Glasgow University at TRECVID 2006

Jana Urban, Xavier Hilaire, Frank Hopfgartner, Robert Villa and Joemon M. Jose
Department of Computing Science
University of Glasgow, UK
{jana,hilaire,hopfgarf,villar,jj}@dcs.gla.ac.uk

Siripinyo Chantamunee and Yoshihiko Gotoh
Department of Computing Science
University of Sheffield, UK
{s.chantamunee, y.gotoh}@dcs.shef.ac.uk

ABSTRACT

In the first part of this paper we describe our experiments in the automatic and interactive search tasks of TRECVID 2006. We submitted five fully automatic runs, including a text baseline, two runs based on visual features, and two runs that combine textual and visual features in a graph model. For the interactive search, we have implemented a new video search interface with relevance feedback facilities, based on both textual and visual features.

The second part is concerned with our approach to the high-level feature extraction task, based on textual information extracted from speech recogniser and machine translation outputs. They were aligned with shots and associated with high-level feature references. A list of significant words was created for each feature, and it was in turn utilised for identification of a feature during the evaluation.

1. INTRODUCTION

Search Task.

This year Glasgow University participated for the first time in the TRECVID search task. We submitted one interactive run and five fully automatic runs. The automatic runs were a combination of text features only (UG_F_1), visual features only (UG_F_2 and UG_F_5) and a combination of feature modalities (UG_F_3 and UG_F_4). The following list describes all submitted runs:

UG_F_1 Text baseline (required)

UG_F_2 Automatic search based on visual features (optimised weighting)

UG_F_3 Graph model based on text query.

UG_F_4 Graph model based on text and visual query examples.

UG_F_5 Automatic search based on visual features (equal weighting)

UG_I_1 Interactive search run (text and visual features).

All our runs where type A (trained on the TRECVID development set only), no use being made of other data, such as the LSCOM concepts.

High Level Feature Extraction Task.

The Sheffield team worked on the high-level feature extraction task. Our approach was aiming at extraction of relevant features based on outputs from automatic speech recognition (ASR) and/or machine translation (MT) systems, underpinned by the mature progress made in the area of text and speech processing. The assumption was that textual data often carried very important information that described the corresponding video shots. However, ASR and MT systems were still far from a human's level, and we were interested to see if ASR errors and translation errors could cause any reduction in performance for the high-level feature extraction task.

The approach benefitted from the ASR and MT dataset provided. We also utilised shot boundary reference in order to segment video into shot-based units. Shots were then aligned with text from ASR and MT systems. Finally, the feature reference was used to build a list of significant words for that feature. We also describe the problems we encountered during the system development, some of which were critical to the system performance.

2. SEARCH

2.1 Features

The basic unit of indexing and retrieval is a video shot. We use the common shot boundary reference and keyframes.

2.1.1 Textual Features

The textual part of Glasgow University's runs were based on the Terrier retrieval system [4], using the standard Terrier stop word list and the Porter stemming algorithm.

For the interactive run (UG_I_1, section 2.3) each shot indexed by Terrier was represented by the English terms (either from the English ASR, or automatic translations) corresponding to the time period of the shot. For the other text-based automatic runs (UG_F_1/3/4), shots were represented by the text from surrounding shots. This was set to a value of six, which indicated that the text from the preceding six, and following six shots were used in generating the textual representation of the shot. This value was set empirically based on the experiments performed on the TRECVID 2005 collection. Some statistics from these two text collections are shown in table 1.

TRECVID'06, November 13–14, 2006, Gberg, Washington, USA.

Table 1: Textual shot statistics

	No surrounding shots	Following/proceeding 6 shots
number of shots	79484	79484
Avg. text length	15.89 terms	203.03 terms
No. empty shots	31583	2174

The use of two different collections for the automatic and interactive runs was based on experience with the TRECVID 2005 collection. With this earlier collection, it was found that using a shot window improved the automatic runs, although during informal interactive evaluation it was felt that this introduced too many repeated shots, from adjacent same time periods in the videos. In interactive retrieval, the user has the ability to browse, via the interface, to surrounding shots, reducing their need to be in the main ranking.

2.1.2 Visual Features

Currently, our retrieval engine based on visual features only does not exploit motion information from video streams. Instead, it retrieves from the provided keyframes only. We used the following five visual feature extractors, all advocated by the MPEG-7 standard:

- Colour layout: a description of the spatial distribution of the colours of an image.
- Contour shape: a compact shape descriptor mixing information on curvature, circularity and eccentricity of the first contour found in the image.
- Dominant colours: the eight most representative colours of the image after quantisation.
- Edge histogram: a description of the distribution of the edges of the image as a histogram with five different types of edges.
- Homogeneous texture: a texture descriptor for a homogeneously textured image based on the energies and the energy deviations of the responses of frequency channels to Gabor filter functions.

