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, diana@site.uottawa.ca

Abstract. For this year's Image CLEF Photo Retrieval task, we investigated the effectiveness of 1) image content-based retrieval, 2) text-based retrieval, and 3) integrated text and image retrieval. We investigated whether the clustering of results can increase diversity by returning as many different clusters of images in the results as possible. Our image system used the FIRE engine to extract image features such as color, texture, and shape from a data collection consisting of about half a million images. The text-retrieval backend used Lucene to extract texts from image annotations, title, and cluster tags. Our results revealed 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 increased precision, but only to a certain extent. Clustering also improved diversity, only in our text-based clustering runs.

1 Introduction

The goal of the ImageCLEF 2009 photo retrieval task is to promote the diversity of search results through presenting relevant images from as many clusters as possible. Such clusters may focus on the location where the image was taken, the subject matter, the event, the time, etc. Our data consists of an unprecedented 498,920 newspaper images, courtesy of the Belga news agency, each containing a picture, a title, a short description of the image, and a time stamp. This data presents many challenges, due to the enormous sizes of image database and the diverse nature of texts to process. Handling a data collection of such size is already a feat on its own. Each of the 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, the 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 see [1].

The University of Ottawa team has developed a system for text-based retrieval, a system for image content-based retrieval, and also an integrated text+image system. 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. We describe two ways to incorporate clusters into the resulting ranking.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part II, LNCS 6242, pp. 157–163, 2010.

2 System Description

2.1 Text-Based Retrieval

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. As a pre-processing step prior to building the index, we converted all words into their stemmed form by running the Porter Stemmer Toolkit [2]. To further improve the search results, we undertook an additional step: query expansion.

2.2 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]. 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 frequently 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*:

$$q_m = \alpha * q_0 + \beta * 1 * \Sigma_j d_j / |D|$$

where:
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);
d_j is the vector representation of a known relevant document in D.

We used the following parameters for Rocchio's method: $\alpha = 1.0$, $\beta = 0.75$, after some experimenting on the training data.

2.3 Image Content-Based Retrieval

The wealth of image data provides us with an excellent opportunity to assess different image retrieval methods. The image dataset 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 filters 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].

Feature extraction was a very lengthy process. In particular, the SIFT extraction takes 10 second per image on an Athlon 64 3.0GHz dual-core system, making the extraction process infeasible. Therefore, we have reduced the size of all images by 50% in the longest edge, while keeping the aspect ratio constant, to allow us to finish the extraction task in reasonable time.

2.4 Integrating Results

2.4.1 Integrating Results from Multiple Image Retrieval Methods

A look in the image-retrieval results indicates that the set of retrieved images varies between different methods. In order to increase the number and the diversity of retrieved images, we have merged the image results from multiple methods by using the following two techniques. Because different image retrieval methods may use different scoring schemes, we need some way to normalize the scores of each ranking, so that the two rankings are directly comparable to each other. We illustrate two techniques of normalization:

Technique 1: Normalize the ranking scores by dividing each score by the maximum score in each topic.

Technique 2: Normalize the ranking scores using variance normalization: (score - average) / variance.

Once the scores have been normalized for both rankings, we can integrate using the following method. For each image appearing in both rankings A and B, we compared its score in the two rankings, and select the maximum score to be the final score in the integrated ranking. If an image appeared in only one ranking, then the score was unchanged.

2.4.2 Integrating Image and Text Retrieval Results

The second step after the integration of image results is to merge them with the text retrieval results. We calculated the integrated “image + text” ranking based on a weighting scheme of 85% text score + 15% image score. This weighting scheme has been tested in the past as an effective scheme [3].

2.5 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, 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 expected to produce the most effective results.

3 Experimental Results

Table 1 lists the results of our runs on the 50 test queries (5 submitted runs, marked with s, plus 8 new runs). For each query, precision at depth R (where R = 5, 10, 20, 30, 100), the mean average precision (MAP), and the cluster diversity (CR) at different depths are reported.

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).

The run ColorSift_1 is an image only run that integrated the results of Color and SIFT, using the image-integration technique 1. The run ColorSift_2 used the image-integration technique 2

The runs TextQE and TextNoQE are text only runs, before clustering, with query expansion and without.

