A new relevance feedback approach for multimedia retrieval

Hanen Karamti\textsuperscript{1}, Mohamed Tmar\textsuperscript{1}, and Anis Benammar\textsuperscript{2}

\textsuperscript{1}MIRACL Laboratory, City ons Sfax, University of Sfax, B.P.3023 Sfax TUNISIA
\textsuperscript{2}REGIM Laboratory, ENIS Soukra km 3.5, University of Sfax, B.P.3038 Sfax TUNISIA

Abstract—Compared to most traditional search engines, information retrieval systems have been devoted in the past few years to relevance feedback (RF). With RF, the user can indicate which documents he finds relevant to his search. RF is an effective solution to improve performance of information retrieval, particularly in multimedia retrieval (image, video).

In this paper we propose a new approach to relevance feedback in video retrieval. We start out by adapting the standard Rocchio \textsuperscript{1}, usually used in textual information retrieval, for video retrieval. This adaptation requires the transformation of image retrieval model based on low-level visual features to a vector space model. Next, we propose an automatic query expansion approach used for image retrieval which operates on high-level features. This approach is based on knowledge resources such as the ontology and the conceptual graph.

The proposed approach is evaluated quantitatively and qualitatively using shots from videos collection of the TRECVID10 \textsuperscript{2}. The obtained results show the effectiveness of our contributions.

Keywords: Relavance Feedback; Vectorization; Rocchio; Query expansion; Conceptual graph; Ontology

1. Introduction

The main objective of video retrieval is to retrieve all images/videos that are similar to a given query in a database of images/videos. The first content-based image retrieval (CBIR) systems \cite{17} propose automatic retrieval methods based on low-level features (color, texture, shape...). These systems allow the processing of images queries, but they do not make it possible to search for images based on their semantic content. This problem is called the semantic gap. A significant improvement of the performance of CBIR systems can be achieved by using relevance feedback, a technique that allows the user to rate the search results. Since 1970, the relevance feedback mechanisms have been widely deployed in text retrieval \cite{8}, \cite{23}. In the vector space model, RF is usually carried out using Rocchio algorithm \cite{8}, \cite{23}, which forms a new query vector ($q_{new}$) from an initial query (represented by a vector noted $q_{old}$) by maximizing its similarity to relevant documents (designated by $R$) and minimizing its similarity to irrelevant documents (designated by $S$).

\begin{equation}
q_{new} = \alpha q_{old} + \beta \frac{1}{|R|} \sum_{d \in R} d - \gamma \frac{1}{|S|} \sum_{d \in S} d
\end{equation}

Where $d$ is document, $\alpha$, $\beta$ and $\gamma$ are positive constants. The RF method in probabilistic models is to select weighted terms based on Robertson/Spark-Jones technique \cite{22}. In CBIR, the main idea is to propose a mechanism that works with positive and negative images which are returned according to Robertson/Spark-Jones weight order \cite{19}. Recently, a RF track was used by CBIR through learning models such as the Support Vector Machine (SVM). SVM is the most popular method; it is used to classify the positive and negative images \cite{16}.

\textsuperscript{1}Rocchio standard is based on a method of relevance feedback found in information retrieval systems which stemmed from the SMART Information Retrieval System around the year 1970.
\textsuperscript{2}The definitive information about this collection can be found at the NIST TREC Video Track web site: http://www-nlpir.nist.gov/projects/trecvid.
In this context, we distinguish the Tow-class SVM, i.e. positive images belong to a class and negative images belong to another class, and both are divided by a hyperplane. We distinguish also the multi-class SVM where the positive and negative images belong to several classes. When they are mixed, it is difficult to find a hyperplane between two classes [4]. This method works with the regions [9], i.e. the user can select the positive and negative regions in an image. After applying relevance feedback, the similarity of each region changes, and the similarity of the image is the combination of all regions.

