SZTAKI @ TRECVID 2009*

Bálint Daróczy Dávid Nemeskey István Petrás
András A. Benczúr Tamás Kiss
Data Mining and Web search Research Group, Informatics Laboratory
Computer and Automation Research Institute
of the Hungarian Academy of Sciences
{daroczyb, ndavid, petras, benczur, kisstom}@ilab.sztaki.hu
http://datamining.sztaki.hu

March 1, 2010

Abstract

We summarize our *fully automatic* approach to the TRECVID 2009 Search task. Our submissions summarized in Table 1 use linear combinations of the following basic techniques.

- **text** ASR text retrieved by the Dutch translation of selected topic terms.
- **image** Similarity of representative frames of shots.
- **face** Face detector output for topics involving people.
- **feature** Total weight of high level feature classifiers considered relevant by text based similarity to the topic. We used the publicly available feature predictions.
- **motion** Motion information extracted from videos where relevant to topic.
- **wide** A variation of text with wider shot neighborhood considered relevant.
- **lattice** Text retrieval based on ASR lattices where available.

The combination of **feature** and **face** together contributed most to the performance of the system. In this experiment the use of **lattice**s, although they were available only for part of the shots, did not improve over the most probable ASR output. The best ASR text based run is **text + wide**, a combination where more distant shots also receive partial score for a matching speech. We notice that the plain linear combination of all scores deteriorated performance. In the paper we measure independent performance of the methods and observe that **feature** alone would have outperformed all of our runs. An improved combination includes **wide** with lower weight.

1 Introduction

In this paper we describe our approach to TRECVID 2009 Search task [15] using fully automatic processing. The data set consisted of 280 hours of video with approximately 96400 shots with the corresponding automatic speech recognition (ASR) transcript. For a subset of the shots, ASR output also included lattice files generated by Huijbregts et al. [5]. We also relied on publicly available annotations for shots produced by participants of High Level Feature Extraction task.

The key feature of our solution is to combine several different text based and content based image retrieval scores. We describe our approaches that rely on the visual part of the topic (image similarity, motion and face detection) in Section 2 and on the topic description (ASR retrieval, relevant high level feature selection, motion and person selection) in Section 3. Finally the combination method and our runs, both submitted and additionally evaluated, are summarized in Section 4.

*This work was supported by the EU FP7 project JUMAS – Judicial Management by Digital Libraries Semantics and by grants OTKA NK 72845 and NKFP-07-A2 *TEXTREND*.

Run ID	MAP	Method
budapest0	0.0163	text + image + feature + $0.1 \cdot$ face + $0.1 \cdot$ motion + wide + lattice
budapest1	0.0225	text + image + $0.1 \cdot$ face + $0.1 \cdot$ motion + wide
budapest2	0.0226	text + image + feature + $0.1 \cdot$ face
budapest3	0.0225	text + image + feature + $0.1 \cdot$ face + $0.1 \cdot$ motion + wide
budapest4	0.0039	text + image
budapest5	0.0051	text
budapest6	0.0050	lattice
budapest7	0.0008	image
budapest8	0.0223	text + image + feature + $0.1 \cdot$ motion
budapest9	0.0062	text + wide

Table 1: Summary of our ten runs submitted.

2 Topic visual content processing

We developed a video processing subsystem. It has the following main functionality:

- Video prefiltering with smoothing and adaptive intensity equalization;
- Low level feature tracker, including
  - optic flow estimation based on corner point tracking [1, 13];
  - automatic motion compensation based on optic flow;
  - shot detection based on motion vectors with adaptive salience level; and
  - motion trajectory analysis of corner points.

We processed each frame and segmented the video into shots. We made also use of the provided shot boundaries. During the processing a motion feature vector was produced for each video frame. This vector contains the following information: detected shot boundary, panning camera motion, global motion vector, global motion histogram. An image frame and a motion vector was associated with each shot. This frame and vector served as input for the further processing.

