

# Exploring Multimedia in a Keyword Space

João Magalhães<sup>1,2</sup>, Fabio Ciravegna<sup>2</sup> and Stefan Rüger<sup>1,3</sup>

<sup>1</sup>Department of Computing  
Imperial College London  
South Kensington Campus  
London SW7 2AZ, UK

<sup>2</sup>Department of  
Computer Science  
The University of Sheffield  
Sheffield S1 4DP, UK

<sup>3</sup>Knowledge Media Institute  
The Open University  
Walton Hall  
Milton Keynes MK7 6AA, UK

([j.magalhaes@imperial.ac.uk](mailto:j.magalhaes@imperial.ac.uk), [fabio@dcs.shef.ac.uk](mailto:fabio@dcs.shef.ac.uk), [s.rueger@open.ac.uk](mailto:s.rueger@open.ac.uk))

## ABSTRACT

We address the problem of searching multimedia by semantic similarity in a keyword space. In contrast to previous research we represent multimedia content by a vector of keywords instead of a vector of low-level features. This vector of keywords can be obtained through user manual annotations or computed by an automatic annotation algorithm. In this setting, we studied the influence of two aspects of the search by semantic similarity process: (1) accuracy of user keywords versus automatic keywords and (2) functions to compute semantic similarity between keyword vectors of two multimedia documents. We consider these two aspects to be crucial in the design of a keyword space that can exploit social-media information and can enrich applications such as Flickr and YouTube. Experiments were performed on an image and a video dataset with a large number of keywords, with different similarity functions and with two annotation methods. Surprisingly, we found that multimedia semantic similarity with automatic keywords performs as good as or better than 95% accurate user keywords.

## Categories and Subject Descriptors

H.3.1 [Content Analysis and Indexing]: Abstracting methods.

## General Terms

Algorithms, Measurement, Experimentation.

## Keywords

Search, multimedia, user keyword annotations, automatic keyword annotations, keyword spaces.

## 1. INTRODUCTION

In the classic multimedia search paradigm the user transforms some information need into a system query, and the system replies with the required information. Unlike text documents, multimedia documents do not explicitly contain symbols that could be used to express an information need. This problem has roots in two different aspects:

- **Richness of multimedia information:** visual and audio information can communicate a wide variety of messages, feelings and emotions; structure adds organization and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

XXX  
XXX

usability.

- **Expressiveness of the user query:** systems have always forced humans to describe their information need in some query language. However, not all information needs are easily expressed.

Multimedia systems are best at processing user queries represented by mathematical expressions, and not everyone has the same skills of expressing ideas, emotions and feelings in such a formal way. While in text retrieval we express our query in the format of the document (text), in multimedia systems this is more difficult. The user is not aware of the low-level representation of multimedia, e.g., colour, texture, shape features, pitch, volume or tones. These low-level feature spaces are ideal to find multimedia documents with similar colours, textures, shapes, etc, but are not adequate to find multimedia by semantic similarity. This scenario calls for a feature space capable of representing multimedia by its semantic content where semantic similarity is easily computed.

Figure 1 depicts the process of computing the semantic similarity  $\text{SemSim}(X, Y)$  between multimedia documents  $X$  and  $Y$ . A multimedia document  $X$  is transformed into the keyword space by the  $p : X \rightarrow X_w$  transformation. In this keyword space, a multimedia document  $X$  is represented by the vector  $X_w$  containing keyword scores. These scores indicate the confidence that a keyword is present in the document. Now, in this keyword space the distance  $\text{dist}_w(X_w, Y_w)$  between vectors  $X_w$  and  $Y_w$  is equivalent to the semantic dissimilarity<sup>1</sup> between documents  $X$  and  $Y$ , i.e.,  $1 / \text{SemSim}(X, Y)$ .

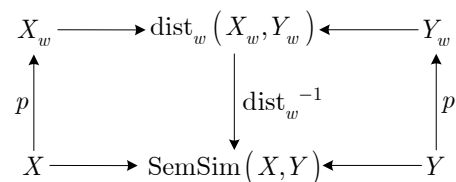


Figure 1 – Commutative diagram of the computation of semantic similarity between two multimedia documents.

Thus, in this paper we study the following aspects of the process:

- Manual versus automatic methods of transforming a multimedia document into the keyword space, i.e., the  $p : X \rightarrow X_w$  transformation.
- Functions to compute the semantic dissimilarity as the distance  $\text{dist}_w(X_w, Y_w)$  between two keyword vectors.

<sup>1</sup> Distance is equivalent to the inverse of similarity: large distances imply low similarity and small distances imply high similarity.

## 1.1 Keywords and Categories

It is in this context that we designed a framework to search multimedia by semantic similarity. As mentioned before, the keyword vectors can be obtained by manual or automatic methods, which we define formally as:

- **User keywords:** a user manually annotates multimedia with keywords representing meaningful concepts present in that multimedia content.
- **Automatic keywords:** an algorithm infers multimedia keywords and a corresponding confidence representing the probability that a given concept is present in that multimedia content.

Figure 2 illustrates some of the images on the Flickr web site annotated by a user with the keyword “London”. These images can be further grouped into themes concerning the same idea: (1) *London touristic attractions*; (2) *London’s river Thames*; (3) *London metro*; (4) *London modern art*. Each one of these themes is a row of images in Figure 2. Formally we define categories as:

- **Categories** are groups of multimedia documents whose content concern a common meaningful theme, i.e., documents in the same category are semantically similar.

The above definitions create two types of content annotations – at the document level (keywords) and at the group of documents level (categories). Because both keywords and categories describe the content of multimedia one would assume that categories can be inferred from keywords. For example, given a query image depicting the *Big Ben* the system would retrieve other images belonging to the same category, “*London touristic attractions*”, and not necessarily visually similar.

In our experimental framework, keywords and categories of

multimedia documents are defined by each collection ground truth: keywords are used to compute semantic similarity and categories are used to evaluate semantic similarity.

