# **Glasgow University at TRECVID 2009**

P. Punitha, Hemant Misra, Reede Ren, David Hannah, Anuj Goyal, Robert Villa, Joemon M Jose {punitha, hemant, reede, anuj, hannahd, villar, jj}@dcs.gla.ac.uk Department of Computing Science, University of Glasgow, UK

# Abstract

In this paper we present our runs submitted to the automatic search tasks of TRECVID 2009. This year, we submitted six runs for the automatic search task. These search runs primarily focused on (1) adaptively ranking the relevance of low level visual features to a topic and fusing those results (2) Estimating topic distribution using the Latent Dirichlet Allocation (LDA) technique during the match in LDA topic space and (3) formation of bag of words for low level features. The automatic runs submitted include, two runs based on MPEG7 visual features, two runs based on visual and high level features and two runs based on bag-of-words derived from SIFT features.

# 1. Introduction

This year Glasgow University submitted six fully automatic runs. The automatic runs included two runs UG-PURun2\_2 and UG-RRRun4\_4, using low level MPEG7 visual features for retrieval. Two runs UG-PURun1\_1 and UG-PURun3\_3 used both visual and high level features. Two runs UG-HERun5\_5 and UG-HERun6\_6 are based on LDA training with bag of words generated from SIFT points. The following list briefs all submitted runs and the features used by them:

**UG-PURun1\_1** Search results using visual features (Color histogram, Edge histogram and Homogenous texture) on a reduced search domain combined with high level features with weighted late fusion.

**UG-PURun2\_2** Search results using MPEG7 visual features (Color histogram, Edge histogram and Homogenous texture), with adaptive feature weighting.

**UG-PURun3\_3** UG-PURun2\_2 combined with high level features.

UG-RRRun4\_4 Search using bag of words generated for each low level visual features.

UG-HERun5\_5 Search results using LDA based image retrieval approach using SIFT features.

UG-HERun6\_6 Search results using LDA based image retrieval approach.

All our runs were of type c, and no other data provided were used for training. All runs were trained on the TRECVid 2009 development set only. All runs used one keyframe per shot for processing.

The remainder of the paper is organised as follows. In section 2, we describe the features used. The details of the submitted runs are given in section 3. Section 4 discusses the results and the paper concludes in section 5.

# 2. Feature Descriptors

# 2.1 Visual Features

MPEG-7 standard features, namely, Edge histogram, Homogenous texture, Colour Structure and Colour Layout features were used as the low level features. In addition to these a simple colour histogram was used in various runs. For the LDA based approach we used the SIFT points for training and retrieval.

# 2.2 High Level Features

Out of the many high level feature extraction runs made available to TRECVID participants, we chose the submissions of the top five performing teams from 2008. In addition to this, we also went through the topic descriptions, and manually performed query expansion to formulate and select the different categories of high level features which are synonyms of the topics description. A frequency of occurrence of the shots in these selected categories was computed to rank the shot as highly relevant for the topic.

#### 3. Search Methodologies

In this section we explain the approaches used in the various runs we submitted.

#### 3.1 Automatic Runs

This section explains briefly the methods employed for the various automatic runs submitted.

#### 3.1.1 Visual Features Based runs (UG-PURun2\_2)

The proposed method comprises of three stages. The first stage deals with the feature selection mechanism which selects the feature that preserves the diversity of visual features of the query examples. A mechanism to push the relevant yet diverse results towards the top of the result list frames the second stage. Finally, the third stage combines the results originating from various query examples and various features.

#### Feature Selection [15]

Let  $V^f$  represent the feature matrix of Q in feature space  $f \in F$ . Given  $V^f$  for all  $f \in F$ , we compute the following matrices and values. A distance matrix  $D^f_{M \times M}$ , is computed using the distance measure for  $f \in F$  using the best distance measure reported in the literature for that feature [1, 2].