2.1 Image processing

We transformed images into a feature space in order to define their similarity for ad hoc retrieval. For image processing we computed SIFT [10] descriptors. This resulted in approximately three times 500-700 keypoints and the corresponding descriptors in 128 dimensions. We also used six Haar-wavelet based face detector with binary outputs to generate high level face prediction for each image.

2.2 Codebook generation

The number of local descriptors produced for each frame varies from image to image. To have fixed dimensional representation of an image we prepared a visual codebook from the training set. We clustered the feature vectors using K-Means with two thousands cluster centers. After this all feature vectors were quantized according to the cluster centers. We chose the K-Means because of its low computational cost. Using the output of the clustering we computed normal and modified versions of BOV representations for each image.

2.2.1 BOV representation

The Bag of Words generative modeling is a well known technique from text domain [7, 6]. It also has been proved successful in image categorization and retrieval. [3, 14]. The images (shot keyframes) are represented by $D = 2000$ dimensional vector. The best performing BOV term frequency vector consists of the histogram of the codewords from the codebook. We tested four additional descriptors, but none of them improved over the original version:

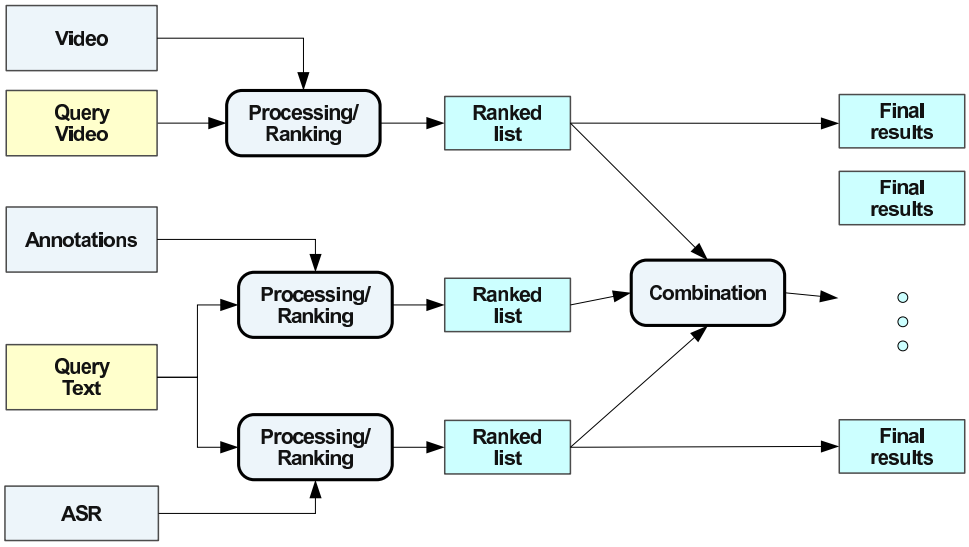

Figure 1: High level flowchart of the processing pipeline. We computed ranked lists from several modalities. We included these list and their various combination in the final ranked lists. The modalities were: video, ASR text, annotation from the High level Feature Extraction task.

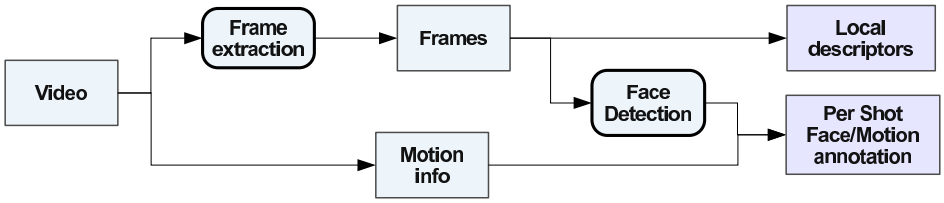

Figure 2: Video feature extraction. The video was segmented into frames automatically. Each shot was associated with one image frame and a characteristic motion vector extracted from that shot. We computed SIFT descriptors in the monochrome and normalized RGB channels. A face descriptor was also computed for each shot.

