FXPAL Interactive Search Experiments for TRECVID 2008

Jeremy Pickens, John Adcock, Matthew Cooper, Andreas Girgensohn
FX Palo Alto Laboratory
Palo Alto, CA 94304
{last name}@fxpal.com

In 2008 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 TRECVID2008, including 4 single user and 2 collaborative runs. The 6 runs are derived from 3 independent interactive trials: 1 collaborative trial, and 2 independent single user trials. These base evaluations are extended by using automatic results derived with different methods.

The complete set of submitted runs in priority order with system names and brief descriptions are listed in Table 1.

Submission	**MAP score**	**Description**
FX-CoHoMm_1	0.148	Collaborative search (Co), automatic results from *House*user's (Ho) MediaMagic (Mm)
FX-CoxxSv_2	0.147	Collaborative search (Co), submission is from the collaborative server (Sv)
FX-AlHoMm_3	0.112	Single user search (Al) in *House*mode (Ho), automatic results from MediaMagic (Mm)
FX-AlHoSv_4	0.109	Single user search (Al) in *House*mode (Ho), automatic results from the collaborative server (Sv)
FX-AlSeMm_5	0.100	Single user search (Al) in *Senate*mode (Se), automatic results from MediaMagic (Mm)
FX-AlxxMm_6	0.076	Baseline single user search (Al) without mediation of collaborative server or program mode (xx), automatic results from MediaMagic (Mm)

Table 1: MAP scores for the search submissions in priority order. MAP scores are shown to 3 digits to expose slight differences which may or may not be statistically significant.

1.1 Single-user runs

We used 6 searchers to complete our 4 single user runs. 2 users used our baseline MediaMagic system and performed 12 topics each to complete a single submission, AlxxMm. The other 4 users each performed 6 topics as *House*and 6 topics as *Senate*, resulting in a complete *House*submission, AlHoMm, and an independent complete *Senate*submission, AlSeMm. The 4th submission, AlHoSv, differs from AlHoMm in the method that is used to fill out the results list at the end of the interactive session.

See section section 4 for details on the nature of the *House*and *Senate*distinction.

1.2 Collaborative runs

We also made 2 multi-user collaborative submissions, CoHoMm and CoxxSv. In these submissions 2 users worked simultaneously to complete each topic. 4 of the 6 searchers who performed the single-user searches were joined into teams and each of the 4 teams performed 6 of the topics (each of the 4 users performed 12

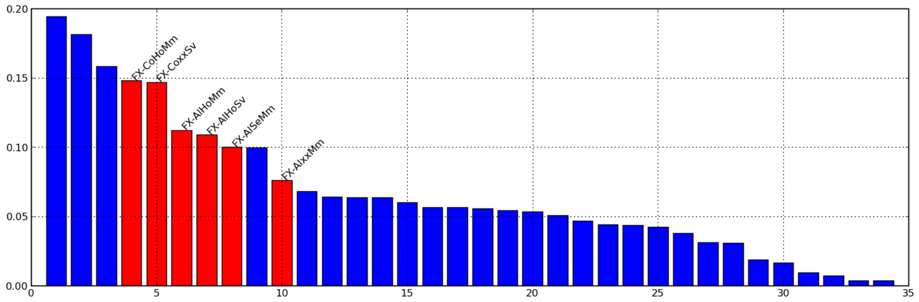


Figure 1: MAP performance of all interactive search submissions with FXPAL submissions labeled.

total collaborative topics). In each two-person team, one user is assigned to be *Senate*and the other, *House*. Teams and roles (*Senate*or *House*) were assigned in a balanced manner.

Table 1 lists the runs in performance order. Figure 2 illustrates the components available in the 3 different types of runs performed.

