Chapter 3

Multimodal Retrieval

Image Retrieval is an active research area because of its application in online photo-sharing like Flickr[1], social search [42], video-sharing sites like YouTube[2] *etc.* However, due to the ever growing size of the web, simple text-based or visual feature-based retrieval may not be sufficient for optimal search results. Accordingly, there has been a gradual shift towards multimodal retrieval which combines features of both text and image content to achieve better efficiency. This stems from the fact that while textual descriptions are easier to process, they provide very little information about image content. On the other hand, the visual content of images can hardly be described in the text. Thus, an extensive source of information remains underutilized [24]. Also, dealing with visual features is computationally intensive

[1]https://www.flickr.com/
[2]https://www.youtube.com/?gl=IN

and time-consuming. Hence, conventional approaches [106][42] have relied on extracting modalities such as textual and visual features separately and combining them. Visual features are low-level features which can be grouped into (a) Local (interior border, color, texture *etc.*) and (b) Global (shape, contour *etc.*) features [56]. Textual features are generally regarded as high-level features. The combination (also called *fusion*) of these features can be done in two ways– *Early Fusion* and *Late Fusion*. In the feature level or early fusion approach, the features extracted from input data are first combined and then sent as input to a single analysis unit. While in the decision level or late fusion approach, the analysis units first provide the local decisions that are obtained based on individual features. The local decisions are then combined using a decision fusion unit to make a fused decision vector that is analyzed further to obtain a final decision [5].

Once the features are extracted and the fusion strategy is determined, the next step usually involves computing proper weights for combining the features [64]. Although a significant amount of work has focused on correlating textual and visual information, the *semantic gap* between the high-level information need of users and commonly employed low-level features continues to be a challenge. Semantic representation of documents is not easy to manifest, particularly when non-textual features are involved. Different query formulation techniques such as query-by-example, query-by-sketch, query-by-humming, *etc.* have been suggested in the paper [23] to bridge the semantic gap. Whenever queries are involved, *Relevance Feedback*[3] plays a pivotal role [28][103]. The way a query is formulated often dictates the retrieval

[3] A comprehensive survey on relevance feedback can be found in [74].

results. However, it is not always possible to expect the same level of expertise from an end user as that of a system developer. In such cases, *query reformulation* can prove to be of great use [40]. Among all the query reformulation techniques, query expansion (QE) is highly popular [63]. But query expansion itself is not a trivial task and a detailed study is given in the work [13]. Belkin *et al.* [8] observe that query length is positively related to increased search effectiveness. While adding unrelated terms to the query may dilute the focus of the query, appending key concepts (or keyphrases) may enhance the efficiency.

3.0.1 Motivation

Myoupo *et al.* [65] have inspired us to take up query reformulation more seriously for multimodal retrieval. For textual query expansion, we have employed two well-established keyphrase extraction techniques. The first method of keyphrase extraction utilizes *tf-idf* [60] to capture the most frequent terms in top ranked documents. The second method uses keyphrase extraction algorithm algorithm, KEA [95], which employs four features to extract keyphrases from a document. Moreover, we propose a new graph-based keyphrase extraction model incorporating mutual information that exists between words. For combining text and image features, we have adopted Fisher-LDA [94] as presented in the paper [64] to determine the optimum combination weights. We rank the relevant documents (along with its associated images) based on the final score computed for each query-document pair. We

show with the help of experiments that textual query expansion has a positive effect on image retrieval as well. We believe that this is the first study that has helped us in identifying textual query expansion (using keyphrases) as a mode of improving multimodal image retrieval[4]. Thus, drawing a direct comparison with other existing works as mentioned earlier is difficult owing to the variegation in methods, platforms and datasets.

3.1 Baseline Framework

In this section, we discuss the details of our baseline. The procedure adopted for baseline, except preprocessing has been followed in all the three expanded query retrieval approaches discussed later. Let us begin by giving an overview of the topic set. A snapshot of the topic set is provided in Figure 3.1. Proposed multimodal retrieval approach primarily consists of four steps.

3.1.1 Preprocessing

3.1.1.1 Text

Initially, we need to preprocess the dataset available to remove any unnecessary and redundant data to suit our requirements. The steps to do so are as follows:

[4]The work reported in this chapter was published as the paper entitled "Multimodal Retrieval using Mutual Information based Textual Query Reformulation" in Expert Systems with Applications, Volume 68, February 2017, Pages 81-92, SCI Impact Factor:3.928, ISSN 0957-4174, DOI:http://dx.doi.org/10.1016/j.eswa.2016.09.039.

```
<topic>
  <number> 94 </number>
  <title xml:lang="en"> roller coaster wide shot </title>
  <title xml:lang="de"> Weitwinkelaufnahme von Achterbahnen </title>
  <title xml:lang="fr"> plan large d'une montagne russe </title>
  <image> rollercoaster1.jpg </image>
  <image> rollercoaster2.jpg </image>
  <image> rollercoaster3.jpg </image>
  <image> rollercoaster4.jpg </image>
  <image> rollercoaster5.jpg </image>
  <narrative>Photos of roller coasters that show a big part of its
  rails and loops are relevant. Photos that show only a close-up of
  its cars/wagons or a small detail of the roller coaster are not
  relevant. </narrative>
</topic>
```

FIGURE 3.1: Snapshot of Topic set

1. For each image in the topic set, we remove all the non-English language text from its accompanying document[5].

2. All the stopwords are removed from each document.

3. Then, we remove all unintelligible and special characters (except punctuation marks such as '.', ',' *etc.*) to refine the text.

4. We extract only the ⟨title⟩ (denoted henceforth by t) and ⟨narrative⟩ (denoted henceforth by n) part of each topic. Further, we retain only the positive part of ⟨narrative⟩ and discard the negative part. *Positive* part means the phrases or sentences in a narrative in which the user specifies *what she wants*. Contrarily, *negative* part means the phrases in a narrative in which the user specifies *what she does not want*. These positive parts of narratives are combined to form n_p. A thorough justification for this exercise can be found in the paper [71], where

[5]From here onwards we refer accompanying text article as 'text document' and 'document' interchangeably

the authors have applied a similar technique for text retrieval. In our case, the extraction and labeling of negative and positive part were done manually to avoid any classification error.

To the best of our knowledge, this method of considering the narratives has not been applied to image retrieval yet. In Section 6.3, we show with the help of experiments that our hypothesis holds true.

5. Finally, we combine title t and n_p to form the baseline text query q_T.

3.1.1.2 Image

We did not preprocess the images as the features were readily available. The low-level visual features include both local features and global features (texture, color and edges). Here, *Surf* [7] plays the global descriptor while *cime* [86] and *telp* ([15]) are two local descriptors. Specifically, *cime* is used for classifying pixels into either interior or border and *telp* is used for texture and color. Features are extracted from visual vocabularies as well as from visual queries and represented using bag-of-visual-word model.

