Exploiting External Knowledge to Improve Video Retrieval

David Vallet[1,2] Iván Cantador[1,2] Joemon M. Jose[1]

[1] Department of Computing Science
University of Glasgow
Glasgow, Scotland, United Kingdom, G12 8QQ
{dvallet, cantador, jj}@dcs.gla.ac.uk

[2] Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
28049, Madrid, Spain
{david.vallet, ivan.cantador}@uam.es

ABSTRACT

Most video retrieval systems are multimodal, commonly relying on textual information, low- and high-level semantic features extracted from query visual examples. In this work, we study the impact of exploiting different knowledge sources in order to automatically retrieve query visual examples relevant to a video retrieval task. Our hypothesis is that the exploitation of external knowledge sources can help on the identification of query semantics as well as on improving the understanding of video contents.

We propose a set of techniques to automatically obtain additional query visual examples from different external knowledge sources, such as DBPedia, Flickr and Google Images, which have different coverage and structure characteristics. The proposed strategies attempt to exploit the semantics underlying the above knowledge sources to reduce the ambiguity of the query, and to focus the scope of the image searches in the repositories.

We assess and compare the quality of the images obtained from the different external knowledge sources when used as input of a number of video retrieval tasks. We also study how much they complement manually provided sets of examples, such as those given by TRECVid tasks.

Based on our experimental results, we report which external knowledge source is more likely to be suitable for the evaluated retrieval tasks. Results also demonstrate that the use of external knowledge can be a good complement to manually provided examples and, when lacking of visual examples provided by a user, our proposed approaches can retrieve visual examples to improve the user's query.

Categories and Subject Descriptors

H.3.3 Information Search and Retrieval – *retrieval models, search process*. H.5.1 Multimedia Information Systems – *video*.

General Terms

Algorithms, Experimentation.

Keywords

Video retrieval, images, query by example, semantics.

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

In recent years, there has been an unprecedented increase on the creation and consumption of video digital information. With an ever growing amount of video content available on the Web for casual and professional users, there is a need to facilitate video retrieval on large collections. Studies in the area of video retrieval have been focusing on this problem from the mid 90's [1][7]. The challenge is exacerbated by the fact that, unlike text documents, video contents have a multimodal nature – videos contain information such as speech, text, visual and spatiotemporal metadata – which has to be taken into consideration by a video retrieval system.

Current Web video retrieval systems (e.g., YouTube[1]) rely solely on user textual metadata in order to provide a text-based search interface. In this context, the problem of these community-based online video systems is twofold: users are often reticent to provide manual annotations, and the quality of these annotations is in many cases questionable [8].

The exploitation of low-level features as well as image retrieval has proved to be one important approach for video retrieval. However, low-level features suffer from the semantic gap problem, where low-level information often does not match with the real semantics associated to a video [10]. In the literature, the semantic gap has been tried to be alleviated by the extraction of high-level features, which are expressed in terms of semantic concepts belonging to thesauri. One representative example is the Large Scale Ontology for Multimedia (LSCOM) [13]. These thesauri model concepts such as "road" or "person", and normally use concept detectors based on training data [18], which are difficult to scale on large and dynamic collections, and thus have impeded their integration into Web services. Rather than exploiting these features in isolation, the most successful approaches have been based on the combination of low- and high-level features in a single video retrieval process [9][17], suggesting that all features play an important role when building a video retrieval system.

Low-level features are still an important source of information, and are widely used by video retrieval systems. Differently from high-level features, low-level features do not rely on trained classifiers, and are more easily scalable. Their main drawback is that a set of visual examples relevant to the task at hand must be provided to the retrieval engine. For instance, TRECVid (TREC Video Retrieval Evaluation) [16] is an annual initiative to evaluate video retrieval systems that provides a set of image and video shot examples for each of its video retrieval topics (tasks), with which we can simulate a scenario where a user provides a set of visual

[1] http://www.youtube.com/

examples. The video retrieval systems evaluated in this workshop often analyse the above visual examples in order to obtain low-level features that are used by their retrieval processes.

In a real scenario, however, these approaches might be a burden to the user, as there is a requirement to find and provide the system with the visual examples. To address this problem, some systems attempt to ease the user's effort by using an interactive retrieval approach [17], or by providing a way in which the user can sketch a visual example [5]. Other systems attempt to retrieve these visual examples automatically, without interaction from the user. This can be done by using other available features such as text or high-level concepts in a pseudo-relevance fashion [1]. The systems execute an image retrieval process, generally using a textual representation of the user's information needs, which returns a set of documents that can be considered relevant for the user's text query, and thus can be used as visual examples of the low-level feature extraction processes. This approach requires the collection to contain additional features, which are not always available, especially on Web collections.

