PKU-ICST at TRECVID 2012: Instance Search Task

Yuxin Peng, Xiaohua Zhai, Jian Zhang,

Chang Yao, Tianjun Xiao, Nianzu Li, and Xiaodi Luo

Institute of Computer Science and Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We participate in all two types of instance search task in TRECVID 2012: automatic search and interactive search. This paper presents our approaches and results. In this task, we mainly focus on exploring the effective feature representation, feature matching, re-ranking algorithm and query expansion. In feature representation, we adopt two basic visual features and five keypoint-based BoW features, and combine them to represent effectively the frame image. In feature matching, multi-bag SVM is adopted since it can make full use of few query examples. Moreover, we conduct keypoint matching algorithm on the top ranked results. It is effective yet efficient since only top ranked results are concerned. In re-ranking stage, we observe that the top ranked videos always contain a few noisy videos. To eliminate such noise, we proposed a re-ranking algorithm based on semi-supervised learning to refine the top ranked results. In query expansion, we automatically crawl extra training images from Flickr according to the names of query instance. We achieve the good results in both tasks. Official evaluations show that our team is ranked 2nd on automatic search and 1st on interactive search.

1 Overview

In instance search task of TRECVID 2012, we participate in all two types: automatic search and interactive search. We submitted 4 runs for the instance search task of TRECVID 2012, including 3 runs for automatic search and 1 run for the interactive search. The evaluation results of our 4 runs are shown in Table 1.

Table 1: Results of our submitted 4 runs on Instance Search task of TRECVID 2012.

Type	ID	MAP	Brief description
Automatic	F_X_NO_PKU-ICST-MIPL_1	**0.220**	B+S+C+M+R+F
	F_X_NO_PKU-ICST-MIPL_3	0.189	B+S+C+M+R
	F_X_NO_PKU-ICST-MIPL_4	0.173	B+S+C+O+M+R
Interactive	I_X_NO_PKU-ICST-MIPL_2	**0.270**	S+H

In automatic search, our team is ranked 2nd in all 23 teams (our best run ranks the third among all 79 runs of 23 teams, and the first two runs belong to one team). In interactive search, our run is ranked 1st. Table 2 gives the explanation of brief description in Table 1. The framework of our system for instance search task of TRECVID 2012 is shown in Figure 1.

Table 2: Description of our methods.

Abbreviation	Description
B	**B**asic feature
S	**S**ift feature
C	**C**olor Sift feature
O	**O**pponent Sift feature
M	Keypoint **m**atching
R	**Re**-ranking based on semi-supervised learning
F	Query expansion with **F**lickr images
H	**H**uman feedback