A correlation matrix  $C_{M \times M}^{f}$  is computed using equation (1),

$$C_{ij}^{f} = \frac{d_f^k(\sum AB) - (\sum A)(\sum B)}{\sqrt{d_f^k(\sum A^2) - (\sum A)^2} \sqrt{d_f^k(\sum B^2) - (\sum B)^2}}$$
(1)

Here, A and B are the feature vectors of queries  $q_i$  and  $q_j$  and  $d_f^k$  is the dimension of feature  $f_k$ . From the above matrices,  $C^f$ ,  $D^f$  and  $V^f$ , the statistical values,  $S_C^f$ ,  $S_D^f$ , and  $S_V^f$  that is, the standard deviation of the respective matrices are calculated. In addition to these,  $P_C^f$ ,  $P_D^f$ ,  $P_V^f$ , the p-value of the t-statistic of the correlation matrices  $C^f$ ,  $D^f$ ,  $V^f$  respectively is also computed. We use the t test to determine whether the means and variances of the two data sets are distinct assuming that the data have a normal distribution using equation (2). The probability that a value of the t statistic is greater than or equal to that observed would happen by chance, if the two sets of data were drawn from the same population, is computed as the p-value by transforming the correlation to create a t statistic having M-2 degrees of freedom, where M is the number of samples i.e., the number of query examples. The confidence bounds are based on an

asymptotic normal distribution of  $0.5 * \log \left(\frac{1 + C_{AB}^f}{1 - C_{AB}^f}\right)$ , with an approximate variance equal to 1/(M-3).

$$t = \frac{A - \bar{B}}{\sigma_{AB} / \sqrt{2/M}} \tag{2}$$

where,  $\sigma_{AB} = \sqrt{\frac{\sigma_A^2 + \sigma_B^2}{2}}$  and  $\bar{A}$  and  $\bar{B}$  is the mean of A and B.

Since the basic idea behind using many query examples for retrieval is a mutual concession to retrieve more diverse results and support semantic retrieval, we argue that features having larger variance should be preferred for that topic. To be not biased to a single feature, we order the features based on the six values,  $P_C^f$ ,  $P_D^f$ ,  $P_V^f$ ,  $S_C^f$ ,  $S_D^f$ , and  $S_V^f$ . Higher values of  $S_D$ ,  $S_C$ ,  $P_C$ ,  $P_D$ ,  $P_V$  and lower values of  $S_V$  indicate that feature *f* is more useful in fetching more distinct and diverse results.

## **Cluster Based Re-ranking**

Clustering of images based on low-level features offers an easier option to introduce diversity into this set of search results. Clustering similar images together and then reordering the list by choosing one image at a time from each cluster, pushes a variety of images towards the top of the result list, thereby increasing the chances of better precision. The result list can also be re-ordered as and when a new cluster is updated. For this purpose, we propose a single linkage clustering based on Kruskal's graph theoretic greedy approach [3].

Kruskal's approach is a greedy approach and is often used to find minimum spanning trees. A minimum spanning tree of a weighted graph G(V, E, W) is a connected graph G'(V, E', W') with all n vertices of graph G, but with only a subset of n-1 edges which have minimal weighted edges. The problem of finding a minimum spanning tree is to find such a minimally connected tree of a graph. Each graph is represented by an adjacency matrix with each cell having the weights on the edges. Kruskal's algorithm works by selecting the minimum value in the weight adjacency matrix and hence selecting that minimum weighted edge and the vertices connected by this chosen edge. The process is repeated by choosing the next minimum entry in the matrix, such that the edge chosen connects to at least one new vertex that is not already chosen, until all vertices in the original graph are chosen.

