

Cross-modal retrieval by text and image feature biclustering*

András Benczúr István Bíró Mátyás Brendel
Károly Csalogány Bálint Daróczy Dávid Siklósi

Data Mining and Web search Research Group, Informatics Laboratory
Computer and Automation Research Institute of the Hungarian Academy of Sciences
{benczur, ibiro, mbrendel, cskaresz, daroczyb, sdavid}@ilab.sztaki.hu

Abstract

We describe our approach to the ImageCLEF-Photo 2007 task. The novelty of our method consists of biclustering image segments and annotation words. Given the query words, we may select the image segment clusters that have strongest cooccurrence with the corresponding word clusters. These image segment clusters act as the selected segments relevant to a query. We rank text hits by our own tf.idf based information retrieval system and image similarities by using a 20-dimensional vector describing the visual content of image segments. Here relevant image segments were selected by the biclustering procedure. Images were segmented by a home developed segmenter. We used neither query expansion nor relevance feedback; queries were generated automatically from the title and the 0.1 weighted description words.

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 Management]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Cross-modal retrieval, image annotation, biclustering, co-clustering

1 Introduction

In this paper we describe our approach to the ImageCLEF Photo 2007 evaluation campaign [7] over the IAPR TC-12 Benchmark [8] with reduced annotation text. The key feature of our solution is to combine text based image retrieval and content based image retrieval. Moreover, a biclustering algorithm to form interrelated clusters of image segments and annotation words.

Our CBIR method is based on segmentation of the image and on the comparison of features of the segments. The biclustering algorithm is used to filter relevant segments.

*This work was supported by a Yahoo! Faculty Research Grant and by grants *MOLINGV* NKFP-2/0024/2005, NKFP-2004 project Language Miner <http://nyelvbanyasz.sztaki.hu>.

We use the Hungarian Academy of Sciences search engine [2] as our information retrieval system. Biclustering is used to extend our method so, that segments are filtered according to the estimated relevance-relation to the query-text.

1.1 Related results

Many of the existing CBIR systems rely on so called blobs, regions, or segments, see [5, 11, 4, 10] for example. The specialty of our method is our special segmentation method and the combining of CBIR with biclustering and text based IR.

Biclustering is used in a wide variety of applications. For example documents-words clustering [9] or gene expression clustering [1].

2 The base text search engine

We use the Hungarian Academy of Sciences search engine [2] as our information retrieval system. Its ranking algorithm was developed originally for HTML pages, it uses a *tf.idf* based ranking with increased weights for query words within URL, anchor text, title and additional HTML elements.

We created small HTML-like documents from the textual information of the images. These contain the title, description and the location part of the annotations. The ranking algorithm uses different weights depending on which part of the document contains the query term. As in the original HTML search engine we removed the stop words from the documents. We used a wider range of stop words than usually because words like "photo" and "image" are frequently used in annotations but does not have distinctive meaning in this task.

The queries which were submitted to the search engine were constructed from the the title and narrative parts of topics. We removed the stop words described above first and sentences containing the phrase "not relevant" were also removed from the narrative. The resulting query is the concatenation of the remaining words of the title and the narrative parts. The words from the narrative part are marked, as the ranking algorithm uses different weights for the title and narrative query word hits.

The relevance score is computed by the weighted combination of the following:

- *tf.idf* based ranking which takes the proximity of query terms into account [12, 3] and uses document length normalization [14] and uses different weights depending on the location of the query terms inside the document (title, description or location).
- Number of different query terms in the document.

This ranking method had several parameters: weights of the *tf.idf* score and the number of query words, weights of the title, description and location hits and the weight of the title and narrative query terms. After several runs with different set-ups, we found that we get the best-looking result if the weight of the number of query terms is much higher than the *tf.idf* score. In other words we rank the documents with respect to the number of query terms, then use the *tf.idf* score to rank those documents that have the same number of query terms.

Since many topics have location reference, we get the best results if the weight of hits inside the location is much higher than the weights of title and the description. The query terms from the topic title have ten times higher weight than the narrative terms.