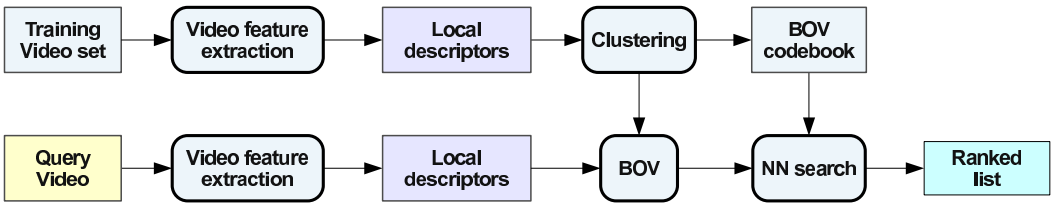

Figure 3: Flowchart of the Bag of Visual Words representation algorithm. First, We generated a visual codebook from the local descriptors of images computed in the video processing step. For codebook building we used K-Means clustering.

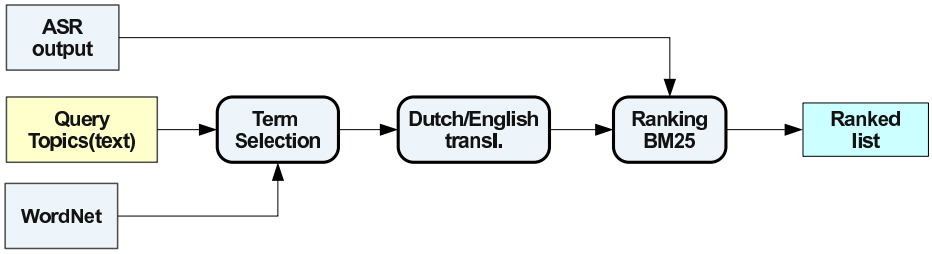

Figure 4: ASR based ranking generation

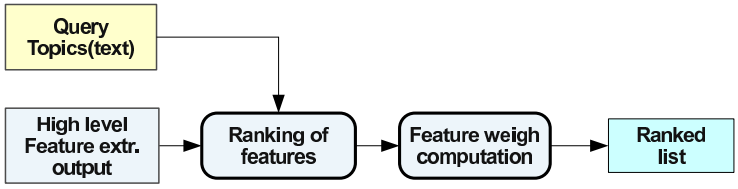

Figure 5: High level feature based ranking generation

- "Soft" BOV vector: $x^s = \sum_j \frac{x_{ij}}{\sum_i x_{ij}}$, where x_{ij} denotes the $i-$th column of the j-th quantized feature vector of one image. Here $i = 1, \ldots, 2000$ and j is typically between 500–2000.
- Inverse "Soft" BOV vector: $x^i = \sum_j 1 - \frac{x_{ij}}{\max_i x_{ij}}$.
- Normalized "Soft" BOV vector: $x^{ns} = \frac{x^s}{\sum_k x^s_k}$. where x^s_k is the element of $x^s = (x^s_0, x^s_1 \ldots x^s_D)$, that is a D dimensional vector.
- Normalized Inverse "Soft" BOV vector: $x^{ni} = \frac{x^i}{\sum_k x^i_k}$ where x^i_k is the element of $x^i = (x^i_0, x^i_1 \ldots x^i_D)$

We tested different distance metrics for nearest neighbor image similarity search (χ^2, L1, L2). In the final experiments we used the best performing cosine distance, $D_{\cos}(x, y) = \arccos \frac{xy}{\|x\|\|y\|}$.

3 Topic text based retrieval

We utilized the available textual information: the automatic speech recognition (ASR) output, the results of the high level feature extraction track and the topics themselves. We have implemented a text processing subsystem for this task. It consists of the following modules:

- Preprocessing
- Text retrieval
- Annotation analysis

We used text retrieval to find the relevant parts in the ASR output and the corresponding high level features for the topics. The results were used as-is in our text-based runs. They also formed the base of our combination runs.

The output of the annotation analysis was used to decide if motion processing and face recognition are necessary for a particular topic and to weight the results of those subsystems.

We used the Hungarian Academy of Sciences text retrieval engine [4] that is based on Okapi BM25 [12] with proximity weights [11, 2].