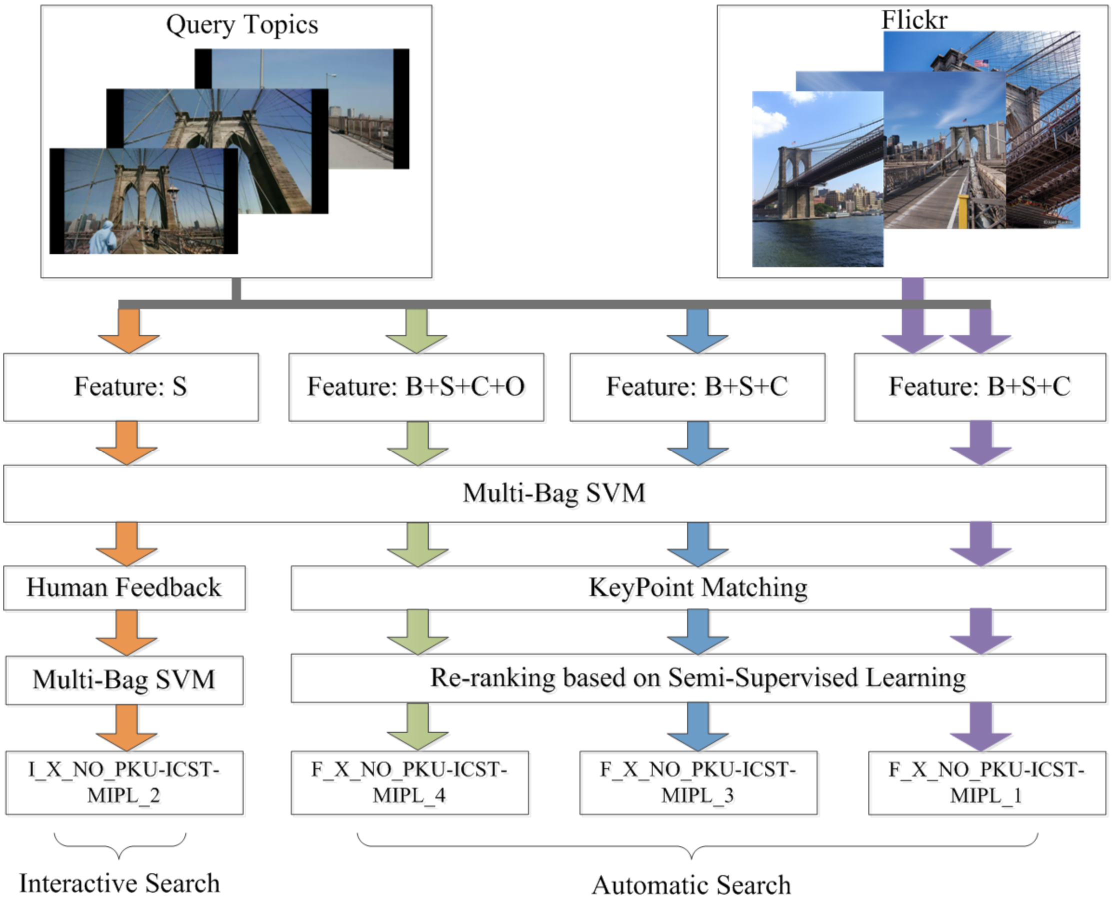


Figure 1: Framework of our instance search approach for the submitted four runs.

2 Feature Representation

We use two kinds of features for the instance search tasks, namely basic visual features and keypoint-based BoW features.

2.1 Basic visual features

We extract two basic visual features namely CMG(Color Moment Grid) and LBP(Local Binary Pattern) from each keyframe image. The details of these visual features are given as follows:

(1) **CMG** (756-d): the image is divided into sub-images by 1x1, 3x3, 5x5 and 7x7 grid in the CIE-Lab color space. The color moments of the 1st, 2nd and 3rd order are extracted from these sub-images in each channel.

(2) **LBP** (1475-d): it depicts the relationship of the center pixel and P equally spaced pixels on a circle of radius R in a gray-scale image. We first divide the gray-scale image into sub-image by a 5x5 grid, and then choose a neighborhood size of 8(P=8) equally spaced pixels on a circle of radius 1(R=1) that form a circularly symmetric neighbor set with "uniform" patterns.

2.2 Keypoint-based BoW features

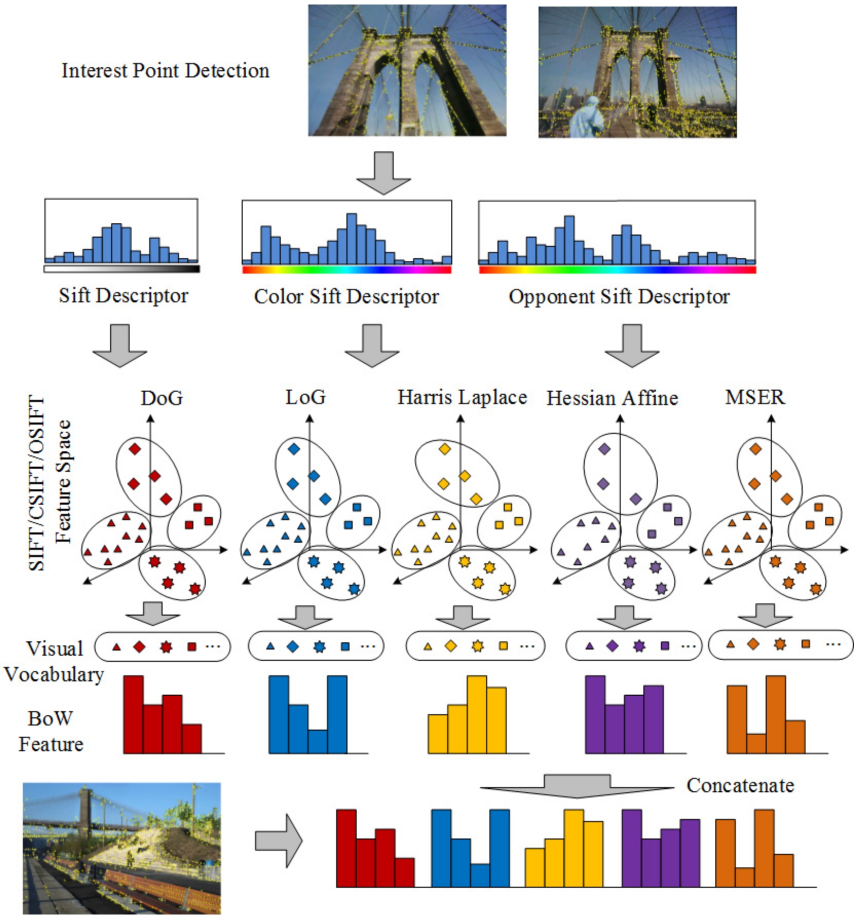


Figure 2: Combination of BoW features based on detectors and descriptors.