3 The content based IR system

The task of the CBIR part was to help annotation based retrieval with a content based method. Query images had to be classified into query topics. For each topics 3 sample images were given. For each query image we had to compute a distance, which measures how similar the query image is to the 3 sample images in the topic. Segment, region or blob based image similarity is a common method in content based image retrieval, see for example [5, 11, 4, 10]. Respectively, the basis

of our method is to find segments on the query image, which are similar to the segments in the sample-images. Image segmentation in itself is a widely researched and open problem itself. We used an image segmenter developed by our group to extract segments from the query images. Similarity of complex objects is usually measured feature-based. This means the the similarity of the objects is defined by the similarity in a certain feature-space.

$$dist(S_i, S_j) = d(F(S_i), F(S_j)) : S_i \in S(Q), S_j \in S(I) \quad (1)$$

where $S(q)$ is the set of segments of the query image Q , and $S(I)$ is the set of segments of the sample image I , $dist$ is the distance function of the segments, d is a distance function in the feature space (usually some of the conventional metrics in the n -dimensional real space), F is the function, which assigns features to objects and segments. We extracted features, like mean color, size, shape information, and histogram information. Our histograms had 5 bins in each channel. Altogether a 20-dimensional, real valued feature-vector was extracted for each of the segments. The same features were extracted for the segments of the sample-images. The features of the segments were written to a file. This way we obtained a data-base of the topic-samples, containing features of segments.

3.1 Segmentation

Most of the traditional digital image segmentation methods were based on the fact that we have no further information aside the position of pixels and their color. Fifteen or less regions are usual considered enough for processing with a Content Based Image Retrieval(CBIR) system properly.

In computer graphics we use many color representations. The most common color space is RGB. Because the equivalence of components in this three dimensional space we can use the Euclidean distance to measure the similarity of pixels P_1 and P_2 :

$$diffRGB(P_1, P_2) = \sqrt{(R_{P_2} - R_{P_1})^2 + (G_{P_2} - G_{P_1})^2 + (B_{P_2} - B_{P_1})^2} \quad (2)$$

We know that the human visual system (HVS) is more sensitive to the brightness of pixels than to the additional color. According to this we can use a different distance:

$$diffY(P_1, P_2) = 0.3 * | R_{P_2} - R_{P_1} | + 0.59 * | G_{P_2} - G_{P_1} | + 0.11 * | B_{P_2} - B_{P_1} | \quad (3)$$

The basic idea is that we transform the image into a graph. Let $G = (V, E)$ be an undirected graph where $\forall v_i \in V$ corresponds to a pixel in the image, and the edges in E connect certain pairs of neighboring pixels. The early graph-based representations contained only edges between neighboring pixels, recent works define a full graph with a more complex distance function. This graph-based representation of the image reduces the original proposition into a graph cutting challenge.

The adequate cut is the normalized cut (Ncut developed by Shi and Malik[13]). Unfortunately Ncut is one of the NP-hard problems so they could only approach the optimal cut. The subjective quality of the minimal cut found by the early graph-based CBIR systems is far weaker than that of the result of nearly normalized cut-based CBIR systems. However, old normalized cut-based methods are too slow for real-time or practical utilization. Felzenszwalb and Huttenlocher made a very efficient and linear algorithm that yields a result near to the optimal normalized cut[6]. Notably, they did not limit the number of segments inside an image.

With our restrictions, the graph-based algorithms are not the only option. One of the well-known methods is pyramid-based image segmentation. Using the Gaussian-Laplacian pyramid we can define a connection between pixels on neighboring levels and inside a level. Generally the pyramid-based segmentation methods use six or more levels with typically small thresholds. Pyramid-based algorithms have advantages over graph-based methods, such as computation time, but they offer subjectively worse quality - they suffer from the so-called blocking problem, and perform badly with complicated picture regions.

When we decided to develop our own segmentation method we wanted to achieve a relatively low running time, and to preserve a quality comparable to the graph-based algorithms used as reference. To improve the efficiency of our implementation we used some functions from the OpenCV library (www.intel.com/technology/computing/opencv/) for resizing, smoothing the image and to calculate the Sobel gradient. We resized all images to a fixed resolution to eliminate the downsides of different image resolutions. Gaussian-based smoothing helped us cut down high frequency noise. As pre-segmentation we built a three-level Gaussian-Laplacian pyramid to define initial pixel groups. The original pyramid-based method which uses brightness distance was modified to eliminate the blocking problem. After these, we had groups of 16 pixels maximum. To detect complex segments, we implemented a modified Felzenszwalb and Huttenlocher[6] graph-based method with an adaptive threshold system using Euclidean distance to prefer larger regions instead of small, noise-like parts of the image.

Algorithm *Segmentation* (I_{src}, τ_1, τ_2)

τ_1 and τ_2 are threshold functions. Let I_2 be the source image, I_1 and I_0 are the down-scaled images. Let $P(x, y, i)$ be the pixel $P(x, y)$ in the image on level i (I_i). Let $G = (V, E)$ be an undirected weighted graph where $\forall v_i \in V$ corresponds to a pixel $P(x, y)$. Each edge (v_i, v_j) has a non-negative weight $w(v_i, v_j)$.