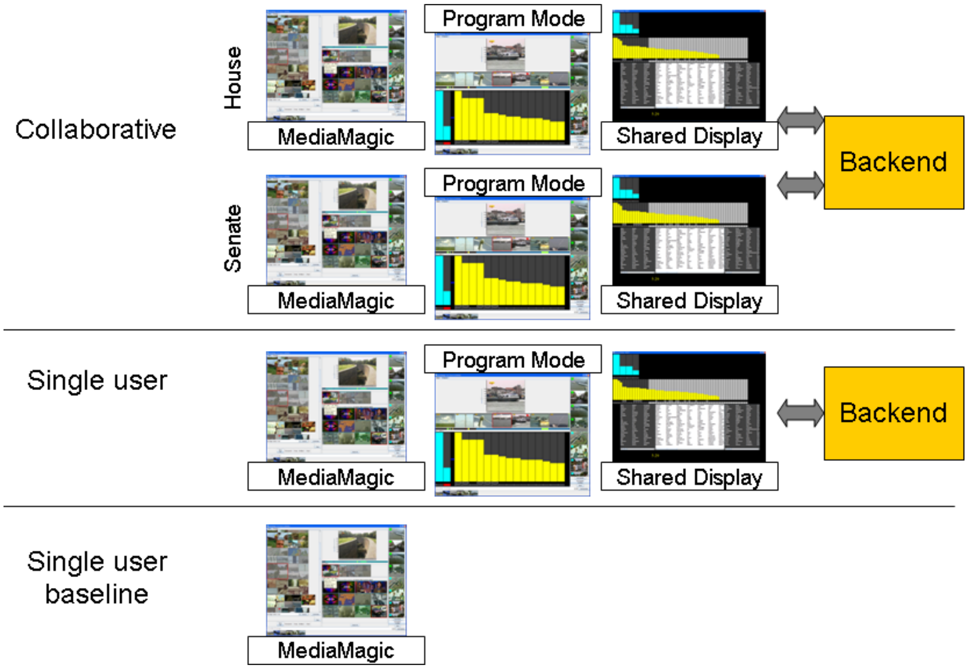


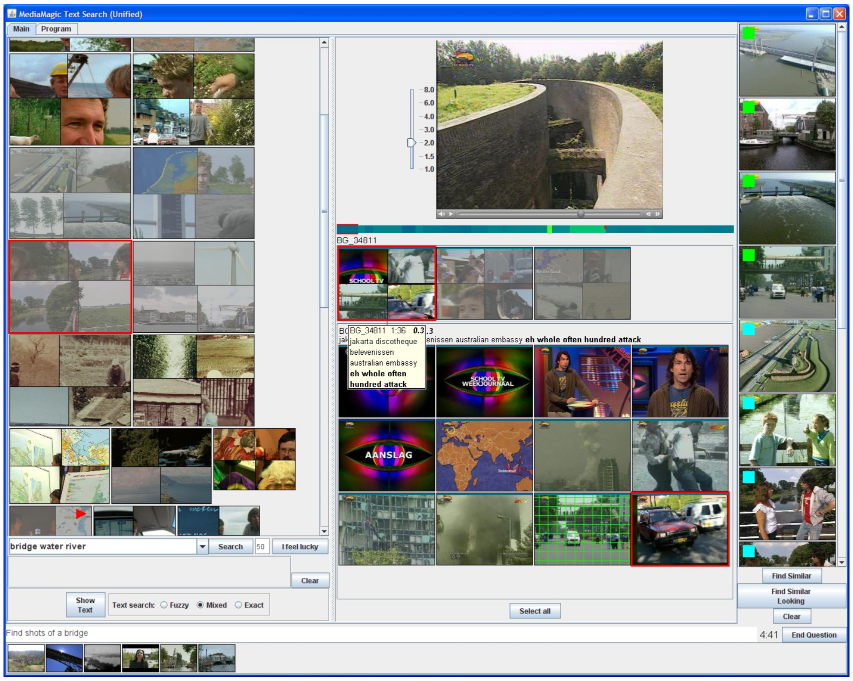
Figure 2: Graphical comparison of the components used in collaborative (CoHoMm, CoxxSv), single-user (AlHoMm, AlHoSv, AlSeMm), and single-user baseline runs (AlxxMm).

2 Overview of Results

The collaborative runs achieved approximately 30% higher MAP than the best single user run with a negligible observed difference between using the MediaMagic automatic query component or the collaborative backend's priority ranking. The *House*mode single user run appears to hold an edge over the *Senate*single user run, though the statistical significance of this result has not yet been evaluated. Meanwhile, the baseline system without program mode, shared display, and the aid of the collaborative backend lags well behind the next best single user result attaining only 3/4 of the MAP of the next worst single user run and only half of the MAP of the collaborative system.

The system components are described below in greater detail.