## 1.2 Contributions

The contributions of this paper can be summarized as:

- Proposing a high-level feature space to represent multimedia by a vector of keywords
- Comparing manual and automatic methods of computing keyword vectors and their influence in the accuracy of search by semantic similarity
- Analyzing the effectiveness of different similarity functions in the proposed keyword space
- Finding that multimedia semantic-similarity with automatically annotated keywords perform better than 95% accurate user keywords but is still below completely accurate user keywords

Section 3 exposes our idea of keyword space, followed by the implementation description of our semantic-multimedia search system. Section 4 describes how keyword vectors are computed with a naïve Bayes algorithm (automatic keywords) or are obtained from the ground truth labels of the collection (user keywords). We then apply noise to the user keywords to simulate different levels accuracy (100%, 95%, 90%, 85% and 80%). Once documents are represented in the keyword space the user can select or submit a query document (Section 5). A semantic similarity function is used to find documents from the same unknown category. Section 6 presents the tested similarity functions: cosine similarity, Minkowski distance, Kullback-Leibler divergence, and Jensen-Shannon divergence. Experiments were done on Corel Images and TRECVID2005 data.

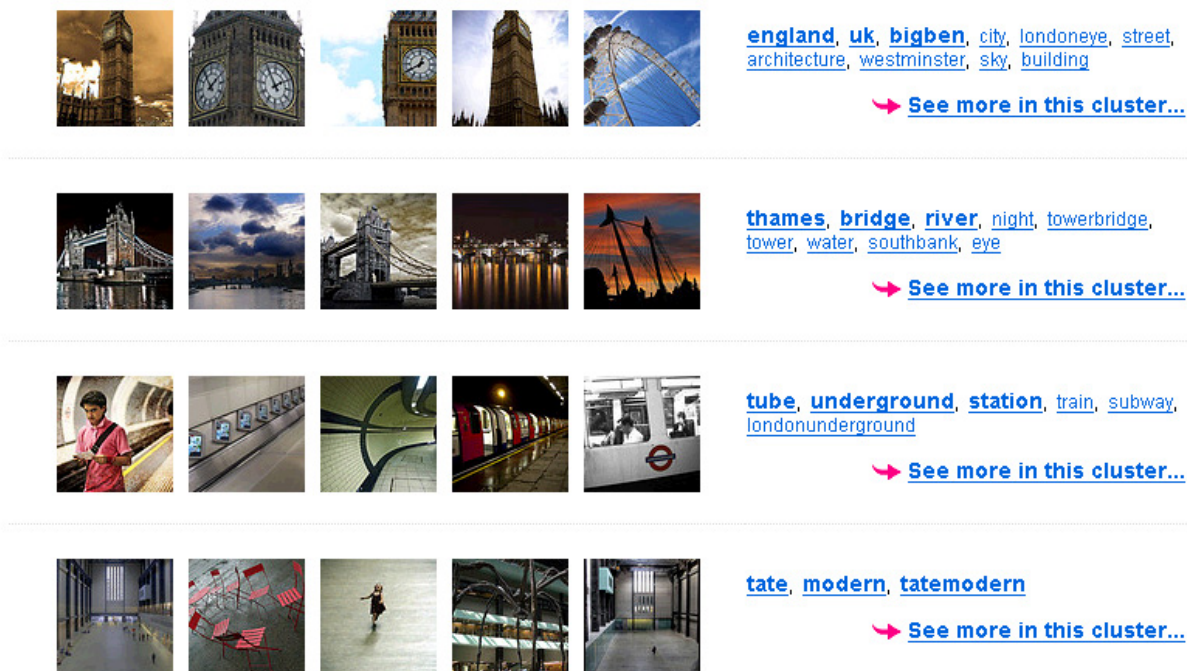


Figure 2 – Example of Flickr images annotated with the keyword London.

## 2. RELATED WORK

Searching multimedia by semantic similarity has been a problem in Computer Science for many years that has been tackled with different types of paradigms: some approaches have processed multimedia at feature level; others have exploited user interaction to refine the user query; while some have explored a combination of these paradigms, see [11] for a recent survey.

### Content based Systems

Early research in this area produced systems where users would provide a multimedia example of what they wanted to search for, e.g., QBIC [5]. This type of system works well when we want to search for images that are visually very similar to the query image. Going one step further, relevance feedback systems allow the user to compose a set of visual positive examples that are different instances of the same category. The system is still not aware of any keyword because it represents images by their low-level features: it relies on the user interaction to establish the link between multimedia low-level features and categories.

In most relevance feedback literature these links are initialised with some predefined set of weights and updated by an iterative algorithm based on the feedback from the user. Relevance feedback tries to iteratively specify the semantic characteristics of the intended results by adding semantically relevant examples and removing semantically non-relevant examples from the working model. Yang et al. [27] implemented a relevance feedback algorithm that works on a semantic space created from image clusters that are annotated with the most frequent keyword in that cluster. Semantic similarity is then computed between the examples and the image clusters. Lu et al [13] proposed a relevance feedback system that annotates images with the previously described heuristic and updates these semantic relations according to the user feedback. The semantic links between the documents and the keywords are heuristically updated or removed, if appropriate. Zhang and Chen [29] followed an active learning approach, and He et al [8] applied spectral methods to learn the semantic space from the user feedback. Other relevance feedback approaches have been proposed by Zhou and Huang [30], Chang et al. [2] and Wang and Li [26].