These features were selected as the top performing ones in experiments made on previous TRECVID collections. In addition, we used our own implementations of colour histograms, both in L*a*b* and HSV spaces. Altogether the dimension of the visual features used is 497.

2.2 Automatic Runs

We have submitted five fully automatic runs, each of which are described in the following sections. The results of the submitted runs are summarised in table 2, which compares mean average precision (MAP), precision at 10 (P(10)), precision at total relevant shots (P(NR)) and recall averaged over all topics. The MAP results per topic are shown in table 3.

2.2.1 Text Baseline – UG_F_1

The baseline run, UG_F_1, is based on the retrieval results obtained from Terrier on the text index as described in section 2.1.1. The BM25 retrieval model was used, together with automatic query expansion with the number of expansion documents set to ten, and the number of terms used in the expansion set to eight. Each shot was represented by the text both corresponding to the shot in time, plus the text in the previous and following six shots, as described in section 2.1.1.

Finally, we re-ranked the results by applying a "shot weighting window". In this step, each result shot conferred a proportion of its score to the preceding and following five shots from the same video. The following weighting window was applied: $w = [1, 0.9, 0.75, 0.5, 0.25]$. Let s_0 denote the current shot with corresponding score v_0 and s_{-1}, s_{-2},... s_{-5} denote the five preceding shots, while s_{+1}, s_{+2},... s_{+5} denote the five following shots. Then applying the weighting window to s_0 will result in the following adapted scores: $s_{\pm i} = v_{\pm i} + w[i] * v_0$ for $1 \leq i \leq 5$. The weighting window was again set empirically based on the training performed on the TRECVID 2005 collection.

Tables 2 and 3 summarise the results of all submitted search runs. The baseline text run appears roughly in line to other submissions, suggesting a similarity of approach with other participating organisations.

2.2.2 Visual Features – UG_F_2/5

Runs labelled UG_F_2 and UG_F_5 have both been obtained by resorting only to the visual features described in section 2.1.2. Due to lack of means and time, we did not exploit the video streams themselves, and fell back on the keyframes provided with the collection. Given a particular query, the methodology we used is the following:

- the 58 example images provided by NIST were complemented by extracting keyframes from the 113 video examples, so as to provide about $n = 10$ example images per query. Extraction of keyframes was done automatically, the keyframes being drawn from each video sample with uniform probability.
- The previous set of images is considered as the positive set S^+ of images. In the same manner, we form a pseudo-negative set S^- by randomly choosing n sample images in the whole set of keyframes.
- Given a feature ϕ, we rank an image I according to the following score:

$$s_\phi(I) = 1 - \frac{\Delta^2_\phi(I, I^+)}{\Delta^2_\phi(I, I^+) + \Delta^2_\phi(I, I^-)}$$

  where

  $I^- = \arg\min_{J \in S^-} d(\phi(I), \phi(J))$,

  $I^+ = \arg\min_{J \in S^+} = \arg\min_{J \in S^+} d(\phi(I), \phi(J))$,

  d is the Euclidean distance, and Δ_ϕ is the Mahalanobis distance – the covariance matrix of which being estimated per feature ϕ, but using training data.
- The final score used for ranking is then defined as a linear combination of the scores per feature:

$$S(I) = \sum_{i=1}^{N} \lambda_i s_{\phi_i}(I)$$

  where N is the total number of features, and the λ_i coefficients are either taken equal to 1 in run UG_F_5, or as to minimise the integral classification error on training data in run UG_F_2.

The training data used was the TRECVID 2005 collection and its related ground truth. We did not optimise S and Δ from any other source.

Our intention in submitting the two runs UG_F_2 and UG_F_5 this year was to estimate the potential of the visual features alone in a retrieval process. With no real surprise, they performed poorly (see tables 2 and 3). We observed a behaviour for this year similar to that obtained on the TRECVID 2005 collection:

- a few topics performed well: topic 195 (soccer goal) and 187 (helicopter in flight) this year, and Topics 156 (tennis court), 171 (soccer goal), and 155 (map of Iraq) from the 2005 collection, sometimes with precision over 40%, because the example images provided represent only one object, and that their semantic is low.
- all other queries performed very poorly, mainly because the two former assumptions are not met any longer.

