A Unified, Modular and Multimodal Approach to Search and Hyperlinking Video

John Preston
jlp1g11@ecs.soton.ac.uk

Jonathon Hare
jsh2@ecs.soton.ac.uk

Sina Samangooei
ss@ecs.soton.ac.uk

Jamie Davies
jagd1g11@ecs.soton.ac.uk

Neha Jain
nj1g12@ecs.soton.ac.uk

David Dupplaw
dpd@ecs.soton.ac.uk

Electronics and Computer Science, University of Southampton, United Kingdom

ABSTRACT

This paper describes a modular architecture for searching and hyperlinking clips of TV programmes. The core component of the system consisted of analysis of sections of transcripts based on a textual query. Results show that search is made worse by the addition of other components, whereas in hyperlinking precision is increased by the addition of visual features.

1. INTRODUCTION

The 2013 MediaEval search an hyperlinking task [1] tackles two problems; search across and within video collections, and hyperlinking of short video segments relevant to a given anchor segment. This paper describes the system we built to address the two tasks.

2. ARCHITECTURE

For the search sub-task, the system was to take a query and return a series of clips as results. This necessitated extracting relevant or otherwise appropriate sections from programmes. To facilitate this, individual programmes were generalised to functions of interest over time, where the real value at a given time indicated the instantaneous relevance of that point within the context of the current query. This paradigm permitted a modular approach to the system architecture, where individual modules can be applied to add, remove, or modify a set of timelines.

Each module operates by either building a probability density function (PDF) across a timeline or by adjusting the factor by which a timeline was scaled (so indicating an overall increase in the relevance of one timeline compared to another). PDF's were generated by placing Gaussian functions at the point of interest on the timeline. The variance of each Gaussian was proportional to the length of the segment of interest.

2.1 Transcript module

The transcript module was the most important component of the system, doing most of the heavy lifting when it came to determining the relevant sections of a programme. The module searched for keywords (taken from the query string) across all transcripts of a certain kind (LIMSI, LIUM, or subtitles). Matches were extracted from each transcript in turn, and a binary tree of these hits was then built using agglomerative clustering. The tree was walked, and when a cluster's separation (calculated as the distance between the average values of the cluster's left and right children) fell below a specified threshold, the cluster was used to build a Gaussian whose amplitude was calculated from

$$\alpha = \frac{|W|}{|Q|} \sum_{w \in W} \mathrm{boost}(w)\,\mathrm{idf}(w)$$

where W was the set of keywords in the transcript, Q was the set of all possible keywords from the query, $idf : W \rightarrow \mathbb{R}$ was a function mapping each keyword on to its inverse document frequency, and $boost : W \rightarrow \mathbb{R}$ was a function mapping each keyword on to its boost in the query. Additionally, the true amplitude was scaled by the normalised score returned by the search engine when searching for transcript documents matching the query. Thus the amplitude of the Gaussian captures the relevance of all keywords in the cluster with respect to the document, as well as how completely the cluster covers the set of all possible query terms. The Gaussians were centred on the midpoint of the range covered by the cluster, and the parameter c of the Gaussian was chosen as one third of the temporal size of the cluster plus 60 seconds.

2.2 Other modules

The *synopsis* and *title* modules increased the scale factor of any timelines whose synopses or titles matched the query by an amount derived from the search engine's score for the query in those fields and for the programmes corresponding to those timelines. The *channel filter* module was implemented which performed some naïve NLP on the query: if it was found that a channel was mentioned in the query then any timelines corresponding to programmes on other channels were removed from the timeline set. The *concept* module looked in the query text and at visual cues for known concept detections that could be added to timelines. The amplitude for the concept module's Gaussians was determined from the normalised confidence for each concept detection, and the width was a constant 5 seconds.

Additionally, another module worked purely visually, finding shots that were visually similar to existing shots with high confidence. For each programme, the most stable keyframe of each shot was extracted and SIFT features were calculated. Each SIFT feature was hashed using locality-sensitive hashing (LSH) and a graph was constructed where the vertices were keyframes and edges were created if pairs