3.1 Topic text preprocessing

We have found the topic descriptions in their original form unsuited to be used as text queries. Firstly, they contain many words that describe information about the video, not its content. Secondly, words that contain no information by themselves, such as prepositions and numbers, would greatly lower search performance. Therefore, a preprocessing phase was needed.

We parsed the topics with the Stanford lexical parser [8, 9]. This enabled us to determine the part-of-speech (POS) tags of all words in the topics and to stem them accordingly. We also extracted the phrase and dependency structure of the topic descriptions.

To identify the meta-description in the topics, (e.g. statements about the video itself, as opposed to its content; the "Find shots of" instruction also falls in to this category) we assembled a list of words that might take part in such descriptions, based on previous TRECVID topics. Then we removed these words from the topics, together with any words governed by them in the phrase and dependency structure.

Next, we filtered the remaining words based on their POS tags. Our assumption was that nouns would make the best query words, followed by verbs and adjectives. Especially in case of the ASR output, it seemed unlikely that the narration would describe what was happening in the video, let alone with the same words as in the topic descriptions. However, if people were interacting with things, there was a chance they would refer to them. Based on these considerations, we assigned the following weights to the words: nouns 1, verbs 0.5, adjectives 0.2; all other words were dropped from the query.

Since the ASR output is in Dutch, we translated the queries using online dictionaries word-by-word. As a result, several synonyms were produced for every English word. For a group of alternate translations we have only kept the highest tf.idf in every document. We did not retrieve English queries in the machine translation available for the TRECVID 2009 data.

3.2 Window width for relevance

We loaded the speech acts as the documents to the search engine. A speech act consists of the words in the ASR output that have the same speech id. We run a regular text-based search with the translated queries. This produced a scored list of speech acts. We then calculated shot scores as follows: if a shot overlaps with a speech act, it receives its full score. Otherwise, the score is reduced proportionally to the distance between the speech and the shot; if the two were more than sixty seconds apart, no score was awarded. We used the resulting ranked shot list in the **text** run.

We also created a wide shot list. The only difference to the process described above is that shots received scores of speech acts as far as 300 seconds.

3.3 Retrieval in ASR lattices

In comparison to the most likely utterance, we have implemented a retrieval method working directly on speech lattices for some of the TRECVID 2007-2008 data where lattices from Sound and Vision [5] were available. For remaining speech data we used the most likely text as fallback mechanism.

When processing lattices, for each node w of the lattice we computed the likelihood of all paths reaching term w. When computing likelihood we normalized over out-edges. By this procedure we have obtained a probability for every word w that we used instead of the tf score. We also implemented an algorithm that takes proximity as in [11, 2] into account by counting likelihood and path length. Due to the lack of meaningful expressions in the topic text we turned this feature off in the experiments.

3.4 High level video feature relevance

The topics were also matched against the 20 high level features made available from the feature extraction track. The feature descriptions served as documents in the search engine. This provided us with a scored list of possibly relevant features for every topic. To determine the relevance of a shot to a topic, we computed the sum of these scores, weighted by the results of the high level feature extraction track for the shot.

The result of this method was used in the **feature** list, which, along with the text and image runs, served as the base for the combination runs. This list was not submitted for evaluation, though later it proved to provide the best results (see Table 2). To evaluate the strength of the features, in addition

MAP	Method
0.0003	motion
0.0003	face
0.0021	image Normalized “Soft” BOV
0.0021	image “Soft” BOV
0.0023	image Normalized Inverse “Soft” BOV
0.0023	image Inverse “Soft” BOV
0.0038	image BOV
0.0050	lattice
0.0051	text
0.0053	lattice + wide
0.0064	text + wide
0.0378	features
0.0382	features + $0.2 \cdot$ face
0.0381	features + $0.2 \cdot$ motion
0.0418	features + $0.2 \cdot$ wide

Table 2: Results of the post-submission experiments with explanations given in the Abstract.

we show two kinds of manual strategies to determine the weight of a topic–feature pair. In the “strict” strategy we gave weight 1.0 only to those pairs linked grammatically and connected by meaning. And in the “lax” strategy we also gave weight 0.5 for those pairs where we could find an interpretation between the topic and the feature. By the “strict” strategy we have only 15 topic–feature relations resulting in a MAP 0.0922 based on the relevant topics (13 out of 24) and MAP 0.0499 on the full set of topics. The MAP of “lax” based on 58 relations is 0.0545 (22 relevant topics) and 0.0499 on the full set. Our experiments showed if we manually assign weights between the topics and the high level features the results are significantly better than in our automatic strategy.

