu.tw/ cjlin/libsvm/.