3.1.2 Indexing and Matching Score calculation

- All the text documents are indexed using Terrier IR ([81]) in its default setting of Vector Space Model[6]. No indexing was performed over the images since their features were available apriori in the dataset.

[6] http://terrier.org/docs/v4.0/quickstart.html

- Now, for each text query-document pair we calculated a matching score $\mathcal{S}_T$ using cosine similarity ([27]). Similarly, for each query-image feature pair we calculated matching scores $\mathcal{S}_{V_{cime}}$, $\mathcal{S}_{V_{telp}}$ and $\mathcal{S}_{V_{Surf}}$ for *cime*, *telp* and *Surf* features respectively[7].

3.1.3 Learning Combination Parameters

For this step, we divide the entire topic set into 80:20 ratio for training and testing (40 topics for training and 10 topics for testing) and performed a 5-fold cross validation. Then, using the training set, we learn the optimal weight for combination parameters for each modality.

Assigning proper weights through learning is a delicate issue when there are more than one modalities involved. A modality's weight should be determined based on its descriptive ability [54]. Usually, text-based image retrieval performs better than image-based image retrieval [21]. This implies that text information should be given more priority over visual information while assigning the combination weight.

Moulin *et al.* [64] focus on various weight learning schemes: *MAP Optimization*, *Combination parameter optimization* by *Fisher Linear Discriminant Analysis* for text-image combination and provides a comparative measure of their performances. Textual and visual scores are combined linearly when more than one visual vocabulary are present. MAP Optimization and Fisher-LDA are used for learning the combination parameter, δ. They establish that Fisher-LDA performs better than

[7] T denotes text and V denotes visual features

other combination models in multimodal retrieval. Therefore, we adopt Fisher-LDA for weight learning in our proposed model.

The Fisher-LDA learning model: Given a document-query matrix $\mathbf{X}$ of dimension $|D| \times |Q|$, it can be subdivided into relevant set X_r and non-relevant set $X_{\bar{r}}$ with respect to each query q. If $\mathbf{C} =< \delta_1, \delta_2, ..., \delta_{|N|} >$ is the coefficient vector of score vector $\mathbf{X}_q =< X_1, X_2, ..., X_{|N|} >$ then the discrimination function can be written as Equation 3.1:

$$C_0 = \mathbf{C}^t \mathbf{X}_q = \sum_{i=1...|N|} \delta_i X_i \tag{3.1}$$

The associated covariance matrix, Cov, of C_0 can be decomposed by Huygens theorem into within class matrix M_w and between class matrix M_b. Definition of these three matrices are given in Equations 3.2, 3.3 and 3.4 as follows:

$$Cov = \left(\frac{1}{|\mathbf{X}_q|}\right) \sum_{i=1}^{|\mathbf{X}_q|} (x_i - \mu)^t (x_i - \mu) \tag{3.2}$$

$$M_b = \left(\frac{1}{|\mathbf{X}_q|}\right)(|X_r|(\mu_r - \mu)^t(\mu_r - \mu) + |X_{\bar{r}}|(\mu_{\bar{r}} - \mu)^t(\mu_{\bar{r}} - \mu)) \tag{3.3}$$

$$\begin{aligned} M_w = & \left(\frac{1}{|\mathbf{X}_q|}\right)\sum_{x_i \in X_r}(x_i - \mu_r)^t(x_i - \mu_r) \\ & + \left(\frac{1}{|\mathbf{X}_q|}\right)\sum_{x_i \in X_{\bar{r}}}(x_i - \mu_{\bar{r}})^t(x_i - \mu_{\bar{r}}) \end{aligned} \tag{3.4}$$

where μ, μ_r and $\mu_{\bar{r}}$ denote the mean data vectors given by Equations 4.1, 3.6 and 3.7 respectively:

$$\mu = \left(\frac{1}{|\mathbf{X}_q|}\right)\sum_{i=1}^{|\mathbf{X}_q|} X_i \tag{3.5}$$

$$\mu_r = \left(\frac{1}{|X_r|}\right)\sum_{x_i \in X_r} x_i \tag{3.6}$$

$$\mu_{\bar{r}} = \left(\frac{1}{|X_{\bar{r}}|}\right)\sum_{x_i \in X_{\bar{r}}} x_i \tag{3.7}$$

According to Fisher-LDA, maximization of the ratio $\left(\frac{\mathbf{C}^t M_b \mathbf{C}}{\mathbf{C}^t Cov \mathbf{C}}\right)$ best separates the relevant and non-relevant classes[8]. We adopt the same approach to obtain the best possible combination of weights for our experiments. These parameters are δ_T for text and $\delta_{V_{cime}}$, $\delta_{V_{telp}}$ and $\delta_{V_{Surf}}$ for visual modality, respectively. The values corresponding to each set of training data are listed in Table 3.2.

3.1.4 Ranking

Now, we use the combination weights learnt above in the testing phase. For each query a final score $\mathcal{S}_{final}$ is calculated as given in Equation 3.8

$$\mathcal{S}_{final} = \sum_{f} \delta_f \mathcal{S}_f \tag{3.8}$$

where $f \in \{T, V_{cime}, V_{telp}, V_{Surf}\}$. Based on this final score, the documents are ranked corresponding to each query.

[8] 't' in Equations 3.1 through 3.4 denotes transpose.

3.2 Query Reformulation

Query Expansion using relevant documents is a widely accepted approach for query reformulation in text retrieval [77][59]. Possible expansion terms can be deduced from the content of these relevant documents and deduced terms can be ranked by some measure that describes how useful the terms might be in attracting more relevant documents [13]. An expanded text query can be typically compared against