3 System Description



BG_34811

BG_34811 1:36 0.3

jakarta discotheque belevenissen australian embassy **eh whole often hundred attack**

bridge water river 50

Text search: ○ Fuzzy ● Mixed ○ Exact

Find shots of a bridge 4:41

Figure 3: MediaMagic main interface. A shot is shown with a green, cyan, or red overlay to indicate that is has been marked relevant, maybe relevant, or not relevant to the topic. Query results as thumbnails on the far left, media player, program timeline, and detail of the currently selected story in the middle, and relevant result thumbnails in a column on the far right.

The underlying MediaMagic interface, depicted in Figure 3 is largely unchanged from previous years [4, 5, 6, 7] and is described in detail in section A. In brief the interface provides for text search over the translated transcripts, segmentation of shots into story units for viewing query results and navigating programs, pivoting off of shots or stories with text, visual, and semantic similarity queries. It serves as both the sole interface for the baseline single user run (FX-AlXXmm) and as the basic component of our enhanced single user system and two-user collaborative search system.

In 2008 we extended the baseline system with several components:

- 'Maybe' judgements (available in all evaluated systems)
- Program Mode (not used in baseline system AlxxMm)
- Mediating backend (not used in baseline system AlxxMm)
- Shared display (not used in baseline system AlxxMm)

3.1 Maybes

In all evaluated systems we adopted an extra relevance rating of 'maybe relevant'. As an alternative to simply rating a shot as relevant or not, the user may choose to rate it as 'maybe relevant' or 'definitely relevant'. Shots marked as 'maybe relevant' are kept in the results list after shots marked as 'definitely relevant'. In the displayed screenshots (see Figure 4 and Figure 3, the shots marked as 'maybe relevant' are shown with a cyan overlay, those marked as 'definitely relevant' are shown with a green ovlerlay. By providing this multi-valued relevance scoring we intend to improve precision by allowing the user to assure that the shots about which they are most sure are at the top of the results, followed by those about which they are uncertain.

3.2 Program Mode

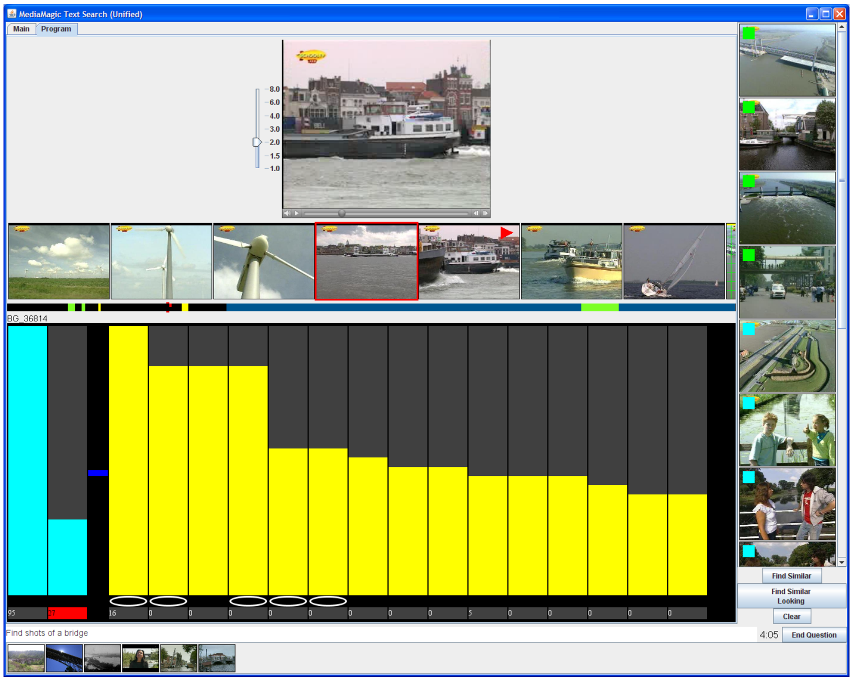


Figure 4: Program mode interface. A keyframe/video viewer is available at the top. The shots of the current program are shown in time-order underneath, and underneath that the bars (which correspond to programs) provide a visualization of which programs have yielded the most relevant results so far (blue bars) or the cumulative score the program has received so far during this topic (yellow bars). Ellipses underneath indicate programs which contain stories which have been frequently retrieved, but not yet examined by the user. Shots judged relevant are arranged vertically on the right.