Further work is based on the negative relevance feedback through learning negative examples [18]. Other researches are based on descriptors [12]. The role of those approaches is to determine the importance of each descriptor involved in the CBIR. In the beginning, the importance of each descriptor is initialized with the same value. Then it is updated throughout the retrieval session. During each iteration, the user evaluates the displayed images and decides whether they are relevant or irrelevant to his need. The system uses the evaluated images as training images and decides whether they are relevant or irrelevant to the document. Then the system gives the user need. The system uses the evaluated images as training images and decides whether they are relevant or irrelevant to the document and the queries on a vector space with dimension $n$. The $d = (w(1, d), w(2, d) \cdots w(n, d))$ is the vector associated to the document $d$, when, $w(i, d) \in [0, 1]$ is the weight often $t_i$ in the document $d$.

It is the same for the query, $q$ represents the vector noted $q = (w(1, q), w(2, q) \cdots w(n, q))$. When $w(i, q) \in [0, 1]$ is the weight often $t_i$ in query $q$. The term weighting is defined with the term frequency (Tf) and the inverse document frequency (Idf) [8].

The most popular similarity measures are:

**Cosinus:**

$$\text{Cos}(q, d) = \frac{\sum_{i=1}^{n} w(i, q) \times w(i, d)}{\sqrt{\sum_{i=1}^{n} w(i, q)^2} \times \sqrt{\sum_{i=1}^{n} w(i, d)^2}}$$

(2)

**Jaccard:**

$$C(q, d) = \frac{\sum_{i=1}^{n} w(i, q) \times w(i, d)}{\sqrt{\sum_{i=1}^{n} w(i, q)^2} + \sqrt{\sum_{i=1}^{n} w(i, d)^2} - \sum_{i=1}^{n} w(i, q) \times w(i, d)}$$

(3)

**Dice:**

$$\text{Dice}(q, d) = \frac{2 \times \sum_{i=1}^{n} w(i, q) \times w(i, d)}{\sqrt{\sum_{i=1}^{n} w(i, q)^2} + \sqrt{\sum_{i=1}^{n} w(i, d)^2}}$$

(4)

**OverLap:**

$$\text{Overlap}(q, d) = \min \left(\sqrt{\sum_{i=1}^{n} w(i, q)^2}, \sqrt{\sum_{i=1}^{n} w(i, d)^2}\right)$$

(5)

### 3.1.2 Vectorization process

As the classical vector space model, we can represent the image query $Q$ by a vector of low-level features:

$$\bar{Q} = (q_{c1}, q_{c2} \cdots q_{ck}, q_{f1}, q_{f2} \cdots q_{fp})$$

Where $k$ is the number of features extracted by the descriptor CLD $^3$ (Color Layout Descriptor), $p$ is the number of features extracted by the descriptor EHD $^4$ (Edge Histogram Descriptor). $q_c$ is the color feature extracted by CLD descriptor and $q_f$ is the shape feature extracted by EHD descriptor.

$^3$A color layout descriptor (CLD) is designed to capture the spatial distribution of color in an image. The feature extraction process consists of two parts: grid based representative color selection and discrete cosine transform with quantization [11].

$^4$Edge Histogram Descriptor (EHD) is proposed for MPEG-7 expresses only the local edge distribution in the image [5].
The keyframes in the database are represented by an 
$n \times m$ matrix $M$:
\[
M = \begin{pmatrix}
I_{1,1} & I_{1,2} & I_{1,3} & \cdots & I_{1,m} \\
I_{2,1} & I_{2,2} & I_{2,3} & \cdots & I_{2,m} \\
& \ddots & \ddots & \ddots & \ddots \\
I_{n,1} & I_{n,2} & I_{n,3} & \cdots & I_{n,m}
\end{pmatrix}
\]

Where $n$ is the number of database keyframes, $m = k + p$ 
is the number of matrix columns, $I(j, c_k)$ is the $k^{th}$ 
color feature $c$ extracted from keyframe $j$ and $I(j, f_k)$ is the $k^{th}$ shape feature $f$ extracted from keyframe $j$ with $j \in \{1, \ldots, n\}$.

