ESTIMATING MISSING FEATURES TO IMPROVE MULTIMEDIA RETRIEVAL

Abraham Bagherjeiran *Nicole S. Love* *Chandrika Kamath*

Lawrence Livermore National Laboratory, Livermore, CA 94551

ABSTRACT

Retrieval in a multimedia database usually involves combining information from different modalities of data, such as text and images. However, all modalities of the data may not be available to form the query. The results from such a partial query are often less than satisfactory. In this paper, we present an approach to complete a partial query by estimating the missing features in the query. Our experiments with a database of images and their associated captions show that, with an initial text-only query, our completion method has similar performance to a full query with both image and text features. In addition, when we use relevance feedback, our approach outperforms the results obtained using a full query.

Index Terms— multimedia information retrieval, text and image mining

1. INTRODUCTION

A common problem in multimedia retrieval is that of finding a face for a name so that we can determine if the face is present in other images without the associated name. A related problem is one where we want to associate a name with a face so we can determine if the name appears in any text documents.

This task of finding a name given a face or a face given a name in a database of documents, each containing one or more captioned images, can be difficult. If we consider the entire text of the document, we may find names unrelated to the faces in the images. A solution is to consider just the caption as it often provides a concise summary of the events in the image. But, this has drawbacks as well. If we query using the name of a person, we may get incorrect results when either the caption contains the name, but the associated image does not contain the person or the image of the person is present, but the caption does not include the name. Alternatively, focusing on the image has the well known problem of recognizing faces, given the variation due to changing illumination, different poses and view angles, changing hairstyles, and the presence or absence of makeup or accessories. It therefore makes sense to exploit both the text and the image to improve the retrieval performance.

UCRL-CONF-225087: This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Ideally, we would expect the best results when we use image and text information in both the query and in the retrieval process. Unfortunately, in many cases, we have only a partial query, where either the text or the image is missing. Recent work on combining text and images has focused mainly on the computationally expensive technique of modeling the joint probability distribution between words and image regions [1]. In this paper, we present an approach to estimating missing features which is both simple and computationally inexpensive. We focus on the specific problem of retrieving documents composed of images of faces and their associated captions given either a name or an image containing a face, and show how we can improve the retrieval results.

2. EXTRACTING TEXT AND IMAGE FEATURES

We represent each item in our database by a feature vector $v = \left(\hat{I}, \vec{t}\right)$, where $\hat{I}$ are features extracted from a face image I and $\vec{t}$ is the text feature vector derived from the associated caption. If an image has several faces in it, each face in the image has a copy of the text features, but different image features derived from the individual faces.

2.1. Text Features

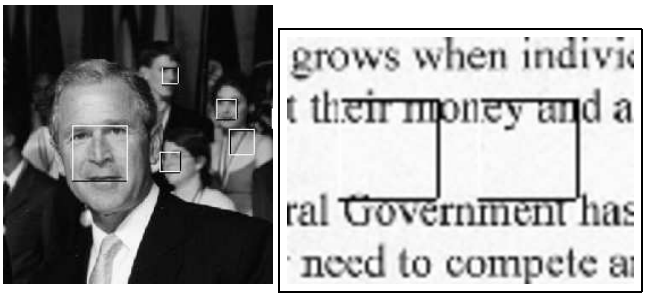
Given a caption, we extract the text features using the script, `doc2mat` [2], which removes common words and finds the root words or tokens in the caption. For example, the caption "President George W. Bush..." becomes the set of tokens $w = \{\text{presi}, \text{georg}, \text{bush}\}$. We process the tokens using the term frequency inverse document frequency (TFIDF) representation from text retrieval [3]. This represents the text part of the feature vector, $\vec{t} = (t_1, \ldots, t_{|T|})$, as:

$$t_i = \begin{cases} 1 - \frac{\log_2 df_i}{\log_2 n} & c_i \in c \\ 0 & c_i \notin c \end{cases}$$

where T is the set of all tokens in the data set and $c_i \in T$ is the ith token in the set of all tokens for the data set, c is the current set of tokens for the caption, n is the number of documents in the data set, and df_i is the number of documents that contain the token c_i.