Another automatic approach consists of accessing to external collections, such as Flickr[2], to retrieve relevant visual examples, based on their associated metadata. This strategy has already been explored by previous video retrieval systems (e.g., [6][15]), especially for the automatic search tasks of TRECVid. However, on video retrieval there is not a formal study to this methodology yet, nor a study on the impact of the used external collection. Furthermore, to date, these approaches do not seem to have fully exploited the metadata structures of the available collections.

In this paper, we aim to analyse the role of these external Knowledge Sources (KS) for providing relevant visual examples, without extra effort from the user, unlike manually providing examples, or without techniques that require collections indexed with additional features, such as pseudo-relevance feedback. We study three different external KSs: 1) a highly structured, collaborative built KS, with a low semantic coverage in multimedia sources, such as DBPedia[3]; 2) a folksonomy-based KS, freely defined by users, with a greater coverage, such as Flickr; and 3) a low structured KS such as Google Images[4], but with a high coverage (the Web).

We present different methodologies which exploit the above KSs in order to automatically retrieve visual examples relevant to a given retrieval task. We suppose this task does have a textual representation (i.e., in form of a textual query), as this is the most common information need representation for video retrieval, and the most familiar representation for users.

Focusing on TRECVid 2007 and TRECVid 2008 collections, we use the automatically retrieved visual examples as a source of low-level features to be used in the video retrieval process. We are thus able to analyse the role of each external KS on providing relevant visual examples, and compare them with the provided TRECVid visual examples, which can be considered as a set of visual examples manually provided by a user. Furthermore, we analyse whether these external resources do provide comparable results to those from examples manually provided by a user, and whether they can be a good complement to the manual visual examples.

Our research can be summarised in the following hypotheses:

H1. External knowledge sources available online contain visual examples which can complement or mitigate the lack of visual examples provided manually by a user.

H2. The underlying semantics available in some external KS can be exploited to retrieve even more meaningful visual examples.

The following sections are organised as follows. Section 2 provides background information on the use of external KS for video retrieval, relevant to this work. Section 3 presents our framework for external KS exploitation, and the different visual example retrieval methodologies adopted for the three selected KSs: DBPedia, Flickr and Google Images. Section 4 describes the video retrieval strategies that make use of the visual examples obtained by our framework. Sections 5 and 6 expose experimental setup and results, respectively. Finally, Section 7 concludes with some discussion and future research lines.

2. BACKGROUND

The exploitation of external knowledge is a relatively new research direction in multimedia information retrieval. External knowledge can be a set of collaborative annotations, an additional media collection from Web services, or a domain related formal knowledge base, e.g. WordNet [12] or a specific ontology. In this work, we exploit external KSs as image retrieval services, in order to collect relevant visual examples to be used in video retrieval tasks. Most image retrieval services accept textual keywords as input. Here, we present a technique that makes use of a more structured KS, DBPedia, which allows building semantic queries.

The applications of techniques that exploit external KSs can be roughly categorised into two groups: 1) obtaining visual examples relevant to a specific task; and 2) providing extra ground truth for relevance estimation. Using an external KS is a direct solution to alleviate the problem of insufficient visual examples, used either for training or retrieval purposes. Some systems submitted to TRECVid have exploited KSs such as Google, Flickr or YouTube in order to retrieve further visual examples. The results of these systems, however, do not clarify nor allow an in-depth analysis of the effect that the KS has in the retrieval process. Snoek et al. [18] collect Web images to train the video search system MediaMill. Olivares et al. [15] spread manual annotations across Flickr's image collection in order to develop effective concept detectors for image and video retrieval. Both works show that the diversity of images in such repositories makes the approaches not as effective as expected. This is mainly because current image retrieval services are solely based on textual features such as caption or user annotations. Even so, Olivares et al. are able to filter the metadata existing in Flickr to enhance a text-based image retrieval engine, proving thus how external knowledge can be successfully exploited to improve text-based searches in image retrieval. In this work, we propose to exploit more the semantics and structure present on KSs, focusing on video retrieval.