MediaEval 2013 Workshop, October 18-19, 2013, Barcelona, Spain

Table 1: Results for the search task

Run code	MRR	mGAP	MASP
S_M_Mod	0.208	0.0973	0.113
U_M_Mod	0.141	0.0812	0.0587
I_M_Mod	0.149	0.0828	0.581
S_MV_ModCon	0.146	0.0743	0.726
U_MV_ModCon	0.0808	0.0542	0.0401
I_MV_ModCon	0.0746	0.0412	0.0208
S_MV_ModConLSH	0.117	0.0652	0.0533
U_MV_ModConLSH	0.0510	0.0383	0.0211
I_MV_ModConLSH	0.0723	0.0431	0.0221

of key-frames contained colliding features [3]. The module operated by finding sections of timelines corresponding to shots whose integrals exceeded a threshold (i.e. shots already deemed relevant by the preceding modules), and added Gaussians centred on the shots whose keyframes were directly connected to this keyframe on the LSH graph. The base amplitude of the Gaussians was determined as the fraction of functions under which the two keyframes collided to the largest number of collisions, and the true amplitudes were additionally scaled by the integral of the shot from which the graph traversal originated. A constant width of 60 seconds was used.

2.3 Hyperlinking

The hyperlinking sub-task was viewed as an extension of the search sub-task, where the search query was an anchor as opposed to a string. Thus hyperlinking was addressed by slightly modifying the search architecture.

A module was written to construct textual queries from anchors by extracting the portion of a transcript throughout the anchor segment. This allowed the use of the transcript, synopsis, title, and channel filter modules from the searcher architecture without any modification.

The concept module was modified to find any concept detections during the anchor segment and to bring in other sections of video matching these concepts, whereas in the searcher architecture concepts were inferred from the query text.

The LSH-SIFT graph module was modified to operate by searching for similar frames starting with the frames in the anchor segment, whereas before this module was purely expansionary, operating on timeline sections that were already of high confidence.

2.4 Tools and techniques

In order to facilitate searching of transcripts, synopses, and programme titles, these features were indexed using Apache Lucene[1]. SIFT features were extracted from keyframes using OpenIMAJ[2] [2] and Hadoop[3].

3. RESULTS

The results from the search task are summarised in Table 1. Runs using the subtitles gave the best performance in each category, which is understandable due to the more accurate nature of subtitles compared with speech-to-text transcripts. It is interesting to observe that the performance of the system decreased as additional features were brought in, which may indicate that these additional modules were not scaled properly, or otherwise very noisy in the context of the query. This is surprising for concept detection, as in the search task concepts were directly picked from the query.

Table 2: Results for the hyperlinking task

Run components	Retr.	P5	P10	P20
Subs	5412	0.513	0.483	0.390
Subs, concepts	7510	0.560	0.517	0.425
Subs, concepts, LSH	7515	0.573	0.520	0.428

Table 2 shows the results for the hyperlinking task. It can be seen that as additional features were added, the precision of the system increased, which may be a result of concept detection and LSH graph searching occurring directly on a segment of video, as opposed to extracting concepts from a textual query and using LSH as a purely expansionary tactic. Additionally, baseline results indicate that this architecture, at least in its current configuration, is more oriented towards hyperlinking than search.

4. CONCLUSION

The system implemented served much better at the hyperlinking sub-task than the search sub-task; this was slightly unexpected as the performance on the development data was higher. This may in part be due to fundamental limits to core aspects of the architecture, namely the transcript module: textual queries have a low bandwidth and describe many features that are not discernible from a programme's transcript, and so a more complex approach is required to improve performance. Additional NLP on queries, along with person detection (i.e. face recognition/verification), could significantly improve performance in the search domain, and we would like to explore this further in the future.

For hyperlinking, results indicate that a naïve approach using visual features such as concept detections can readily improve the performance of such systems. Additionally, results show that transcript segments from anchors provide a rich source of textual information for building a search base.

5. ACKNOWLEDGMENTS

The described work was funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements 270239 (ARCOMEM), and 287863 (TrendMiner).

6. ADDITIONAL AUTHORS

Additional author: Paul H. Lewis (phl@ecs.soton.ac.uk)

7. REFERENCES

[1] M. Eskevich, G. J. Jones, S. Chen, R. Aly, and R. Ordelman. The Search and Hyperlinking Task at MediaEval 2013. In *MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.

[2] J. S. Hare, S. Samangooei, and D. P. Dupplaw. OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *ACM MM'11*, pages 691–694. ACM, 2011.

[3] J. S. Hare, S. Samangooei, D. P. Dupplaw, and P. H. Lewis. Twitter's visual pulse. In *ICMR'13*, pages 297–298, New York, NY, USA, 2013. ACM.

[1] `http://lucene.apache.org`

[2] `http://openimaj.org`

[3] `http://hadoop.apache.org`