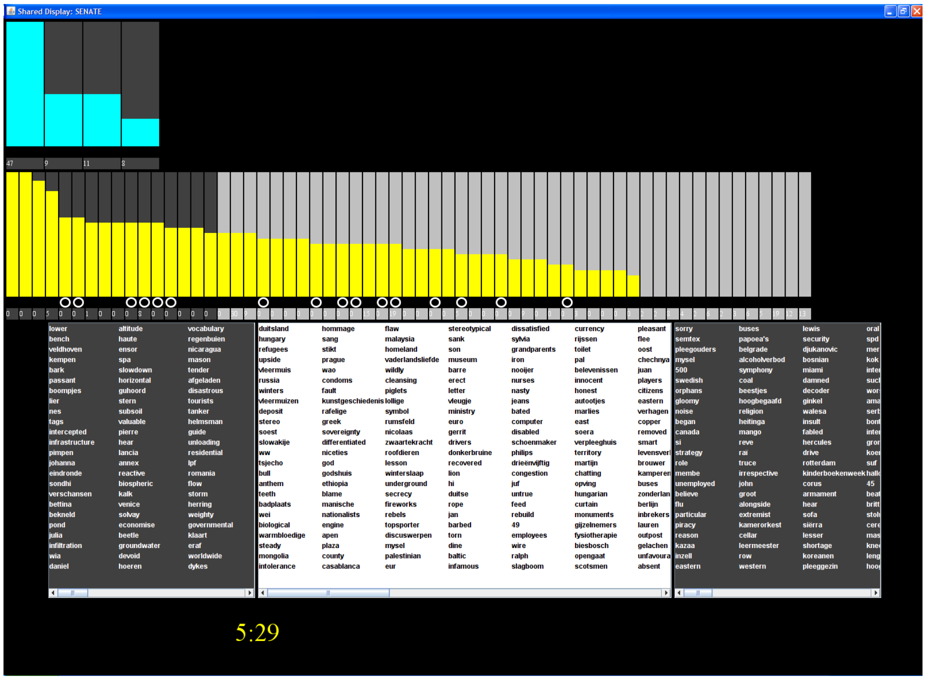
A new addition to the interface in 2008 is the addition of a "Program Mode" view. This is shown in Figure 4. This mode is accessed at searcher's discretion through a tab on the main interface pane and provides a streamlined interface for reviewing all the shots within a single program or video file. It also provides information to the user about which programs may be most fruitful to subject to exhaustive examination and a quick way to select a promising program for review.

3.3 Shared Display

A shared display provides information to the searcher about the distribution of terms relevant to the current search session and the distribution of retrieved and judged video material. In particular it identifies the programs which have provided the greatest number of relevant shots so far, and the programs which have the highest *Senate*or *House*scores (see section 4) which are expected to be promising avenues of exploration. The *Executive*program list and the term suggestions are common between *House*and *Senate*users, while the *Legislative*list of programs is dependent on the scoring mode: *Senate*or *House*(as described below).

3.4 Collaborative Backend

The collaborative mediation component of our system maintains a model of which shots, stories, or programs are likely relevant to the current topic based on the actions of the user over the topic session. All actions of the user at the MediaMagic interface (queries, query results, relevance judgements, viewed shots and stories) are communicated to the backend. There the query results are mediated and possibly altered by the backend before being returned to the interactive components. As a topic session progress the backend aggregates the actions of the searcher(s) to provide feedback to the user in program mode and in the shared display about the distribution of material retrieved and judged during the search session. Relevance judgements and visited status of shots and stories are also shared between clients in the collaborative setting through the



lower	altitude	vocabulary
bench	haute	regenbuien
veldhoven	ensor	nicaragua
kempen	spa	mason
bark	slowdown	tender
passant	horizontal	afgeladen
boompjes	guhoord	disastrous
lier	stern	tourists
nes	subsoil	tanker
tags	valuable	helmsman
intercepted	pierre	guide
infrastructure	hear	unloading
pimpen	lancia	residential
johanna	annex	lpf
eindronde	reactive	romania
sondhi	biospheric	flow
verschansen	kalk	storm
bettina	venice	herring
bekneld	solvay	weighty
pond	economise	governmental
julia	beetle	klaart
infiltration	groundwater	eraf
wia	devoid	worldwide
daniel	hoeren	dykes