On similar lines, if the images in the result list are regarded as the vertices, our interest is in re-arranging these vertices depending on their distances with every other image in the result list. If the result list consists of certain number of clusters which are visually distinct, then the distance between any two images belonging to the same cluster will possess smaller distance values when compared to the distance between images belonging to different clusters. However, this demands the design of a clustering mechanism that can cluster images in the result list as expected. This entails a prior knowledge about how many clusters are to be formed or must be computed adaptively which is a separate research topic. We do not prefix the number of clusters; instead, using Kruskal's approach, we start from a pair of images and stop when all images in the search result list are already entered into the re-ranked list (or a prefixed number of clusters is formed). For our purpose, an affinity matrix (which is also referred to as a weight adjacency matrix) is computed for the documents in the result list, using the appropriate distance measure for each feature. Given A, the affinity matrix of a graph representing the images and their distances G(V, E), where V refers to the set of images in the search result list and E represents the set of weighted edges (distance values). A set of clusters  $C = \{c_1, c_2, c_3, \dots c_c\}$  could be created such that  $\bigcap_{i,j=1 \mid i \neq j}^c (c_i, c_j) = \emptyset$  and  $\bigcup_{i=1}^c c_i = V$ . It could be argued that complete linkage would be better for clustering similar images. However, our goal here is not to cluster similar images but, to move one instance of each distinct cluster of images present in the top 1000 results towards the top of the list. Using a complete linkage clustering connects an image to an already existing cluster, which is not our aim. Instead, we prefer, to grow possible seed points in a sequential manner originating/belonging from/to different clusters of images present in the result. In the worst case, there could be a single cluster which grows sequentially.

#### **Automatic Result Fusion**

We compute a vote for each image in the result list, images retrieved for all features are highly ranked, and then the images retrieved for two higher weighted features are ranked next highest and so on. If there is no common image in the result lists due to different features, the final result is obtained by combining the unique results from different features in a round robin fashion, by following the highest to lowest feature weights.

#### **Indexing Structure**

In this section, we will briefly describe the indexing structure we used, which combined with the classification method described above.

First, a set of low-level visual features (color histogram, edge histogram, and homogenous texture) of each image in the collection are extracted, then the collection is composed of the id of each image or key-frame and a list of multidimensional vectors representing the description of each visual feature.

To construct the indexing structure, first, we use a random projection clustering algorithm that is adapted to a high dimensional data space. This clustering algorithm combines the ideas from random projection techniques and density-based clustering in the following steps. Firstly, we project data on D random orthogonal lines. Then, we run a density clustering on each projection. A selection of the ``best" projections is done based on the information given by the projection based number and density of clusters. Finally, the ``best" projections are combined to find regions that might contain clusters. The final clustering is obtained

by selecting the detected regions that have a density higher than a threshold. More details about this approach can be founded in [4].

For every cluster, the minimum bounding rectangle is kept in memory and used to normalize the data space. Then, for each point p of every cluster, we find the pyramid  $P_y$  that contains the point with a simple computation of distance between the point and every centre of the pyramid bases. The next step determines the trunk T with respect to the height of the point from the top of the pyramid containing this point. T and  $P_y$  are concatenated to create an (integer) index  $I_p$  for each point of the data space. A recursive call on all trunks containing more than S points is made. S is a parameter fixed by the user or with respect to the data set size. The data dimensionality in the recursive algorithm is reduced to D-1. At this phase, we have obtained a principal index  $I_p$  for each point and multiple recursive indexes  $I_r$  for points that passed through a recursive call. Finally, those multiple indexes are put in a variant of a B+ tree structure, that is, a B+ tree that can contain more than one index for a point.

## 3.1.2 High level and Visual Features combined runs (UG-PURun1\_1 and UG-PURun3\_3)

The high level features provided by NIST were used in our run **UG-PURun3\_3**. The high level features from the top five teams (from last year – REGIMVID, PKU-ICST, IBM, NII.tv2009.HLF, Marburg) were selected for this purpose. From each team, the run of highest priority is extracted and then final high-level features are generated by voting from these five runs.

For each topic example, a result list containing 1000 shots are generated for each feature as explained in Section 3.1.1. The importance of each feature for a topic is calculated as explained in the previous section. Treating each example with equal importance and with the above computed preference for each feature, the shots retrieved for the visual feature and also present in the appropriate annotation list is ranked high in the final result list generated for **UG-PURun3\_3**. If any of the list was exhausted early and there were no more common shots in the lists, each list was traversed in a round robin fashion to pick the remaining shots to make the list consist of 1000 relevant shots.

