# TRECVID 2010 Known-item Search (KIS) Task by I$^2$R

Lekha Chaisorn, Kong-Wah Wan, Yan-Tao Zheng
Yongwei Zhu, Tian-Shiang Kok, Hui-Li Tan, Zixiang Fu, and Susanna Bolling

**Institute for Infocomm Research (I$^2$R), A*STAR**
**1 Fusionopolis Way, Connexis Tower, Singapore**

## ABSTRACT

The KIS task can be regarded as an extreme case of target-specific video search, in which the query aims to uniquely locate a single true answer. Locating the unique video for a query, however, poses new challenges over existing information retrieval approaches. Our participation in TRECVID this year focuses on how to adapt traditional information retrieval, specifically video search, methods to KIS in both automatic and interactive setting. In automatic KIS, as there exists a single true answer for each query, the input queries are expected to present distinctive information locating a unique entity but not a broad topic covering a number of relevant videos. Therefore, query formulation is one of our focuses in automatic KIS. On the other end of the spectrum, our emphasis in interactive KIS is two-fold. First, an intuitive and user-friendly user interface is developed to facilitate the browsing of returned videos. As the query is usually specific, we postulate that searchers can quickly reject most of the negative videos after seeing a few keyframes of the video. This premise of “fast rejection” motivates us to leverage the storyboard to pre-visualize a video. When users can not reject a returned video as negative, he/she may indicate it as a relevant one. By collecting a number of relevant videos, the searchers can perform relevance feedback to refine the retrieval and continue the search. The automatic and interactive KIS achieve MAP of 0.454 and 0.727 respectively, showing the effectiveness of the proposed methods.

## 1. INTRODUCTION

The known-item search (KIS) task is to retrieve a unique video that the searcher has known and seen before in a video corpus. The KIS task can be regarded as an extreme case of target-specific video search, in which the query aims to uniquely locate a single true answer. Compared to traditional video search, KIS simulates the real-life video search scenario in a more practical setting. The information need of a video searcher can usually be satisfied by a single answer video, rather than a list of relevant videos. Locating the unique video for a query, however, poses new challenges over existing information retrieval approaches. Our participation in TRECVID this year focuses on how to adapt traditional information retrieval, specifically video search, methods to KIS in both automatic and interactive setting.

Automatic KIS takes a text-based query and returns a ranked list of possible answers. As there exists a single true answer for a query, the input queries are expected to present distinctive information locating a unique entity but not a broad topic covering a number of relevant videos. Therefore, query formulation is one of our focuses in automatic KIS. Traditional video search usually expands the query to match more videos/documents, broadening the list of relevant answers. This solution, however, may not be effective in KIS. Considering a query has one unique answer, the more returns by query expansion may not increase the retrieval recall, but unfavorably bring in more noise. To achieve more distinctive KIS query, we refine the query by formulating query phrases and weighing query terms. A semantic query phrase searches for a group of words functioning as a single unit, and thus yielding more distinctive and specific retrieval. Word terms in query carry different importance. By weighing different terms, we can alleviate the vagueness of redundant terms and emphasize on the distinctive ones in the retrieval. In addition of refining queries, we leverage the multi-modalities, including ASR, OCR and text metadata, for better retrieval. Specifically, OCR is found to provide important description of video contents, when ASR and text metadata can not. Among all modalities, the video metadata presents unique characteristics of standardized structure with proper scheme and syntax. By leveraging this structure information, we can describe the video content in different granularities, and thus enabling more effective retrieval.

On the other end of the spectrum, our emphasis in interactive KIS is two-fold. First, an intuitive and user-friendly user interface is developed to facilitate the browsing of returned videos. As the query is usually specific, we postulate that searchers can quickly reject most of negative videos after seeing a few keyframes of the video. This premise of "fast rejection" motivates us to leverage the storyboard to pre-visualize a video. When users can not reject a returned video as negative, he/she may indicate it as a relevant one. By collecting a number of relevant videos, the searchers can perform relevance feedback to refine the retrieval and continue the search.