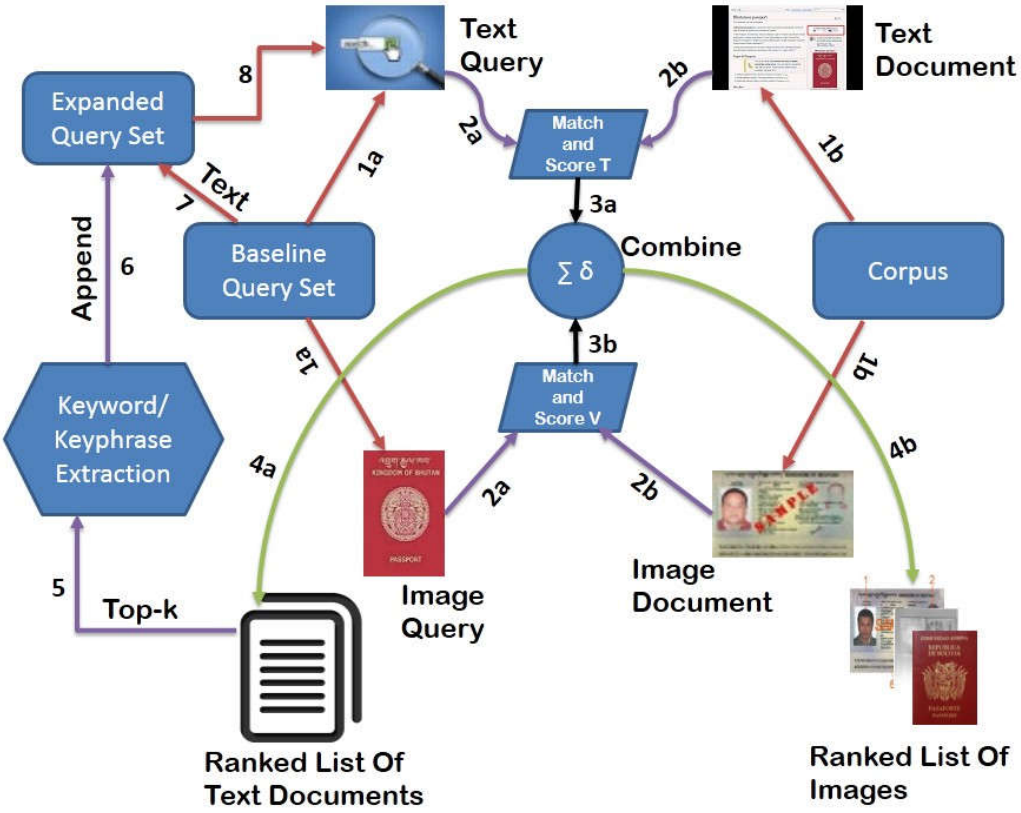

FIGURE 3.2: High-level diagram of proposed multimodal retrieval model

the textual description of the visual information. In our work, we have used Pseudo-Relevance Feedback[9] for Query Expansion. We have taken a few informative terms

[9] http://nlp.stanford.edu/IR-book/html/htmledition/pseudo-relevance-feedback-1.html

from top-k retrieved documents against each query and appended these highly informative terms into the original query. For extracting keywords/keyphrases from relevant documents, we have adopted two well-established models in addition to proposing one as described in the next subsections.

A high-level representation of the hitherto mentioned process is depicted in Figure 3.2. The steps are numbered from 1 to 8 for each cycle. It is to be noted that substeps a and b are carried out simultaneously. Each cycle consists of the following steps:

1a: An initial query consisting of both text and image (or any one of them), is selected from the baseline query set.

1b: From the corpus, all the documents and images are fetched using the baseline system.

2a: Both text and image query are then broken down into features and fed into the matching and scoring function.

2b: Similarly, the text documents along with its image counterparts are split into its features and fed into the matching function.

3a: The score for matching text documents is generated in this step and forwarded to the fusion procedure.

3b: Matching images are scored and forwarded to the fusion step.

4a: & **4b:** A ranked list of documents and its counterpart images is generated based on the scores.

5: From the top-k text documents of the ranked list, keyphrases/keywords are extracted using one of the three methods.

6: & **7:** The extracted keyphrases/keywords from the previous step are appended to the original text query, to form the expanded text query.

8: The final expanded query is again fed as an input query to fetch a new set of results.

This complete cycle can be repeated as long as the query does not saturate (explained later in Section 3.5.1)

3.2.1 tf-idf based Query Expansion Model (TQEM)

A common text query expansion technique is to include the terms from the documents having highest frequency. However, the raw frequency is rarely used as a measure of the importance of key terms. Instead, tf-idf [60] is used to identify useful keywords. We use Lemur TFIDF [104] version. Algorithm 1 gives an insight into the procedure. It is to be noted that, the documents of the corpus has already been preprocessed as part of the baseline framework (Section 3.1.1). Our reduced corpus D_r comprises of the top-5 relevant documents that were retrieved per query using the baseline method[10]. This was done to avoid any random words to be picked as a keyword.

The set of keywords extracted using this algorithm are then ranked on the basis of tf-idf score and only top five unique keywords are then used for query expansion. In

[10]This is applicable for the next two approaches as well

Algorithm 1: tf-idf based keywords $\mathcal{K}$ extraction

Data: Preprocessed Corpus D_r, Empty vector $\mathbf{f}$
Result: Keywords $\mathcal{K}$

```
for ∀w_i ∈ D_r do
    f[i] ← tf-idf(w_i) ;
end
Sort(f)                          ▷ In descending order;
j ← 1;
for j < 5 do
    if w_n = w(f[j]) then
        K.append(w_n);
    end
end
```

the seventh step of the algorithm $w(f[j])$ stands for the term corresponding to *tf-idf* value of $f[j]$. Once the expanded query is generated, it is then used for computing similarity with the documents using the process explained in Section 3.1.2. The procedure of learning combination parameters and ranking of documents is explained earlier.

3.2.2 KEA based Query Expansion Model (KQEM)

KEA [95] is an algorithm for extracting keyphrases from text documents. It can be either used for free indexing, where keyphrases are selected from the document itself or for indexing with a controlled vocabulary. According to the authors of KEA, there are two fundamentally different approaches to the problem of automatically generating keyphrases for a document: (a) keyphrase assignment and (b) keyphrase extraction. Both these approaches use machine learning methods and require a set of documents with keyphrases for training purposes. Keyphrase assignment attempts

to select the phrases from a controlled vocabulary that best describe a document. The vocabulary provides uniform access to clusters of related documents grouped under a single keyphrase. The training data associates a set of documents with each phrase in the vocabulary and builds a classifier for each phrase. Before extracting keyphrases from new documents, KEA first creates a model that learns the extraction strategy from manually indexed documents. For each document in the input directory KEA requires a corresponding file with manually assigned keyphrases.

Given the list of the candidate phrases, KEA marks those that were manually assigned as positive example and the rest as negative examples. By analyzing four feature values (stated below) for positive and negative candidate phrases, a model is computed, which reflects the distribution of feature values for each phrase. A new document is processed by each classifier and assigns keyphrases of any model that are classified positively. The only keyphrases that can be assigned are ones that have already been seen in the training data. For extracting keyphrases from new documents, KEA feeds feature values into appropriate models for each candidate phrase and computes its probability of being a keyphrase. The final set of keyphrases consists of phrases with the highest probabilities. A few key points of KEA are presented below.

1. **Additional Preprocessing**

   (a) *Documents* : KEA preprocesses the entire document set and keeps them under a directory.

(b) *Thesaurus* : It has an option for additional vocabulary.

(c) *Extracting Candidates* : This method calculates predefined n-grams for matching vocabulary with a thesaurus.

2. **Features:** KEA considers the following four features for extracting keyphrases.