On the other hand for **UG-PURun1\_1**, we manually listed the high level features supplied by NIST, and mapped the synonyms for each topic description. The high level feature list was not limited to the top performing group results. Now generating one list for a topic using results of different synonyms of the topic description does not give the provision of rating a shot on top by using the likelihood values. This is because the likelihood values are generated for different high level features with different training sets for each feature. Therefore we again followed a frequency count to rate a shot highly relevant to the topic. The topic examples were matched with this small domain to re-rank the list. If the list was less than 1000, the results generated from low level feature (Section 3.1.1) were used to make up the list of 1000 relevant shots.

#### 3.1.3 Bag of Words based runs (UG-RRRun4\_4, UG-HERun5\_5 and UG-HERun6\_6)

#### Feature Subspace selection based run

For **UG-RRRun4\_4** we try to exploit the approaches in statistical information retrieval for feature subspace selection. As Zhai [7] argues that relevance is closely associated with the distribution density of documents in a collection, normalizing feature distributions is an effective method to enhance the discrimination between documents and a query. We therefore propose the computation of feature terms by using the hypothesis of uniform term distribution [6]. The extraction of a feature term is a projection from a multiple valued N-dimensional variable to an integer, i.e. clustering which assigns class labels to data samples [5]. For example, the classification of a RGB colour into four classes can be denoted as  $[0,255]^3 \rightarrow \{0,1,2,3\}$ . This process can be further described as a projection to a boolean vector of integer appearance,  $[0,255]^3 \rightarrow \{0,1,2,3\} \rightarrow \{0,1\}^4$  for the appearance of class label 0, 1, 2 and 3. We symbolize such a projection as a function  $h: [0, K]^N \rightarrow \{0,1, ..., M-1\} \rightarrow \{0,1\}^M$  where K denotes the range of a feature and M the number of classes. We regard these integers as feature terms. In this game, we consider the one dimension case, where N=1. This is because dimensions in a low-level feature are usually regarded independent from each other.

For a collection D, the frequency of a feature term  $f_t$  is the times that a feature falls into a given value interval  $t \in [0, M)$ .

$$f_t = \|D_t\|, D_t = \{d|h(d) = t, d \in D\}$$

Where d is a document in D. The probability of a feature term t is  $p(t) = \frac{f_t}{\sum_0^{M-1} f_i}$ . The maximum entropy criterion is used to decide on the number of feature terms. In the experiments, we employ five low-level features, including colour histogram, colour layout, edge histogram, homogenous texture and colour structure. The BM25 model is then used to estimate the relevance [5]. This framework can be extended to accept high-level concepts and other term-like inputs such as SIFT features. We will try the extended framework in next year's TRECVID.

## LDA based retrieval runs

Both runs **UG-HERun5\_5 and UG-HERun6\_6** used LDA based retrieval. LDA [8, 10] has been successfully employed for various tasks in the past such as language model adaptation [9], detecting semantic coherence of a document [11] etc. It has also been applied for the image retrieval task in earlier TRECVid evaluations [12, 14]. LDA creates a generative and unsupervised topic model which builds upon the assumption that every document is represented by a topic distribution and every topic has an underlying word distribution.

To applying LDA to the task of image retrieval, we first need to describe each image as a "bag of words" representation. In order to do so, each image is represented in terms of some region based features such as SIFT. Once the SIFT features are obtained for each image, we cluster the features from all the images. In our experiments, to save on computational cost, we clustered only 2% of the relevant test data from TRECVid 2008 into 10,000 clusters. We employed the K-Means algorithm to cluster the data and obtained cluster centers. Then each image from TRECVid 2009 (test and query both) was represented in terms of these cluster centers. Thus we obtained a "bag of words" representation for each image in the collection.