2.2.3 Combination of Features – UG_F_3/4

Finally, we have experimented with combining textual and visual features in a graph-model [6]. The graph is constructed in the following way:

Nodes

- Each shot is represented as a node in the graph.
- Each term from the textual index (see above) is represented as a node in the graph.
- The visual features are represented by one node per feature type per shot.

Edges

- Each shot node is linked to all term nodes of the terms it is annotated with.
- Each shot is linked to the respective feature nodes (one feature node per feature type).
- The visual feature nodes of the same type are linked to its top k neighbours.

The original graph-model, the Image-Context Graph (ICG), proposed in [6] was formulated for image retrieval tasks. For the visual representation of shots, we have only considered a static global image given by the keyframe for each shot. In addition, the ICG contains contextual links between images (or shots) based on their usage. Such usage information was not available for the TRECVID collection since we have not performed prior user experiments.

Querying in the ICG is implemented using the theory of random walks [3]. The details are explained in [6]. Essentially, querying involves choosing a set of starting or query nodes, and then computing the stationary distribution, π, of the ICG based on the restart vector. The images (shots) are then ranked by their score in π, which represents the probability of reaching this image node from the set of starting nodes.

The ICG has two parameters: k the number of nearest neighbours; and α the weighting factor of start nodes versus the graph-structure in the random walk computation. k is set to 5 and α to 0.7. Further, the visual feature node links are weighted by a weight combination determined on a per query basis (see [6]).

We have experimented with two different ways of choosing the initial starting nodes: using the terms given in the topic description; or using both the terms and the visual examples.

Query by Text – UG_F_3.

In order to choose suitable query terms, we issued a text query consisting of the topic description to the text retrieval engine (see above). We performed pseudo relevance feedback (top 10 documents returned) and added the 8 best scoring terms to the original query.

Rather than setting one restart vector containing all the query terms as explained above, this run is based on issuing one (random-walk) query per term. The final results are obtained by merging the individual result lists using the voting approach [6].

Query by Text and Visual Examples – UG_F_4.

The image and video examples in the topic description are not indexed in the graph (this choice was made to reflect a realistic scenario where a user can query by an arbitrarily chosen example not necessarily contained in the collection). Therefore, we need to choose suitable starting nodes that are most similar to the given examples. Therefore, we have issued a visual query with all the topic example to the visual index (see above) and used the nodes corresponding to the shots returned amongst the top 10.

We used the original set of terms (without query expansion) as the textual query nodes. This time we constructed *one* restart vector from both visual example and terms.

The results in tables 2 and 3 show that UG_F_4 performs slightly better than UG_F_3, although neither can improve on the text-baseline. In hindsight, both the use of query expansion (QEX) and the choice of issuing one random walk per query term rather than using *one* overall restart vector containing all query terms, have decreased the performance of the ICG in run UG_F_3. The MAP score of the original run submitted was 0.0183. A run using one overall restart vector and QEX results in a MAP score of 0.0242, while using one overall restart vector *without* QEX results in a MAP score of 0.0315. This shows that the ICG can improve the text-baseline ($MAP = 0.0298$).

2.3 Interactive Run

Based on the TRECVID 2006 guidelines, an interface for the retrieval system was developed based on both textual and visual features. Users can trigger retrieval cycles, browse through returned keyframes which represent video shots, play and scroll through the actual video file. For query refinement, users can give explicit relevance feedback.

For indexing and retrieving the text, we used the Terrier system, described in section 2.1.1. Textual and visual features are combined using a voting approach [6] when giving explicit relevance feedback.

2.3.1 The Interface

Our interface was modelled on similar video retrieval systems such as [1] and [2]. Figure 1 shows a screenshot of the developed system. It can be divided into three parts: the Search Panel, the Result Panel and the Playback Panel. The Search Panel contains a field for entering the query (which also allows boolean algebra). The Panel also contains an "Expand Query" button which will open a new window for relevance feedback (Query Expansion Window).