### Systems based on Automatic Keywords

While the previous approaches are not aware of multimedia keyword annotations, a new type of systems that explores this extra information has already flourished in the multimedia community. These systems allow the user to query with one or more keywords, which are used to search for multimedia content annotated with them. The initial annotation of multimedia content with keywords can be done manually (user keywords) or with some learning-algorithm and/or heuristic-rules (automatic keywords). Automatic algorithms are attractive as they only demand a low analysis cost when compared to the manual alternatives. Automatic image annotation algorithms are mostly based on some statistical modelling technique of image low-level features. Several techniques to model a keyword with different types of probability density distributions have been used: Feng and Manmatha [4] proposed a Bernoulli model with a vocabulary

of visual terms for each keyword, Yavlinsky et al. [28] deployed nonparametric density estimation, Carneiro and Vasconcelos [1] a semi-parametric density estimation. Automatic multimedia keyword annotation has also been an active area of research: Snoek et al. [24] explore temporal synchronization to combine the multi-modal patterns, Monay and Gatica-Perez explore dependencies across different media [17], while Magalhães and Rüger [15] developed a multimodal maximum entropy framework. The above methods extract features from the multimedia itself, but other, heuristic techniques rely on metadata attached to the multimedia: for example, Lu et al [13] analyse HTML text surrounding an image and assign the most relevant keywords to it. We follow Magalhães and Rüger's [15] approach to generate automatic keywords for its simplicity, scalability and relatively high precision.

The described family of techniques allows multimedia applications to work at a semantic level by extracting the keywords from both multimedia database documents and user query examples. This is already a big step from previous approaches towards more semantic applications but in some cases (if not most cases) it still might be too limiting.

### Keyword based Multimedia Similarity

The above types of approaches can produce good results but it puts an extra burden on the users who now have to describe their information need in terms of all possible instances and variations or express it with keywords. In both cases users may find limits in terms of their creativity, expressiveness or patience in reformulating their queries. Thus, in these cases users should be able to formulate a query with a *semantic example* of what they want to retrieve. Of course, the example is not semantic per se but the system will look at its annotations instead of its low-level characteristics (e.g. colour or texture). This means that the system will infer a vector of keywords from an image and use the keyword vector to search for images represented by similar vectors. Moving away from implementing query by semantic example as relevance feedback, Rasiwasia proposed a framework to compute the semantic similarity with a distance metric that ranks images according to the keywords of the current query [20, 21]. They start by computing keyword annotations with an algorithm based on a hierarchy of mixtures [1]. They then compute the semantic similarity as the Kullback-Leibler divergence. Tesic et al. [25] address the same problem but replace the Kullback-Leibler divergence as the semantic similarity by an SVM. The SVM views the provided examples as positive ones and samples negative examples randomly from regions of the feature space where the positive examples have low probability. Their results show good improvements over text-only search. Following these steps, Natsev et al. [19] explored the idea of using keyword-based query expansion to re-rank multimedia documents. They discuss several types of methods to expand the query with visual keywords. Another approach to query expansion in multimedia retrieval by Haubold et al. [6], uses lexical expansions of the queries. Semantic distances between words is also explored by Smeaton and Quigley [23] to perform query expansion. They show that this technique offers a substantial improvement over traditional IR techniques. Note that these approaches limit their methods to automatic keywords and do not

consider user keywords as we do on this paper.

Another interesting and related work is the study by Hauptman et al. [7] to identify the number of keywords that is required to fill the semantic gap. They use a topic search experiment to assess the number of required keywords to achieve a high precision retrieval system – their study suggests 3,000 keywords. This study associates the success of semantic-multimedia IR to a single factor (number of keywords) and leaves several different aspects aside such as similarity functions and different querying paradigms.

### 3. KEYWORD SPACES

Our goal is to devise a feature space capable of representing documents according to their semantics. In this setting we represent a multimedia document as

$$d = (d_f, d_w), \quad (1)$$

where  $d_f$  corresponds to the document low-level features and  $d_w$  to the document keyword annotations. These two representations form two distinct feature spaces, e.g., in the first case an image is represented by its texture or colour features, in the second case the same image is represented by its semantics in terms of keywords. A keyword space for searching multimedia by semantic similarity is defined by the following properties:

- **Vocabulary:** defines a lexicon

$$V = \{w_1, \dots, w_T\} \quad (2)$$

of  $T$  keywords used to annotate multimedia documents.

- **Multimedia keyword vectors:** a multimedia document  $d$  is represented by a vector

$$d_w = (d_1, \dots, d_T) \in [0,1]^T \quad (3)$$

of  $T$  keywords from the vocabulary  $V$ , where each component  $d_i$  corresponds to the likelihood that keyword  $w_i$  is present in document  $d$ .

- **Keyword vectors computation:** the keyword vector can be computed automatically or provided by a user. Section 4 discusses and compares both methods.
- **Semantic dissimilarity:** given a keyword space defined by the vocabulary  $V$ , we define semantic dissimilarity between two documents as

$$\text{dissim}_w : [0,1]^T \times [0,1]^T \rightarrow \mathbb{R}_0^+, \quad (4)$$

the function in the  $T$  dimensional space that returns the distance between two keyword vectors. Section 6 presents several distance functions.

Given the above definitions it is easy to see that for a query example  $q = (q_f, q_w)$  and a candidate document  $d = (d_f, d_w)$ , the semantic similarity between documents is computed as the inverse of the dissimilarity  $\text{dissim}_w(q_w, d_w)$  between the corresponding keyword vectors.

The lexicon of keywords corresponds to dimensions of the keyword space, allowing documents to be represented with

varying types of information according to the type of keyword (e.g., visual concepts, creation date, authorship). In searching semantic multimedia it is important that the semantic space accommodates as many keywords as possible to be sure that the user’s idea is represented in that space without losing any meaning. Thus, automatic systems that extract a limited number of keywords are less appropriate. This design requirement leads us to the research area of high-dimensional spaces. The structure of the space, i.e. the way keywords interact with each other, is defined by the distance function of that space. Distance functions are crucial in computing the semantic similarity between two multimedia documents – they define keyword independence and dependence. For example, the Euclidean distance considers keywords to be independent while graph-based metrics take keyword dependence into account.

In this paper we limit the lexicon of keywords to a set of  $T$  visual and multimodal concepts that are present in images and video clips.

### 4. KEYWORD VECTORS COMPUTATION

Data points in the keyword space correspond to a vector of keywords for each multimedia document – the way these vectors are computed is application dependent.