We trained the LDA model on the test data using Gibbs sampling algorithm described in [10] and obtained the two parameters of the LDA model, topic distribution in each image ( $\theta$ ) and word distribution in each topic ( $\phi$ ). The usual approach followed in the literature is to compute the conditional probability of a query image, q, given the test images, and then find the top matching test images to the query image.

$$P(q|Image_i) = \prod_{\nu=1}^{V} P(q_{\nu}|Image_i) = \prod_{\nu=1}^{V} \left[\sum_{t=1}^{T} \theta_{ti} \phi_{\nu t}\right]^{q_1}$$

In the above equation, V is the number of cluster centers, 10,000 in our case, and T is the number of LDA topics, which we kept as 50 in this study.  $\theta_i$  is the topic distribution for test *Image<sub>i</sub>* learned during LDA training and  $q_v$  is the number of times cluster center v has occurred in the query. In this formulation, each cluster center in a query is independent of every other cluster in that query while computing the conditional probability. The semantic relation between the cluster centers in a query is not exploited in this case. A remedy is to project the query also in the LDA topic space and then compute the similarity between the query and the images in the LDA topic space using a measure such as KL-divergence.

In this work, we projected the queries into the LDA topic space before computing the similarity between the query and the test images in the LDA topic space. Projection of the queries into the LDA topic space can be done by the method proposed in [9, 11]. We have provided two variations of the proposed approach, in the first approach we selected the top matches by round-robin (UG-HERun6\_6) and in the second approach (UG-HERun5\_5) we used voting to find the best match.

## 4. Results

Table 1: Resultant performance of various UG runs							
Run ID	MAP	<b>P(10)</b>	R-prec	infAP	Recall		
UG-PURun1_1	0.0094	0.1542	0.0245	0.0094	0.0395		
UG-PURun2_2	0.0047	0.0958	0.0309	0.0045	0.0562		
UG-PURun3_3	0.0119	0.1333	0.0424	0.0118	0.0626		
UG-RRRun4_4	0.0003	0.0208	0.0063	0.0003	0.0129		
UG-HERun5_5	0.0122	0.0833	0.0347	0.0113	0.0806		
UG-HERun6_6	0.0132	0.0833	0.0358	0.0121	0.0785		



Figure 1 Comparison of MAP, P10, R-prec and infAP for runs submitted by UG

Table	2:	MAP	per	topic
-------	----	-----	-----	-------

Topic	UG-	UG-	UG-	UG-	UG-	UG-	Best of	Best in
-	PURun1_1	PURun2_2	PURun3_3	RRRun4_4	HERun5_5	HERun6_6	UG	TRECVID09
269	0.0062	0.0072	0.0006	0.0005	0.0000	0.0000	0.0072	0.191
270	0.0198	0.0125	0.0790	0.0000	0.0331	0.0102	0.079	0.355
271	0.0009	0.0019	0.0007	0.0001	0.0012	0.0014	0.0019	0.202
272	0.0000	0.0007	0.0041	0.0005	0.0001	0.0000	0.0041	0.133
273	0.0012	0.0005	0.0076	0.0001	0.0002	0.0001	0.0076	0.257
274	0.0007	0.0021	0.0004	0.0001	0.0004	0.0006	0.0021	0.085
275	0.0006	0.0005	0.0002	0.0003	0.0004	0.0006	0.0006	0.019
276	0.0007	0.0022	0.0021	0.0000	0.0009	0.0006	0.0022	0.579
277	0.0046	0.0025	0.0004	0.0010	0.0001	0.0005	0.0046	0.222
278	0.0200	0.0090	0.0208	0.0004	0.0151	0.0117	0.0208	0.294
279	0.0000	0.0001	0.0000	0.0000	0.0000	0.0002	0.0002	0.006
280	0.0001	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.043
281	0.0466	0.0016	0.0085	0.0006	0.0001	0.0003	0.0466	0.111
282	0.0003	0.0024	0.0017	0.0001	0.0001	0.0023	0.0024	0.061
283	0.0001	0.0018	0.0070	0.0001	0.0004	0.0005	0.007	0.074
284	0.0044	0.0016	0.0582	0.0001	0.0016	0.0045	0.0582	0.346
285	0.0002	0.0376	0.0242	0.0006	0.2228	0.2693	0.2693	0.488
286	0.0001	0.0006	0.0002	0.0004	0.0022	0.0011	0.0022	0.209
287	0.0093	0.0057	0.0029	0.0002	0.0043	0.0010	0.0093	0.27
288	0.0000	0.0013	0.0016	0.0001	0.0002	0.0002	0.0016	0.038
289	0.0035	0.0139	0.0022	0.0019	0.0065	0.0063	0.0139	0.145
290	0.1068	0.0019	0.0571	0.0000	0.0022	0.0053	0.1068	0.369
291	0.0001	0.0003	0.0003	0.0001	0.0005	0.0003	0.0005	0.023
292	0.0001	0.0062	0.0067	0.0000	0.0001	0.0000	0.0067	0.018