duitsland	hommage	flaw	stereotypical	dissatisfied	currency	pleasant
hungary	sang	malaysia	sank	sylvia	rijssen	flee
refugees	stikt	homeland	son	grandparents	toilet	oost
upside	prague	vaderlandsliefde	museum	iron	pal	chechnya
vleermuis	wao	wildly	barre	nooijer	belevenissen	juan
russia	condoms	cleansing	erect	nurses	innocent	players
winters	fault	piglets	letter	nasty	honest	citizens
vleermuizen	kunstgeschiedenis	lollige	vleugje	jeans	autootjes	eastern
deposit	rafelige	symbol	ministry	bated	marlies	verhagen
stereo	greek	rumsfeld	euro	computer	east	copper
soest	sovereignty	nicolaas	gerrit	disabled	soera	removed
slowakije	differentiated	zwaartekracht	drivers	schoenmaker	verpleeghuis	smart
ww	niceties	roofdieren	donkerbruine	philips	territory	levensver
tsjecho	god	lesson	recovered	drieënvijftig	martijn	brouwer
bull	godshuis	winterslaap	lion	congestion	chatting	kamperen
anthem	ethiopia	underground	hi	juf	opving	buses
teeth	blame	secrecy	duitse	untrue	hungarian	zonderlan
badplaats	manische	fireworks	rope	feed	curtain	berlijn
wei	nationalists	rebels	jan	rebuild	monuments	inbrekers
biological	engine	topsporter	barbed	49	gijzelnemers	lauren
warmbloedige	apen	discuswerpen	torn	employees	fysiotherapie	outpost
steady	plaza	mysel	dine	wire	biesbosch	gelachen
mongolia	county	palestinian	baltic	ralph	opengaat	unfavoura
intolerance	casablanca	eur	infamous	slagboom	scotsmen	absent

sorry	buses	lewis	oral
semtex	papoea's	security	spd
pleegouders	belgrade	djukanovic	mer
mysel	alcoholverbod	bosnian	kok
500	symphony	miami	inter
swedish	coal	damned	suc
orphans	beestjes	decoder	wor
gloomy	hoogbegaafd	ginkel	ama
noise	religion	walesa	serb
began	heitinga	insult	bon
canada	mango	fabled	inter
si	reve	hercules	gron
strategy	raï	drive	koe
role	truce	rotterdam	suf
membe	irrespective	kinderboekenweek	hall
unemployed	john	corus	45
believe	groot	armament	beat
flu	alongside	hear	britt
particular	extremist	sofa	stol
piracy	kamerorkest	siërra	cere
reason	cellar	lesser	mas
kazaa	leermeester	shortage	kne
inzell	row	koreanen	leng
eastern	western	pleeggezin	hoo

Figure 5: Shared display interface. The *Executive*component is shown in blue at the top and indicates which programs have contained the greatest number of relevant shots so far. The *Legislative*component is shown in yellow in the middle and indicates which programs have more stories which are repeatedly retrieved by queries but not yet viewed by the user. In the same session this component will appear differently to the *Senate*and *House*searchers. The ellipses indicate programs with the highest cumulative retrieval score aggregated across all searches. The text boxes at the bottom show the most frequent terms from *Executive*, *Legislative*and *Judicial*program categoris (see section 4).

backend so that each searcher can benefit from the relevant video already found by the other searcher and avoid re-examining material already viewed by the other searcher.

In several submissions (CoxxSv and AlHoSv) we opted to use shots drawn from the backend's priority list to fill out the TRECVID submission rather than shots from the MediaMagic client's similarity search (described in subsection A.4).

4 Collaborative Mediation and Relevance Clustering

Our system begins with the observation that relevant shots tend to cluster by program. That is, shots are not uniformly distributed across programs, but tend to cluster in a relatively few number of programs. For example, in the 2007 TRECVID dataset, it is possible to achieve over 50% recall (on average across all topics) with shots from the 3 best programs.

Our goal, then, is to find those programs with the highest available number of relevant shots. We do this by dynamically categorizing each program into one of two programs: *Executive*and *Legislative*. The *Executive*programs are those in which at least one shot has been marked relevant by the user. The *Legislative*programs are those in which at least one shot has been retrieved, but no shot has been marked relevant. A third category of programs, *Judicial*, is the set of programs from which no shots have been even retrieved by a query. The idea is that, before the search session ends, the user will revisit those programs with the highest *Executive*scores and search through those programs in greater detail. Thisfocused examination should allow the user to find all the other relevant shots in that program that had not been found through direct search. The score for a program that has been categorized as *Executive*is as follows for a shot denoted s and a program denoted p :