## 2. RELATED WORK

Video search has been one of the defined tasks under TRECVID benchmarking and evaluation for several years. Relying on textual information solely from metadata is able to give a reasonable result, but it is not good enough. Etter [1] further enhance the system by focusing on query expansion, using external knowledge bases such as Wikipedia titles and images. To further improve the performance, most researches approach video search by incorporating concept-based retrieval and ASR text on top of information from metadata. Such approach can be found in [2], which matches concepts model with queries before using content-based retrieval. Elleuch et al. [3] implemented three sub-systems for automatic search, consisting of text extraction from video, visual feature detection and lastly audio feature detection. Face detection and global features along with color layout and texture features are used in [4] and [5]. In addition to HLFs/concepts, Zhao et al. and Zha [6][7] uses visual-example based retrieval with SVM, followed by weighting the combinations as multimodal fusion. Other works based on multimodal fusion can be found in [8] and [9]. A concept detector in [10] [11] is trained for each query using SVM and KNN, before selecting the concepts using visual features and text descriptions. Cao et al. [12] and Zheng et al [13] make use of multiple modality feedback strategies, including the visual-based feedback, concept-based feedback and community-based feedback. By re-ranking the retrieval results using face detector after making use of learning-based retrieval and weighted fusion for selecting relevant concepts, Peng et al. in [14] was able to attain excellent performance in both automatic and manual search.

## 3. AUTOMATIC KIS SYSTEM (uKISS)

In automatic search, we follow the trend of harnessing the collateral text and content-based related information cues such as high level visual features (HLF), audio (music, speech, silence), language, automatic speech transcripts (ASR), and OCR in the video corpus. While there is a significant departure in the nature of the video content (user submitted web videos are more diverse and noisier) from previous years (professionally edited news/documentary video have more structures and predictability), we nevertheless expect that the approach to exploit the complementary text and related content-based cues in the video corpus, will continue to yield the best dividend in achieving good retrieval performance. Figure-1 shows our overall retrieval framework for automatic KIS.

### 3.1 Text modality

Notwithstanding the increased noise in the user-submitted descriptive text (in the accompanying XML metadata files), we see them as providing the most salient semantic cues for matching to a textual query.

From the 122 training queries provided by NIST, we observe that a simple keyword matching approach can achieve a baseline AP (Average Precision) retrieval of about 0.25. On closer inspection, we note that in some cases of low precision result, the retrieval is distracted by noise in both the query statement and the XML metadata text. Examples of such noise are: These noises come in the following form: (a) miss-spelling (e.g "hollwed") and joined words (e.g. "classwork"); (b) mismatch of word stem (e.g. "dance" and "dancer"); and (c) spurious retrieval of words that are unlikely to be present in the XML text (e.g. "…brown-hair, with glasses, and wearing a blue shirt…"). Hence our main strategy is to build on the textual keyword matching baseline by applying extensive pre-processing and parsing techniques to both the XML metadata files and the query statements.