3.5 Video events

Topic descriptions provide information we could use in conjunction with the output of our face and motion detectors.

Some queries required that the shots contain people. We assembled a list of synonyms for “person” and “human” using WordNet. Topics that contain any of these words were flagged. When evaluating these topics, a **face** score was awarded for shots, which equals to the output of the face detector.

We assumed that motion may be another valuable indicator. We assigned a motion score for each topic in the following way: for every verb among the query words, we divided the number of senses that represent motion in WordNet by the number of all senses, and then we summarized these values. We multiplied this score by the output of the motion detector to assign the **motion** score for shots.

These scores are inadequate to form the base of an independent run. However, we used them in the combination lists.

4 Results

We summarize the results of our submissions in Table 1 and our post-submission experiments in Table 2. In our official runs, the combination of text retrieval, image similarity, relevant feature scores and a downweighted face detection performed best, closely followed by additional combinations with motion detection and wide window text retrieval.

In the post-submission analysis we find highest contribution of the features themselves, a potential run that we did not submit. The performance of the features can further be improved by downweighted combination. Reduced linear weight values can now be guessed by the poor performance of the remaining methods. In this respect ASR retrieval has a contribution in MAP increase higher than its own MAP score.

5 Conclusions

In our experiments the best performing method was the automatic detection of connections between the high-level video features and the topics. In combination we successfully applied ASR retrieval and our face detector. We believe that the highest potential of improvement lies in our image retrieval methods that have only moderately been adapted to the TRECVID search track. We did not deploy cross-modal feedback mechanisms that usually outperform the simple linear combination of the scores.

References

[1] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm, 2000.

[2] Stefan Büttcher, Charles L. A. Clarke, and Brad Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *SIGIR '06*, pages 621–622, New York, NY, USA, 2006. ACM Press.

[3] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[4] Bálint Daróczy, Zsolt Fekete, Mátyás Brendel, Simon Rácz, András Benczúr, Dávid Siklósi, and Attila Pereszlényi. Cross-modal image retrieval with parameter tuning. In Carol Peters, Danilo Giampiccol, Nicola Ferro, Vivien Petras, Julio Gonzalo, Anselmo Peñas, Thomas Deselaers, Thomas Mandl, Gareth Jones, and Nikko Kurimo, editors, *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the Cross-Language Evaluation Forum*, Lecture Notes in Computer Science, Aarhus, Denmark, September 2008 (printed in 2009).

[5] M. Huijbregts, R. Ordelman, and F. de Jong. Speech-based annotation of heterogeneous multimedia content using automatic speech recognition. 2007.

[6] Huma Lodhi Huma, Craig Saunders, Nello Cristianini, Chris Watkins, and Bernhard Scholkopf. Classification using string kernels. *Journal of Machine Learning Research*, 2:563–569, 2002.

[7] Thorsten Joachims, Fachbereich Informatik, Fachbereich Informatik, Fachbereich Informatik, Fachbereich Informatik, and Lehrstuhl Viii. Text categorization with support vector machines: Learning with many relevant features, 1997.

[8] Dan Klein and Christopher D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*. December 2002.

[9] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[10] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Corfu, Greece, 1999.

[11] Yves Rasolofo and Jacques Savoy. Term proximity scoring for keyword-based retrieval systems. In *ECIR*, pages 207–218, 2003.

[12] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. In *Document retrieval systems*, pages 143–160. Taylor Graham Publishing, London, UK, UK, 1988.

[13] J. Shi and C. Tomasi. Good features to track. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[14] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003. IEEE Computer Society.

[15] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.