(a) *tf-idf* : Candidate phrases that have high *tf-idf* value are more likely to be keyphrases.

(b) *First occurrence* : Terms that tend to appear at the start or at the end of a document are more likely to be keyphrases.

(c) *Length of a phrase* : It considers phrases which are of preferable length.

(d) *Degree of Node* : Number of semantically related phrases is treated as the degree of the phrases. Higher degree phrases are preferred over lower ones.

3. **Learning Model:** Before extraction KEA undergoes through a training stage. It needs to learn extraction model from manually indexed documents.

4. **Keyphrase Extraction :** In the final step, most suitable candidate phrases are selected as keyphrases based on feature values and learning strategy.

Thus, we fed our preprocessed corpus to KEA and it returned a set of ranked keyphrases. We selected the top five keyphrases to expand the baseline queries.

An interesting find of this exercise was that the expanded queries performed significantly well over the baseline as indicated in Tables 3.3 and 3.4. Consequently, we checked the images that were retrieved along with the associated documents and found out that in many cases, the query expansion models were superior. Thus, it reaffirmed our hypothesis that textual query expansion alone can produce better image retrieval results. In the next section, we discuss our proposed algorithm along with the differences it has compared to the above two approaches. Also, we show that our approach outperforms the other two in both text and image retrieval.

3.2.3 Mutual Information based Query Expansion Model (MIQEM)

There are many approaches for keyphrase extraction from text documents. One of the unsupervised methods includes Graph-based ranking in which a graph is built from the input documents and its nodes are ranked according to their importance [62]. Each node of the graph corresponds to a candidate keyphrase from the document and an edge connects two related candidates. We borrow this notion of representing *relatedness* from the graph-ranking method, but build the graph in a different fashion altogether. Existing methods require candidate keyphrase to be identified prior to ranking.

We hypothesize that if we can build a graph of all the words embedded in a set of documents, we will be able to capture the semantic and syntactic similarity among

words effectively. Simple co-occurrence calculation may not give us a complete picture of the word distribution in the corpus. Instead, we build a graph of all the words in the document with words as nodes and their edges connect the nodes which are correlated through semantics and syntax. Algorithm 2 gives an insight into the graph building procedure.

The essence of any keyphrase extraction technique is to determine the most significant terms from a corpus. For measuring the significance of a term, frequency is an established feature. But, as stated in the literature, frequency is not always the best metric. Instead, a probabilistic feature may measure the significance better. A probability based feature that can capture the term significance could be a product of probability of occurrence of a term and the amount of information that the term carries [2]. In light of this, we propose a Mutual Information based probabilistic model for keyphrase extraction. The main step in this model is the construction of a *word graph* which represents the measure of interaction between any two words/terms. Here, the measure of interaction is the mutual information between any word pair. Unlike other frequency based approaches which provide most significant terms, our method provides most significant *keyphrases*. Notably, using only top-5 documents as a corpus lends us the advantage of cutting down the complexity of constructing the word graph as explained below.

Our proposed algorithm is broadly divided into three steps:

1. ***Construction of Word Graph:*** The intuitive idea behind the construction

of Word Graph is that two words that co-occur more than one time may share some information together. Mutual information compares the probability of observing any two random variables x and y together (the joint probability) with the probabilities of observing x and y independently (chance) [16].

$$I(x, y) \propto \log \frac{P(x, y)}{P(x)P(y)} \tag{3.9}$$

If there is a genuine association between x and y, then the joint probability $P(x, y)$ will be much larger than chance $P(x)P(y)$, and consequently $I(x, y) >> 0$. If there is no interesting relationship between x and y, then $P(x, y) \approx P(x)P(y)$, and thus, $I(x, y) \approx 0$.

This means that if two or more words occur concurrently in more than one document and have a high degree of correlation, it will be reflected in their mutual information. In other words, any two or more co-occurring terms with high mutual information is a candidate keyphrase. Steps 5 to 17 of the algorithm performs this task. First, we compute the probability of occurrence of each term w_i in the POS tagged corpus $\hat{D}$. Let it be denoted by $p(w_i)$.

Next, for each document in the corpus, we compute the mutual information between any two terms w_i and w_j, denoted by $\mathcal{I}(i, j)$.

$$\mathcal{I}(i,j) = p(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{3.10}$$

where $p(w_i, w_j)$ is the joint probability of w_i and w_j. Now, in the word graph, each of the terms (words) in the corpus is a node ($V =< w_1, w_2..., w_n >$). There exists an edge between any two nodes v_i and v_j if mutual information $\mathcal{I}$ of the corresponding terms w_i and w_j is greater than zero. In other words, the terms in the graph are connected by the strength of their correlation and co-occurrence. It should be noted that such a word graph is undirected.

2. ***Semantic Enrichment of Word Graph:*** Existing keyphrase extraction approaches takes into account various text features like co-occurrence frequency. However, usually, they ignore the semantic relation between two words which may not necessarily occur in the same document. For a text query, these relationships may prove to be crucial in determining appropriate keyphrases. For example, one document may contain the words "car" and "Volkswagen" frequently while another document may contain "vehicle" and "Volkswagen". When looked at separately, these two documents may appear different since these words may have different frequencies. But, if we could somehow harness the fact that "car" and "vehicle" are related to each other, then the phrase "car Volkswagen vehicle" would supposedly be more prudent in identifying

relevant documents.

We try to incorporate this knowledge of *relationship* through WordNet[11]. If v_j is related to a term v_i in the word graph G according to WordNet, then we introduce edges between v_j and v_k (for all k), where k is the index of nodes to which v_i was previously connected. The weight of the edge (v_j,v_k) is same value as that of (v_i,v_k). It should be noted that we have ignored the type of relation that exists between two nodes, since relations are symmetric, which is reflected by the non-directionality of the graph G. The resultant graph $\bar{G}$ is more enriched in terms of the number of connections between terms (nodes). This process of enrichment is carried in step $18 - 20$ of Algorithm 2.

3. ***Extraction of Keyphrases:*** Next step is to identify candidate keyphrases $\mathcal{K}_c$. For any keyphrase to be a candidate, it has to satisfy all the rules as stated below:

   i. Each candidate must have four or fewer terms in it. This limitation arises from the fact that, statistically, the length of keyphrases are not more than four terms in any document. Also, having a lengthier keyphrase will unnecessarily lengthen the query.

   ii. The first term must be a noun or an adjective and have a non-zero degree in the word graph.

[11] https://wordnet.princeton.edu/

iii. Each of the following terms must have a positively weighted edge with the previous term.

iv. If a term has more than one outgoing edges, the algorithm selects the edge with the highest edge weight[12]. Thus, we follow a greedy approach.

The search for a term to be included in the keyphrase stops when there are no more terms left to traverse, or the sentence ends with a punctuation mark. This process is repeated so that each term in the word graph is traversed at least once.