The initial $i$ keyframes returned by the system are represented by the reference $i \times m$ matrix $R$:
\[
R = \begin{pmatrix}
I_{1,1} & I_{1,2} & I_{1,3} & \cdots & I_{1,m} \\
I_{i,1} & I_{i,2} & I_{i,3} & \cdots & I_{i,m} \\
& \ddots & \ddots & \ddots & \ddots \\
I_{n,1} & I_{n,2} & I_{n,3} & \cdots & I_{n,m}
\end{pmatrix}
\]

To calculate the similarity between query and keyframes, we must, on the one hand, calculate the similarity between query and keyframes, with $S(Q, I_j)$ the score of similarity between query $Q$ and reference keyframe $I_j$ where $j \in \{1, \ldots, i\}$:
\[
Sim(Q, R) = \langle S(Q, I_1), S(Q, I_2), \ldots, S(Q, I_i) \rangle
\]

On the other hand, we must calculate the similarity between reference keyframes and database keyframes, where $S(I_i, I_j)$ is the similarity between the $i^{th}$ image in the database and the $j^{th}$ reference keyframe:
\[
Sim(M, R) = \begin{pmatrix}
S(I_1, I_1) & S(I_1, I_2) & \cdots & S(I_1, I_i) \\
S(I_2, I_1) & S(I_2, I_2) & \cdots & S(I_2, I_i) \\
& \ddots & \ddots & \ddots \\
S(I_i, I_1) & S(I_i, I_2) & \cdots & S(I_i, I_i)
\end{pmatrix}
\]

Then, we can construct the new query vector $Q_{\text{new}}$ defined by the vector $Sim(Q, R)$ in the new search space $Sim(M, R)$ containing the new keyframes $(\mathbf{I}_{\text{new}1}, \mathbf{I}_{\text{new}2}, \ldots, \mathbf{I}_{\text{new}m})$ with the same size as $Q_{\text{new}}$.

Therefore, we have:
\[
\mathbf{I}_{\text{new}n} = \langle S(I_n, I_1), S(I_n, I_2), \ldots, S(I_n, I_i) \rangle.
\]

So a new search is triggered and a new vector result $\mathbf{R}_2$ is constructed. To calculate the distance between vectors, we use the Overlap (equation 5) metric:
\[
\mathbf{R}_2 = (\text{Overlap}(Q_{\text{new}}, \mathbf{I}_{\text{new}1}) \ldots \text{Overlap}(Q_{\text{new}}, \mathbf{I}_{\text{new}n}))
\]

### 3.1.3 Integration of standard Rocchio

To improve the results returned by vectorization technique, we integrate the standard Rocchio (equation 1). We put $\gamma$ to 0, and we evaluates the system with different values of $\alpha$ and $\beta$. The must appropriate values of $\alpha$ and $\beta$ are respectively 0.3 and 0.7.

The first $n$ images in the vector result $R_2$ will be added to the initial query to build a new query (equation 6). This process will be reiterated until the user satisfaction.

### 3.2 Conceptual query expansion for image search

In this section, we are going to explore how semantic relations can be used to expand a query for concepts describing an image [14]. Query expansion consists in adding some synonym or relative words into the query set of original keywords to improve the recall and the precision of information retrieval. Traditional methods of query expansion did not make adequate use of semantic relations between query keywords. They often give bad results on recall and precision. These methods can be improved by using knowledge resources such as ontology and conceptual graph.

In this paper, a novel approach for query expansion is presented. The main idea of the approach is to add to the query a set of keywords which are extracted from the representation of the relations between concepts. In the first step, a conceptual graph is constructed based on LSCOM ontology.

In the second step, keywords are extracted by the intersection of connected subgraphs representing each video from the result. Finally, all keywords extracted will be added to the initial query. The features extracted from images are saved in indexes. We also used an XML file containing a set of concepts derived from semantic indexing key frames [15].

### 3.2.1 Construction of conceptual graph

A Conceptual graph is a graph where nodes are concepts and edges indicate the relationship between them. In this section, we explain the steps we followed to create this graph. We have constructed the conceptual graph using a lexicon of 130 concepts from the LSCOM ontology, used during the last session of the evaluation campaign TRECVID10. The table 3.1 shows an overview of LSCOM concepts.