We explore the keypoint-based BoW(Bag-of-Word) features to represent each keyframe image. In our method, the extraction of keypoint-basd BoW features includes three steps:

(1) Detect keypoints using five detectors from the images, and use three descriptors to present the regions of those keypoints.

(2) Use k-means algorithm to cluster the keypoints into 1000 clusters, and form a visual vocabulary with the cluster centroids.

(3) Adopt soft-weighting[5] method to assign keypoints to multiple nearest visual words (centroids), where the word weights are determined by keypoint-to-word similarity. The normalized histogram of visual words forms a BoW feature vector.

In step (1), we adopt five complementary detectors to detect the keypoints from images: Difference of Gaussian (DoG) [1], Laplace of Gaussian(LoG)[1], Harris Laplace[2], Hessian Affine [3], and MSER [4]. For each detector, we use following three descriptors to generate three Bow features: 128-dimension SIFT descriptor[1], 192-dimension ColorSIFT descriptor [7], and 384-dimension OpponentSIFT descriptor [6]. As shown in Figure 2, for each combination of detector and descriptor, a 1000-dimension feature vector is generated separately. Different BoW features and basic features are concatenated to form the final feature in different runs as described in Table 1.

3 Feature Matching

In feature matching, multi-bag SVM is adopted since it can make full use of few query examples. Moreover, we conduct keypoint matching algorithm on the top ranked results. It is very effective yet efficient since only top ranked results are concerned.

The query examples are considered as positive samples. Due to the fact that only few shots are relevant with the topics in the test data set, we adopt the random sampling of test data as negative examples. A problem of learning-based method is that there are too few positive samples and too many negative samples. In our approach, we use MBSVM algorithm to handle this imbalanced problem, the algorithm details are presented in Figure 3 and diagram is shown in Figure 4.

(1) Over-sample the positive samples: Duplicate the positive sample set P for $(PCopy - 1)$ times and get a new set of positive samples P' with $PCopy \times PN$ samples, where *PN* is the number of positive samples in P before over-sampling.

(2) Under-sample the negative samples: Randomly select $NPR \times PCopy \times PN$ negative samples, and combine them with the over-sampled positive sample set P' to form a bag. That is to say, in each bag, the number of negative samples is *NPR* times as the number of positive samples, where *NPR(negative-to-positive-ratio)* is a parameter to control the degree of data imbalance in each bag. A model is trained by *LibSVM* for each a bag, where *RKF* kernel is used with default parameters.

(3) Repeat the above step (2) for *BagNum* times, where *BagNum* is a parameter specifying the number of bags. Then for each shot in the test data set, the *BagNum* prediction scores given by different models are averaged to form the final result. Notice that the negative samples in each bag are selected without repetition, that is, the negative samples are totally different in these bags. This ensures that we can make full use of the most of negative samples.

Figure 3: our algorithm for learning-based retrieval.

Totally, there are three important parameters in MBSVM algorithm: *PCopy*, *NPR* and *BagNum*. Experiments show that *PCopy=100*, *NPR=5* and *BagNum=5* can achieve good performance in both the accuracy and efficiency, while *PCopy* needs to be set according to the number of frames extracted from each video clip in the query examples.

We use keypoint matching method based on SIFT descriptor to further improve the performance. Since keypoint matching is time consuming, we only conduct keypoint matching algorithm on the 1000 top ranked videos, which is effective yet efficient.