2.2. Image Features

The image features are obtained by first finding faces in an image, and then extracting low-level features. Figure 1(a) shows an example of the results obtained using Mikolajcyzk's face detector [4]. The five face regions, including one which is clearly not a face, are outlined using a square box. Another typical error occurs with images of text documents where non-faces, with a similar intensity distribution as the eyes in a face, are identified as faces as shown in Figure 1(b).



(a) Pres. Bush at an event (b) A portion of a letter by Pres. Bush

Fig. 1. Examples of regions detected as faces

Once a candidate face is detected, it is scaled to 128×128 pixels and low-level image features are extracted. These features are general and can be extracted easily from any image as they do not entail finding facial features such as the eyes. We found the best results when the image feature vector, $\hat{I}$, consists of the following features concatenated:

1. **Angular radial transform** [5] which projects a face onto a set of orthogonal basis functions in 12 angular and 6 radial directions (*71 features*)
2. A normalized uniform **histogram** of the pixel intensities of the face (*16 features*)
3. The **Gabor** features which are the mean and standard deviation of Gabor filtered images of 5 scales and 6 orientations [5] (*60 features*)
4. The features derived from the **gray level co-occurrence matrices (GLCM)** obtained using a *quantized face* of 16 intensities. (*20 features*)
5. The **power spectrum** features [6] as well as the block averages of the square of the Fourier transform into 4 by 4 blocks (*20 features*)

These low-level image features are combined to generate, $\hat{I}$, with *187* image features for a face.

3. THE RETRIEVAL PROCESS

We create a feature vector for each data item (*i.e.* face) in the database by concatenating the image and text features. We then normalize the feature vectors so that each feature is in the range $[0, 1]$. Given a query, the retrieval process returns the closest k items to the query, with ties broken randomly. We use a weighted cosine distance to measure the similarity between the feature vectors representing the query and the items in the data set. This function computes the angle between the two weighted feature vectors x and y:

$$d(x, y) = \arccos(\cos(Wx, Wy)),$$

where W is a diagonal matrix of weights. Typically, the weights are normalized to sum to 1.

3.1. Completion of Partial Queries

We consider two approaches to estimate the missing features in partial, image-only or text-only, queries.

3.1.1. Simple query completion method

A simple approach to completing a partial query assumes that the user's query does not adequately express their true intent. The query is refined iteratively to improve the retrieval results [3]. Let the initial partial query be $q = \langle a_q, ? \rangle$ where a_q are the features provided by the user and ? indicates the missing features. In the first step, we ignore the missing features and return a set of k documents $R(q) = \{r_1, \ldots, r_k\}$, where $r_i = \langle a_i, m_i \rangle$ and k is typically less than the size of the database n. The m_i are the features in r_i that were missing from the query. Unlike the query, each retrieved document has values for all features. The original query is then updated to form a new query $\hat{q} = \langle a_q, \hat{m}_q \rangle$. This consists of the original query values a_q and an estimate of the missing values $\hat{m}_q = \frac{\sum_j \gamma_j m_j}{\sum_j \gamma_j}$, which are a weighted average of the values for the retrieved documents $R(q)$, where $\gamma_j = \delta_j e^{-\alpha \hat{d}(r_j, q)}$ is proportional to the similarity of each retrieved document r_j, and

$$\delta_j \quad = \quad \begin{cases} 1 + \beta & j \leq k^\star \\ 1 & j > k^\star. \end{cases}$$

As the top $k^\star$ documents tend to be the most relevant, we increase their weight by $\delta \in [1, 2]$ such that $0 < \beta < 1$. The parameter $\alpha \geq 1$ weights the distance values such that even the least relevant documents have weight > 0.001. We used parameters $\alpha = 2$, $\beta = 0.1$, and $k^\star = 100$ for our experiments, which we determined empirically. The distance function $\hat{d}$ is d scaled to be in the range $[0, 1]$:

$$d(r, q) \quad = \quad \begin{cases} d\left(\langle a_r \rangle, \langle a_q \rangle\right) & \text{if } q = \langle a_q, ? \rangle \\ d\left(\langle a_r, m_r \rangle, \langle a_q, \hat{m}_q \rangle\right) & \text{otherwise} \end{cases}$$

where d is the attribute-level distance.

The query is repeatedly updated using the documents retrieved in response to the previous query until the results converge or a maximum number of iterations is reached.