Due to the huge size of the collection, we used one frame per shot for our runs. Table 1 shows the results of various runs submitted by UG. The run UG-PURun1\_1 has the best performance in terms P10. The results generated in this run combined the low level features with the high level features. A smaller collection was comprised by pooling all shots annotated with the synonyms of the topic description. A search based on low level features was executed in this small collection to rank these results. If the number of shots in this collection was less than 1000, then the results from only the visual features were appended to the tail of the list. Since, the top of the list definitely consisted of the shots annotated with high level features, it resulted with better precision. UG-PURun3\_3, UG-HERun5\_5, UG-HERun6\_6 have almost the same MAP. This suggests that, the LDA based retrieval using SIFT feature performs equally or even better than the run based on combining low level features and the top results of high level features.



Figure 2 Comparison of MAP for each topic due to various UG runs

The MAP computed for each topic due to the different approaches presented in the paper is as listed in Table 2 and graphically represented in Figure 2. It can be seen from Figure 2, that the LDA based approach performs extremely well for the topic #285, which is 'Find shots of printed, typed, or handwritten text, filling more than half of the frame area'. The search result on the smaller domain of only high level features first, performed well for the topic #281' "Find shots of two more people, each singing and/or playing a musical instrument" and #290 "Find shots of one or more ships or boats, in the water". The run which combined the results from low level features and high level features UG-PURun3\_3 performed well for the topic #270 "Find shots of a crowd of people, outdoors, filling more than half of the frame area", #278 "Find shots of a building entrance", and #284 "Find shots of a street scene at night". The results show that, it is not easy to achieve good results with one system, one approach and one specific feature. The system, the features and the parameters needs to be adaptively changed for each topic. However, we are still in the process of analysing why our approach performed for some specific topics and failed for others.

The experimental performance of UG-RRRun4\_4 was not as expected. This seems to be, because we limit the scope to low-level global features and do not consider the modelling of hidden concepts. Note that the framework of bag-of-words can be easily extended to accept high-level concepts and other term-like inputs such as SIFT features. The performance will be improved by hierarchical document modelling and the use of hidden concepts. First, there are many approaches to present a visual document. Global features, e.g. colour and texture, describe an image from a rough scope and SIFT-based features focus on image details. Properly combining these features may improve the modelling of visual documents and thus improve retrieval performance. Second, bag-of-words representation allows the modelling of hidden concepts by statistical approaches such as LDA and pLSA. This will alleviate the scarcity problem of known high-level concepts. In summary, we tried the bag-of-words approach for the first time and wish to improve this approach in the coming TRECVid's.



Figure 3 Comparison of the best MAP of UG submissions and the best MAP of all submissions to TRECVID2009, fully automatic search task.

We also compared the best results of our runs for various topics with that of the best results for each topic from all the runs submitted to the automatic search task. The graphical representation of this comparison is shown in Figure 3. Further analysis can be explored only after the details of various methods are published.