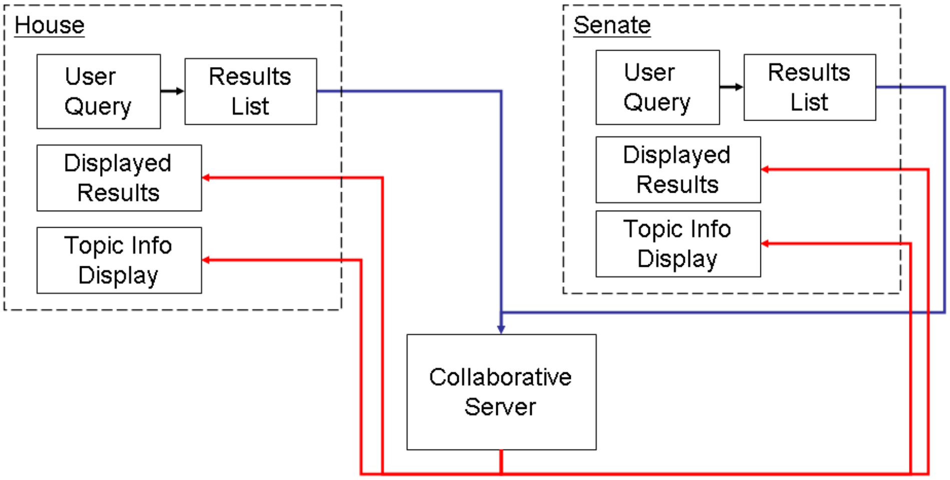


Figure 6: Collaborative system flow showing two searchers with *Senate*and *House*roles. The backend provides different *Legislative*outputs based on the searcher role.

$$\sum_{s \in p} \begin{cases} 1.0 & , \quad s \in Relevant \\ 0.5 & , \quad s \in Maybe \\ 0.0 & , \quad \text{otherwise} \end{cases}$$

The *Legislative*programs, on the other hand, are those which have been retrieved by the user at some point during the search session, but in which no shots have been marked relevant because they have not yet been examined by the user. These are programs that we would like to use to rerank future user queries, with the goal of obtaining a more accurate sampling of the distribution of relevant shots across programs. Programs with evidence of high retrieval activity could indicate a larger likelihood of relevance.

We have designed two fundamental types of *Legislative*program scoring, which we name *House*scoring and *Senate*scoring. Our inspiration for these algorithms comes from the legislative branch of the U.S. government, which is a representative body for a federation of distinct territories, or states. In this legislature, the voice of every state in the union is represented in two different manners. In the House, a state receives voting power that is directly proportional to the size of the population in that state. More people means more voting power. In the Senate, on the other hand, every state receives exactly the same number of votes (two). The idea behind this bicameral legislature is that the same people need to be represented in two different manners to ensure balance of power. The House ensures that populations are represented proportionally, which is a fair criterion on a representative democracy. The Senate, on the other hand, helps protect smaller states against the tyranny of the majority. By giving every state exactly the same voting power, large states have a more difficult time forcing their will on smaller states.

By analogy, we "represent" retrieval scores by these same two methods. Every time a shot is retrieved, via a user query, there is a score associated with that retrieval. No matter what the query type (text, latent semantic text, concept, image query, or mixture of any of the above), the retrieval scores are normalized to the [0..1] range. A program is then given either a *House*or *Senate*score, based on the sum of either a *House*or *Senate*interpretation of that score.

$$HouseScore(p) = \sum_{s \in p} \sum_{s \in QR} \begin{cases} Score(s) & , \quad s \in p \wedge s \notin Viewed \wedge s \notin Nonrel \\ 0 & , \quad otherwise \end{cases}$$

$$SenateScore(p) = \sum_{s \in p} \begin{cases} 1.0 & , \quad s \in p \wedge s \notin Viewed \wedge s \notin Nonrel \\ 0 & , \quad otherwise \end{cases}$$

For a program denoted p and a shot denoted s, the set of all query results denoted QR and the score of shot s in the current query result $Score(s)$. The set of viewed shots is denoted $Viewed$ and the set of shots marked non-relevant is denoted $Nonrel$.

The *House*scoring is an example of proportional representation; more retrieval activity corresponds directly with a higher score. A program's score is based not only on the original strength of the match between its constituent shots and every user query, but if a shot is retrieved multiple times, each time adds to the total score. On the other hand, the *Senate*score only gives a single vote to every shot in the program. No matter of the original retrieval score for that shot was 0.96 or 0.003, and no matter if it was retrieved once or half a dozen times, it is given a score ("vote") of 1.0.