[2] http://www.flickr.com/

[3] http://dbpedia.org/

[4] http://images.google.com/

In the TRECVid 2006 workshop, Liu et al. [14] use an extra collection of ABC news as additional ground truth to re-rank video documents. They argue that a real video collection may offer a strong ground truth, and expel semantic ambiguity around the manipulated TRECVid collection. Nevertheless, although the usage of an extra collection as a reference seems to be plausible, it results in additional computation cost, and makes the retrieval performance dependent on the quality of the used collection. This leads to the problem of quality prediction on the query as well as a document collection [6]. In this work, we also analyse if external knowledge can enhance, or even substitute, a set of visual examples manually provided by a hypothetical user.

3. EXPLOITATION OF EXTERNAL KNOWLEDGE TO OBTAIN RELEVANT VISUAL EXAMPLES

A content-based video retrieval system aims at supporting the user to retrieve a sequence of videos whose contents should satisfy a number of personal interests, needs or requirements. The success of searching such videos depends, among other things, on the fact of formulating a clear and meaningful query.

Since content-based information retrieval systems deal with the search of visual objects, it seems natural to conduct that search using examples of such objects. In fact, many content-based information retrieval systems follow a query-by-example (QbE) approach. Nonetheless, in this context, the user is required to pick one or more video examples beforehand. When the user does not exactly know which video shots he is looking for, or the dimension of the search space is very large, as is often the case, this approach might not be feasible.

Facing these problems, strategies based on query-by-text (QbT) allow the user using keywords to express high level semantic concepts that should appear in the video sequences to retrieve, and are difficult to describe through QbE. Thus, queries are formulated in the form "retrieve videos which contain [keywords]", and videos have to be annotated with semantic concepts corresponding to all possible keywords the user might introduce.

Textual annotation of videos represents then a new battlefield. Videos are difficult to be annotated automatically, and users could manually perform this task. However, since it is a really tedious labour, it cannot be done reliable by a single person. As has been demonstrated recently in Web 2.0 applications, such as YouTube, Yahoo Videos[5], Metacafe[6], Revver[7] or Daily Motion[8], the community can play an important role to annotate on-line videos, and multimedia content retrieval based on collaborative social tagging is being extremely successful.

Hence, it turns out that both QbE and QbT strategies are needed. In our proposal, the user provides a textual query to describe the semantic concepts that should appear in the videos he would be interested in. Instead of looking for these concepts analysing directly the video content, we propose to explore external collaboratively annotated image repositories in order to collect a set of images potentially relevant for the user's query. Then, applying a QbE strategy, these images are compared with keyframes of the videos available in the system, and those videos with the keyframes most "similar" to the above images are finally retrieved.

Following this approach, we combine the benefits of QbE, QbT and social tagging techniques. First, we take advantage of the high descriptive power of querying by example. Second, we provide the user with an easy way to express his multimedia information needs. Finally, we mitigate the problem of lacking of video annotations making use of the community tagging and categorisation efforts.

In the next subsections, we describe the architecture of our proposal, and the external KSs with which we have empirically tested the proposal.

3.1 Architecture

We study the exploitation of two public available collaborative KSs with large image collections: DBPedia and Flickr.

DBPedia is a Semantic Web gateway which collects data from Wikipedia[9] encyclopaedia. Wikipedia articles consist mostly of free text, but also contain different types of structured information, such as info-boxes, categories, images and links to external Web pages. Much of this structured information is indexed by DBPedia, which serves as a basis for enabling sophisticated queries against Wikipedia content. As of November 2008, the DBpedia dataset describes more than 2.6 million "things", including people, places, companies, etc. These descriptions are completed with more than 600K related images. Given a certain concept, we propose to obtain the images associated to its correspondent DBPedia entity. Making use of the DBPedia semantic relations of this entity with other entities, we will also obtain images of related concepts.

On the other hand, Flickr is an image hosting website which allows users to share and annotate (tag) personal photographs. In this case, the meta-information of the images is given by the social tags introduced by the photograph owners. As November 2008, Flickr claims to host more than 3 billion images. Given a certain concept, we propose to match it to one or more social tags in order to retrieve images related to that concept. The set of tags within individual user and item profiles, together with tag popularity, will be used to rank the matched tags and retrieved images.

The quality of the images obtained from DBPedia and Flickr for our video retrieval proposal will be compared against the quality of those images that are retrieved by a less structured KS: Google Images, a well-known QbT-based image search service. The details of this comparison are described in Section 5.