Google Query (Name)	# of Documents
George H. W. Bush	162
President Bush (G. W. Bush)	219
Jeb Bush	140
William Jefferson Clinton	211
Hillary Rodham Clinton	286
Bill Gates	219
Aaliyah	227
Ben Affleck	182
Andre Agassi	187
Christina Aguilera	214
non-faces	2497
unknown faces	1366

Table 1. Data set used in out experiments.

3.1.2. Query completion with relevance feedback

In the simple query completion, we assume, often incorrectly, that the first few documents are the most relevant ones. However, if the user can specify which of the returned documents are more relevant [3], we can estimate the missing values using only this relevant subset $R'(q) \subseteq R(q)$ where $R'(q) = \{r \mid r \in R(q) \wedge cl(r) = cl(q)\}$ and $cl(x)$ is the class label of a document x. This class label mimics user relevance feedback. Now, δ_j becomes:

$$\delta_j = \begin{cases} 1 & r_j \in R'(q) \wedge j \leq k^\star \\ 0 & \text{otherwise} \end{cases}$$

where $k^\star = 100$ is the maximum number of documents to consider in estimating the missing features. We limit the number of documents as a user is unlikely to have the attention span to provide feedback on all the retrieved documents. Here, the updated query, $\hat{q}$, replaces the missing values and updates the original query values a_q as well, resulting in the new query: $\hat{q} =< \hat{a}_q, \hat{m}_q >$. If $R' = \emptyset$, we revert to the simple completion method without relevance feedback.

4. EXPERIMENTAL RESULTS

We conducted our experiments using a data set of 5910 documents comprised of images and their associated captions. The documents were obtained using the Google Images Agent and the ten text queries in Table 1. 42% of the data are non-faces, 23% are unknown faces and the remaining 35% are divided among ten known faces. The non-faces correspond to errors in the face detector. We chose a low threshold to ensure few faces were missed. This lowered the precision, increasing the number of non-faces detected. We also manually labeled the documents to evaluate the quality of the results and to mimic relevance feedback.

Table 2 illustrates the contributions of the text and image features using the simple k-nearest neighbor classifier with $k = 1$. We note that text features alone cannot distinguish faces from non-faces, while the image features can easily do so. Further, text features distinguish between individuals better than image features, i.e. fewer known faces are mis-labeled as another known face. Combining the image and text features improved the accuracy by about 10%. The text-only case benefits as the image features distinguish faces from non-faces and the image-only case benefits as the text features help in distinguishing names.

features	*% correct*	*% F labeled NF*	*% KF labeled AKF*
Text	55.3%	32.5%	6.1%
Image	56.2%	4.4%	37.6%
Image + Text	66.2%	6.2%	7.9%

Table 2. Results of the 1-NN classifier using image and/or text features, including the % known faces correctly labeled, the % of face documents classified as non-face, and the % of known face documents mis-labeled as another known face.

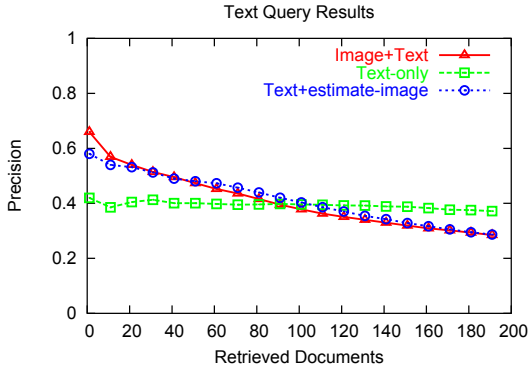
We consider 50 queries, 5 per known face. An image query was the image of a face, a text query was a unique descriptive caption (name or title), and a full query combined the two. We used precision-scope curves to evaluate our retrieval algorithms. Scope is the number of retrieved documents and precision is the ratio of the number of relevant documents retrieved to the scope. We used 20 iterations of query completion without relevance feedback or until the queries converged (queries differ by less than 0.001). For query completion with relevance feedback, we used only 2 iterations, each with 200 documents, as we do not expect the users to provide feedback on a large number of documents.

The results for text-only queries (Figure 2(a)) are relatively flat due to a large number of documents equidistant to the query. By estimating the image features, we can break ties among documents with identical captions, resulting in performance close to that of the full query. When we include relevance feedback (Figure 2(b)), the text query with estimated image features again performs as well as the full query and exceeds the full query results with no relevance feedback.