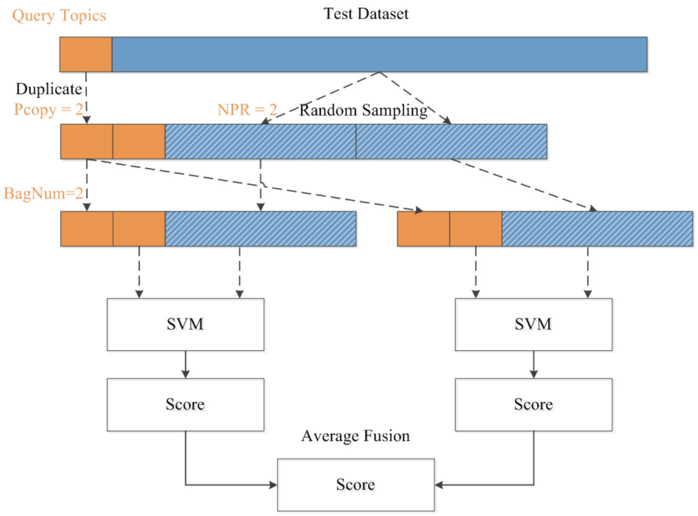


Figure 4: Diagram of MBSVM algorithm, where Pcopy=2, NPR=2 and BagNum=2.

4 Re-ranking

In re-ranking stage, we observe that the top ranked videos always contain a few noisy videos. Figure 5 shows an example of query "*Stonehenge*". Most of the top ranked videos are correct and they look similar to each other. To eliminate such noise, we proposed a re-ranking algorithm based on semi-supervised learning to refine the top ranked results, which can make full use of the data distribution information. The detail of our algorithm is described in Figure 6.

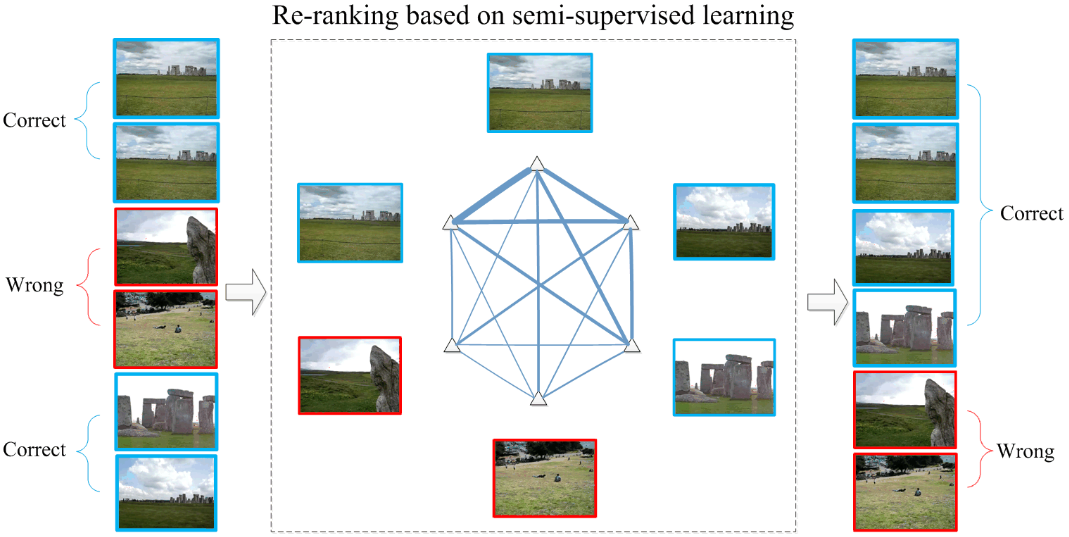


Figure 5: Results of query "*Stonehenge*". The top ranked videos always contain a few noisy videos. Most of the top ranked videos are correct and they look similar to each other. To eliminate such noise, we proposed a re-ranking algorithm based on semi-supervised learning to refine the top ranked results, which can make full use of the data distribution information.

(1) Given the data matrix of 1000 top ranked videos F and L, where F_i stands for the feature vector of a frame image and L_i stands for the video ID of vector F_i, $\mathrm{i} \in \{1,2,\dots,\mathrm{n}\}$ where n > 1000 means there are *n* frames from 1000 videos.

(2) Initialize the affinity matrix W with all zeros, and update as following:

$$\mathrm{W}_{\mathrm{i,j}} = \frac{F_i \cdot F_j}{|F_i| \cdot |F_j|}, \mathrm{i,j} \in \{1,2,\dots,\mathrm{n}\}, \mathrm{i} \neq \mathrm{j}. \tag{1}$$

(3) Generate the *k*-NN graph:

$$\mathrm{W}_{\mathrm{i,j}} = \begin{cases} W_{ij}, & F_i \in kNN(F_j); \\ 0, & otherwise. \end{cases} \tag{2}$$

kNN(F

kNN(F_j) stands for the set of k-nearest neighbors of F_j.