The general architecture of the proposal is depicted in Figure 1. The user provides the system a natural language query describing the contents of the videos he wants to retrieve, and the system returns a ranked list of videos, in which the ranking scores are similarity values between the video contents and the given input query. In our experiments, the user input is simulated through a subset of natural languages queries extracted from TRECVid collections.

[5] http://video.yahoo.com/

[6] http://www.metacafe.com/

[7] http://revver.com/

[8] http://www.dailymotion.com/

[9] http://www.wikipedia.org/

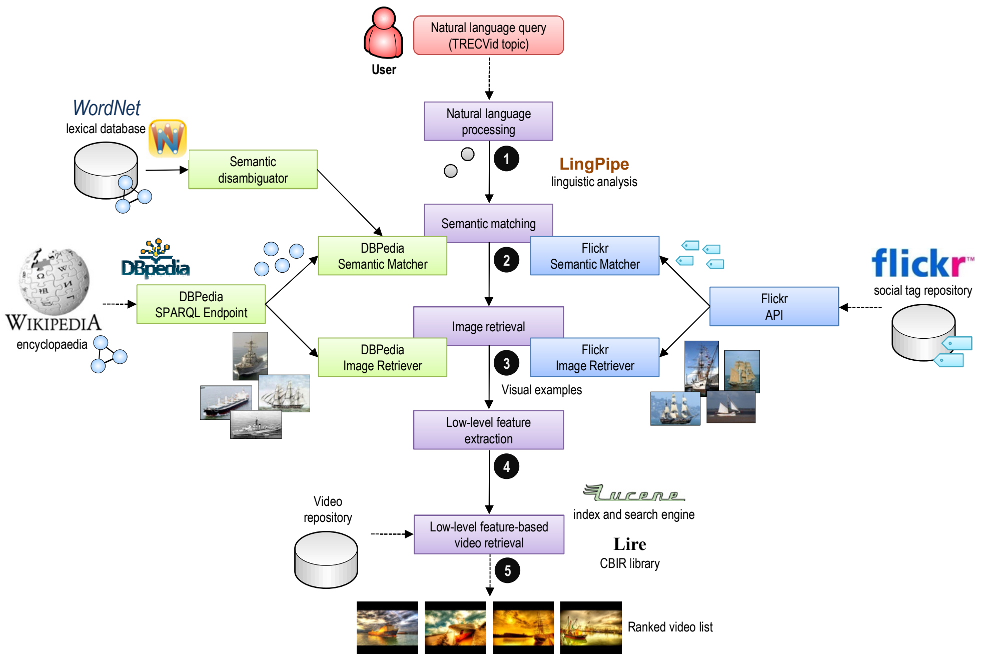

Figure 1. General architecture of the proposal

The whole video retrieval process is divided in five steps, numbered in the figure.

1. The user provides the system with a text query describing the video content he is interested in. A Natural Language Processing (NLP) module[10] extracts the noun entities appearing in the query. Because it is not the focus of this work, we do not explain this process herein, and just mention that it is able to accurately identify common and proper, single and compound nouns. From now, we will refer to these identified noun entities as “concepts” of the query.
2. The extracted concepts are passed to a module which matches them with semantic entities (i.e., DBPedia entities and Flicker social tags) belonging to the external KSs.
3. Once the semantic entities are identified, several heuristics, which depend on the KSs, are performed by an image retriever in order to return ranked lists of images which are annotated with the above entities.
4. The gathered images are analysed, and some of their low-level features (e.g., colour, shape, texture) are obtained.
5. Following a QbE strategy, the low-level features of the images are compared with those of the video keyframes, already indexed. Based on these comparisons, and following a ranking combination technique, the system finally assigns ranking scores to the videos in order to filter and sort them for the user.

In the rest of this section, we explain in more detail the semantic matching and image retrieval processes (steps 2 and 3) for each of the used KSs. Steps 4 and 5, low-level feature extractor and low-level feature video retrieval, are described in Section 4.

3.2 Knowledge sources

For each of the KSs explored in this work, in order to obtain sets of images related to a list of concepts (expressed in the form of text keywords), several tasks have to be carried out.