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 them and to all the text runs.

To integrate text and image runs, we used the integrated image ranking Sift plus Color and we further integrated them with text retrieval ranking. For this, we selected the top 5 hits from the image ranking, and combined their scores with their text scores.

The run ImgTextQE is the run that adds text retrieval results to the results of integrating Color and Sift by technique 1, including query expansion.

ImgTextQE_ClustersInterLeaved adds interleaved clustering to the previous run, while ImgTextQE_ClustersNonInterleaved is similar but the clustering method is non-inter-leaved.

Finally, the run ImgTextNoQE uses image and text, no query expansion, and no clustering.

Table 1. Our results. Modality indicates whether the retrieval is based on image features (I) or texts (T) or both (IT). P stands for precision. MAP stands for mean average precision. For each image run, only one feature was investigated. CR means cluster diversity. The submitted runs are marked with (s). The last two runs are the best runs in the competition, submitted by other groups.

Run Name	Modality	P@5	P@10	P@20	P@30	P@100	CR@5	CR@10	CR@20	CR@30	CR@100	Num rel. retr.	MAP	F_1
Color (s)	I	0.14	0.10	0.07	0.06	0.04	0.23	0.24	0.25	0.27	0.54	507	0	0.14
Sift (s)	I	0.13	0.09	0.07	0.06	0.04	0.18	0.22	0.24	0.27	0.34	318	0	0.13
Tamura (s)	I	0.13	0.09	0.07	0.06	0.04	0.17	0.19	0.21	0.28	0.36	400	0	0.12
ColorSift_1	I	0.15	0.10	0.07	0.06	0.03	0.17	0.17	0.19	0.21	0.31	331	0	0.13
ColorSift_2	I	0.12	0.09	0.07	0.07	0.04	0.14	0.16	0.19	0.22	0.30	612	0	0.12
TextQE	T	0.82	0.78	**0.74**	**0.72**	0.65	0.39	0.46	0.56	0.65	0.81	12915	0.29	0.58
TextNoQE	T	0.72	0.74	0.72	**0.72**	**0.66**	0.36	0.43	0.49	0.55	0.70	**15701**	**0.41**	0.55
ClustersInterLeaved (s)	T	0.41	0.50	0.57	0.60	0.63	0.47	**0.64**	**0.81**	**0.83**	0.89	12920	0.27	0.56
Clusters_NonInterLeaved (s)	T	0.79	0.76	0.72	0.71	0.65	**0.48**	0.55	0.66	0.74	**0.89**	12915	0.29	**0.64**
ImgTextQE	I T	**0.83**	**0.79**	**0.74**	**0.72**	0.65	0.43	0.50	0.59	0.67	0.82	12915	0.29	0.61
ImgTextQE Clusters InterLeaved	I T	0.43	0.51	0.56	0.60	0.63	0.33	0.49	0.66	0.69	0.81	12920	0.27	0.50
ImgTextQE ClustersNonInterLeaved	I T	0.80	0.77	**0.74**	**0.72**	0.65	0.40	0.48	0.57	0.63	0.82	12915	0.30	0.59
ImgText NoQE	I T	0.73	0.74	0.72	**0.72**	**0.66**	0.39	0.47	0.51	0.57	0.71	**15701**	**0.41**	0.57
InfoComm	T	0.85	0.85	0.83	0.83	0.80	0.59	0.67	0.69	0.72	0.86	14658	0.43	0.75
Xerox-SAS	I T	0.82	0.79	0.76	0.74	0.72	0.74	0.82	0.83	0.86	0.89	10635	0.29	0.81

3.1 Discussion of the Results

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, suggesting that perhaps the top 5 images have the potential to increase performances when they are combined with the text retrieval system.

We investigated the effect on retrieval score when the image retrieval score is combined with the text retrieval score. Our results show that for each text-only run, the corresponding image + text run always improves the precision scores, and respectively the F-measure, by 0.01 - 0.02. The improvements were the result of an 85% text + 15% image weighting scheme and when only the top 5 images were integrated. Altering the weighting scheme in favor of images or incorporating more images have resulted in lower precision scores. Our results confirm that images can in fact boost retrieval performances consistently in all experiments, even though improvements are small in many cases. Image ranking seems to only improve the position of relevant documents, but does not add more relevant documents.