Programs ordered by the *House*method will be different than those ordered by the *Senate*method. The *House*tends to favor programs with an overall large amount of retrieval activity, even if that activity is concentrated in a small number of shots. The *Senate*, on the other hand, tends to favor programs with a wider diversity (larger number) of distinct retrieved shots, even if the original retrieval score of those shots is low.

A searcher, whether working alone or collaboratively, is assigned one of these two roles: *House*or *Senate*. When that user runs a query, the system does not just return the ranked list of shots. Instead, it look at the global ordering of programs, sorted by the user's role (*House*or *Senate*score) and then places at the top of the ranked list those shots in the ranked list that are found in the highest-scored programs. This promotion of shots from programs with higher Legislative scores is the way in which the user is able to get a better, ongoing sampling of shots from programs with potentially higher numbers of relevant documents. There is no guarantee that the promoted shots are the most relevant shots, for a particular (text, image, etc.) query. But by sampling from these highly-scored programs, the user has a higher chance of discovering programs with relevant shots.

Appendices

A MediaMagic

The MediaMagic interface, depicted in Figure 3 was designed for efficient browsing and rich visualization of search results, and is largely unchanged from previous years [4, 5, 6, 7]. It serves as both the sole interface for the baseline single user run (FX-AlXXmm) and as a component of our enhanced single user system and two-user collaborative search system.

A.1 Data pre-processing

We perform a completely automatic pre-processing step to identify topic or story units to augment the reference shot boundaries [8]. 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 [3] as described in [9].

In preparation for interactive operation, text indices are built for both the shot-level and story-level segmentations using Lucene [10] (for keyword search) and our latent semantic indexing system (for fuzzy text search). Color correlograms [11] are pre-computed for each shot thumbnail image and used for image similarity search. Also for each shot thumbnail image, color histograms and SURF [13] descriptors are computed as input features for our concept detection.

A.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 [10] 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 [17], 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 [11]. Any shot thumbnail in the interface can be dragged into the query bar (Figure 3)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.

A.3 Interface Elements

The interactive search system is pictured in Figure 3.

Shots are visualized with thumbnails made from the keyframes computed using the reference shot segmentation and histogram features as in section B. 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.

Overlays are used to provide several cues. A gray overlay on a story icon indicates that it has been previously visited (see Figure 3). A red overlay on a shot icon indicates that it has been marked not relevant by the user. A green overlay on a shot icon indicates that it has been included in the results set. A cyan overlay indicates that it has been marked as possibly relevant. 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 program timeline.

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.

A.4 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 by the MediaMagic client 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 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 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).

The collaborative mediation component of our system maintains its own idea of which shots are likely relevant to the current topic. In several submissions we opted to use shots drawn from this list to fill out the submission rather than shots from the MediaMagic client's similarity search.

B Concept Similarity

We use SVM-based concept detectors for the 35 lscom-lite concepts to provide concept-based shot and story similarity measurements. Each shot has an associated 35 element vector describing the posterior probability of each of the high-level features. For concept distance between two shots we use the mean absolute distance (normalized L1) between their concept vectors.

B.1 Detector construction

We used the TRECVID 2007 sound and vision test data (2008 development data) labeled with the 2007 high level feature extraction task results to train our detectors.

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 subsection A.1 and [13].

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 [18]. 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.

B.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 subsection A.4. 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.

References

[1] F. Chen, M. Cooper, and J. Adcock. A Simplified Approach to Rushes Summarization. *TRECVID BBC Rushes Summarization Workshop at ACM Multimedia'08*, Vancouver, Canada, 2008.

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

[3] 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

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

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

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

[7] J. Adcock, J. Pickens, M. Cooper, L. Anthony, F. Chen, and Pernilla Qvarfordt. FXPAL Interactive Search Experiments for TRECVID 2007. *Proceedings of TRECVID 2007*, 2007.

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

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

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

[11] 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.

[12] Sheng Tang, et al. TRECVID 2008 High-Level Feature Extraction By MCG-ICT-CAS *Proc. TRECVID 2008 Workshop*, Gaithersburg, MD, USA , Nov. 2008.

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

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

[15] 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

[16] 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

[17] 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

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