In some applications, keyword vectors  $d_w$  are extracted automatically from captions, Web page text, or low-level features. In this paper we implemented a machine learning algorithm  $p_A$  that computes keyword vectors from low-level features:

$$p_A : d \rightarrow d_f \rightarrow d_w \quad (5)$$

The machine learning algorithm supports a large number of keywords so that the keyword space can wrap the semantic understanding that the user gives to a document. This is in line with the requirement for highly expressive descriptions of multimedia, i.e., large number of keywords.

In other type of applications, keyword vectors  $d_w$  are extracted manually from the document content by a user  $p_U$ , i.e.,

$$p_U : d \rightarrow d_w. \quad (6)$$

The user inspects the document to verify the presence of a concept and annotates the document with that keyword if it is present. This introduces several ambiguities rooted on user’s understanding of the keywords and criterion to decide the keyword presence in the content.

The next two sections describe the implemented automatic keyword computation and the method to obtain user annotations.

#### 4.1 Automatic keyword annotations

In this section we describe how to estimate a probability function  $p$  that automatically computes the vector

$$d_w = (p(w_1 | d_f), \dots, p(w_T | d_f)), \quad (7)$$

of  $T$  keyword probabilities from the document’s low-level features  $d_f$ . Following the approach proposed by Magalhães and Ruger [14, 15], each keyword  $w_i$  is represented by a nave Bayes

model. The model allows expressing multimodal information as described in the following sections.

#### 4.1.1 Keyword Models

Keywords are modelled as text and visual data with a naïve Bayes classifier [15]. In our approach we look at each document as a unique low-level feature vector  $d_f = (f_1, \dots, f_M)$  of visual features (Section 4.1.2) and text terms (Section 4.1.3). The naïve Bayes classifier results from the direct application of Bayes law and independence assumptions between terms in a document:

$$p(w_j | d_f) = \frac{p(w_j) \prod_{i=1}^M p(f_i | w_j)}{\sum_{i=1}^T p(w_i) p(d_f = (f_1, \dots, f_M) | w_i)}. \quad (8)$$

A document can be represented as an event model of term presence or term count, leading to the choice of a binomial or multinomial model respectively [16]. We choose the multinomial distribution as the binomial distribution is too limiting given the probabilistic nature of visual and text features. Defining  $\bar{w}_j$  as the not-presence of keyword  $w_j$  we can formulate naïve Bayes in the log-odds space,

$$\log \frac{p(w_j | d_f)}{p(\bar{w}_j | d_f)} = \log \frac{p(w_j)}{p(\bar{w}_j)} + \sum_{i=1}^M p(f_i | d) \log \frac{p(f_i | w_j)}{p(f_i | \bar{w}_j)}, \quad (9)$$

which casts it as a linear model that avoids decision thresholds in annotation problems.

#### 4.1.2 Visual Data Processing

Three different low-level visual features are used in our implementation: marginal HSV distribution moments, a 12 dimensional colour feature that captures the histogram of 4 central moments of each colour component distribution; Gabor texture, a 16 dimensional texture feature that captures the frequency response (mean and variance) of a bank of filters at different scales and orientations; and Tamura texture, a 3 dimensional texture feature composed by measures of image coarseness, contrast and directionality. The images are tiled in 3 by 3 parts before extracting the low-level features. More details can be found in [15].

#### 4.1.3 Text Data Processing

Text feature spaces are high dimensional and sparse. To reduce the effect of these two characteristics, one needs to reduce the dimensionality of the feature space. We use mutual information to rank text terms according to their discriminative properties. See [15] for details.

### 4.2 User keyword annotations

Manual annotations done by real end-users are sometimes random, incomplete or incorrect for several reasons: the user might not be rigorous, users have different understanding of the same keyword, or it might be the result of spam annotations. A professional annotator produces better quality annotations – thus, in a real scenario one would expect to have user keywords with

accuracies below 100%. Following this reasoning, we use professional annotations to generate user keywords with different levels of accuracies:

- Generate completely accurate user keywords from the professional annotations of the collection;
- Given the professional annotations, replace 0%, 5%, 10%, 15% or 20% of the annotations by incorrect ones to simulate different levels of user keywords accuracies (this is done to both positive and negative annotations).

This procedure can also be seen as the simulation of an improved automatic annotation algorithm. Automatic annotation algorithms are not completely accurate and we do not foresee that a new algorithm will achieve a high-level of accuracy in the near future. Thus, this can also be seen as a forecast to what can be achieved with an improved automatic algorithm.

## 5. QUERYING THE KEYWORD SPACE

User queries can include keywords, multimedia examples, and arbitrary combinations of keywords and multimedia examples. The algorithm that parses the user request produces query vectors in the keyword space with the same characteristics as multimedia document vectors. For the objectives of this paper we only need to consider single example queries. Thus, for each query, the system analyses the submitted example and infers a keyword vector with the automatic algorithm

$$p_A : q \rightarrow q_f \rightarrow q_w, \quad (10)$$

or a user provides the keywords present in the example, i.e.,

$$p_U : q \rightarrow q_w. \quad (11)$$

Query examples are converted into keyword vectors with the methods already described in section 4.

## 6. KEYWORD VECTORS DISSIMILARITY

In this section we discuss the dissimilarity functions used to compute the semantic similarity between two multimedia documents. The dissimilarity functions presented in this section assume two different types of spaces: geometric spaces and probabilistic spaces. Thus, all dissimilarity functions assume that either the space is linear or that keywords are independent.

Note that with completely accurate user keywords we isolate the dissimilarity functions from the keyword annotation process. This way we can assess how much of the semantic similarity precision is due to the keyword vector computation method and how much is due to the dissimilarity functions.

The computation of similarity ranks for all documents in a database is an expensive process with linear complexity. Several methods exist to reduce this complexity, as for example sampling [10]. This topic is outside the scope of this paper as we are interested in finding methods to rank documents by semantic similarity with the maximum possible precision.