First, each query concept label has to be matched with semantic entities existing in the KS. Note that a concept label can be part of more than one keyword. In general, a concept label does not appear directly as part of the unique names of the entities. Depending on the KS, an ad-hoc morphologic processing of the concept label will have to be done. Second, once the concept labels have been morphologically modified, and matched with entities, the semantic properties provided by the KS have to be exploited to enhance the retrieved entity set. Finally, the images which are annotated with the final entities have to be ranked. Again, a customised ranking strategy has to be performed.

The subsequent sections describe how we have accomplished the previous three tasks for DBPedia, Flickr and Google Images KSs.

3.2.1 DBPedia

DBPedia is an ontology which stores structured information obtained from Wikipedia, and, making use of Semantic Web technologies, links that information with other KSs, such as OpenCyc, WordNet or DBLP, among others[11].

Its structure basically consists of three elements: classes, instances and properties. Classes can be understood as categories in which the information is organised (e.g., “City”), instances are specific individuals that belong to the classes (e.g., “New York city” as an instance of “City”), and properties are attributes of the classes/instances whose values can be literal values (strings, numbers, etc.) or links to other classes/instances. For instance, “hasPopulationOf” could be a numeric property defined in the class “City”, and whose value would be different for each city.

[10] http://alias-i.com/lingpipe/

[11] http://wiki.dbpedia.org/Interlinking

There are usually two properties that relate classes and instances: “subClassOf” and “type” (instanceOf). “A subClassOf B” means *class A is a subcategory of class B*, and “i type A” means *instance i is an instance of class A*.

Each of the above elements is uniquely identified on the Internet by an URI (Uniform Resource Identifier). In DBPedia, for example, *http://dbpedia.org/resource/New_York_City* is the URI of the instance “New York city”, *http://dbpedia.org/resource/ Category:Cities_in_New_York* is the URI of the class “Cities in New York”, and *http://www.w3.org/2004/02/skos/core#subject* is the URI of a property equivalent to the “type” property.

In our proposal, the concepts of the query have to be matched with entities (classes or instances) of DBPedia. For this purpose, each concept label has to be found in one or more DBPedia URIs. However, an exact matching is often not possible, and some morphologic transformations in the concept label have to be conducted. More specifically, we create several forms of the concept label, and attempt to find them as subparts of the URIs. In order to match DBPedia’s URI format, we change the concept label blank spaces to underscores “_”. We also apply the following transformation in order, stopping whenever a match is found:

- All the characters of the keyword are converted to lower case.
- All the characters of the keyword are converted to upper case.
- All the characters of the keyword except the first one, which is maintained as upper case, are converted to lower case.
- If the keyword is a compound noun, all the characters except the first characters of the keyword tokens are converted to lower case (e.g., “new york” is transformed into “New York”).

This process is done with the singular and plural forms of the keyword (when they do exist). If no entities are found, we apply the same mechanism, but instead of looking for the keyword in the URIs, we search for it in the values of the property *http://dbpedia.org/property/redirect*, which is used to link equivalent entities (e.g., “NYC” redirects to “New York City”). Moreover, if there are no matches yet, we repeat the process with the property *http://www.w3.org/2000/01/rdf-schema#label*, whose values are alternative forms of the entity name (e.g., “Nueva York” is the Spanish label for “New York”).

In some cases, several DBPedia entity URIs are retrieved for a single concept. In order to choose one of them, we make use of WordNet [12]. WordNet is a lexical database and thesaurus that groups English words into sets of cognitive synonyms called “synsets”, provides definitions of terms, and models various semantic relations between synsets.

The local names of the URIs are split into their tokens. For instance, let us suppose that the concept of interest is “orange”, and the local names of the matched URIs are orange_fruit, orange_brand, and orange_river. Their corresponding token lists would be {orange, fruit}, {orange, brand}, and {orange, river} respectively. Then, we look for the concept in WordNet and get its synsets. We also tokenise the synset definitions. For instance, the first WordNet synset of “orange” would be transformed into the token list {yellow, orange, fruit, tree, citrus}. Following the synset order given by WordNet, we compute the intersection between the entity and the synset token lists. When we obtain a non empty intersection (without taking into account the token which is the concept itself), we stop and take the intersected entity as the most likely suitable for the concept. In the previous example, the list {orange, fruit} intersects with {yellow, orange, fruit, tree, citrus} by the token fruit, so the selected DBPedia entity for “orange” is orange_fruit.

It is important to note that we do not perform any disambiguation strategy at query level. It could happen that the real meaning of a concept in a query is not the most likely one. The concept “orange” might refer to the river, and not to the fruit. This issue has not been addressed in this paper, and constitutes an interesting future research line.