The Result Panel is divided into five tabs. The Search Results lists all retrieved video shots ranked. Each retrieved shot is represented via the extracted keyframe. When clicking on one frame, the video shot and additional information will be displayed in the Playback Panel. Under each keyframe, the user can click on radio buttons to explicitly rate the relevance of that particular result. According to their rating (relevant, maybe relevant, not relevant), the keyframes will be displayed in one of the other three tabs (relevance

Table 2: Overall experiment results

Run ID	MAP	P(10)	P(NR)	Recall
UG_F_1	0.03	0.121	0.077	0.148
UG_F_2	0.005	0.075	0.023	0.051
UG_F_3	0.018	0.146	0.052	0.11
UG_F_4	0.021	0.05	0.072	0.143
UG_F_5	0.004	0.1	0.025	0.059
UG_I_1	0.047	0.558	0.076	0.067

Table 3: MAP per topic

Topic	UG_F_1	UG_F_2	UG_F_3	UG_F_4	UG_F_5	UG_I_1
173	0.0132	0.0013	0.0065	0.0139	0.0026	0.0051
174	0.0007	0.0027	0.0008	0.0023	0.0126	0.0106
175	0.0009	0.0004	0	0.0002	0.0004	0.0394
176	0.0118	0	0.0066	0.0012	0	0.0006
177	0.0437	0.0003	0.0092	0.0075	0.0006	0.0222
178	0.1854	0.0001	0.0863	0.1112	0.0002	0.139
179	0.0689	0.0001	0.0224	0.006	0.0001	0.1178
180	0	0	0.0005	0.0002	0.0002	0.0225
181	0.0066	0.0001	0.0034	0.0051	0	0.1559
182	0.0564	0.0015	0.0071	0.0072	0.0115	0.0274
183	0.0094	0.0014	0.002	0.0094	0.0018	0.036
184	0.01	0.0036	0.0026	0.0027	0.0107	0.0165
185	0.0028	0.0002	0.0004	0.0062	0.0015	0.0115
186	0.0028	0.0015	0.0053	0.0025	0.0009	0.012
187	0.0371	0.0094	0.0065	0.0195	0.0005	0.0627
188	0.1341	0.0002	0.0675	0.0549	0.0009	0.0595
189	0	0.0161	0.0006	0.0039	0.025	0.0035
190	0	0.0001	0.0003	0.0006	0.0006	0.0229
191	0.002	0.0027	0.0018	0.0018	0.0057	0.0279
192	0.0011	0.0018	0.0012	0.0017	0.001	0.0034
193	0.0019	0.0002	0.0028	0.0064	0.0025	0.0042
194	0.0185	0	0.0667	0.0664	0.0001	0.1659
195	0.0577	0.0798	0.0291	0.0337	0.0252	0.0849
196	0.0504	0.0023	0.11	0.1355	0.0029	0.0878
all	0.0298	0.0052	0.0183	0.0208	0.0045	0.0475

tabs). Keyframes which have been retrieved in a prior retrieval are displayed in another colour for a better identification. Empty relevance tabs are disabled by default. The number of rated entries is displayed in each title of the tabs. Results can be moved to other tabs by rating them again. The features of the keyframes that are rated relevant will proposed as visual query for the next search in the query expansion window.

The fifth tab (Final Result tab) contains the keyframes that the user considered to be a result for the current search topic. When the user decides to play one video shot, he gets everything displayed in the Playback Panel which is placed on the right-hand side of the graphical user interface. On the top, he sees the selected keyframes in its context – with its neighboured keyframes to the left-hand and the right-hand side. He can obtain additional information about the video (Broadcaster, Program, Country, Date and Language) in moving the mouse over the keyframe. When clicking on the neighboured keyframes, the Playback Panel will be updated displaying the video shot and the additional information.

Underneath these keyframes, the interface displays the automatic speech recognition text of the selected video shot. In the middle of the Panel, the video shot is played. When the shot ends, the video pauses. The user can start and pause the video anytime on clicking on the typical icon under the video. The current playing position is presented with a slider bar. The user can use this bar to navigate in the video file.

On the bottom, the user can either mark a shot as a result or rate the relevance of the shot via buttons. Clicking on one of the four buttons will determine the time stamp of the shot that is currently played, detect the name of the shot in the MPEG-7 file provided by NIST and update the Result Panel. Every played shot is automatically added to the candidate list for the visual query visualised in the query expansion window.

The query expansion window assists the user to refine his query, Figure 2 showing a screenshot. On the top, the panel displays all keyframes the user marked as relevant or which were played during the run. The user can select or unselect each keyframe, indicating whether the keyframe is added as visual query or not. In the middle of the panel, a time span can be set to confine the search according to a date. The system also proposes exact dates, implicitly ascertained from the videos played before. On the bottom, the system suggests query terms that can be added to the query. The terms are taken from the video surrogate of the relevant rated or clicked keyframes or – if no keyframes have been rated or clicked before – from the Top 100 results of the initial query (pseudo relevance feedback). The user can change or add new terms and specify for each term if it has to appear (AND), if it may appear (OR), or if it may not (NOT) be in the video surrogate. In addition, the user can change the weight for each term.