### 6.1 Geometric Space: Minkowski Distances

Similarity metrics in high-dimensional spaces are widely studied

in image retrieval: low-level features have a large number of dimensions and different characteristics. Several measures have been studied in this area such as Manhattan, Euclidean, and Mahalanobis. Howarth and Ruger [9] have shown that for visual features fractional dissimilarity measures (Minkowski distance with  $p < 1$ ) offer a good performance for several types of features. The Minkowski distance between the query example  $q$  and a database document  $d$  is defined as

$$\begin{aligned} \text{dissim}(q_w, d_w) &= L_a(q_w, d_w) \\ &= \left[ \sum_i |q_{w_i} - d_{w_i}|^a \right]^{1/a}, \end{aligned} \quad (12)$$

where the indices  $i$  concern the keyword  $w_i$ , and  $a$  is a free parameter  $a > 0$ . In this paper we use  $a = \{0.5, 1, 2\}$  as different distance measures. This measure requires the vectors  $q_w$  and  $d_w$  to be normalized.  $L_a$  is not a true metric for  $a < 1$  because it violates the triangle inequality; nevertheless it can offer useful dissimilarity values.

## 6.2 Geometric Space: Cosine Similarity

Since we work in high-dimensional spaces, in geometric terms one can define the independence between two vectors as the angle between them. This is the well known cosine similarity which becomes a dissimilarity by taking the difference to 1:

$$\text{dissim}(q_w, d_w) = 1 - \frac{q_w \cdot d_w}{\|q_w\| \cdot \|d_w\|} \quad (13)$$

Geometric correlation is one of the several possible ways to measure the independence of two variables. This metric is equivalent to the Pearson correlation coefficient in statistics; it measures the correlation between two random variables as the strength of their independence.

## 6.3 Probabilistic Space: KL Divergence

In statistics and information theory the Kullback-Leibler (KL) divergence is a measure of the difference between two probability distributions. It is the distance between a "true" distribution, (the query vector), to a "target" distribution, (the document vector). The KL divergence is expressed as

$$\begin{aligned} \text{dissim}(q_w, d_w) &= D_{KL}(q_w \parallel d_w) \\ &= \sum_i p(q_{w_i}) \log \frac{p(q_{w_i})}{p(d_{w_i})}. \end{aligned} \quad (14)$$

In information theory it can be interpreted as the expected extra message length due to using a code based on the candidate distribution (the document vector) compared to using a code based on the true distribution (the query vector). Note that KL divergence is not a true metric as it is not symmetric.

## 6.4 Probabilistic Space: JS Divergence

The Jensen-Shannon (JS) divergence is the symmetrised variant of the KL divergence and provides a true metric to compare two probability distributions:

$$\begin{aligned} \text{dissim}(q_w, d_w) &= D_{JS}(q_w, d_w) \\ &= \frac{1}{2} D_{KL} \left( q_w \parallel \frac{1}{2}(q_w + d_w) \right) \\ &\quad + \frac{1}{2} D_{KL} \left( d_w \parallel \frac{1}{2}(q_w + d_w) \right) \end{aligned} \quad (15)$$

An interesting characteristic of the JS divergence is that one can assign different weights to each distribution, [12]. This makes it particularly useful for decision problems where weights could be the prior probabilities.

## 7. EVALUATION

We will now describe the multimedia semantic similarity experiments done in an image and a video database and discuss the results of our evaluation.

### 7.1 Collections

Experiments were done in an image collection and a video clip collection. Both collections were split into training and test set, and each image/video clip is annotated with a set of keywords and categories. The manual annotations used to train keyword models and to simulate the user keywords were done by professional annotators. This means that real user keyword annotations will be less accurate than our completely accurate user keyword annotations.

#### Corel Images

This dataset was compiled by Duygulu et al. [3] from a set of COREL Stock Photo CDs. The dataset has some visually similar keywords (jet, plane, Boeing), and some keywords have a limited number examples (10 or less). The collection is split into a training set of 4,500 images and a test set of 500 images. Each image is annotated with one to five keywords from a vocabulary of 371 keywords. Only keywords with at least one image both in the test and training set were used, which reduces the size of the vocabulary to 260 keywords. The collection is already organized into 50 image categories, such as *rural France*, *Galapagos wildlife* and *nesting birds*, as illustrated in Figure 7. Despite its small size this collection has often been used in retrieval evaluation scenarios, e.g., [1, 4, 15, 20, 21, 28] and still serves well for comparisons to the state of the art.

#### TRECVID

To test semantic similarity on video data we used the TRECVID2005 data: since only the training set is completely labelled, we randomly split the English training videos into 23,709 training documents and 12,054 test documents. We considered each document to be a key-frame plus the ASR text within a window of 6 seconds around that key-frame. Key-frame keywords have two origins: the standard vocabulary of 39 keywords provided by NIST, plus the large-scale LS-COMM ontology of 400 keywords provided by Naphade et al. [18]. We trained the keyword models on the 39 keywords to form the keyword space and used 8 categories as relevance judgments (ground truth) for evaluation (*landscape*, *weapons*, *politics*, *vehicle*, *group*, *daytime outdoor*, *dancing* and *urban park*). The 8 categories were selected from the LS-COMM ontology as non overlapping keywords with

the other 39 keywords and had an enough number of examples. Note that because TRECVID categories are not annotated at the level of groups of documents we expect to have a lower accuracy in TRECVID when compared to Corel that have meaningful categories.