Among the 3 image features tested, color is the best feature, followed closely by SIFT, and then by the Tamura texture. The low precision scores of SIFT and Tamura runs might be explained by the reduced sizes of image, which may have eliminated too many details needed for accurate retrieval.

In our experiments, we have integrated image rankings from multiple image retrieval methods. Here we evaluate the performance of the two image-integration techniques. As we can see from the runs ColorSift_1 (technique 1) and ColorSift_2 (technique 2), there are both pros and cons associated with each technique. Recall that the two techniques differ in the way we normalize the scores before integration. Where technique 1 normalizes by dividing each score by the maximum, technique 2 uses the variance normalization. Our results for technique 1 (ColorSift_1) show that precision is improved slightly, but it did not retrieve more relevant images than the individual non-integrated rankings. In technique 2 (ColorSift_2), the number of relevant images retrieved increased from 300 to 600 after the integration, but the precision scores @5, @10, @20 have not changed after integration, suggesting that most of additional relevant images retrieved are in the bottom of the ranking. The additional relevant images are quite attractive. In future we need to find a way to move relevant images to the top of the ranking.

Our results also compared non-clustered runs (TextQE) versus clustered runs (Clusters_Inter-, and non-interleaved). Clustered runs always have higher CR scores @5, @10, @20, @30 than non-clustered runs, confirming that k-means clustering under our current settings do in fact increase cluster variety quite a bit. There is, however, a trade-off between precision and cluster diversity, as increased CR scores are often accompanied by lower precision scores. This is most obvious from the comparison of TextQE and Clusters_Interleaved runs, where CR@10 goes up from 0.46 to 0.64, but P@10 goes down from 0.78 to 0.50. This is understandable, because not every document in each cluster is relevant, so when cluster documents are re-inserted to the ranking in interleaving fashion, there is a risk that non-relevant documents appear at the top positions of the ranking. It is worth noting that the run Clusters_NonInterleaved strikes a middle ground between precision and cluster diversity, producing the best overall performance (F score is the highest at 64%).

Also, we found out that the non-clustered run TextNoQE already has a good MAP and precision values (MAP of 0.40, and P@5-P@30 around 0.70). This has the best MAP score. The other non-clustered run TextQE is also good (MAP of 0.29, and P@5-P@30 around 0.80). This shows that text-retrieval itself can achieve very good MAP and precision, and clustering does not improve precision and MAP scores. If anything, clustering may even drop precision. In general the F-scores did not correlate with the MAP scores.

The two best runs in the competition (InfoComm and Xerox-SAS) can be used as gold standards against which we can judge the effectiveness of our system. We notice that even the best runs also exhibited a trade-off between precision and cluster variety. Although our results do not rank among the top in the overall F-measure, when judging precision and cluster diversity independently, our results are very competitive. Our TextQE run looks similar to InfoComm in the P@5 and P@10 scores, and our Clusters_Interleaved run is very close to Xerox-SAS in the CR@10, @20, @30, and @100 scores.

4 Conclusion and Future Work

In this paper, we experimented with several ways to integrated results from different image-based retrieval methods, from text and image retrieval, and two ways to increase cluster diversity. In future work, we can use more advanced image features and we can use probabilistic retrieval as the main retrieval framework.

References

1. Paramita, M., Sanderson, M., Clough, P.: Diversity in photo retrieval: Overview of the ImageCLEFPhoto task 2009. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part II. LNCS, vol. 6242, Springer, Heidelberg (2010)
2. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
3. Inkpen, D., Stogaitis, M., DeGuire, F., Alzghool, M.: Clustering for Photo Retrieval at ImageCLEF 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706. Springer, Heidelberg (2009)
4. Rubens, N.: The application of fuzzy logic to the construction of the ranking function of information retrieval system. Computer Modelling and New Technologies 10, 20–27 (2006)
5. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
6. Deselaers, T., Keysers, D., Ney, H.: FIRE: A Flexible Image Retrieval Engine. ImageCLEF 2004 Evaluation. In: Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 688–698. Springer, Heidelberg (2005)
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, `http://www.codecodex.com/wiki/index.php?` (retreived)