2.3.2 Experimental Methodology

Users.

For the TRECVID experiment, we asked six users (five males and one female) who were *not* familiar with our system to each perform searches for 12 of the 2006 search topics. All the users had a primary degree and some an advanced degree. Most of them watch TV shows on a regular basis and according to their own judgement they have a good knowledge about current affairs in general. All of them claimed to use information systems very frequently. However, they rarely use any digital video retrieval system.

Experimental Design.

The design of the interactive experiment following the official guidelines. Each user had to work on 12 topics of the TRECVID

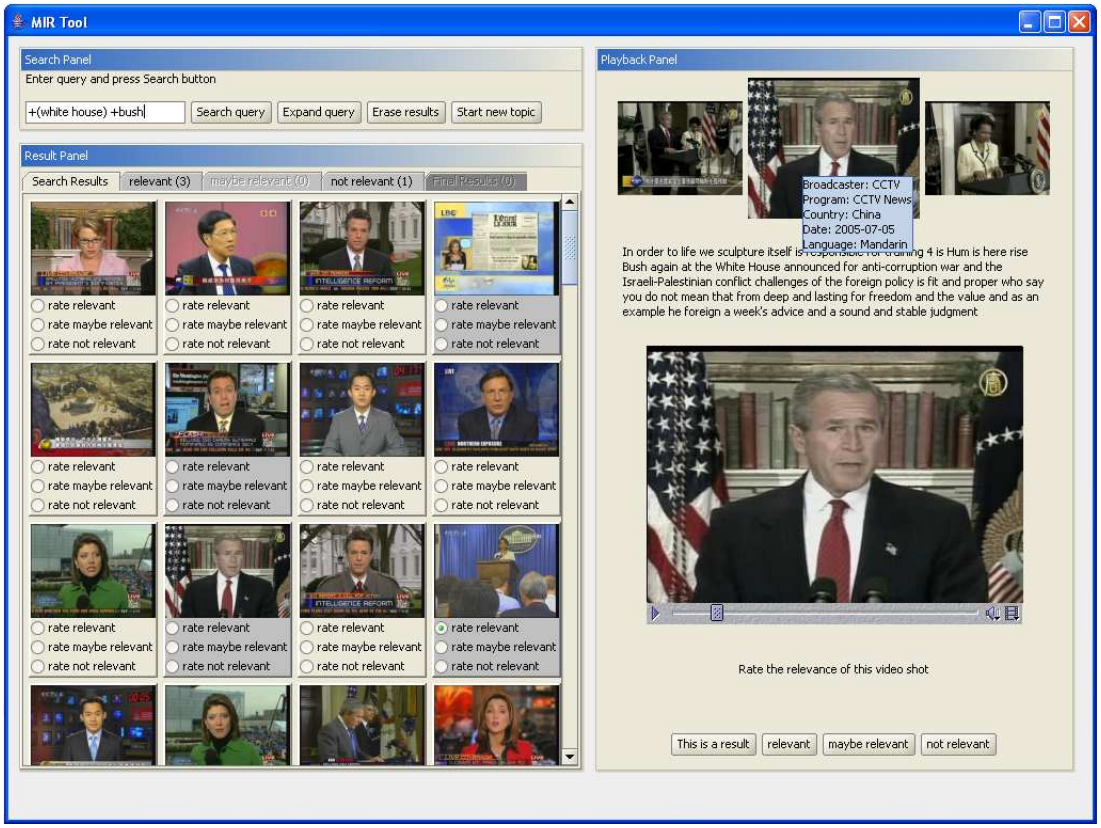

Figure 1: Graphical User Interface

2006 collection and as given by the guidelines they had a maximum time of 15 minutes for each topic. Each searcher took up to three hours each, all of whom carried out the searches in one morning or afternoon session.

For later internal evaluation, each user was asked to fill in several questionnaires:

- Prior to the evaluation, a pre-experiment questionnaire
- After each topic, a post-topic questionnaire was provided
- After the evaluation as a whole was finished, a post-experiment questionnaire was administered

Each of these questionnaires followed that developed by DCU for TRECVID 2004[1]. At the end of each experiment, each user was also informally questioned about their views of the evaluation system. In addition, the retrieval interface logged the actions of the users in a log file. Actions included shots marked as relevant, queries executed, and the interaction with other interface elements such as the keyframe browsing functionality.