## 7.2 Experimental Design

We designed an experimental methodology that allows us to isolate the two aspects that we want to study: semantic dissimilarity functions and comparison between automatic keywords and user keywords. The experiment methodology was as follows:

- 1) Learn the naïve-Bayes model for each keyword on the training set of each collection (260 models for Corel and 39 for TRECVID). Note that we do not reuse the training set.
- 2) Submit a test document as a query example to rank the remaining test examples by semantic similarity
- 3) Compute keyword annotations for both documents and queries with the different algorithms:
  - a) Automatic keywords with the naïve-Bayes algorithm (260 keywords for Corel and 39 for TRECVID)
  - b) User keywords with different accuracies: 100%, 95%, 90%, 85% and 80% (260 keywords for Corel and 39 for TRECVID)
- 4) Rank documents by their semantic similarity to the query example according to a given dissimilarity function:
  - a) Cosine, Minkowski, KL and JS
- 5) The category of the query example is used as relevance judgment to evaluate the rank of documents
- 6) Repeat steps 2 to 5 for all test examples

## 7.3 Results and Discussion

Average precision is the used measure for comparing a ranked set of results to binary relevance judgements. The average precision of a particular query rank is the area under the precision-recall curve of that query. It is calculated by averaging the precision found at every relevant document. The advantage of using average precision as a performance measure is that it gives a greater weight to results retrieved early. Mean average precision (MAP) is defined as the mean of the average precisions of all queries.

### User keywords versus automatic keywords

The MAP upper bound of retrieval by semantic similarity is computed with completely accurate user keywords. This bound is specific for the set of keywords and categories. In the image collection the upper bound is 0.453; in the video-clip collection the upper bound is 0.103. Experiments in the image collection are presented in Table 1, Figure 3 and Figure 4. Table 1 displays the results of searches that rely on user keywords with a varying degree of annotation accuracy (see Section 4.2 for details). Figure 3 shows the MAP of ranking by similarity that uses the naïve Bayes classifier and various dissimilarity functions. For example, using these automated keywords and the Cosine similarity results in a MAP of 0.235, which is on par with or even slightly better than the corresponding one of the 95% correct user keyword (0.226). The same holds for other similarity functions. Figure 4

visually summarises Figure 3 and Table 1. Automatic keywords have roughly the same performance as 95% correct user keyword. As one would expect, they perform better than user keywords with smaller accuracy and worse than using the 100% ground truth. The encouraging news here is that we are comparing a simple automatic annotation algorithm to professional level annotations, and one would expect there to be scope for improvement. Experiments on the video-clip collection are presented on Table 2, Figure 5 and Figure 6. These results reinforce those of the one's on the image collection (the values are lower in this case because the rank length is now 12,054 instead of 500). The summary presented on Figure 6 also shows that the MAP of the automatic keywords 0.054 is on par with the corresponding MAP of the 95% correct user keywords (0.051).

Both experiments show a major change in retrieval precision when the user keyword accuracy goes from 100% to 95 % suggesting that the semantic similarity is highly sensitive to small changes in highly accurate annotations. Another interesting fact is the ranking stability for accuracies under 90%, which implies a more robust behaviour than in the 90%-100% range.

### Similarity functions

With completely accurate user keywords we isolated the similarity functions from the keyword annotation process. This way we could assess how much of the retrieval precision is due to the keyword annotation algorithm and how much is due to the semantic dissimilarity functions. The 100% accurate user keywords establish an upper bound on the MAP that is still below 0.50 for the image collection (and much lower in the video-clips collection). KL and cosine similarity functions were consistently better than the others. No single similarity function was much better than the others. It is also interesting to note that Minkowski distances appear to be the most robust to the probabilistic output of the naïve Bayes keywords extractor.

These observations point to two possible ways of improving semantic similarity distances: increase the number of keywords or investigate alternative similarity metrics. One solution would be the simple application of brute force, hoping to have comprehensive annotations with sufficiently good automatic keyword extractors. Another solution would suggest investigating similarity functions that incorporate keyword interdependencies and are robust to noisy keyword vectors, e.g., [22].

### User keywords and uncontrolled vocabularies

The use of an uncontrolled vocabulary can be a disadvantage for semantic similarity because it causes keyword matching problems at several levels. First, it is never possible to know the correct meaning that a user gives to a keyword (e.g., the keyword *football* means different sports for different cultures). Second, the user might dishonestly annotate a document with a popular keyword to attract other users. Third, users might have different criteria to annotate documents, e.g., some users might rigorously annotate all keywords while others might skip the obvious ones.

Because all these problems do not exist in automatic methods, we believe that the results of the proposed framework show that automatic methods have an important role in the semantic exploration of multimedia content.

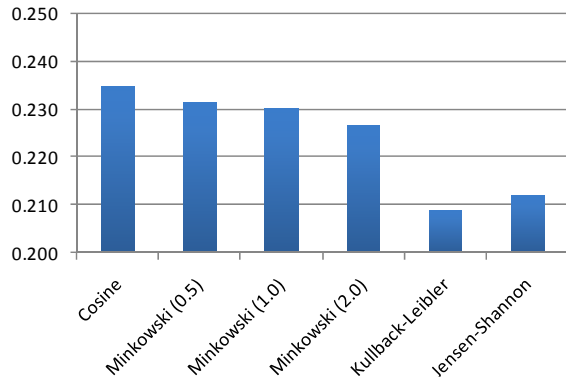


Figure 3 – MAP with automatic keywords (Corel Images).

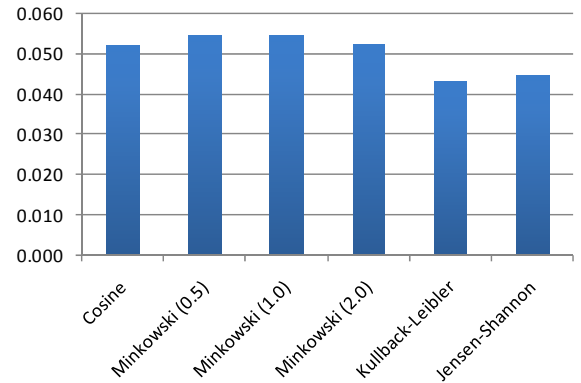


Figure 5 – MAP with automatic keywords (TRECVID).