Figure 1 shows an excerpt of the LSCOM ontology representing the various semantic relations between some concepts of TRECVID10. From these relations, the conceptual graph defined by Figure 2 is constructed.

http://www.lscom.org/
Table 1: Excerpt of concepts used in TRECVID2010

<table>
<thead>
<tr>
<th>TV10_ID</th>
<th>LSCOM_ID</th>
<th>LSCOM_Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>149</td>
<td>Actor</td>
<td>One or more television or movie actors or actresses</td>
</tr>
<tr>
<td>002</td>
<td>181</td>
<td>Adult</td>
<td>Shots showing a person over the age of 18</td>
</tr>
<tr>
<td>003</td>
<td>218</td>
<td>Airplane</td>
<td>Shots of an airplane</td>
</tr>
<tr>
<td>004</td>
<td>125</td>
<td>Airplane_Flying</td>
<td>An airplane flying in the sky</td>
</tr>
<tr>
<td>005</td>
<td>1062</td>
<td>Anchorperson</td>
<td>Anchorperson</td>
</tr>
<tr>
<td>006</td>
<td>202</td>
<td>Animal</td>
<td>Shots depicting an animal (no humans)</td>
</tr>
<tr>
<td>007</td>
<td>246</td>
<td>Asian_People</td>
<td>People of Asian ethnicity</td>
</tr>
</tbody>
</table>

Table 2: Recall values and precision values returned by an initial research with the queries defined

<table>
<thead>
<tr>
<th>Queries</th>
<th>Relevant images</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Male_Person&quot;</td>
<td>117</td>
<td>0.22</td>
<td>0.41</td>
</tr>
<tr>
<td>&quot;Building&quot;</td>
<td>65</td>
<td>0.32</td>
<td>0.3</td>
</tr>
<tr>
<td>&quot;Car&quot;</td>
<td>30</td>
<td>0.76</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Fig. 1: Excerpt of semantic relations resulting from LSCOM ontology

3.2.2 Integration of conceptual graphs for query expansion

After construction of the conceptual graph, we extract the connected subgraphs among all concepts of each video shot returned in the result. In fact, we search a path relating all the concepts of shot video. The relationships between concepts bring new concepts which are not defined in the shot. Furthermore, the connected subgraphs should be intersecting. The intersection between them is a set of new concepts which will be added to the initial query. Let $G(N, R)$ the conceptual graph where $N = \{c_1, c_2, ..., c_n\}$ is the set of concepts and $R \subseteq N^2$ is the set of edges. The video shot returned by the initial result is represented by $V$, where $V \subseteq N$.

We say that a shot $V$ is a connected subgraph if

$$\forall j \in \{1, 2, ..., k - 1\}, (c_j, c_{j+1} \in R) \text{ and } (c_k, c_j \in R).$$

Let $Comp_1, Comp_2, ..., Comp_n$ be the connected subgraphs of $G$ representing returned video shots in the result. $Comp_i$ is the connected subgraph linking all the concepts of video shot $i$, i.e. the concepts from $i$ and the intermediary concepts between them. The query $Q$ is represented by a subgraph in $G$ where $Q = \{q_1, q_2, ..., q_n\}$ is the set of query concepts. If $Q$ is a connected subgraph in $G$ then the new query $T_{new}$ is defined by:

$$T_{new} = \bigcap_{i=1}^{n} Comp_i \cap Q$$

If $Q$ is not a connected subgraph in $G$ then:

$$T_{new} = \bigcap_{i=1}^{n} Comp_i$$

4. Experiments

We evaluate our relevance feedback approach into TRECVID10 collection. We limit our experiments to 186 video designated by 6915 keyframes. The table 2 defines the queries used in our tests, theirs precision values and recall values returned after an initial search.
Table 3: Evaluation of the recall and precision according to the number of reference images