Gaussian-Laplacian Pyramid

1. For every $P(x, y, 1)$ $Join(P(x, y, 1), P(x/2, y/2, 0))$ if $\tau_1 < diffY(P(x, y, 1), P(x/2, y/2, 0))$
2. For every $P(x, y, 2)$ $Join(P(x, y, 2), P(x/2, y/2, 1))$ if $\tau_1 < diffY(P(x, y, 2), P(x/2, y/2, 1))$

Graph-based Segmentation

1. Compute $Max_{weight}(R) = \max_{e \in MST(R, E)} w(e)$ for every coherent group of points R where $MST(R, E)$ is the minimal spanning tree
2. Compute $Co(R) = \tau_2(R) + Max_{weight}(R)$ as the measure of coherence between points in R
3. $Join(R_1, R_2)$ if $e \in E$ exists so $w(e) < \min(Co(R_1), Co(R_2))$ is true, where $R_1 \cap R_2 = \emptyset$ and $w(e)$ is the weight of the border edge e between R_1 and R_2
4. Repeat steps 1,2,3 for every neighboring group (R_1, R_2) until possible to join two groups

This algorithm sometimes does not find relevant parts with low initial thresholds. To find the relevant borders which would disappear with the graph-based method we calculated the gradient image to select important edges from remainders. Typically we detected a hundred or less segments in a regular image.

3.2 The distance function

We used euclidean distance to measure similarity of images. The distance of image Q to sample image I was computed as:

$$dist(Q, I) = \min_I \frac{1}{n} \sum_j \min_i dist(S_i, S_j) : S_i \in S(Q), S_j \in S(I), n = |S(I)| \quad (4)$$

where n is the number of segments in the sample image, S(Q) is the set of segments of image Q, S(I) is the set of segments of the sample-image S. So, we took all the segments in the sample image, and searched for the closest segment in the query image. After this, we summed up all the distances for the sample image. Doing this in the other way would also be possible: for each segment in the query image we could search closest segments in the sample images. We choose our direction, because for each topic the sample images and its segments are fixed, and this way, dist(Q) does not depend on the number of segments in the query image. To be sure, we are using a normalization by n, but found that this way the distances are better. From the 3 sample-images, we took the closest one.

4 Image segment – annotation word biclustering

Clustering is used to group together similar objects and has practical importance in a wide variety of applications such as web-log and market-basket data analysis. The data in this applications is arranged as a co-occurrence table. Most of the clustering algorithms clusters one dimension of the co-occurrence table based on the similarities along the second dimension. Biclustering is simultaneously clusters both dimensions of the table by exploiting the duality between rows and columns.

We used the following biclustering algorithm [9]. Let X and Y be discrete random variables that take values in the sets $\{segments\}$ and $\{words\ of\ annotations\}$ respectively. Let $p(X, Y)$ denote the joint probability distribution between X and Y . Let the k clusters of X be written as: $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k\}$, and let the l clusters of Y be written as: $\{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_l\}$. We are interested in finding maps C_X and C_Y ,

$$C_X: \{x_1, x_2, \dots, x_k\} \mapsto \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k\}, \quad C_Y: \{y_1, y_2, \dots, y_l\} \mapsto \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_l\}.$$

For brevity we write $\hat{X} = C_X(X)$ and $\hat{Y} = C_Y(Y)$ where \hat{X} and \hat{Y} are random variables that are a deterministic function of X and Y , respectively. Finally let $D(p \parallel q)$ denote the *Kullback-Leibler* divergence of probability distributions p and q .

Algorithm *BiClustering* $(p, k, l, C_X^{(0)}, C_Y^{(0)})$

Where $C_X^{(0)}$ and $C_Y^{(0)}$ are some initial partition functions.

1. Compute segment (row) clusters: For each segment x , find its new cluster index as

$$C_X^{(t+1)}(x) = \operatorname{argmin}_{\hat{x}} D \left(\frac{p(x, Y)}{p(x)} \parallel p(Y) \cdot \frac{p^{(t)}(\hat{x}, \hat{y})}{p^{(t)}(\hat{x}) \cdot p^{(t)}(\hat{y})} \right),$$

resolving ties arbitrarily. Let $C_Y^{(t+1)} = C_Y^{(t)}$.

2. Compute word (column) clusters: For each word y , find its new cluster index as

$$C_Y^{(t+2)}(y) = \operatorname{argmin}_{\hat{y}} D \left(\frac{p(X, y)}{p(y)} \parallel p(X) \cdot \frac{p^{(t+1)}(\hat{x}, \hat{y})}{p^{(t+1)}(\hat{x}) \cdot p^{(t+1)}(\hat{y})} \right),$$

resolving ties arbitrarily. Let $C_X^{(t+2)} = C_X^{(t+1)}$.

3. set $t = t + 2$ and go to step 1.

