Clustering for text and image-based photo retrieval at CLEF 2009

Qian Zhu and Diana Inkpen
School of Information Technology and Engineering, University of Ottawa
qzhu012@uottawa.ca and diana@site.uottawa.ca

Abstract

For this year's Image CLEF Photo Retrieval task, we have prepared 5 submission runs to help us assess the effectiveness of 1) image content-based retrieval, and 2) text-based retrieval. We investigate whether the clustering of results can increase diversity by returning as many different clusters of images in the results as possible. Our image system uses the FIRE engine to extract image features such as color, texture, and shape from a database consisting of more than half a million images. The text-retrieval backend uses Lucene to extract texts from image annotations, title, and cluster tags. Our results reveal that among the three image features, color yields the highest retrieval precision, followed by shape, then texture. A combination of color extraction with text retrieval has the potential to increase precision, but only to a certain extent. Clustering also improves diversity in one of our clustering runs.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Managment**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Information retrieval, image retrieval, photographs, text retrieval, k-means clustering, SIFT, Lucene, FIRE

1 Introduction

The goal of this year's Image CLEF event is to promote the diversity of search results through presenting relevant images from as many clusters as possible. Such cluster may focus on the location where the image was taken, the subject matter, the event, the time, etc. Our database consists of an unprecedented 498,920 newspaper images, courtesy of the Belgium news agency, each containing a picture, a title, a short description of the image, and a time stamp. Handling a database of such size is already a feat on its own. Each of our 50 queries consists of up to 3 sample images (each having a description and a picture). We may use the text, the image, or both parts as query for the retrieval task. In addition, our queries are divided into two parts: part 1 (25 queries) provides the cluster titles for each query to help us cluster the results; part 2 does not provide any cluster hints. For more details about the task see [1].

The University of Ottawa team has developed a system for text-based retrieval, and image content-based retrieval. In the sections that follow, we describe each system, compare their retrieval effectiveness, and investigate whether or not clustering helps increase the diversity of results. We have used the k-means clustering algorithm. Then we describe two ways to incorporate clusters into the resulting ranking.

2 System Description

2.1 Text-based retrieval system

This system is running on the Lucene search engine that searches through a document collection based on the frequency of query terms in the document (tf-idf measure). In our system, the image annotations, titles, and tags are indexed. To further improve the search results, we undertook two additional steps: stemming and query expansion.

2.1.1 Stemming

It has been shown that stemming can slightly improve retrieval scores. Stemming means removing the suffixes of words, such as -ly, -ing, -ed, -s, etc, which sometime maybe overlooked by the system. As a pre-processing step prior to building the index, we converted all words into their stemmed form by running the Porter Stemming algorithm [2].

2.1.2 Query Expansion

In addition to using the image title as our query, we also expanded the query using the terms that appear in the description section of the 3 sample images. This was used in last year's competition, and has been shown to work well [3]. However, when this method is used, we should keep in mind that not all terms are introduced to the query equally. Otherwise, irrelevant documents may appear in our ranking due to expanded irrelevant query terms. We therefore give each term some weight, as determined by the frequency of the term in the description tag of 3 sample images. The words that appeared many times will have a higher weight in the expanded query. The LucQE library [4] provides a good implementation of the weighted query expansion done using the Rocchio's method. This method produces the modified query m:

$$\vec{q_m} = \alpha \vec{q_0} + \beta \frac{1}{|D|} \sum_{\vec{d_j} in D} \vec{d_j}$$

where:
$\vec{q_0}$ is the original query vector (i.e., the image title);
D is the set of known relevant documents (i.e., the description of sample images);
$\vec{d_j}$ is the frequency vector for a relevant document j in D.
We used the following parameters from Rocchio's method: $\alpha = 1.0$, $\beta = 0.75$.

2.2 Image content-based retrieval system

The wealth of image data provides us with an excellent opportunity to assess different image retrieval methods. The image database is the largest we have tested to date, and we shall see how our system performed under such a heavy load. Our system extracted 3 image features from each image: color, Tamura texture, and scale invariant feature transform (SIFT) [5].

Of particular interest is the SIFT feature, which is a feature related to shapes. This local image feature extracts particular interest points from the image which are highly distinctive, relatively stable to scale, and invariable to rotations and minor changes in illumination and noises. Images are first applied a Gaussian-blur filter at different levels, producing successively blurred images.

The differences between the blurred images are calculated based on the *Difference of Gaussians (DoG)* technique. And from the extremes of DoG, local interest points are derived. We have found a front-end SIFT extraction tool (called *extractsift*) from the FIRE image retrieval package [6]. This extraction uses Andrea Vedaldi's implementation of the SIFT algorithm [7].

Because feature extraction was a very lengthy process, some time-saving tricks were needed. In particular, the SIFT extraction takes 10 sec/image on an Athlon 64 3.0GHz dual-core system, which is simply too long. So we have reduced the size of all images by 50% to allow us to finish the tasks in reasonable time.

2.3 K-means clustering

To investigate the effect of clustering documents, we employed the k-means clustering algorithm on the documents retrieved from the query-expanded text retrieval system. The version of the algorithm we used can be found in [8]. Only the top 50 retrieved documents participate in clustering, because expanding this clustering range risks introducing irrelevant document to the top of the ranking. Additionally, the clustering is based on the 10 most frequent terms in each document, and the number of clusters (k) is chosen as 10, as well. This combination of settings have been shown to work best, because setting k too high may risk losing precision at the expense of cluster recall, while setting k too low improves precision at the expense of sacrificing cluster variety.