Once all the candidate keyphrases have been extracted from the word graph, we rank them in descending order on the basis of *informativeness*. Informativeness of a keyphrase is measured by summing over all the edge weights of terms present in the keyphrase. After ranking, we select top five keyphrases $\mathcal{K}$ from the set of candidate keyphrases $\mathcal{K}_c$ to be appended to the original baseline query. Line 21 of Algorithm 2 performs this operation.

3.3 Theoretical Comparison of Keyphrase Extraction Models

A brief overview of the salient characteristics of the different keyphrase extraction methods is presented in the previous section. In this section, Table 3.1 exhibits the

[12]If there is a tie between two outgoing edges, we randomly select one of them.

Algorithm 2: Mutual Information based Keyphrase Extraction

Data: POS tagged Corpus $\hat{D}$, a null graph $G(V, E)$
Result: Candidate Keyphrases $\mathcal{K}_c$

for $\forall w_i \in \hat{D}$ **do**
  $p(w_i) \leftarrow ProbOccur(w_i)$;
end
$l \leftarrow 1$;
for $l \leq |\hat{D}|$ **do**
  for $\forall w_i, w_j \in \hat{D}$ **do**
    if $p(w_i, w_j) > 0$ **then**
      $\mathcal{I}(i, j) \leftarrow p(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$;
    end
  end
  $v_i \leftarrow w_i \in \hat{d}_l$ ▷ $\forall v_i \in V$;
end
for $v_i, v_j \in V$ **do**
  if $\mathcal{I}(i, j) > 0$ **then**
    $e_{i,j} \leftarrow \mathcal{I}(i, j)$ ▷ $i \neq j$, $e_{i,j} \in E$;
  end
end
for $\forall v_i \in V$ **do**
  $\bar{v}_i \leftarrow WordNetEnrich(v_i)$;
end
$\bar{G} \leftarrow G(\bar{V}, E)$;
$\mathcal{K}_c \leftarrow TraverseAndExtract(\bar{G})$

findings of the comparison among them.

As stated earlier, We use these different keyphrase extraction techniques in retrieval task where extracted keyphrases are used to reformulate the textual queries through relevance feedback. The comparative retrieved results are presented in the Section 6.3 which show that the retrieval efficiency of expanded query based retrieval models is better than the baseline retrieval model for both the image and text retrieval. Although in the current literature different retrieval models are present, quantitatively we have compared our proposed query expansion based retrieval models against our

Parameters \ **Models**	**TF-IDF based Query Expansion Model (TQEM)**	**KEA based Query Expansion Model (KQEM)**	**Mutual Information based Query Expansion Model (MIQEM)**
Approach	Rule-Based	Supervised Learning	Graph-Based
Features	Term Frequency Inverse Document Frequency	Term Frequency Collection Frequency Relative position of the first occurrence of the term POS tag of a term	Measures of specificity Measures for term weighting Pairwise Mutual Information Informativeness of a keyphrase
Advantages	Efficient and Simple Straightforward encoding	Automated Keyphrase Extraction Relatively better performance Works well for summarizing, browsing, searching and clustering	Exploits relatedness between terms Captures inter-word semantic association Uses greedy approach for keyphrase selection Superior in Performance Effective and Tractable
Drawbacks	Suffers from non-scalability Introduces quantification error	Requires human intervention (labeling keyphrases) Satisfactory results not guaranteed	Dependent on relevance of initial set of retrieved documents

TABLE 3.1: Comparative overview of Keyphrase Extraction Models

baseline model only. This is due to our retrieval model requires ad-hoc datasets with some specific attributes like set of images, collection of associated documents, topic set consisting of text queries, text queries and narratives, relevent retrieval judgments *etc.* Most of the current works do not consider all of those specific attributes. So comparing our proposed retrieval models with the other retrieval systems is out of scope here.

3.4 Experimental Setup

3.4.1 Dataset

The ImageCLEF Wikipedia Retrieval 2011 collection[13] was used in the ImageCLEF 2010-2011 Wikipedia image retrieval evaluation campaigns. This collection consists of 237, 434 images based on the September 2009 Wikipedia dumps. A major portion of this image collection is annotated, while some images are without any annotations. Annotations are provided by Wikipedia users in several languages. The provided textual descriptions of the images are extracted from the Wikimedia common files and from the articles that contain the images. These annotations as well as descriptions are heterogeneous and are in three languages (in English, German and French), except narratives, which are present only in English. Nearly 10% of the collection is annotated in all the three languages. 24% of the images are annotated bilingually and 62% images are annotated in only one language. The rest 6% are either not

[13] http://www.imageclef.org/2011/Wikipedia

annotated or annotated with other languages.

Metadata files have names of the images as available in Wikimedia Common Repository and links to images and their associated descriptions, comments and captions. These metadata files have information about the text associated with images. Wiki11[14] topic-list consists of 50 topics. Each topic contains text queries in three different languages, five visual queries and a narrative. A snapshot of the topic set is provided in Figure 3.1.

Ground truth of the topic list is given in a relevance assessment ('qrel') file. This 'qrel' was generated assuming binary relevance, relying on relevant *vs.* non-relevant from the runs submitted by the participants of previous years.

3.4.2 Specifications

- **Text** We have considered only English language documents, queries and narratives from the dataset. The total number of English documents in this collection is $70,127$. Using information from metadata files, we have mapped text documents with their associated images. In this mapping operation image-id and text-id play a vital role. After preprocessing, query-document matching scores are calculated by Lemur TFIDF model [104].

- **Image** The dataset comprises of three sets of visual features (*cime, telp* and *Surf*) for the entire image collection and image queries represented by bag-of-words. PIRIA, CAE LIST's tools [47] are used to index these image features.

[14] http://medgift.hevs.ch/wikipediaMM/2010-2011/wikipedia_topics_2011.zip

Dimension of indexed *Surf* features is 5000, *telp* features is 576 and *cime* features is 64 respectively for each image.

3.4.3 Evaluation Measures

We have used Wiki11 dataset to test our system. The relevance judgment file provided with the dataset contains a limited number of relevant documents per query. It is to be noted that we have not set any threshold on the number of retrieved documents. So, our end-result is a list of ranked documents. Hence, we have adopted R-Precision[15] instead of ordinary Precision and Recall as a metric to judge the test set results. To evaluate the efficiency of our proposed method against the baseline and other two approaches we have calculated Mean Average Precision (MAP) over 50 given topics.

Another point to be noted is that we have evaluated only the text retrieval using the metrics mentioned above. The images being qualitative in nature, no such metrics were used. Instead, a brief discussion of them is provided in Section 3.5.2.