2.3.3 Results

The results of the interactive evaluation are shown in tables 2 and 3. The results are poor compared to other organisations' interactive search submissions. Based on the feedback provided by users via the questionnaires, informal interviews, and interface logging, the retrieval system had a number of shortcomings:

- The relevance feedback system was slow in operation and therefore did not present an attractive method of carrying out query refinement. Indeed, the interface logs indicate that relevance feedback was used only 12 times by all users in the course of the evaluation.
- From the interviews and logs, it was found that half the users made considerable use of the video browsing functionality to find relevant shots.
- Similar to the automatic runs, the performance of the basic retrieval engine was not good, and did not provide users with good "starting points" in the browsing of the videos.
- There was some confusion with the differences between a shot which is "relevant", "maybe relevant" and a "final result". All users did not use the relevance feedback system significantly, and most users tended to select "final results" rather than relevant and mostly relevant.

Looking at the interactive results, queries which performed poorly in the baseline text-only run tend also to perform poorly in the interactive run. This can be explained by a quirk of the interface design, whereby the only why a user can execute a combined visual and textual query was via the relevance feedback system - which few of the users used. All other queries were text only (the interface as it stands does not allow the user to select example images for use in an explicit query). Finally, we would like to point out that our interactive experiments were performed by novice users who had no prior knowledge of the retrieval system rather than expert users (or system developers).

[1] http://www-nlpir.nist.gov/projects/tv2004/questionnaires.html

Figure 2: Query Expansion Window

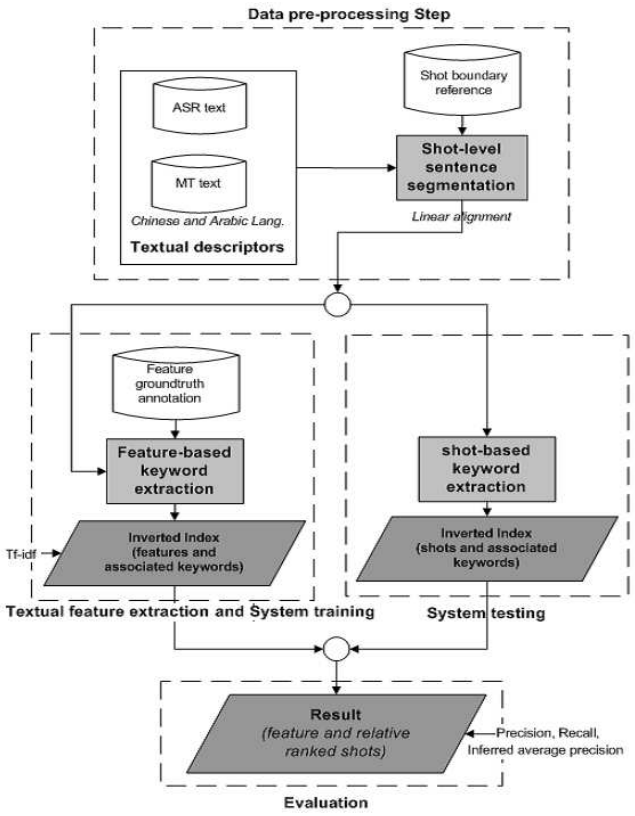

Figure 3: Architecture for high-level feature extraction system.

3. HIGH LEVEL FEATURE EXTRACTION

3.1 Approach

Outputs from ASR and MT systems were rich information sources. It was hoped that, by associated them with feature annotation and shot boundary reference, we would be able to identify many of, if not all, video shots relating to the given features without relying on other modalities. Figure 3 illustrates the architecture of the system. It consisted of several stages — broadly, data pre-processing stage and feature extraction stage (for training the system with 2005 data); the latter was paired with testing stage (with 2006 data). Finally the evaluation was performed by NIST.

3.1.1 Data Pre-Processing

Data pre-processing was concerned with extraction of textual attributes. The textual descriptors were provided, however, it required some pre-processing to put them together, partially due to differences in formatting. ASR and MT data were aligned with shot units by employing speaker time and the shot boundary reference (referred to as 'shot-level sentence segmentation'). It was followed by identification of the most significant words occurred in shots that were labelled with high-level features ('feature-based keyword extraction').

ASR and MT outputs.

The ASR transcripts and translations from Chinese and Arabic sources were provided. Time stamps were used to align words to the individual shots. Stop words were removed and stemming was performed. We encountered several problems. Firstly, the MT texts did not always correspond to the most relevant video scenes. In some cases, a portion of translations or ASR transcripts was lost from the data provided. Not surprisingly, there were shots without any textual descriptors. In the current implementation, these shots could not be processed. We are considering the use of textual information from adjacent shots in order to alleviate the problem. Information from adjacent shots may also be useful for refining the list of the most significant words.