(4) Construct the matrix: $\mathrm{S} = D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with its (*i,i*)-element equal to the sum of the i-th row of W.

(5) Iterate $\mathrm{G}_{\mathrm{t+1}} = \alpha S G_t + (1-\alpha) Y$ until convergence, where G_t denotes the refined result in *t*-th round and we set $G_0 = Y$, α is a parameter in the range (0,1). Y is the initial score list of the frames of 1000 top ranked videos, we set the score of each frame the same as its original video.

Figure 6: re-ranking algorithm based on semi-supervised learning.

5 Interactive Search

In the interactive search, we only adopt SIFT descriptor and two kinds of keypoint detectors: Harris Laplace detector and Hessian Affine detector. Each frame is represented as a 2000-dimension BoW feature vector for efficiency. The detail of interactive search is described as following: Firstly, we retrieve the related 1000 videos by Multi-bag SVM as introduced in Figure 3. Then, we manually annotate about 25 positive or negative results for each topic. According to our observation, we found following three key factors: (1) Positive and negative samples are both helpful, and positive samples are more important than negative samples. (2) Positive samples ranked lower are helpful because they provide much information complementary to query examples. (3) Negative samples ranked higher are helpful because they look similar to positive samples and are easily mistaken.

With those new positive and negative samples, we adopted Multi-bag SVM again to re-train models. In this round, we only predict the 1000 top ranked results from last round for efficiency. Finally, we got the interactive search results and return to users.

6 Conclusion

By participating in the instance search task in TRECVID 2012, we have the following conclusions: (1) effective feature is vital, (2) learning-based similarity measure is a key factor, (3) re-ranking based on semi-supervised learning is helpful, (4) query expansion can improve the performance.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant 61073084, Beijing Natural Science Foundation of China under Grant 4122035, National Hi-Tech Research and Development Program (863 Program) of China under Grant 2012AA012503, National Development and Reform Commission High-tech Program of China under Grant [2010]3044, and National Key Technology Research and Development Program of China under Grant 2012BAH07B01.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision(IJCV)*, vol. 60, no.2, pp.91-110, 2004.

[2] C.G.M. Snoek, K.E.A. van de Sande, O. de Rooij, et al., "The MediaMill TRECVID 2008 Semantic Video Search Engine", *Proceedings of the 6th TRECVID Workshop*, 2008.

[3] K. Mikolajczyk, and C. Schmid, "Scale and affine invariant interest point detectors", *International Journal of Computer Vision(IJCV)*, vol. 60, no. 1, pp. 63-86, 2004.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions", *British Machine Vision Conference(BMVC)*, pp.384-393, 2002.

[5] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval", *ACM international conference on Image and video retrieval(CIVR),* pp.494-501, 2007.

[6] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 32, no.9, pp. 1582-1596, 2010.

[7] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 27, no.10, pp. 1615-1630, 2004.

PKU-ICST at TRECVID 2012: Known-item Search Task

Yuxin Peng, Yunbo Peng, Xiaohua Zhai,

Jian Zhang, Tianjun Xiao, Xin Huang, and Kang Cai

Institute of Computer Science and Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We participate in all two types of known-item search task of TRECVID 2012: automatic search and interactive search. This paper presents our approaches and results. We adopt three kinds of text information, which are XML documents, ASR and OCR. And we index and search the three kinds of pre-processed text individually with Lucene. In addition, the results are combined and re-ranked by two re-ranking approaches. We achieve the good performances, and official evaluation shows that our team is ranked 1st in both automatic search and interactive search.

1 Overview

In known-item search task of TRECVID 2012, we participate in all two types: automatic search and interactive search. We submitted 4 runs, including 3 runs for automatic search, and 1 run for interactive search. The evaluation results of our 4 runs are shown in Table 1. Our team is ranked 1st in both automatic search and interactive search.

Table 1: Results of our submitted 4 runs on KIS task of TRECVID 2012.