3.5 Results and Analysis

In this section we discuss the results obtained using the above mentioned query expansion techniques namely TQEM and KQEM. Also we show how our proposed algorithm fares against the other two. We performed 5-fold cross validation over

[15]Henceforth we refer R-Precision as simply Precision

the total topic set of 50 queries. Thus, each test query set comprised of 10 queries (topics) and the remaining 40 queries formed the training set. Before comparing various techniques against each other, we present the optimal training parameters obtained for all the five test sets. Table 3.2 shows the optimum value for δ for each of the modalities. As we hypothesized, since we are not reformulating the image

Training Set	$\delta_{V_{cime}}$	$\delta_{V_{telp}}$	$\delta_{V_{Surf}}$	δ_T
Set 1	0.00646	0.00882	0.04966	0.19497
Set 2	0.00646	0.00882	0.04966	0.18908
Set 3	0.00646	0.00882	0.04965	0.20204
Set 4	0.00646	0.00882	0.04966	0.44258
Set 5	0.00646	0.00882	0.04965	0.21842

TABLE 3.2: Optimal Weight of Combination Parameters

features there wasn't any noticeable changes in the δ values for the visual features. The change is only in δ_T, which is obvious since the textual queries were expanded.

3.5.1 Comparison of various approaches

Next, we present the comparison results of three approaches TQEM, KQEM and MIQEM against the baseline. In Table 3.3, we compare the approaches when text queries are formulated without narratives. The first five rows of the table depict the Average Precision values of retrieved results for each test set of 10 queries against each of the three query expansion approaches. In the last row, we present the Mean Average Precision (MAP) values over the total set of queries. Clearly, for Mean Average Precision (MAP), MIQEM performs better than baseline, TQEM and KQEM. But that is not the case with average precision as can be seen in Set 4

where KEA performs slightly better than ours. We tried to analyze the reason for the same and found that for some queries in Set 4, the top 5 documents retrieved by baseline method hardly contained any useful keywords. This in turn suggests that, if the documents retrieved using baseline are not of good quality they would attribute to selection of poor keyphrases using MIQEM. So essentially it means that if we pick documents in which words co-occur simply by chance rather than some correlation between them our method may not perform better for every instance. Although, on the whole it still gives better results. Next, we performed the same set of experiments by considering the relevant narratives of text queries.

In Table 3.4, we compare the approaches when text queries consider relevant portion of narratives. MIQEM significantly outperformed in all sets in terms of Average

	Baseline	**TQEM**	**KQEM**	**MIQEM**
Set 1	0.2028	0.2691	0.2721	0.2842
Set 2	0.3198	0.3625	0.3245	0.3658
Set 3	0.2201	0.3207	0.2627	0.3306
Set 4	0.2370	0.2941	0.3076	0.3030
Set 5	0.2222	0.2187	0.2596	0.2713
MAP	0.2404	0.2930	0.2853	**0.3110**

TABLE 3.3: Average Precision per query set and MAP for queries without narration

Precision and MAP. Also, the improvement is statistically significant[16]. So we can conclude that narration of queries play an important role in selection of documents. This point has been neglected by existing TBIR systems. Our experiments show that narratives convey more information and helps model the actual information

[16]We performed Two-sample unequal variance t-test on the complete set of 50 topics.

	Baseline	**TQEM**	**KQEM**	**MIQEM**
Set 1	0.2139	0.3165	0.2874	0.3849
Set 2	0.3051	0.3245	0.3333	0.3765
Set 3	0.2518	0.3097	0.2939	0.3451
Set 4	0.2105	0.3451	0.3676	0.4028
Set 5	0.1841	0.2536	0.2713	0.3183
MAP	0.2331	0.3099	0.3107	**0.3655**

TABLE 3.4: Average Precision per query set and MAP for queries with narration

need. So, if we emphasize on the positive part of the narrative we are bound to get better results. Figures 3.3 through 3.7, exhibit the per query results for each of the five test sets. It can be seen that in most of the cases, our proposed approach gives a better result than the baseline and other two methods. In fact, we gained an improvement of 56.81% over baseline approach and 17.94% and 17.63% over TQEM and KQEM respectively with respect to MAP.

This validates our claim that when the semantic correlation between words is taken into consideration, it improves retrieval efficiency. In few cases, though, such as for query 71, 88, *etc.*, TQEM and KQEM perform at par with our MIQEM. This could be due to the fact that factors like tf-idf played an important role in the selection of these keywords simply because they were very frequent generic keyphrases spread across documents. It could also be attributed to the fact that each of the key terms in our word graph was semantically enriched using WordNet and since WordNet itself is not comprehensive, it could have negatively affected the keyphrase extraction. It might also be the reason why few queries underperformed using our approach. However, the overall gain is statistically significant as the results depict.

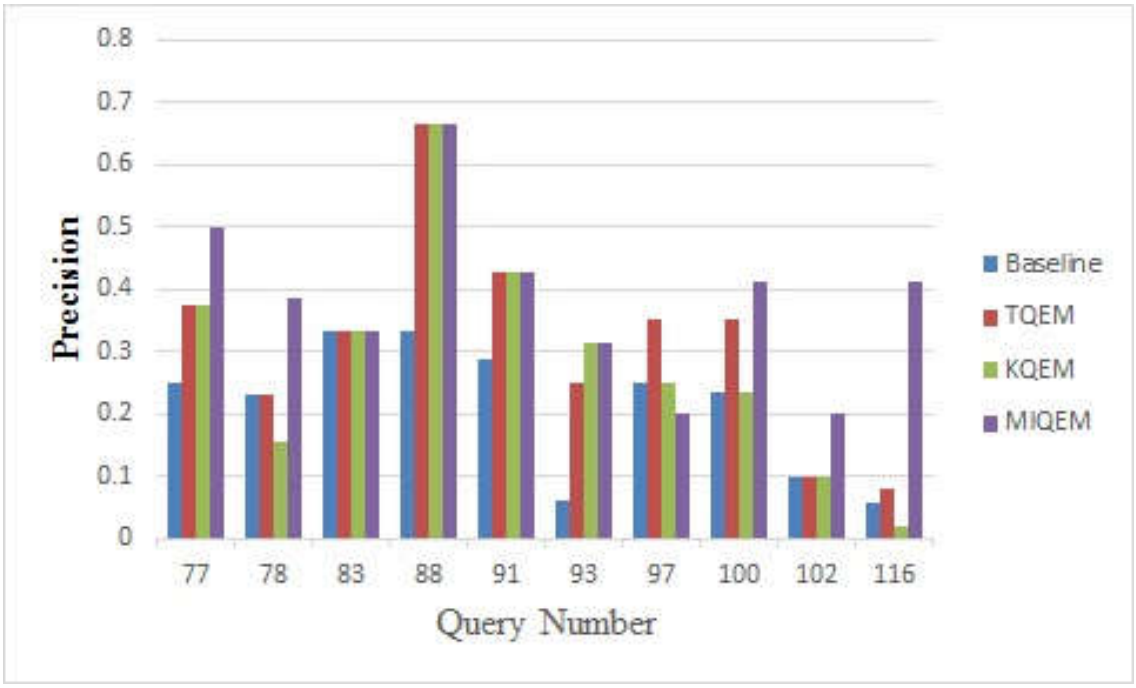

FIGURE 3.3: Per Query Precision Comparison for Set 1

FIGURE 3.4: Per Query Precision Comparison for Set 2

Our next step is to ensure that we are not sacrificing tractability of our algorithm to improve accuracy of the retrieval system. To do this, we carry out the query