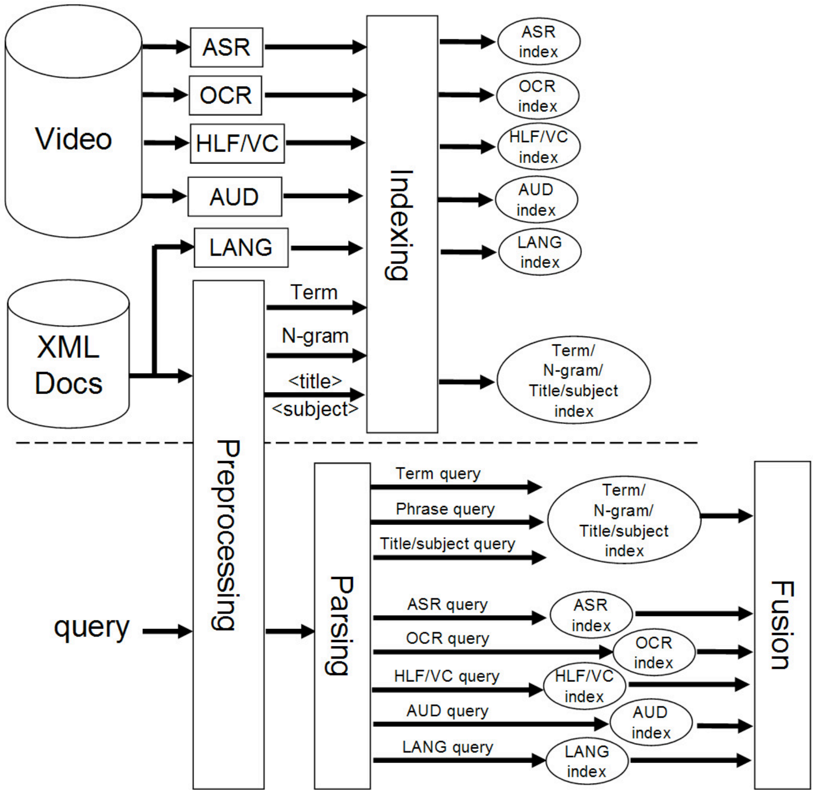


Figure-1 Overall retrieval framework for automatic KIS

*3.1.1 Dependency Parsing and Spell Correction*

The first preprocessing step is to attempt at correcting the miss-spelt words. We use the Aspell (version 0.60.3) tool [26] to first automatically identify words that are potentially miss-spelt. Then, by using the "aspell –a" option, we take the first three correction suggestions, and append onto the document (or query statement). In this way, we are effectively performing a kind of document expansion and query expansion. To prevent spell correction of named entities, we used a dependency parser [15] to generate the POS tags of the query statements, and skip spell correction for Proper-Pronoun (NNP or NNS). We do this only for the query statements, because they are grammatically well-formed. However, we cannot do this for the XML metadata documents because they are mostly not sentence-constructed.

As part of Spell Correction, we also perform an ad-hoc method of analyzing whether the constituent sub-strings of a word are also valid words themselves. For example, the word "talenthotel" and "crabwalk" are corrected by splitting into "talent hotel" and "crab walk" respectively. Using a dictionary of valid words, the splitting algorithm successively scans each character into a buffer, checking whether the sub-string buffer already contains a valid word in the dictionary. If it does, it flushes the sub-string buffer into a candidate word list, resets the buffer and moves to the next character. Note that by imposing the requirement that all constituent sub-strings must also be valid words, words like "flowersbloozz" will not be splitted, even though "flowers" is a valid word.

With all the above modifications, we have found that the retrieval AP score can have about 20% improvement over the first baseline (directly using the original query text as input to the retrieval engine).

*3.1.2 Custom stemming and inflectional normalization*

To cope with the morphological and inflectional endings of words, a term normalization process to reduce words to their linguistic root (such as the Porter Stemming algorithm [16]) is usually needed in an Information Retrieval system. However, the inflection algorithm in the Porter stemmer has some limitations: it does not perform well on irregular verbs (e.g. sing/sang/sung), noun inflection (e.g. child/children, woman/women), and derivation (e.g. economic/economy, move/mover, kind/kindness), etc. This issue is especially important for the XML metadata corpus, because each XML metadata file is short, with average length of about 50 words (taking only the <title>, <subject> and <description> fields). Furthermore, even though the queries are specified as long, most of the query words are spurious and likely not to be found in the XML documents. Hence, the number of relevant keywords in the query statement is very small (e.g. 2-3 relevant keywords).

It is therefore crucial that we have a robust method of resolving inflectional endings. We take a brute-force approach to do this. First, using the POS tags, we extract all the nouns and verbs in the query statements. Using a dictionary of common English inflectional words as in [17] we store each of the nouns/verbs and their inflectional forms in a look-up-table to be used during the indexing process (for both video documents (XML) and the queries).

Using inflectional normalization, we observe that the retrieval AP score can have a further 20% improvement over the spell-corrected system.

An example of a query that has been helped by inflectional normalization is query 72: "*Find the video in which a woman in a blue flowered dress talks about cooking for the elderly. She is in a kitchen and puts two dishes in the oven*." The query word "elderly" is normalized to the same XML text word "elder". Using the standard Porter Stemmer, the two words have different roots.

### *3.1.3 Salient Term/Phrase Selection*

Because the XML documents are short, there are only few relevant keywords to match to a given query. Our final preprocessing step attempts to extract these salient keywords, which can be standalone terms or N-gram phrases. The key is to determine which query terms/phrases are relevant and informative key concepts [25].

The motivation of this preprocessing can be seen using query 302: "*Find the video talking about George Bush and Patriot Act showing people walking on street with tape over their mouths*." In this query, the main objects of search are "George Bush", "Patriot Act" and "people with taped mouths". The phrases that describe the (visual) motion characteristics of the people are secondary. We now make an important conjecture: it is unlikely that the user would detail such motion description in the XML metadata file. Rather, we think that the typical user would likely describe the video using high level concept words such as "George Bush" and "Patriot Act" or "Silent protest", etc. In other words, the entire phrase describing the motion characteristics of the people can be effectively removed from the query statement, with little risk of hurting retrieval performance.