Once we have selected a DBPedia entity, we obtain its corresponding image in Wikipedia. DBPedia uses the property *http://xmlns.com/foaf/0.1/depiction* to provide the URL of such image. The problem then is that only one image is associated to a given entity. In order to obtain more related images, we exploit the semantic relations available in DBPedia. We explain our approach with an example, shown in Figure 2.

Let us suppose that the user has entered the query “find shots of a *boat*”. Let us focus on the concept “boat”, and assume that DBPedia contains information about the concept “boat” in the way depicted in the figure. The entity “Boat” has an image in Wikipedia (linked by the property *foaf:depiction*), and belongs to the category (class) “Boats”, as declared by the property *dbpedia:category* (equivalent to the general property “subClassOf”). To obtain more images of boats, we extract all the subcategories of the class “Boat” following the property *skos:broader*, which can be understood as the inverse relation of “subClassOf”. In the example, we find the subcategories “Racing boat”, “Ferry” and “Sailboat”. Again, following the property *foaf:depiction*, but this time starting from the found subcategories, we retrieve more images. This process is iteratively performed for the subsequent categories in the DBPedia class hierarchy. It is also carried out taking into account the “instanceOf” relations, and might be done based on other arbitrary relations, but this issue is not addressed in this work.

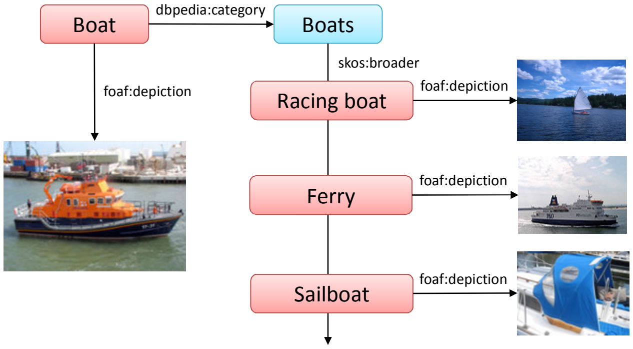

Figure 2. Relations of concept “boat” (extracted from DBpedia)

With the entities related to “Boat”, the query has been extended in such a way that the system takes into consideration different types of boats, thus returning also images that contain racing boats, ferries or sailboats, even though they were not explicitly annotated with the concept “boat”.

It is important to notice here that using DBPedia, the concepts of a query have to be searched separately. Thus, there is no possibility of querying for several concepts that should appear together in a single image, like e.g. in “find shots of a *boat* sailing into the *sunrise*”. This situation does not occur in other KSs such as Flickr or Google Images.

After the images related to a concept are obtained, we can assign them a ranking score, by exploiting the semantic structure available in DBPedia. The score of an image should be based on the proximity of the concept from which the image was retrieved to the initial matched concept. In the previous example, the image retrieved from the entity "Boat" should have a higher score than the images retrieved from the entities "Racing boat", "Ferry" and "Sailboat". It also has to be based on the generality/ambiguity of the associated concept. That is, a concept that belongs to a few classes should have a greater score than those that belong to many classes, since the former is more likely to be more specific (i.e., less ambiguous).

To address these issues, we propose the following heuristic. First, we get all the categories of the corresponding entity. For example, New_York_City belongs to the categories Cities_in_New_York, Former_capitals_of_the_United_States, etc. Then, we split the concept and category names into noun tokens, like "york", "city", "capital", "state", etc. Finally, we count the number of occurrences of entity name tokens with category name tokens, and compute the score as:

$$score(img) = \frac{\#tokenOccurences}{\#categoryTokens} \cdot log_2\left(1 + \frac{1}{\#categories}\right) \in [0,1]$$

As can be observed, the influence of the number of categories in the score value is less than the influence of the token occurrences. Through empirical experimentation we checked this is a convenient consideration.

3.2.2 *Flickr*

Flickr is one of the most popular photo sharing services on the Internet. Registered users are allowed to upload their photos into the system, and manually annotate them with keywords (tags). They can also include a title and a description for each photo.

Flickr does have much more images than DBPedia. However, in many cases, they show personal experiences or artistic works of the users, and do not focus on showing specific objects for definition purposes as DBPedia does. Facing this inherent "noise", our goal is to investigate whether exploiting the meta-information underlying the social tags we are able to identify which images are relevant to a given keyword-based query.