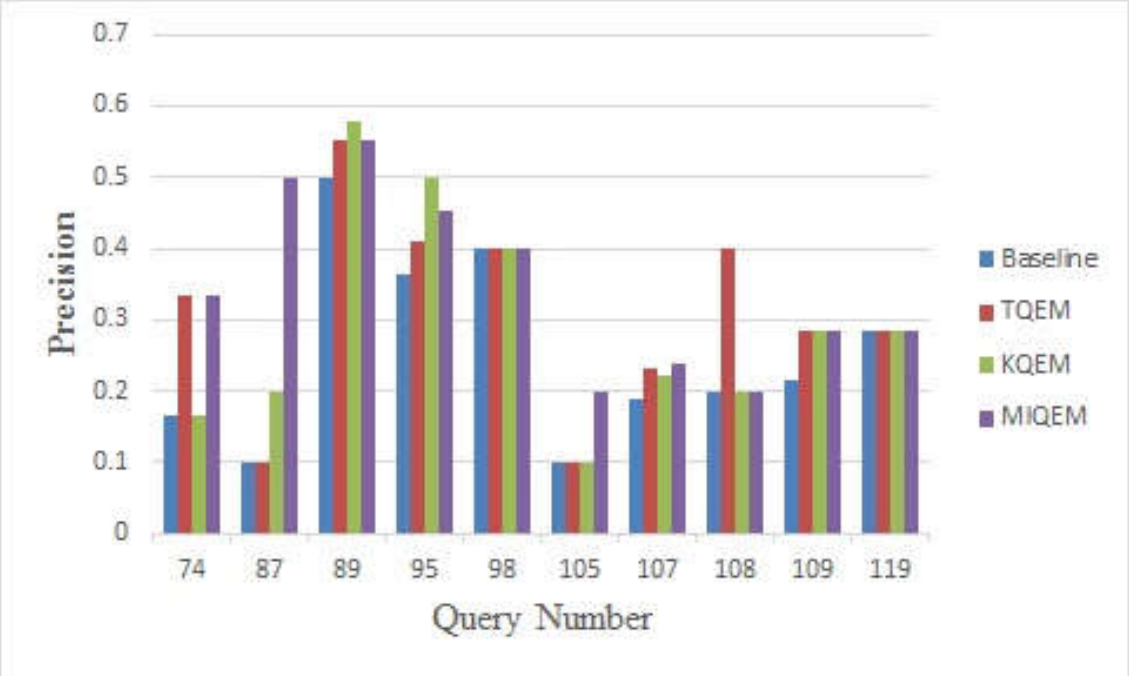

FIGURE 3.5: Per Query Precision Comparison for Set 3

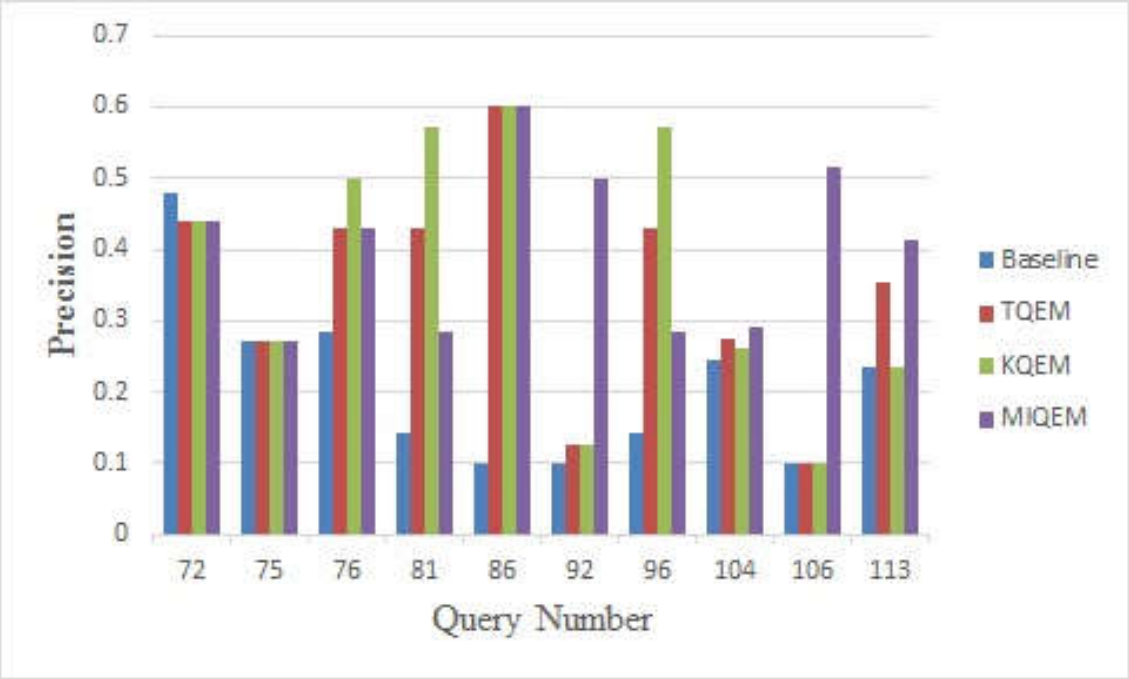

FIGURE 3.6: Per Query Precision Comparison for Set 4

expansion repeatedly to check in which iteration we obtain the same set of results. In other words, we try to find out the saturation point of our query expansion