To illustrate how our conjecture can be automatically computed, we first show in figure-2 the Stanford POS tree of the two queries. In both queries, the main object of search can be easily located by the Proper Noun Penn Treebank tag NNP. Higher weights can be placed on these terms/phrases to request the retrieval engine to place a higher rank on documents matching to these terms/phrases. Furthermore, these terms/phrases are so important that they are likely to be placed under a <title> or <subject> header. So we can also create a dedicated index on these fields and issue these salient terms/phrases to them. As for the spoken phrases, they can be issued to an index created on the spoken transcript.

Each of our conjecture can be written as a decision rule on the POS-tagged query sentence. The end goal is to identify terms and N-gram phrases (N=2, N=3) for weighted/phrase retrieval. We create more conjecture rules by going over the 122 training queries, each time using the XML ground truth (provided by NIST) to return the matched words to facilitate rule building. Because the semantics provided by POS-tags is limited, this process is semi-automatically done. That is, we needed to refine some of the rules. For example, we need to manually encode words like "narrated", "saying", etc, as spoken cues for the phrases to be issued to the ASR index.

Using salient term/phrase extraction, we observe that the retrieval AP score can have a further 10% improvement over the inflectional normalization system.

```
Query 302:
(ROOT
 (S
  (VP (VB Find)
   (NP
    (NP (DT the) (NN video))
    (VP (VBG talking)
     (PP (IN about)
      (NP
       (NP (NNP George) (NNP Bush))
       (CC and)
       (NP (NNP Patriot) (NNP Act))))
     (S
      (VP (VBG showing)
       (NP
        (NP (NNS people))
        (VP (VBG walking)
         (PP (IN on)
          (NP (NN street)))
         (PP (IN with)
          (NP
           (NP (NN tape))
           (PP (IN over)
            (NP (PRP$ their) (NNS mouths))))))))))))
  (. .)))
```

### 3.2 Text related modality

To augment the XML metadata text base, we also incorporate Optical Character Recognition (OCR) and Automatic Speech Recognition (ASR) transcripts from two ASR engines: one from our own institute, and one from LIMSI [18]. Our OCR engine is based on gray-scale segmentation and neural network recognition [19]. Image frames are regularly sampled at 5 frames per second and fed into the OCR engine. To speed up the processing and reduce redundant output, each input image frame is also first checked in the RGB color space to see if it is sufficiently different from the previous frame.

Our ASR engine is based on the HTK speech recognition engine (version 3.4 release) from Cambridge University [27]. We do not have information on the LIMSI ASR engine. We only take the transcript output from [20]. We concatenate the output of both ASR and treating this as a single text source, an ASR index is then created.

Using ASR+OCR to augment the XML text, we observe that the retrieval AP score can have a further 15% improvement over the salient term/phrase detection system.

Here are two sample queries whereby the ASR has helped in retrieval. Query-208 is as follow: "*Find the video with a brown-haired young woman in a sleeveless brown shirt and blue jeans standing surrounded by a white background, telling people they need to go onlyine and register to vote*.". Note that the word "onlyine" is miss-spelt. Using aspell correction, the suggested correction is "online". Using this suggestion, the XML ground truth text has little matched keywords. However, the ASR has "online", "register" and "vote". This causes the correct video ID to be returned to the top-rank.

Another query-0017 is as follow: "*Find the video of President Bush standing near sea vessels with Coast Guard members talking about his pride of the Coast Guard, immigration, and security issues*." The XML ground truth text has matches "Bush". But the ASR further has "Coast Guard". This causes the correct video ID to be returned to the top-rank.

Here is one sample query whereby the OCR has helped in retrieval. Query-10 is as follow: "*Find the video of large cannons with camoflauge being fired in the mountains*". Note again that the "*camoflauge*" is miss-spelt. Despite the correction to "camouflage", the XML ground truth has no match to any of the query words. However, the OCR transcript detects the word "cannon" in the opening frames of the video. This causes an exact match to the "cannon" query word, causing the video ID to be returned as top-rank.

## 3.3 Non-text modality

### *3.3.1 High Level Visual Features (HLF)*

Many prior studies have found that the incorporating of high-level visual semantic features (HLF) have improved retrieval results [8, 11, 12]. We continue this trend by using the HLF results contributed by Columbia University [9]. In these results, 130 HLF detectors are defined and applied to the official shot key-frames provided by NIST. We adopt a bag-of-visual-HLF and multi-label approach, in that a shot key-frame can have multiple HLF labels (accepting all HLF detections above a threshold of 0.2), and the entire video is represented by the collective presence of HLF labels, disregarding from which shots are these HLF labels are detected.