Common shot boundary reference.

The shot boundary reference was released by the TRECVID organiser. The news story is considered as a concatenation of individual video portions. The frames within one motion-camera normally describe the same story. A story may be produced by including all frames from one continuous unit of video. Therefore, shot-level segmentation can provide a reasonable structure for the contents of video.

Feature annotation.

Using the feature annotation, we should be able to identify shots that describe the features. The annotation for the TRECVID 2005 data was provided by the MedialMill team [5]. 101 features were annotated for 169 hours of Arabic, Chinese and the US broadcast news, out of which 39 features were involved in this year's task. A number of shots is extracted for each feature and associated with ASR and MT texts using time stamp information. We realised that there existed shots that did not match the annotated feature. This had very serious effects on the performance of the system.

3.1.2 Textual Feature Extraction

For each word, the *tf-idf* score was calculated. The procedure produced a ranked list of the most significant words for individual high-level features. We found 6 297 significant words for 39 features (161 words per feature on average)[2]. Note that we examined the use of subsets (say, 70% or 85%) instead of using the complete set of significant words for testing. It was found that there was no significant difference in terms of precision and recall. In practice, a subset might have been sufficient because it could save space and processing time.

3.2 Experiment

3.2.1 Experimental Design

We derived a list of the most significant words from the TRECVID 2005 data, using the annotation of high-level features produced by the MedialMill team [5], as reference. ASR transcripts and MT texts were aligned with corresponding shots and the standard textual feature extraction techniques were applied. For evaluation the TRECVID 2006 dataset was utilised. It comprised of 158.6 hours of video in three languages including English, Chinese, and Arabic.

We completed a single run for all 39 high-level features, using the text-based system described earlier. First, occurrences of significant words were examined in shot units. When the extracted words were significant enough, shots were associated with the corresponding high-level feature. The final result was a list of ranked shots classified by individual features.

3.2.2 Results and Discussion

Our submission was evaluated by NIST using the inferred average precision. Figure 4 shows the results that compare our scores with minimum, median and maximum scores. On average, our submission resulted in precision for 2000 shots at 0.0119 and for 100 shots at 0.0480. In total, 475 shots were identified correctly out of 9074 groundtruth shots. As a result, the inferred average precision was calculated as 0.005.

Problem caused by the erroneous annotation of high-level features.

As noted earlier, we noticed that, for the TRECVID 2005 data, there existed a number of shots that did not match the annotated high-level features. This has caused a serious effect on our system. We are still investigating the extent of this problem.

Problem caused by news contents.

The system was developed from the TRECVID 2005 data, and then applied to the 2006 data. Because the system relied on occurrences of particular sets of words, changes in news contents from 2005 to 2006 certainly has some effect on the performance.

Problem caused by alignment.

Time stamps were utilised to align ASR and MT text to shot segments. Our assumption was that, within a shot, significant words would occur that described that particular shot. Clearly, this assumption was not quite correct. There were many occasions that some words could be strongly related to the next or the previous shot. For example, there were cases in which anchors who appeared in a studio shot were talking about the contents of a report in the next shot. We are currently experimenting with the alignment using speaker information.

[2]Stemming and stopping were applied at the earlier stage.

Problem caused by the number of features.

We have applied the same approach to all 39 high-level features. The question is – would it be possible to apply a single scheme to many different kinds of features? Clearly, we might be able to achieve better by focusing on one particular feature at the cost of the rest of features. But that luxury cannot always be expected. For the current submission, we developed a system solely based on textual information. It is likely that the overall performance would be improved by combining multiple approaches for multiple modalities, and now we are looking at this direction.

4. CONCLUSIONS

The Glasgow University team submitted five automatic and one interactive run. In the automatic runs, text-only, visual-only and feature combination based on a graph model were compared. The visual runs were based on a combination of global MPEG-7 descriptors. As expected, these global features alone performed poorly. The baseline text run appears roughly in line to other submissions, suggesting a similarity of approach with other participating organisations. We also experimented with a novel combination approach based on a graph model incorporating both visual and textual features. Although the combination runs submitted officially did not improve the text-baseline, an unofficial run based on the simplest setting of the graph led to a slight improvement. Finally, the interactive experiment provided much feedback into the design of the interface, for current and future refinement.

The Sheffield University team used information derived from ASR and MT data for the high-level feature extraction task. During the system development, we have encountered several problems, some of which were critical to the system performance. We are currently analysing the results obtained, aiming at further development in the area.

5. ACKNOWLEDGMENTS

The research leading to this paper was supported by the European Commission under contracts FP6-027026 (K-Space) and FP6-027122 (Salero).

