IPL at CLEF 2012 Medical Retrieval Task

Spyridon Stathopoulos, Nikolaos Sakiotis, Theodore Kalamboukis

Department of Informatics
Athens University of Economics and Business, Athens, Greece
spstathop@aueb.gr tzk@aueb.gr

Abstract. This article presents an experimental evaluation on using Latent Semantic Analysis (LSA) for searching very large image databases. It also describes IPL's participation to the image CLEF ad-hoc textual and visual retrieval for the medical task in 2012. We report on our approaches and methods and present the results of our extensive experiments applying data fusion on the results obtained from LSA on several low-level visual features.

Key words: LSA, LSI, CBIR

1 Introduction

The continuous advances of internet and digital technologies, as well as the rapidly increasing multimedia content used by modern information systems, have imposed a need for an efficient system for organizing and retrieving content from large multimedia collections. However, the performance in image retrieval is still very far from being effective for several reasons: computational cost, scalability and performance.

In our runs this year we have experimented with Latent Semantic Analysis (LSA), a technique that, although has been used successfully in many applications in the domain of text retrieval, [1] it has not experienced similar success in CBIR. The main reason being the density of the $features \times images$ matrix, C, generated in image retrieval, in contrast to textual retrieval where the $term \times document$ matrix is sparse. As a result, complexity cost of SVD is raised to prohibitively high levels for both, space and computational time.

In this article we give an overview of the application of our methods to ad-hoc medical retrieval and present the results of our submitted runs. Our efforts this year were concentrated on applying the LSA method to a number of low-level visual features and then using data fusion techniques on the SVD transformed low rank approximation of images to enhance retrieval. We explore a what we call SVD-bypass technique to factor the feature matrix by solving a much smaller in size eigenproblem of the term correlation matrix CC^T instead of solving the SVD of matrix C. This method proved to be a much more efficient and scalable solution for large data sets.

In the next section, we describe our approach and in the following sections we present the submitted IPL's runs on textual and visual retrieval with their

corresponding descriptions and results. Finally, in the last section we conclude the remarks of this work with propositions for future research.

2 Visual Retrieval

According to the traditional use of LSA in information retrieval a *term-by-document* matrix, C, is first constructed and an SVD analysis is then performed on this matrix. However as stated before, the feature matrix in the case of image retrieval, is a dense matrix. This increases the computational costs of the SVD analysis to prohibitive levels for large image databases. A typical example for our database this year, for the color layout feature will produce a matrix of size $11288 \times 305000 \approx 30$GB (in double precision) which makes the SVD impossible to solve with our computer resources. In our LSA implementation we solve the eigenproblem of the feature correlation matrix CC^T instead. This matrix, for a suitable representation of the images is of a moderate size, demanding less storage and the eigenvalue problem of CC^T can be solved much faster than the SVD factorization of the matrix C. We then approximate the feature matrix taking only the k-largest eigenvalues and corresponding eigenvectors of matrix C, for a suitable value of k.

2.1 Preprocessing of the data

It is well known that the representation of a digital image depends on several factors, from its resolution to color models etc. In a collection of images, it is highly possible that there will be important variations considering these characteristics. Thus, each image undergoes through several transformations before the feature extraction step. In our case we have applied the following transformations.

1. Size normalization. All images are re-scaled to the same size.
2. Transformation to gray-scale images.
3. Tile splitting. Each image is split into equal-sized, non-overlapping cells we reference to as tiles.

2.2 Feature Extraction and Selection

The vector representation of the images was based on three low-level features of MPEG-7, Scalable Color (SC) with 64 coefficients per tile, Color Layout (CL) having 192 coefficients per tile and the Edge Histogram (EH) feature. Experiments on CLEF 2011 image collection showed that, the extraction of the edge histogram per tile had a negative impact on retrieval performance, thus this feature was extracted from the whole image instead. All the features were extracted using the Java library Caliph&Emir of the Lire CBIR system [2].

Finally, a simple histogram with 32 levels of gray colors was extracted from each tile. To increase the discriminating power of the histogram, we remove the levels with high frequency and normalize the remaining histogram values for all

images. At the same time, all histogram levels with a total frequency above 80% are considered stop-words and thus, they are removed. We refer to this feature as Color Selection Histogram (CSH).