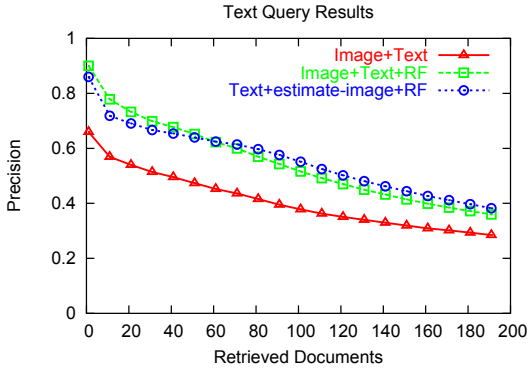
In contrast, the image-only queries (Figure 2(c)) have a low precision, as the image features are poor at distinguishing one person from another. A simple estimation of the text features improves the results only slightly as a large number of incorrect documents are combined to form the estimated features. When the correct documents are used through relevance feedback (Figure 2(d)), the completed query improves performance beyond the full query results and approaches the performance of the full query with relevance feedback.

5. CONCLUSIONS

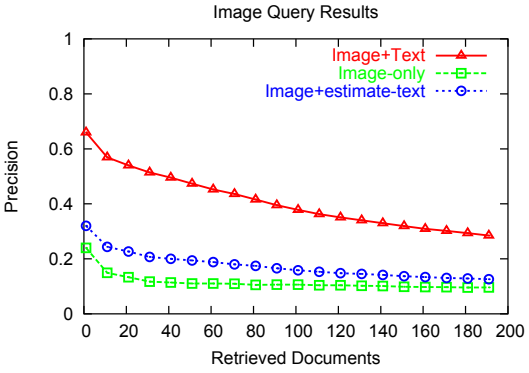
In this paper, we investigated two query completion methods for the completion of partial, text-only or image-only, queries in a multimedia retrieval application. A simple method to es-



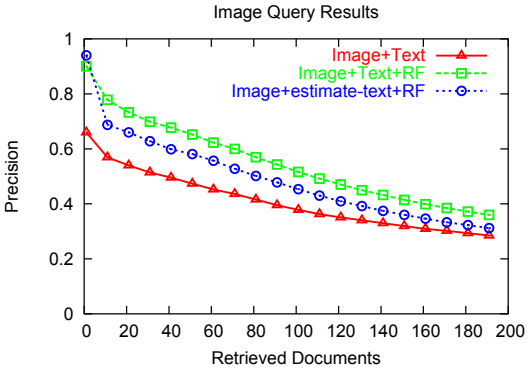
(a) text-only query with image features estimated



(b) text-only query with image features estimated using relevance feedback



(c) image-only query with text estimated



(d) image-only query with text estimated using relevance feedback

Fig. 2. Precision-scope curves for completion of text-only and image-only queries

timate the missing features allowed us to exploit the strengths of each - the text features which are good at distinguishing individuals with unique names and the low-level image features which can distinguish faces from non-faces. The use of relevance feedback led to further gains, exceeding the performance of a full query with both text and image features.

6. ACKNOWLEDGMENTS

We thank Dale Slone for the scripts to obtain the images and associated captions and Krystian Mikolajczyk for the face detector software. The software used in this work was developed by the Sapphire team.

7. REFERENCES

[1] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in *Proceedings of the 26th International Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2003, pp. 119–126, ACM Press.

[2] G. Karypis, "Cluto: A clustering toolkit.," Tech. Rep. 02-017, University of Minnesota, Department of Computer Science, November 2003.

[3] M. Ortega, K. Porkaew, and S. Mehrotra, "Information retrieval over multimedia documents," Tech. Rep. Technical Report, University of California at Irvine, 1999.

[4] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," *European Conference on Computer Vision*, vol. 1, pp. 69–82, 2004.

[5] S. Newsam and C. Kamath, "Comparing shape and texture features for pattern recognition in simulation data," in *SPIE Electronic Imaging*, San Jose,CA, January 2005, pp. 106–117.

[6] M. Augusteijn, L. Clemens, and K. Shaw, "Performance evaluation of texture measures for ground cover identification in satellite image by means of neural network classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 3, pp. 616–626, 1995.