User keywords accuracy	100%	95%	90%	85%	80%
Cosine	0.446	0.226	0.162	0.151	0.142
Minkowski (0.5)	0.438	0.181	0.152	0.146	0.141
Minkowski (1.0)	0.438	0.181	0.152	0.146	0.141
Minkowski (2.0)	0.438	0.181	0.152	0.146	0.141
Kullback-Leibler	0.453	0.224	0.160	0.150	0.143
Jensen-Shannon	0.436	0.226	0.162	0.151	0.146

Table 1 – MAP with different user keywords accuracies (Corel Images).

User keywords accuracy	100%	95%	90%	85%	80%
Cosine	0.098	0.051	0.053	0.047	0.040
Minkowski (0.5)	0.095	0.051	0.040	0.036	0.032
Minkowski (1.0)	0.095	0.051	0.040	0.036	0.032
Minkowski (2.0)	0.095	0.051	0.040	0.036	0.032
Kullback-Leibler	0.103	0.064	0.062	0.050	0.042
Jensen-Shannon	0.095	0.051	0.040	0.036	0.033

Table 2 – MAP with different user keywords accuracies (TRECVID).

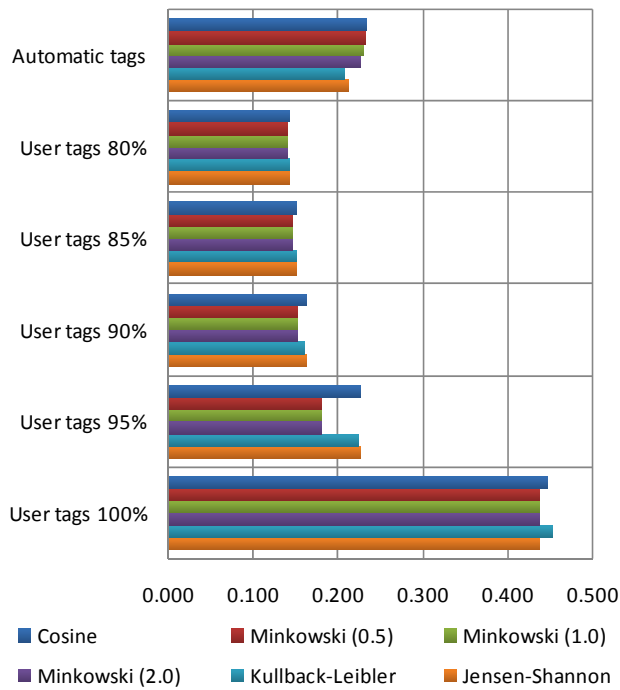


Figure 4 – MAP with different user keywords accuracies and automatic keywords (Corel Images).

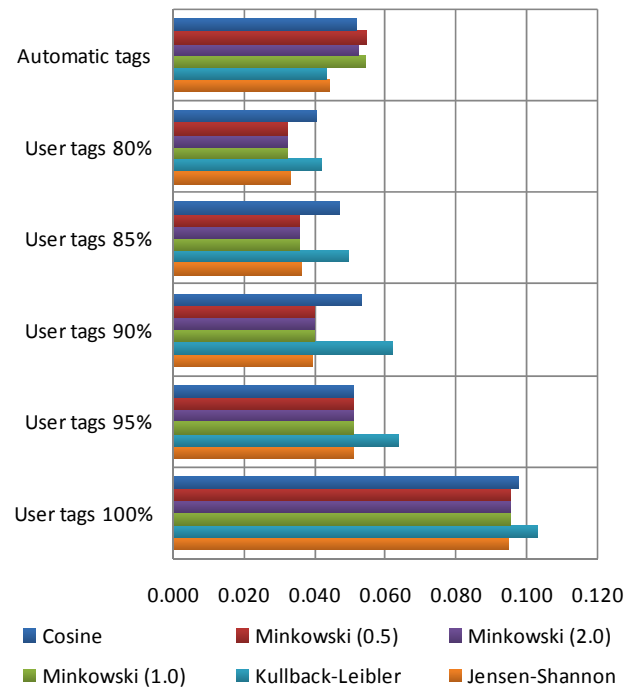


Figure 6 – MAP with different user keywords accuracies and automatic keywords (TRECVID).












Image category: rural France	Image category: Galapagos wildlife	Image category: nesting birds
 <p data-bbox="272 474 440 499">barn;buildings;field;</p>	 <p data-bbox="764 474 854 499">crab;rocks;</p>	 <p data-bbox="1203 474 1328 499">birds;nest;tree;</p>
 <p data-bbox="277 741 435 766">buildings;field;tree</p>	 <p data-bbox="727 741 889 766">giant;rocks;tortoise</p>	 <p data-bbox="1170 741 1357 766">barn;birds;nest;wood;</p>
 <p data-bbox="272 1008 440 1031">castle;hills;sky;stone</p>	 <p data-bbox="764 1008 854 1031">birds;nest</p>	 <p data-bbox="1195 1008 1333 1031">birds;nest;water;</p>

Figure 7 – Example of image keyword-categories relationships with different complexities.

### Semantic relevance

Assessing the user information needs from an example is always a difficult task. In this paper we assumed that the information need can be represented by a set of keywords extracted from the example and evaluated with categories. The commonly used measures of precision and recall use a binary relevance model to identify relevant and non relevant documents. However, in the current scenario the relevance of a document is difficult to measure because semantic relevance is gradual. The problem is even more complex for several reasons, e.g., for a particular query an image with one matching keyword might be more meaningful than an image with two matching keywords; an image might belong to different categories but only one category is the required one. Figure 7 illustrates some of these situations: the images on the *rural-France* category can illustrate other unknown categories, e.g. *old-buildings*; images on the categories *nesting birds* and *Galapagos wildlife* overlap semantically in many aspects. Thus, evaluating semantics should be done at a finer scale: rank correlation measures might be a better way of evaluating semantic similarity.