2.3 Construction of the feature-correlation matrix CC^T

As we have already mentioned the matrix C is full in the case of CBIR and so it is the matrix CC^T. This matrix multiplication is the most intensive part of the computations and memory demanding. In our implementation we overcome all these problems by splitting the matrix C into a number of blocks, such that each block can be accommodated into the memory ($C = (C_1, C_2, ..., C_p)$) and calculate CC^T by:

$$CC^T = \sum_{i=1}^{p} C_i C_i^T \tag{1}$$

After solving the eigenproblem of the feature-correlation matrix CC^T, the k largest eigenvectors, say U_k, and the corresponding eigenvalues, are selected. The original feature vectors are then projected into the $k-th$ dimensional space using the transformation

$$y_k = U_k^T y \tag{2}$$

on the original vector representation of an image y.

2.4 Data Fusion

For the data fusion task we used a weighted linear combination of the results obtained by using the LSA method on different features, as defined by :

$$SCORE(Q, Image) = \sum_i w_i score_i(Q, Image) \tag{3}$$

were $score_i$ denotes the similarity score of an $Image$ with respect to a feature i. The weight of each feature type is determined as a function of its performance [3]. The w_i's were estimated by the square of the Mean Average Precision (MAP) values attained by the corresponding feature on the CLEF '11 collection [4]. These values are listed in Table 1.

Table 1. MAP Values of each feature for the CLEF '11 collection.

Feature	**MAP**
Scalable Color	0.0043
Color Layout	0.0133
Color Selection Histogram	0.0023
Edge Histogram	0.0111

3 Textual Retrieval

This year's collection contains a subset of PubMed of 305000 images from PubMed. A detailed description of the collection in given in the overview paper in [5]. Since records have the same structure as in previous CLEF collections over the past years, we followed the same steps in the textual retrieval task as in CLEF 2011. Each record is identified by a unique figureID, which is associated with: the title of the corresponding article, the article URL, the caption of the figure, the pmid and the figure URL. From the pmid we downloaded the MeSH terms assigned to each article.

Our retrieval system was based on the Lucene [1] search engine. For indexing we removed stop-words and applied Porter's stemmer. For the multi-field retrieval the weights of the fields were assigned at indexing time. We kept the same structure of the database as in CLEF 2009, 2010 and 2011. This year we used only the default scoring function [2] which was best performing at the 2011 CLEF Ad-Hoc retrieval.

Also we use the same weights for the fields as in the last three years [6, 7]. We used two sets of weights: one, that was estimated empirically on the CLEF 2009 collection and a second set where the weights were estimated by the value of the Mean Average precision estimated on the CLEF 2010 collection.

[1] http://lucene.apache.org/

[2] http://lucene.apache.org/core/old_versioned_docs/versions/3_5_0/api/core/org/apache/lucene/search/Similarity.html

4 Experimental Results

4.1 Results from Textual Retrieval

This year we've submitted a total of six runs using different combinations of fields and corresponding weights. In Table 2 we give the definitions of our textual runs and in Table 3 their corresponding results.

Table 2. Definitions of IPL's runs on textual retrieval.

Run ID	Description
IPL_ATCM	Abstract, Title, Caption and Mesh terms all in one field.
IPL_TCM	Title, Caption and Mesh terms all in one field.
IPL_A10T10C60M2	Abstract, Title, Caption and Mesh in 4 fields with weights 1, 1, 6, 0.2 respectively
IPL_A1T113C335M1	Abstract, Title, Caption and Mesh in 4 fields with weights 0.1, 0.113, 0.335, 0.1 respectively
IPL_T10C60M2	Title, Caption and Mesh in 3 fields with weights 1, 6, 0.2 respectively
IPL_T113C335M1	Title, Caption and Mesh in 3 fields with weights 0.113, 0.335, 0.1 respectively

Table 3. IPL's performance results from textual retrieval.

Run ID	MAP	GM-MAP	bpref	p10	p30
IPL_A1T113C335M1	0.2001	0.0752	0.1944	0.2955	0.2091
IPL_A10T10C60M2	0.1999	0.0714	0.1954	0.3136	0.2076
IPL_T10C60M2	0.188	0.0694	0.1957	0.3364	0.2076
IPL_TCM	0.1853	0.0755	0.1832	0.3091	0.2152
IPL_T113C335M1	0.1836	0.0706	0.1868	0.3318	0.2061
IPL_ATCM	0.1616	0.0615	0.1576	0.2773	0.1742

4.2 Results from Visual Retrieval

For the visual retrieval task, we've also submitted a total of six runs, using different values for the parameter k which defines the selected number of eigen-values and vectors to use for indexing and retrieval. For all runs a data fusion on various features was applied, using the weights acquired from runs of each individual feature with the CLEF's 2011 collection. In Table 4 we give the definitions of our runs and in Table 5 their corresponding results.