We modified the above algorithm at the first step by combining KL-divergence with euclidean-distance on the segment features as follows. Let $f_1(x), \dots, f_s(x)$ denote the features of a segment x . Let $c_i(\hat{x})$ be the average of $f_i(x)$ ($i = 1, \dots, s$) where $x \in \hat{x}$. With this notations the modified first step is

$$C_X^{(t+1)}(x) = \operatorname{argmin}_{\hat{x}} \left\{ D \left(\frac{p(x, Y)}{p(x)} \parallel p(Y) \cdot \frac{p^{(t)}(\hat{x}, \hat{y})}{p^{(t)}(\hat{x}) \cdot p^{(t)}(\hat{y})} \right) + \sqrt{\sum_{i=1}^s (f_i(x) - c_i^{(t)}(\hat{x}))^2} \right\}$$

5 Image distance with bicluster information

We extended the CBIR method by using the above described biclustering method. It was supposed that these biclusters indicate some semantical connection between the features and the text. This was used to refine the CBIR method. When the segments of the 3 sample-images were examined, we tried to filter them. All the segments were examined if there is a word in a corresponding textual bicluster, which occurs in the title of the topic. The title of the topic was a short description. If there is at least one word in common, then we assumed, that the segment is semantically connected to a word, which is relevant. Consequently, we included this segment in our computation. In the rare case, when all of the segments of a sample-image was excluded, we included them all again, since an empty set of segments would not do anything good. The distance measure was done the same way with the filtered segments.

6 Results

Table 1 shows the results of the text based, content based and the mixed method. The results were evaluated by the ImageCLEF organizers using the following measures: Mean Average Precision (MAP), Precision at 10/20/30 (P10, P20, P30), Binary Preference (BPREF), Geometric Mean Average Precision (GMAP).

Results using only the visual information are very poor. However, when combined with the text-based ranking it yields significant improvements in every measures.

	MAP	P10	P20	P30	BPREF	GMAP
text + visual	0.2132	0.3267	0.2733	0.2478	0.1962	0.0441
text only	0.1880	0.2833	0.2350	0.2033	0.1675	0.0386
visual only	0.0138	0.0467	0.0433	0.0367	0.0240	0.0019

Table 1: Performance of the three basic method evaluated by different measures

Conclusion and future work

We have demonstrated that the biclustering of image segments and annotation words can improve the performance of an IR system.

Our future work to be done includes to improve the content based IR behind the results, to test various settings of the segmentation and the biclustering algorithm. And finally, to use query expansion and feedback in order to demonstrate whether the method can improve performance over the state of the art.

References

- [1] Ron Shamir Amos Tanay, Roded Sharan. Biclustering algorithms: A survey. Oxford Univ Press, 2002.
- [2] András A. Benczúr, Károly Csalogány, Eszter Friedman, Dániel Fogaras, Tamás Sarlós, Máté Uher, and Eszter Windhager. Searching a small national domain—preliminary report. In *Proceedings of the 12th World Wide Web Conference (WWW)*, 2003.
- [3] Stefan Büttcher, Charles L. A. Clarke, and Brad Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 621–622, New York, NY, USA, 2006. ACM Press.
- [4] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1026–1038, 2002.
- [5] Yixin Chen and James Z. Wang. Image categorization by learning and reasoning with regions. *J. Mach. Learn. Res.*, 5:913–939, 2004.
- [6] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.
- [7] Michael Grubinger, Paul Clough, Allan Hanbury, and Henning Müller. Overview of the ImageCLEFphoto 2007 photographic retrieval task. In *Working Notes of the 2007 CLEF Workshop*, Budapest, Hungary, September 2007.
- [8] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselears. The IAPR TC-12 benchmark - a new evaluation resource for visual information systems. In *International Workshop OntoImage 2006: Language Resources for Content-Based Image Retrieval, held in conjunction with LREC'06*, pages 13–23, 2006.
- [9] Dharmendra S. Modha Inderjit S. Dhillon, Subramanyam Mallela. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, 2003.
- [10] Qin Lv, Moses Charikar, and Kai Li. Image similarity search with compact data structures. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 208–217, New York, NY, USA, 2004. ACM Press.

- [11] B. G. Prasad, K. K. Biswas, and S. K. Gupta. Region-based image retrieval using integrated color, shape, and location index. *Comput. Vis. Image Underst.*, 94(1-3):193–233, 2004.
- [12] Yves Rasolofo and Jacques Savoy. Term proximity scoring for keyword-based retrieval systems. In Fabrizio Sebastiani, editor, *ECIR*, volume 2633 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2003.
- [13] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern and Machine Intelligence*, 22:888–905, 2000.
- [14] Amit Singhal, Chris Buckley, Mandar Mitra, and Gerard Salton. Pivoted document length normalization. Technical report, Ithaca, NY, USA, 1995.