It is important to mention that clustered documents are re-inserted into the ranking in a way that increases the diversity of results. Two ways of doing this are proposed:

1) **Cluster-by-cluster**: Clusters are ranked in descending order by the average similarity score of documents in the cluster. Then, documents in the top scored cluster are all inserted to the ranking, followed by the next top score cluster, etc.

2) **Interleaved**: Again clusters are ranked in descending order. Differently from above, only one document from each cluster is inserted into the ranking at a time. When all clusters have contributed at least one document to the ranking, the method begins inserting the second document into the ranking. This is what we hope to be the better way.

3 Experimental Results

Table 1 lists the results of our 5 submitted runs on the 50 test queries. For each query, precision at depth R (where R=5, 10, 20, 30, 100), the mean average precision (MAP), and the cluster recall (CR) at different depths are reported.

Table 1. Results of the five submitted runs. Modality indicates whether the retrieval is based on image features or texts. P stands for precision. MAP stands for mean average precision. For each image run, only one feature was investigated. CR means cluster diversity. The run Color was based on full-size images which had a maximum of 512px in either width or height. The runs Sift and Tamura were based on 50% reduced size images. For the two text-based runs, both runs involved k-means clustering (k=10). Clusters_Interleaved used the interleaved way to insert clusters into the ranking, whereas Clusters_NonInterleaved used the other way, as described previously. Stemming and query expansion were also applied to the text runs.

Run Name	Modality	P@5	P@10	P@20	P@30	P@100	MAP
Color	Image	0.14	0.10	0.07	0.06	0.04	0
Sift	Image	0.13	0.09	0.07	0.06	0.04	0
Tamura	Image	0.13	0.09	0.07	0.06	0.04	0
Clusters_Interleaved	Text	0.41	0.50	0.57	0.60	0.63	0.27
Clusters_NonInterleaved	Text	**0.79**	**0.76**	**0.72**	**0.71**	**0.65**	**0.29**

Run Name	CR@5	CR@10	CR@20	CR@30	CR@100
Color	0.2279	0.2396	0.2499	0.2745	0.5417
Sift	0.1790	0.2177	0.2384	0.2730	0.3412
Tamura	0.1707	0.1894	0.2059	0.2792	0.3610
Clusters_Interleaved	0.4735	**0.6356**	**0.8074**	**0.8340**	0.8871
Clusters_NonInterleaved	**0.4785**	0.5518	0.6621	0.7414	**0.8937**

4 Discussion of Results

Due to the limitation of 5 submission runs per participant, we were not able to try out the combination of text and image retrieval systems. But we anticipate that the combination run will generate improved scores, because of the following evidences.

First, it is evident that content-based image retrieval alone cannot achieve good performance, because the precision values are simply too low. However, we notice that the precision at depth 5 is the highest for image retrieval, suggesting that perhaps these documents could be added to the text results to boost its performance.

Second, we often have the same text retrieval score for two or more retrieved documents, which makes ranking difficult. If the image retrieval score is combined with text retrieval using careful weighting, it becomes much easier to assign the ranking. Previous work shows that a weighting scheme of 85% text score + 15% image score raises the precision by about 0.03, comparing with text-only retrieval system [3].

Among the 3 image features tested, color is the best feature, followed closely by SIFT, and then by the Tamura texture. The striking similarities between the precision of SIFT and Tamura runs seem to reflect a baseline performance that might resemble random retrieval. Alternatively, their low precision scores might be explained by the reduced sizes of image, which might have eliminated too many details needed for extractions.

Lastly, clustering of documents can tip the ranking to favor either precision (all documents of a cluster are located at top positions) or diversity (interleaving insertion of cluster documents). This is clearly seen in the Clusters_Interleaved and Clusters_NonInterleaved runs.

References

[1] Paramita M, Sanderson M, Clough P: Diversity in photo retrieval: overview of the ImageCLEF Photo task 2009. *CLEF Working Notes* 2009, Corfu, Greece, 2009.

[2] Porter M.F.: An algorithm for suffix stripping. *Program* 1980, 14(3):130–137.

[3] Inkpen D, Stogaitis M, DeGuire F, Alzghool M: Clustering for Photo Retrieval at Image CLEF 2008. *CLEF Working Notes* 2008, Aarhus, Denmark, 2008.

[4] Rubens N: The application of fuzzy logic to the construction of the ranking function of information retrieval system. *Computer Modelling and New Technologies* 2006, 10:20–27.

[5] Lowe, D: Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision* 1999, 2:1150–1157.

[6] Deselaers T, Keysers D, Ney H: FIRE – Flexible Image Retrieval Engine. ImageCLEF 2004 Evaluation. *CLEF Workshop* 2004, 3491:688–698.

[7] Vedaldi A: An open implementation of the SIFT detector and descriptor. *UCLA CSD technical report* 2007.

[8] Sivaraman S: K-means cluster analysis algorithm implementation in Java, retrieved from http://www.codecodex.com/wiki/index.php?title=K-means_cluster_analysis_algorithm