Type	ID	Mean Inverted Rank	Brief description
Automatic	F_A_YES_PKU-ICST-MIPL_2	**0.419**	F_A_YES_PKU-ICST-MIPL_3+ Query Expansion
	F_A_YES_PKU-ICST-MIPL_3	0.317	F_A_YES_PKU-ICST-MIPL_4+ Re-ranking
	F_A_YES_PKU-ICST-MIPL_4	0.313	XML Documents+ASR+OCR
Interactive	I_A_YES_PKU-ICST-MIPL_1	**0.792**	F_A_YES_PKU-ICST-MIPL_2

The 4 runs are described as follows:

- F_A_YES_PKU-ICST-MIPL_4: three kinds of text information are adopted, which are XML documents, ASR and OCR.
- F_A_YES_PKU-ICST-MIPL_3: re-rank the results of F_A_YES_PKU-ICST-MIPL_4 by two re-ranking approaches.
- F_A_YES_PKU-ICST-MIPL_2: adding query expansion to F_A_YES_PKU-ICST-MIPL _3.
- I_A_YES_PKU-ICST-MIPL_1: human feedback based on the results returned by F_A_YES_PKU-ICST-MIPL_2, which is the interactive search.

The framework of our KIS system of TRECVID 2012 is shown in Figure 1.

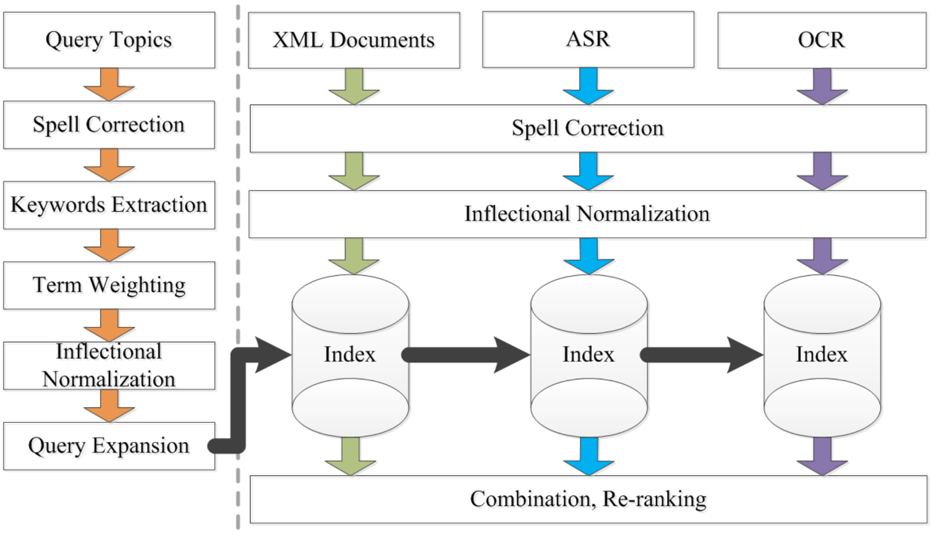


Figure 1: Framework of our KIS system.

2 Text-based Search

We adopt three kinds of text information, which are XML documents, ASR and OCR. In XML documents aspect, we process the XML documents and topics by using the following approaches: spell correction, POS-based keyword extraction, topics term weighting, inflectional normalization and query expansion. These approaches are described in detail as follows.

(1) **Spell Correction:** We find that many words in topics and XML documents are spelt incorrectly, which decrease the performance. We correct the miss-spelt words in topics and XML documents using Aspell [5]. We get the first correction suggestion and append it onto the topics and XML documents [1].

For example, the ground truth of topic 0901 in description region is "*a simple stop motion animation using paperl, mark making tools and saw dust*". The documents will return correctly if "*paper*" is appended according to "*paperl*". Some missing spaces between words are also corrected, for example, topic 1130 "*Find the video showing bursts of white light on mostly black backgrounds which then shows those bursts somewhat clearer showing faint pictures of people and the title "Tracegarden #11" at the end.*". "*Tracegarden*" is split into "*trace garden*".

(2) **POS-based Keywords Extraction:** To extract the keywords, we use Stanford Parser [2] to get

the POS tag and train the weights of different tags by the data of last year. Since the major sentences of XML documents are incomplete, we skip this processing step for XML documents and do this only for topics.

For example, topic 0947 "*Find the video that shows people sitting on a stage in a panel, a large movie screen and a woman wearing a brown and white sweater at a podium with a sign saying Wizards in blue.*" is tagged as below: "*Find/VB the/DT video/NN that/WDT shows/VBZ people/NNS sitting/VBG on/IN a/DT stage/NN in/IN a/DT panel/NN ,/, a/DT large/JJ movie/NN screen/NN and/CC a/DT woman/NN wearing/VBG a/DT brown/JJ and/CC white/JJ sweater/NN at/IN a/DT podium/NN with/IN a/DT sign/NN saying/VBG Wizards/NNP in/IN blue/NN ./.*" We can easily find that the word "*Wizards*" is more important than other words like "*saying*", "*large*", "*panel*" etc.