Because not all 130 HLF detectors are equally accurate, we need to weight the detection reliability of the HLF output. We do this by manually inspecting the top-100 ranked key-frames of each of the 130 HLF detectors. We use the precision amongst these returned results as a measure of the detection accuracy of the HLF detectors. For example, the top 3 HLF detectors with the highest precision are "Outdoor" (0.82), "Face" (0.79) and "Sky" (0.74). We arbitrarily use these retrieval precision as the result fusion weight. Furthermore, using the ground truths from the 122 training queries, we collate the HLF detections from these ground truths. For each of these HLF concepts, we increment it by 1 if it is a concept that we think is semantically relevant. Otherwise, we decrement it by 1. By going over the 122 ground truths, this ad-hoc method provides us further with a way to weight the reliability of the HLF detectors.

For a given query, we use a heuristic rule-based approach to determine which HLF concepts are relevant. For example, if the word "ROAD" ("CAR") appears in the query, the "ROAD" ("CAR") HLF concept will be relevant. Each relevant HLF concept will be weighted according to their retrieval precision described in the last paragraph.

### *3.3.2 Audio*

We attempt basic audio classification into silence, music and others. A short time energy derived feature is used to separate the silence from the non-silence segments while a pitch presence derived feature [21], which measures the presence of stable harmonic pitch content, is used to separate the musical segments from the non-musical segments. The underlying generalization is that the musical segments, comprising vocals or instrument performances, display stable harmonic pitch content. The thresholds are experimentally set and heuristics are used in the classification into the three categories.

Each video is uniformly sampled into 8 segments of equal length. The audio detectors are then applied to each of the 8 segments. In each segment, the audio class with the highest detection confidence is accepted. Hence, there are 8 audio-class detections for every video, augmenting the video representation using a bag-of-audio-class.

For a given query, we again use a heuristic rule-based approach to determine which audio classes are relevant. For example, if the word "SING" appears in the query, the "MUSIC" audio class will be relevant.

### *3.3.2 Language*

Most XML text descriptions given are in English. But some are of types Spanish, French, etc. There are queries that can be classified as Language related queries, for example, query 0064, "Find the video of man with dark hair speaking Spanish and wearing dark shirt". We simply use keyword spotting and look for the phrase "… speaking <language>". Thus, we need language information to index each XML text. We use language classification tool in [28] to classify all XML description into the respective Language of type

English, Spanish, French, etc. The MAP score of 0.454 is achieved by the contribution of language information together with audio features.

### 3.4 Indexing and Retrieval Engine

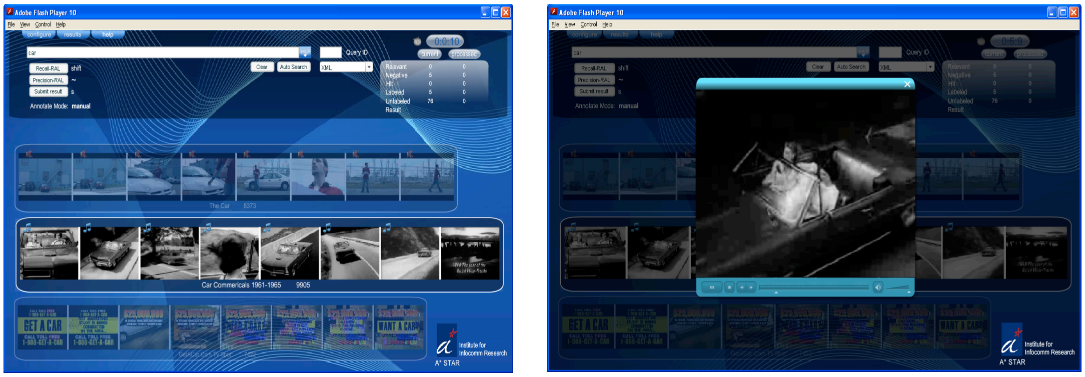
For text indexing and retrieval, we used the CLucene search engine [22]. Lucene is a free/open source informational retrieval library that uses the inverted index and tf-idf weighting scheme. The index stores mapping from each term in a content to the documents that contain it. Given a directory, the indexing process creates an instance of each document within the directory and populates it with Fields that consists of name and value pairs. An index writer is then used to add each document instance into the index which will represent the whole directory. More details on lucene indexing and searching process can be found in [23].