## 5. Conclusions

The Glasgow University team submitted 6 fully automatic runs this year. Out of these six runs submitted, five runs except UG-RRRun4\_4 exhibit median results. For some topics the results were notably above the median. From observations, for our own submissions, the techniques performed differently for different topics. For instance, the LDA based approach performed extremely well for topic 285 amongst the other submissions UG-PURun1\_1 performed well for topics 290 and 281; and UG-PURun3\_3 gave better results for topic 284. We are now analysing the reasons behind why the techniques performed better for different topics and a possible mechanism of combining these techniques.

#### 6. Acknowledgements

The research leading to this paper was supported by European Commission under contracts and FP6-027122(SALERO).

#### References

- 1. Manjunath, B. S., Salembier, P., Sikora, T.: 2003, Introduction to MPEG-7 Multimedia Content Description Interface. Wiley Publications. (2003).
- Haiming Liu, Dawei Song, Stefan M. Rüger, Rui Hu, Victoria S. Uren.: Comparing Dissimilarity Measures for Content-Based Image Retrieval. AIRS 2008: 44-50
- 3. Deo N.: 2004, Graph theory with applications to engineering and computer science. Prentice Hall Publishers.
- Thierry Urruty, Chabane Djeraba, and Dan A. Simovici.: 2007, Clustering by random projections, in Advances in Data Mining. Theoretical Aspects and Applications. 7th Industrial Conference, ICDM 2007, Leipzig, Germany, July 14-18, Petra Perner, Ed. 2007, vol. 4597 of Lecture Notes in Computer Science, pp. 107–119, Springer.
- 5. Reede Ren and Joemon M. Jose.: 2009, Query Generation From Multiple Media Examples. CBMI 2009, pp138-143.
- 6. Gianni Amati and Cornelis Joost Van Rijsbergen.: 2002, Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems (TOIS), Vol. 20, No. 4, pp357-389.
- 7. Cheng Xiang Zhai and John Lafferty.: 2006, A risk minimization framework for information retrieval. Information Processing and Management, Vol. 42, No. 1, pp31-55.
- 8. David M. Blei, Andrew Y. Ng, and Michael I. Jordan.: 2002, Latent Dirichlet allocation, in Dietterich, Thomas G., Suzanna Becker, and Zoubin Ghahramani, editors, Advances in Neural Information Processing Systems (NIPS), volume 14, pages 601–608, Cambridge, MA. MIT Press.

- 9. Aaron Heidel, Hung-an Chang, and Lin-shan Lee.: 2007, Language model adaptation using latent Dirichlet allocation and an efficient topic inference algorithm, in Proceedings of eurospeech, Antwerp, Belgium.
- 10. Thomas L. Griffiths, and Mark Steyvers, 2004, Finding scientific topics, in Proceedings of the National Academy of Sciences, 101 (supl 1):5228–5235.
- 11. Hemant Misra, Olivier Cappe, and Francois Yvon.: 2008, Using LDA to detect semantically incoherent documents. In Proceedings of conll, Manchester, U.K.
- 12. James Philbin, Ondřej Chum, Josef Sivic, Vittorio Ferrari, Manuel Marin, Anna Bosch, Nicholas Apostolof, and Andrew Zisserman.:, 2007, Oxford trecvid 2007 -notebook paper. In Proceedings of the TRECVID 2007 Workshop.
- 13. Shan Jin, Hemant Misra, Thomas Sikora, and Joemon M. Jose.: 2009, Automatic topic detection strategy for information retrieval in spoken documents. In Proceedings of WIAMIS, London, U.K.
- Sheng Tang, Jin-Tao Li, Ming Li, Cheng Xie, Yi-Zhi Liu, Kun Tao, Shao-Xi Xu.: 2008, TRECVID 2008 High-level Feature Extraction by MCG-ICT-CAS. TREC Video Retrieval Evaluation Online Proceedings, TRECVID, 2008.
- 15. P. Punitha, Joemon Jose, Anuj Goyal, Topic Prerogative Feature Selection using Multiple Query Examples for Automatic Video Retrieval, 19-23 July, SIGIR 2009, Boston, Massachusetts, USA.