(3) **Topics Term Weighting:** Since all the KIS topics in TRECVID 2012 are available, so all the topics are regarded as a document collection and the number of topics containing this term are also considered in TF-IDF scheme. The detail of our algorithm is described in Figure 2.

a) We use cosine distance to calculate the relevant score:

$$score(q,d) = \frac{\overrightarrow{V(q)} \cdot \overrightarrow{V(d)}}{\left|\overrightarrow{V(q)}\right| \left|\overrightarrow{V(d)}\right|} \quad (1)$$

where *q* is topic, *d* is XML document, *V(q)* is the weight vector of query *q* and *V(d)* is the weight vector of document *d*. The definition of $\overrightarrow{V(q)}$, $\overrightarrow{V(d)}$, $\left|\overrightarrow{V(q)}\right|$, $\left|\overrightarrow{V(d)}\right|$ are described in step b).

b) We use TF-IDF scheme to compute *V(q)* and *V(d)*. Different from the default formula given by Lucene, we also consider the number of topics containing this term. Thus:

$$\overrightarrow{V(q)} = \sum_{t\ in\ q} w(t,q) = \sum_{t\ in\ q} tf(t,q) \times idf(t,D) \times idf(t,Q) \quad (2)$$

$$\overrightarrow{V(d)} = \sum_{t\ in\ d} w(t,d) = \sum_{t\ in\ d} tf(t,d) \times idf(t,D) \quad (3)$$

$$\left|\overrightarrow{V(q)}\right| = \sqrt{\Big(\sum_{t\ in\ q} tf(t,q) \times idf(t,D) \times idf(t,Q)\Big)^2} \quad (4)$$

$$\left|\overrightarrow{V(d)}\right| = \sqrt{\sum_{t\ in\ d} w(t,d)^2} = \sqrt{\sum_{t\ in\ d} 1^2} = \sqrt{|d|} \quad (5)$$

where *D* is the set of all XML documents, *Q* is the set of all topics and we ignore the weight of each term for $\left|\overrightarrow{V(d)}\right|$.

c) Finally, we get the scoring formula:

$$score(q,d) = \frac{\sum_{t\ in\ q}\big(tf(t,d) \times tf(t,q) \times idf(t,D)^2 \times idf(t,Q)\big)}{\sqrt{\big(\sum_{t\ in\ q} tf(t,q) \times idf(t,D) \times idf(t,Q)\big)^2 \times |d|}} \quad (6)$$

Figure 2: Topic term weighting algorithm.

(4) **Inflectional Normalization:** Due to few words exist in XML documents regions (<title>, <description>, <subject>, <keyword>, <keywords>), the information provided by XML documents is limited. To match more words, we get the inflectional normalization of words in topics and XML documents with a well-formed dictionary from [6]. We store the possible transformation of words in topics into a look-up table, and obtain the original form of words

in both topics and XML documents.

(5) **Query Expansion:** The ontology is constructed to expand the query topics.

In ASR aspect, we use the donated ASR data [3]. The processing steps of ASR data are similar to XML documents, including spell correction, POS-based keyword extraction, topics term weighting and inflectional normalization. In OCR aspect, we automatically detect and recognize the text in videos. Then, we do the same processing steps as ASR. Finally, these three kinds of text information are indexed with Lucene individually, and then we parse the topics and get the retrieval results.

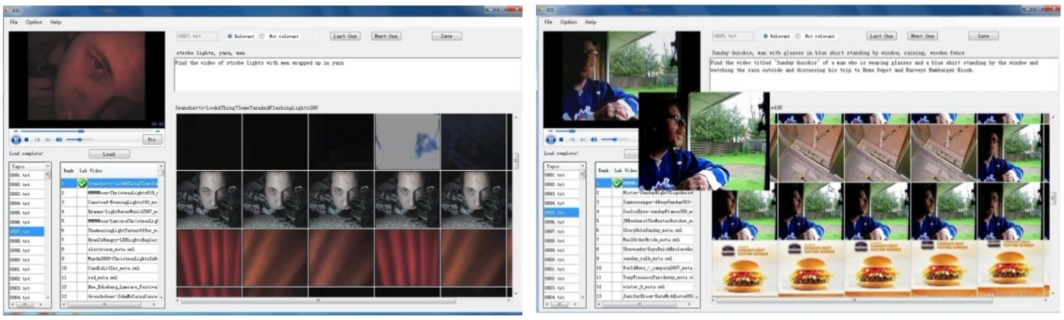
3 Re-ranking

To further improve the performance, we also apply two re-ranking approaches in our system as follows:

(1) **Black/White Video Detection:** Some topics aim at finding a black/white video, such as topic 1181 "*Find the World WarII-era black and white video of the B-60 airplane taking off.*" Thus a black/white video detector [4] is developed to detect whether a video is color or black/white. Firstly, we judge whether a keyframe is black/white. Secondly, if the percentage of black/white keyframes of a video is greater than a threshold, it is regarded to be black/white. Finally, if a topic aims to find a black/white video, all the black/white videos are ranked before the color videos.