## 8. CONCLUSIONS

This paper addressed the problem of exploring multimedia by semantic similarity in a keyword space. Managing multimedia by their keywords and categories is a complex task involving a long chain of information processing algorithms. We presented

experiments to analyze two aspects of the process: (1) the influence of the accuracy of user keyword annotations versus automatic keyword annotation algorithms and (2) functions to compute semantic similarity. Our evaluation allows us to draw the following conclusions regarding multimedia semantic similarity:

- Automatic keyword annotations perform better than 95% accurate user keywords but is still below completely accurate user keywords
- User keywords show that similarity is highly sensitive to extremely accurate keyword annotations (the difference between 95% and 100% correct keywords)
- User keywords show that similarity is robust to errors for averagely accurate keyword annotations (range between 80% and 95%)
- All considered distance functions perform similarly.

We believe that the results of the proposed framework show that automatic methods have an important role in the semantic exploration of multimedia content.

These conclusions together with the experiments results shed some light on the problem of semantically comparing two multimedia documents. These experiments also suggest two hypotheses: rank correlation evaluation would provide better measures to design a semantic similarity function; and graph based metrics might perform better by exploring keyword dependencies.

## ACKNOWLEDGMENTS

This work was partially funded by the X-Media and Pharos projects sponsored by the Commission of the European Communities as part of the Information Society Technologies programme under grant numbers IST-FP6-026978 and IST-FP6-45035.

## 9. REFERENCES

- [1] G. Carneiro and N. Vasconcelos, "Formulating semantic image annotation as a supervised learning problem," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.
- [2] S.-F. Chang, W. Chen, and H. Sundaram, "Semantic visual templates: linking visual features to semantics," in *Int'l Conference on Image Processing*, Chicago, IL, USA, 1998.
- [3] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *European Conf. on Computer Vision*, Copenhagen, Denmark, 2002, pp. 97-112.
- [4] S. L. Feng, V. Lavrenko, and R. Manmatha, "Multiple Bernoulli relevance models for image and video annotation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Cambridge, UK, 2004, pp. 1002-1009.
- [5] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *IEEE Computer*, vol. 28, pp. 23-32, Sep 1995.
- [6] A. Haubold, A. Natsev, and M. Naphade, "Semantic multimedia retrieval using lexical query expansion and model-based re-ranking," in *IEEE Int'l Conference on Multimedia and Expo Toronto*, Canada, 2006.
- [7] A. Hauptmann, R. Yan, and W.-H. Lin, "How many high-level concepts will fill the semantic gap in news video retrieval?," in *ACM Conference on Image and Video Retrieval*, Amsterdam, The Netherlands, 2007.
- [8] X. He, O. King, W.-Y. Ma, M. Li, and H.-J. Zhang, "Learning a semantic space from user's relevance feedback for image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 39-48, Jan 2003.
- [9] P. Howarth and S. Rüger, "Fractional distance measures for content-based image retrieval," in *European Conference on Information Retrieval*, Santiago de Compostela, Spain, 2005.
- [10] P. Howarth and S. Rüger, "Trading accuracy for speed," in *Int'l Conf. on Image and Video Retrieval Singapore*, 2005.
- [11] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 2, pp. 1-19, February 2006.
- [12] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inform. Theory*, vol. 37, pp. 145-151, January 1991.
- [13] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang, "A unified framework for semantics and feature based relevance feedback in image retrieval systems," in *ACM Multimedia*, Los Angeles, CA, USA, 2000, pp. 31-37.
- [14] J. Magalhães and S. Rüger, "High-dimensional visual vocabularies for image retrieval," in *ACM SIGIR Conf. on research and development in information retrieval*, Amsterdam, The Netherlands, 2007.
- [15] J. Magalhães and S. Rüger, "Information-theoretic semantic multimedia indexing," in *ACM Conference on Image and Video Retrieval*, Amsterdam, The Netherlands, 2007.
- [16] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *AAAI Workshop on Learning for Text Categorization*, 1998.
- [17] F. Monay and D. Gatica-Perez, "Modeling Semantic Aspects for Cross-Media Image Indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1802-1817, October 2007.
- [18] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *IEEE Multimedia Magazine*, vol. 13, pp. 86-91, 2006.
- [19] A. Natsev, A. Haubold, J. Tesic, L. Xie, and R. Yan, "Semantic concept-based query expansion and re-ranking for multimedia retrieval," in *ACM Conference on Multimedia Augsburg*, Germany, 2007.
- [20] N. Rasiwasia, P. Moreno, and N. Vasconcelos, "Bridging the gap: Query by semantic example," *IEEE Transactions on Multimedia*, vol. 9, pp. 923-938, August 2007.
- [21] N. Rasiwasia, N. Vasconcelos, and P. Moreno, "Query by semantic example," in *CIVR*, Phoenix, AZ, USA, 2006.
- [22] N. Sebe, M. S. Lew, and D. P. Huijsmans, "Toward Improved Ranking Metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1132-1143, October 2000.
- [23] A. F. Smeaton and I. Quigley, "Experiments on using semantic distances between words in image caption retrieval," in *ACM SIGIR Conf. on research and development in information retrieval*, Zurich, Switzerland, 1996.
- [24] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. M. Smeulders, "The semantic pathfinder: using an authoring metaphor for generic multimedia indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1678-1689, October 2006.
- [25] J. Tesic, A. Natsev, and J. R. Smith, "Cluster-based data modelling for semantic video search," in *ACM Conference on Image and Video Retrieval Amsterdam*, The Netherlands, 2007.
- [26] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 947-963, September 2001.
- [27] C. Yang, M. Dong, and F. Fotouhi, "Semantic feedback for interactive image retrieval," in *Int'l Multimedia Modelling Conference*, Singapore, 2005.
- [28] A. Yavlinsky, E. Schofield, and S. Rüger, "Automated image annotation using global features and robust nonparametric density estimation," in *Int'l Conf. on Image and Video Retrieval*, Singapore, 2005.
- [29] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," *IEEE Transactions on Multimedia*, vol. 4, pp. 260-268, Jun 2002.
- [30] X. S. Zhou and T. S. Huang, "Unifying keywords and visual contents in image retrieval," *IEEE Multimedia*, vol. 9, pp. 23-33, Apr-Jun 2002.