Table 4. Definitions of IPL's runs on visual retrieval.

	Run ID	**Description**
R1:	IPL_AUEB_DataFusion_EH_LSA _SC_CL_CSH_64seg_20k	Edge Histogram and LSA with k=20 on 64 tiles for Scalable Color, Color layout Color Selection Histogram
R2:	IPL_AUEB_DataFusion_EH_LSA _SC_CL_CSH_64seg_50k	Edge Histogram and LSA with k=50 on 64 tiles for Scalable Color, Color layout Color Selection Histogram
R3:	IPL_AUEB_DataFusion_EH_LSA _SC_CL_CSH_64seg_100k	Edge Histogram and LSA with k=100 on 64 tiles for Scalable Color, Color layout Color Selection Histogram
R4:	IPL_AUEB_DataFusion_LSA _SC_CL_CSH_64seg_20k	LSA with k=20 on 64 tiles for Scalable Color, Color layout Color Selection Histogram
R5:	IPL_AUEB_DataFusion_LSA _SC_CL_CSH_64seg_50k	LSA with k=50 on 64 tiles for Scalable Color, Color layout Color Selection Histogram
R6:	IPL_AUEB_DataFusion_LSA _SC_CL_CSH_64seg_100k	LSA with k=100 on 64 tiles for Scalable Color, Color layout Color Selection Histogram

Table 5. IPL's performance results from visual retrieval.

Run ID	**MAP**	**GM-MAP**	**bpref**	**p10**	**p30**
R4	0.0021	0.0001	0.0049	0.0273	0.0242
R3	0.0018	0.0001	0.0053	0.0364	0.0258
R1	0.0017	0.0001	0.0053	0.0227	0.0273
R6	0.0017	0.0002	0.0046	0.0364	0.0212
R2	0.0011	0.0001	0.004	0.0136	0.0136
R5	0.0011	0.0001	0.0039	0.0091	0.0121

5 Conclusions and Further work

We have presented a new approach to LSA for CBIR replacing the SVD analysis of the feature the matrix C $(m \times n)$ by the solution of the eigenproblem for the matrix CC^T $(m \times m)$. The method overcomes the high cost of SVD in terms of memory and computing time. In addition, in all the experiments, of which only a small part was submitted officially this year, the optimal value of the approximation parameter was less than 50 which makes the method attractive for fusion with several low level features. Certainly our approach is promising and has created new research directions that need further investigation. The image representation has an impact on LSA performance and a more systematic research on that direction is currently under progress. Also the eigenvalues of the matrix CC^T follow a Zipfian distribution with the k-th largest values been well separated giving small residual vectors to machine accuracy, which give us an evidence on the stability of the calculated eigenvectors. More work is currently underway in order to determine the stability of the proposed method.

References

1. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS **41**(6) (1990) 391–407
2. Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java cbir library. In El-Saddik, A., Vuong, S., Griwodz, C., Bimbo, A.D., Candan, K.S., Jaimes, A., eds.: ACM Multimedia, ACM (2008) 1085–1088
3. Wu, S., Bi, Y., Zeng, X., Han, L.: Assigning appropriate weights for the linear combination data fusion method in information retrieval. Inf. Process. Manage. **45** (July 2009) 413–426
4. Kalpathy-Cramer, J., Müller, H., Bedrick, S., Eggel, I., de Herrera, A.G.S., Tsikrika, T.: Overview of the clef 2011 medical image classification and retrieval tasks. In: CLEF (Notebook Papers/Labs/Workshop). (2011)
5. Müller, H., de Herrera, A.G.S., Kalpathy-Cramer, J., Fushman, D.D., Antani, S., Eggel, I.: Overview of the imageclef 2012 medical image retrieval and classification tasks. In: CLEF 2012 working notes, Rome, Ital,2012 (2012)
6. Gkoufas, Y., Morou, A., Kalamboukis, T.: Ipl at imageclef 2011. In: CLEF (Notebook Papers/LABs/Workshops). (2011)
7. Gkoufas, Y., Morou, A., Kalamboukis, T.: Combining textual and visual information for image retrieval in the medical domain. The Open Medical Informatics Journal **5** (2011) 50–57