| n  | "Male_Person" | | | | "Building" | | | | "Car" | | | |
|----|--------------|----|----|----|--------------|----|----|--------------|----|----|--------------|----|----|--------------|
|    | Recall | Precision | | | Recall | Precision | | | Recall | Precision | | | Recall | Precision |
| 10 | 0.29 | 0.43 | | | 0.35 | 0.36 | | | 0.73 | 0.34 | | | | |
| 15 | 0.27 | 0.38 | | | 0.37 | 0.38 | | | 0.80 | 0.38 | | | | |
| 20 | 0.28 | 0.40 | | | 0.43 | 0.43 | | | 0.77 | 0.43 | | | | |
| 25 | 0.30 | 0.44 | | | 0.48 | 0.48 | | | 0.77 | 0.36 | | | | |
| 30 | 0.28 | 0.41 | | | 0.48 | 0.48 | | | 0.77 | 0.36 | | | | |
| 35 | 0.31 | 0.44 | | | 0.50 | 0.50 | | | 0.83 | 0.40 | | | | |
| 40 | 0.31 | 0.44 | | | 0.55 | 0.56 | | | 0.83 | 0.40 | | | | |
| 45 | 0.31 | 0.44 | | | 0.55 | 0.56 | | | 0.87 | 0.40 | | | | |

4.1 Assessment of vectorization approach and Rocchio adaptation

4.1.1 choice of references number

Table 3 shows that the recall increases proportionally to the number of references (n) for the "Building" query, but it depends on n for "Male_Person" and "Car" queries. We show that the recall decreases when n varies between 10 and 30. Thus, it takes a growing path. Therefore, we can conclude that the recall depends on the chosen number of references.

To identify the most appropriate value of n, we calculate the precision of these three queries when n is between 10 and 45. Table 3 shows that when n reaches the value 40, the precision becomes constant.

In what follows, we set n to 40.

4.1.2 Evaluation Metrics

To choose the best similarity measure, we evaluate four arithmetic functions: Cosinus, Jaccard, Dice and Overlap. We notice a large number of images returned from all the queries using the Overlap function (figure 3). Therefore, we can conclude that the system returns the most relevant images using the Overlap function to calculate the similarity between the query and images vector.

4.1.3 Assessment Rocchio

For the Rocchio adaptation, we evaluated the α and β values for both queries "Male_Person" and "Building". In Table 4, we find that "Building" query has the best precision results (0.67) where α and β values equal respectively 0.3 and 0.7. Another precision value is 0.65 when α equals 0.7 and β equals 0.3. For "Male_person" query, the high value of precision equals 0.42 when α and β are respectively 0.8 and 0.2 or 0.7 and 0.3. From these values, we can conclude that when α is set to 0.7 and β to 0.3, the precision value is better than the initial result search (precision = 0.41 for "Male_Person" query, precision = 0.3 for "Building" query).

The recall also reaches its maximum value for the same values of α and β.

Figure 4 summarizes the quality of the system responses before and after the RF (by integrating the vectorization technique only or by integrating the Rocchio standard with vectorization technique). From this histogram, we notice that when we apply the standard Rocchio, the precision value is improved compared with the initial result returned by the system, but it remains dependent on the quality of...
vectorization results.

4.2 Assessment of expansion query

The quality of the result after the integration of semantic relations is evaluated by two queries. We see that recall value and precision value increased compared with the initial result, which proves the effectiveness of our approach (see table 5).

Table 6 shows the new concepts found by query expansion technique.

<table>
<thead>
<tr>
<th>Queries</th>
<th>Relevant terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Adult, Single_Person</td>
</tr>
<tr>
<td>Car</td>
<td>Vehicle, Ground_Vehicle</td>
</tr>
<tr>
<td>Bicycling</td>
<td>Bicycles</td>
</tr>
</tbody>
</table>

5. Conclusion

In this article, we presented a study of the context of relevance feedback for multimedia retrieval. The mechanisms deriving from RF have been widely used in textual information retrieval. However, their induction in the process of video retrieval is underdeveloped. In addition, video data causes famous problem of semantic gap.

Our approach is to adapt a standard Rocchio, usually used if textual information retrieval, for the multimedia information retrieval. This adaptation required additional phases such as vectorization. We improve the semantic interpretation of queries through the induction of such knowledge structures such as ontologie and conceptual graph. Such structures were used to expand the user query to improve its expression.

To increase the effectiveness of our approach, it is possible to make some improvements and changes. It would be interesting to explore the LSI concept in the multimedia retrieval process. As for the knowledge structures, in order to induce the process of query expansion, it would be particularly interesting to take into account the context notion that can be deduced from existing correlations between concepts.

References