6. REFERENCES

[1] E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G. J. F. Jones, H. Le Borgne, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N. E. O'Connor, N. O'Hare, S. Rothwell, A. F. Smeaton, and P. Wilkins. TRECVID 2004 Experiments in Dublin City University. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

[2] R. Jesus, J. Magalhães, A. Yavlinski, and S. Rüger. Imperial College at TRECVID. In *TRECVid 2005 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Maryland, 14-15 November 2005*, 2005.

[3] L. Lovasz. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:353–398, 1993.

[4] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on Information Retrieval (ECIR 05), Santiago de Compostela, Spain*, 2005.

[5] C. G. M. Snoek, M. Worring, J. C. van Gemert, J. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the ACM International Conference on Multimedia, Santa Barbara, USA*, 2006.

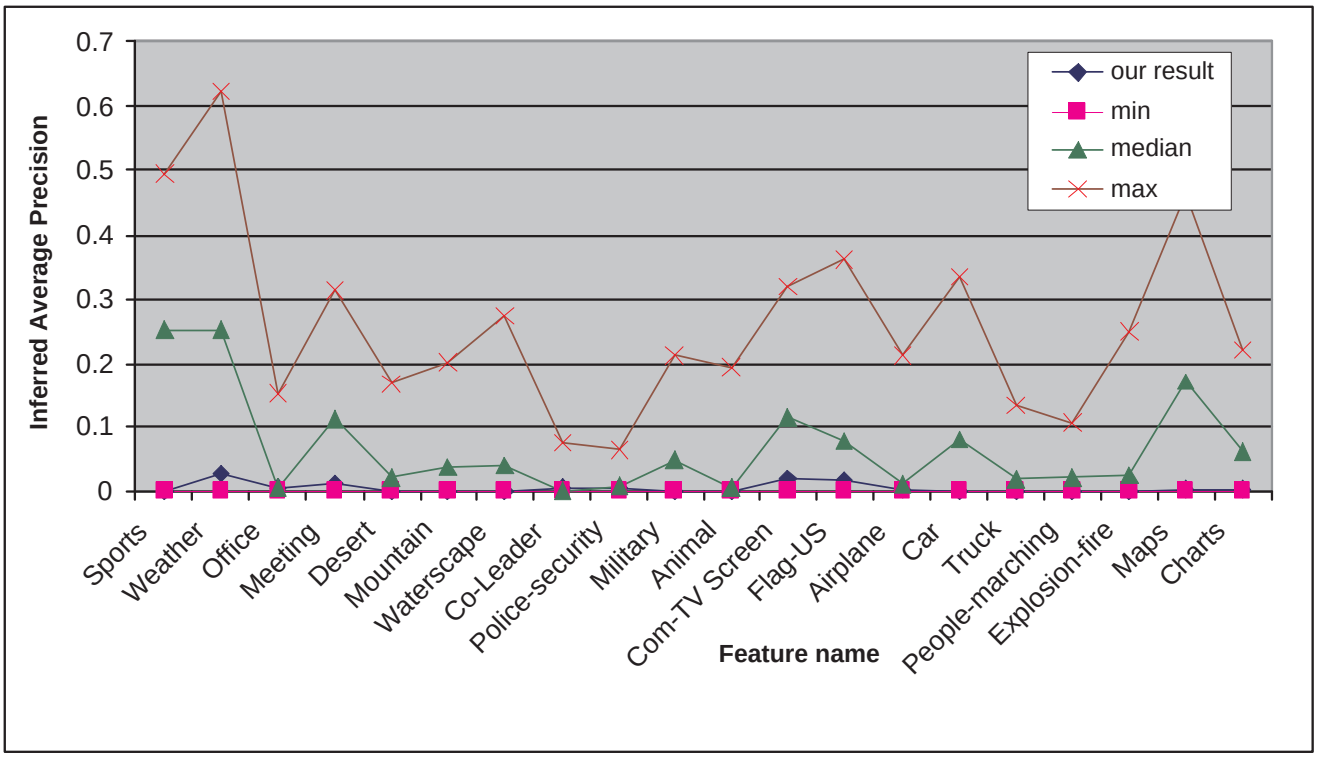

Figure 4: The inferred average precision scores for selected 20 features.

[6] J. Urban and J. M. Jose. Adaptive image retrieval using a graph model for semantic feature integration. In *Proc. of the 8th ACM SIGMM Int. Workshop on Multimedia Information Retrieval (MIR'06)*. ACM, 2006.