(2) **Video Language Detection:** Some topics aim at finding a video in specific language, such as topic 1015 "*Find the video of a seated man with a beard talking in Spanish about 'hipnosis'. Framed documents hang on a green wall behind him. Classical music is playing. A rotating black and white spiral appears.*" Its target is to find a video with "*man with a beard talking in Spanish*". We detect the possible languages of the XML documents by Google Translate. If a topic aims to find a video in a specific language, all the videos with XML documents in this language are ranked before the videos with XML documents not in this language.

4 Interactive KIS System



strobe lights, yarn, men

Find the video of strobe lights with men wrapped up in yarn

Seanshatto-LookAThing?SomeYarnAndFlashingLights296

Sunday Quickie, man with glasses in blue shirt standing by window, raining, wooden fence

Find the video titled 'Sunday Quickie' of a man who is wearing glasses and a blue shirt standing by the window and watching the rain outside and discussing his trip to Home Depot and Harveys Hamburger Kiosk.

(a) (b)

Figure 3: (a) shows user interface of interactive KIS system, (b) popups an enlarged image when mouse moves onto each keyframe.

An efficient user interface is developed for the interactive KIS system, as shown in Figure 3. A storyboard shows part of the keyframes. The UI popups an enlarged image when mouse moves onto one keyframe, which assists the users to look into the details. Furthermore, if not so sure, the users can view the video and listen to the audio with an embedded windows media player conveniently. After deciding whether a video is correct or not, the users can click “Relevant” or “Not Relevant” to give their opinion.

The UI is simple and intuitive. After reading a topic, the users can reject an irrelevant video just by going through a few keyframes. For example, if the users want to get a video of an airplane, after taking a glance at the keyframes which are all about soccer, they know explicitly it is irrelevant to the topic. In fact, most of the videos are completely irrelevant to a given topic. In this way, a novice can reject an irrelevant video in a few seconds. In our interactive KIS system, 19 of the 24 topics are found, yielding a mean inverted rank of 0.792.

5 Conclusion

By participating in the KIS task in TRECVID 2012, we have the following conclusions: (1) the fusion among varieties of modals information is vital, (2) POS-based keywords extraction, topics term weighting, inflectional normalization and query expansion are key factors, (3) the re-ranking approaches can further improve the search performance.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant 61073084, Beijing Natural Science Foundation of China under Grant 4122035, National Hi-Tech Research and Development Program (863 Program) of China under Grant 2012AA012503, National Development and Reform Commission High-tech Program of China under Grant [2010]3044, and National Key Technology Research and Development Program of China under Grant 2012BAH07B01.

References

[1] Lekha Chaisorn, Kong-Wah Wan, Yan-Tao Zheng, Yongwei Zhu, Tian-Shiang Kok, Hui-Li Tan, Zixiang Fu, and Susanna Bolling, “TRECVID 2010 Known-item Search (KIS) Task by I2R”, *Proceedings of the 8th TRECVID Workshop*, 2010.

[2] Dan Klein and Christopher D. Manning, “Accurate Unlexicalized Parsing”, *Proceedings of the 41st Meeting of the Association for Computational Linguistics(ACL)*, pp. 423-430, 2003.

[3] J.L. Gauvain, L. Lamel, and G. Adda, “The LIMSI Broadcast News Transcription System”, *Speech Communication*, vol. 37, pp. 89-108, 2002.

[4] Xin Guo, Yuanbo Chen, Wei Liu, Yuanhui Mao, Han Zhang, Kang Zhou, Lingxi Wang, Yan Hua, Zhicheng Zhao, Yanyun Zhao, and Anni Cai, “BUPT-MCPRL at TRECVID 2010”, *Proceedings of the 9th TRECVID Workshop*, 2011.

[5] “GNU Aspell”, http://aspell.net/

[6] “Kevin’s Word List Page”, http://wordlist.sourceforge.net/