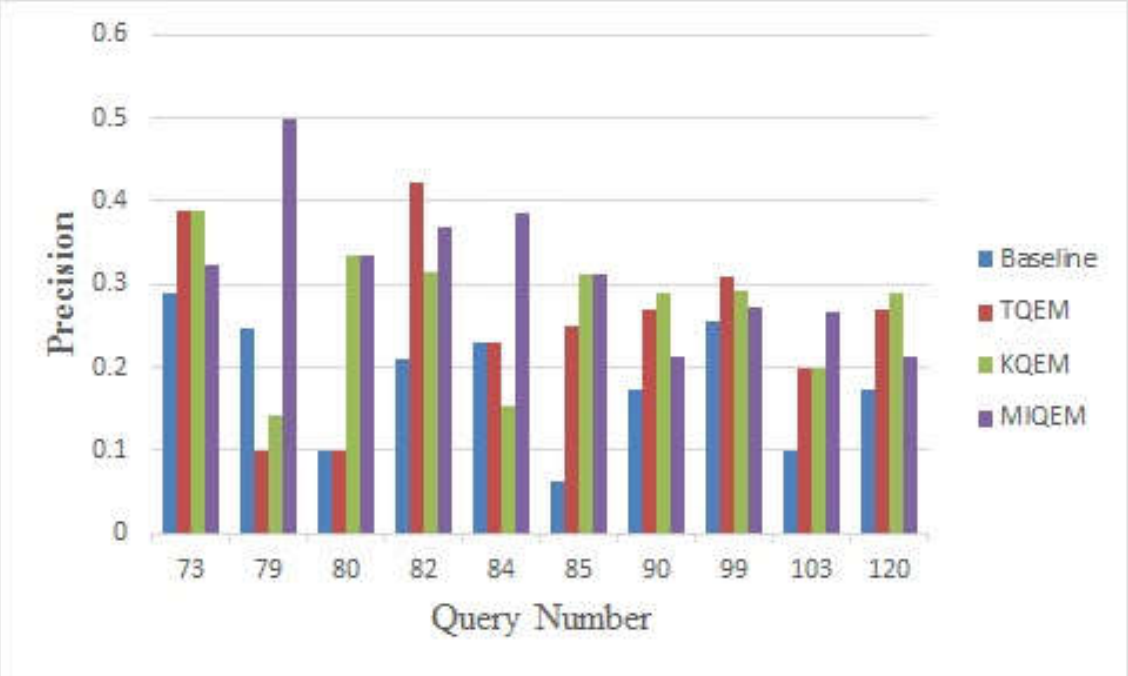

FIGURE 3.7: Per Query Precision Comparison for Set 5

Method	Iterations to query saturation
TQEM	7
KQEM	8
MIQEM	8

TABLE 3.5: Convergence Test

method. Table 3.5 lists the values obtained for the three approaches.

As can be seen, both KEA based query expansion and our approach converged on the 8th iteration, which underlines the fact that our algorithm performs no worse than KQEM. The above results validate our approach's efficiency compared to baseline and other two algorithms. Next, we explore whether our method improves image retrieval efficiency as well.

3.5.2 Comparison of image retrieval results

The left-hand side of Figure 3.8 shows few text queries and corresponding image queries. The retrieved results are shown on the right. A 'red' wrong (cross) symbol in the images denotes that it is not relevant with respect to the given query while a 'green' right (check) symbol denotes it as relevant. It is to be noted that we have shown only the top most retrieved image for each method (since each text document is associated with multiple images). Although there are no clear-cut winners here, going by the quality and narrative description, we see that MIQEM performs at par and even better in some cases compared to other approaches.

Let us consider the first query (text+image) *i.e.* "roller coaster wide shot". The query and the narrative in the topic set are as follows:

Topic No.: 94

Query: *roller coaster wide shot*

Narrative: *Photos of roller coasters that show a big part of its rails and loops are relevant. Photos that show only a close-up of its cars/wagons or a small detail of the roller coaster are not relevant.*

The text query alone is not adequate to judge the correctness or relevance of the images retrieved. Although when the narrative is considered we get considerably better precision. As can be seen, from the images retrieved, the baseline image is completely irrelevant. The one retrieved by TQEM is relatively correct but hard

FIGURE 3.8: Comparison of retrieved images

to identify given the particular narrative. Again, KQEM draws out a close shot of a roller coaster which is irrelevant according to the narrative. But, using MIQEM we get the best result among all the approaches. Almost similar reasoning holds for the rest of the images. Although in few cases, TQEM and/or KQEM performs at par with MIQEM, but MIQEM consistently gives correct results without fail.

So, we can safely claim the fact that judicious textual query reformulation can improve image search results. It also reaffirms the fact that considering narratives as part of the query can lead to better efficacy.

Concluding Remarks

In this section, we have presented the experimental results and analyzed them. In

a nutshell, our proposal seeks to exploit the hitherto untapped potential of text associated with images in conjunction to image retrieval. Through experiments we have established the following facts:

- Text query reformulation can positively affect image retrieval.
- Keyphrase extraction as a mode for query expansion in multimodal retrieval is an uncharted area.
- While narratives have been shown to be an essential component of a query for effective text retrieval, it remained unexplored for images.

All of the above facts were inferred from our study. Next, we proposed a new keyphrase extraction model to counter the fallacies of the two mentioned algorithms (TQEM and KQEM). Our model captures the semantic association between words which was realized in the form of a word graph embedded with Mutual Information. To measure the quality of the keyphrases (reflected through retrieval efficiency) that were extracted we performed experiments whose results are presented in Table 3.3. A noticeable improvement of 29.36% in retrieval efficiency (MAP) was registered over the baseline framework; while an increase of 6.20% and 9.01% was seen over TQEM and KQEM respectively. Even though this is fairly encouraging, addressing the aforementioned third fact further refined the text query thus rendering a significant boost of 56.81%, 17.94% and 17.63% over baseline, TQEM and KQEM, respectively. Such a substantial improvement was achieved without any compromise in tractability as our proposed algorithm was shown to saturate within at most eight steps. Barring

a few discrepancies, the increment in efficiency was uniform over all the topics as is suggested in the graphs and the fact that the MAP of MIQEM was statistically significant. While all the above facts were shown to be true for associated text, it was also reflected in the retrieved set of images as is explained in Section 3.5.2.

3.6 Discussion

Query reformulation using query expansion has been successfully applied to text retrieval. However, in this paper, we analyze and conclude that if query expansion is used judiciously, it can also lead to significant improvement in image retrieval. We combined the textual and visual features using optimal combination parameters learned by employing Fisher-LDA and the combined query was shown to improve image retrieval performance. To expand the textual query we used two approaches– one using the top scoring tf-idf terms from top-k documents and the other approach using KEA. We considered the relevant part of narratives as part of the baseline query since it has been shown to improve text retrieval performance. Moreover, we proposed a new model called Mutual Information based Query Expansion Model (MIQEM) that outperforms both the established algorithms significantly. Keyphrases were generated using a graph-based model enriched semantically by WordNet. A greedy approach to intelligently select keyphrases was also employed. The novelty of our proposed approach is that it is entirely unsupervised and effectively captures the semantic association between words, thus facilitating efficient

image and text retrieval. We believe that this is the first work that studies query expansion using keyphrase extraction in light of multimodal retrieval and also puts forward a new algorithm for the same. An exhaustive set of experiments performed on the ImageCLEF 2010-11 dataset revealed that our proposed scheme outperforms all the three approaches significantly, thus corroborating our claim.