### 3.5 Experimental Results

Below table summarizes the retrieval performance by our uKISS for each modality combination. From the able, we can see that, the incremental of salience features to the system has gradually improved the system. And after we incorporate audio + language features, the system achieves the final MAP of 0.454.

| | Average Precision | |
|---|---|---|
| Cue index combination | 122 training queries | 300 test queries |
| XML (Baseline) | 0.25 | 0.23 |
| + POS-based Spell Correction | 0.26 | 0.27 |
| +Inflectional Normalization | 0.33 | 0.32 |
| +POS-based salient term/phrase | 0.37 | 0.35 |
| + ASR+OCR | 0.42 | 0.43 |
| + HLF | - | 0.443 |
| + Audio+LANG | - | 0.454 |

**Table 1: Present the results obtained from our system performed on the 122 and 300 queries respectively. Note that, the features HLF and Audio + LANG is not used on the 122 queries.**



(a) (b)

**Figure 3. The user interface of interactive KIS system (a), and (b) shows the video play function of the UI.**

### 4. INTERACTIVE KIS SYSTEM (iKISS)

An efficient and intuitive user interface is of paramount importance to facilitate the browsing of returned videos in the interactive know-item search. In general, the query is expected to be highly specific and informative, enabling searchers to quickly reject most negative returned videos. To pre-visualize the video content, we exploit the storyboard, namely a sequence of keyframes in chronicle order. Moreover, the UI supports video and audio play, as shown in Figure 3.

After issuing a query, the UI will display the returned list of videos, in which the storyboard of a video is presented at one row. By navigating down the returned list, the searchers can indicate the video as a “hit”, i.e. the target answer or a “relevant”, i.e., the relevant and uncertain videos. With a few “relevant” videos labeled, searcher can perform the relevance feedback to refine the returned list. In this process, the features from relevant videos are used to perform query expansion based on metadata, HLF, ASR, etc. Our interactive KIS system (iKISS) achieves MAP of 0.727 for expert user, and 0.682 for novice users respectively.

## 5. CONCLUSION

We employ text and content-based related information cues such as high-level visual features (HLF), audio (music, speech, silence), language, automatic speech recognition (ASR) transcripts, and OCR to index each of the test videos. Our Automatic KIS system takes a text-based query and returns a ranked list of possible answers. Query formulation is one of our focuses in automatic KIS. We refine the query by formulating query phrases and weighing query terms functioning as a single unit. And thus it yields more distinctive and specific retrieval. In addition of refining queries, we leverage the multi-modalities, including ASR, OCR and text metadata, for better retrieval. Among all modalities, the video metadata presents unique characteristics of standardized structure with proper scheme and syntax. By leveraging this structure information, we can describe the video content in different granularities, and thus enabling more effective retrieval. As for our interactive KIS, an intuitive and user-friendly user interface is developed to facilitate the browsing of returned videos. As the query is usually specific, we postulate that searchers can quickly reject most of negative videos after seeing a few keyframes of the video. This premise of “fast rejection” motivates us to leverage the storyboard to pre-visualize a video. When users decide not to reject a returned video as negative, he/she may indicate it as a relevant one. By collecting a number of relevant videos, the searchers can perform relevance feedback to refine the retrieval and continue the search.

## REFERENCES

[1] KB Video Retrieval at TRECVID 2009
David Etter -- KB video retreival

[2] PicSOM Experiments in TRECVID 2009
Mats Sjöberg, Ville Viitaniemi, Markus Koskela, Jorma Laaksonen -- Helsinki University of Technology, Finland

[3] REGIM at TRECVID2009: Semantic Access to Multimedia Data
Nizar Elleuch, Issam Feki, Anis Ben Ammar, Hichem Karray, Adel M. Alimi -- Research Group in Intelligent Machines, Tunisia

[4] Brno University of Technology at TRECVid 2009
Petr Chmelař, Vítzslav Beran, Adam Herout, Michal Hradiš, Ivo Řezníček, Pavel Zemčík -- Brno University of Technology, Czech Republic

[5] Kobe University at TRECVID 2009 Search Task
Kimiaki Shirahama, Chieri Sugihara, Yuta Matsuoka, Kana Matsumura, Kuniaki Uehara -- Kobe University

[6] BUPT-MCPRL at TRECVID 2009
Zhicheng Zhao, Yanyun Zhao, Zan Gao, Xiaoming Nan, Mei Mei, Hui Zhang, Heng Chen, Xu Peng, Yuanbo Chen, Junfang Guo, Anni Cai -- Beijing University of Posts and Telecommunications, China

[7] Zheng-Jun Zha, Linjun Yang, Tao Mei, Meng Wang, Zengfu Wang: Visual query suggestion. ACM Multimedia 2009:

[8] The MediaMill TRECVID 2009 Semantic Video Search Engine
C.G.M. Snoek, K.E.A. van de Sande, O. de Rooij, B. Huurnink, J.R.R. Uijlings, M. van Liempt, M. de Rijke, J.M. Geusebroek, Th. Gevers, M. Worring, A.W.M. Smeulders, D.C. Koelma -- University of Amsterdam, The Netherlands

M. Bugalho, I. Trancoso -- Lisboa, Portugal
F. Yan, M.A. Tahir, K. Mikolajczyk, J. Kittler -- University of Surrey, UK

[9] VIREO/DVMM at TRECVID 2009: High-Level Feature Extraction, Automatic Video Search, and Content-Based Copy Detection
Chong-Wah Ngo, Yu-Gang Jiang, Xiao-Yong Wei, Wanlei Zhao, Yang Liu, Shiai Zhu -- Video Retrieval Group (VIREO), City University of Hong Kong
Jun Wang, Shih-Fu Chang -- Digital Video and Multimedia Lab (DVMM), Columbia University

[10] National Institute of Informatics, Japan at TRECVID 2009
Duy-Dinh Le, Sebastien Poullot, Xiaomeng Wu, Michael Nett, Michael E. Houle, Shinichi Satoh -- National Institute of Informatics, Japan
Michel Crucianu -- CEDRIC CNAM, France

[11] Yantao Zheng, Shi-Yong Neo, Tat-Seng Chua, Qi Tian: Probabilistic optimized ranking for multimedia semantic concept detection via RVM. CIVR 2008: 161-168

[12] TRECVID 2009 of MCG-ICT-CAS
Juan Cao, Yong-Dong Zhang, Bai-Lan Feng, Lei Bao, Ling Pang, Jin-Tao Li, Ke Gao, Xiao Wu, Hon-Ttao Xie, Wei Zhang, Zhen-Dong Mao -- Institute of Computing Technology, Chinese Academy of Sciences, China

[13] Yantao Zheng, Shi-Yong Neo, Xiangyu Chen, Tat-Seng Chua: VisionGo: towards true interactivity. CIVR 2009

[14] PKU-ICST at TRECVID2009: High Level Feature Extraction and Search
Yuxin Peng, Zhiguo Yang, Lei Cao, Jian Yi, Ning Wan, Yuan Feng, Xiaohua Zhai, En Shi, Hao Li -- Institute of Computer Science and Technology, Peking University, China.

[15] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.

[16] "The Porter Stemming Algorithm",
http://snowball.tartarus.org/algorithms/porter/stemmer.html

[17] "Kevin's Word List Page", http://wordlist.sourceforge.net/

[18] "The Computer Sciences Laboratory for Mechanics and Engineering Sciences", http://www.limsi.fr/

[19] Robust identification code recognition system. United States Patent 6339651

[20] "TRECVID 10", http://www.audiosurf.org/trecVID10/

[21] H. Tan, Y. Zhu, L. Chaisorn, S. Rahardja, "Audio onset detection using energy-based and pitch-based processing" in Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp3689-3692.

[22] "CLucene-0.9.21b distribution", http://lucene.apache.org

[23] M. McCandless, E. Hatcher and O. Gospodnetić, "Lucene in Action", Manning Publication, 2010

[24] J. Park and W. Croft, Query term ranking based on dependency parsing of verbose queries, In Proc ACM SIGIR, pp 829-830, 2010

[25] M. Bendersky and W. B. Croft, Discovering key concepts in verbose queries. In Proc. ACM SIGIR, pp 491-498, 2008

[26] "GNU Aspell", http://aspell.net

[27] "HTK Speech Recognition Toolkit", http://htk.eng.cam.ac.uk/

[28] "Google AJAX Language API – Google Code", http://code.google.com/apis/ajaxlanguage/