Flickr offers two image search modalities. The first one, called "search by text" from now on, looks for matches between the query keywords and the terms appearing in the personal text descriptions of the images. On the other hand, the second one, called "search by tag" from now on, looks for matches between the query keywords and the social tags of the images. The experiments explained in Section 5 explore both alternatives.

In our approach, and in opposite to DBPedia approach, given a textual query, instead of searching images related to several concepts separately, a query launched to Flickr search service will contain all its identified semantic concepts. Because of that, no processing of singular and plural forms is performed. Thus, for example, the query "find shots of several *boats* sailing into the *sunrise*" is transformed into "boats sunrise", and not into the two independent queries concepts "boat" and "sunrise".

The transformed query is provided to Flickr search service. Then, the first M retrieved images are ranked based on their social tags as follows. We assume the images annotated with the most popular tags should be assigned high scores. Popular tags represent a shared vocabulary among users, and are likely to refer to general (commonly accepted) concepts. The score given to an image is:

$$\mathrm{score(img)} = \frac{\sum_{t \in tags(img): n_t \geq avg(n_T)} \mathrm{n_t}}{\sum_{\mathrm{t \in T}} \mathrm{n_t}}$$

where T is the set of tags which are annotations of the M retrieved images, $\mathrm{n_t}$ is the number of times the tag t appears in the annotations of the retrieved images, and $\mathrm{avg(n_T)}$ is the average number of times the tags of T appear in the image annotations.

In this approach, we do not conduct any disambiguation strategy. We assume the fact of having a set of keywords together in a single query enables the semantic disambiguation of the involved concepts.

3.2.3 *Google Images*

Similarly to Flickr, Google Images allows the user to query for several concepts at the same time. The information is not structured, and the retrieval of the images is based on a matching of the query keywords with the terms surrounding the images in the Web pages where they are placed.

We have not developed any strategy to treat the queries, nor reorder the results obtained from Google. We thus consider this service as a highly unstructured KS. Our hypothesis is exploiting the semantic structures available in DBPedia and Flickr, the latter as a result of the social collaborative tagging, we are able to retrieve visual examples of higher quality. Although in Section 5 we compare the quality of each KS, our understanding is that the presented approaches are complementary.

We have to mention that we also made experiments with Yahoo! Images[12] service, but because the results obtained with it were quite worse than Google's, we decided to not include it in this paper.

4. RETRIEVAL STRATEGIES

In this section, we briefly present the two video retrieval strategies analysed in this work. These strategies follow a QbE approach by using as input the visual examples obtained from the different external KS exploitation techniques presented in the previous section.

The first retrieval strategy uses the visual examples to search across the video collection. This strategy will allow us to evaluate the quality of the obtained visual examples, and thus assess the different approaches to external KS exploitation. The second retrieval strategy complements a set of manually provided visual examples with the automatically retrieved visual examples. The analysis of this strategy will give us clues on the possibility of using external KS to complement visual examples manually provided by users.

4.1 External Knowledge Retrieval

This strategy launches low-level feature-based retrieval processes using the visual examples obtained from an external KS. For each visual example, the results of those processes are aggregated into a single result list. More details, specific to the experimental setup, can be found in Section 5.1.

One of the faced problems was to set up a limit on the number visual examples to use, as some external KSs can retrieve hundreds or even thousands of visual examples for a given query. Using our development collection, we set a maximum number of 50 visual examples to be used in the video retrieval process. This limit value was also applied to the second retrieval strategy.

[12] http://images.search.yahoo.com/

4.2 Improving Manual Visual Examples with External Knowledge

This strategy exploits visual examples collected from an external KS to re-rank the results obtained using a set of manually provided visual examples. The idea of this approach is to give a higher importance to the user's visual examples, and use external visual examples as a complement for the former. This approach might be appropriate when the manually provided examples are no sufficient for the query, or no suitable for expansion.

The retrieval strategy is as follows. Given a set D of video documents to rank, and a set V of visual examples provided by the user, we launch a retrieval process which scores each document $d \in D$ with a normalised score value $s(d,V) \in [0,1]$. We then create a final result set $R(V) = \{d_1, d_2, \dots, d_N\}$ containing the top N ranked documents. In a second stage, a retrieval process is performed using the set of visual examples EV obtained from the external KS. This retrieval process, however, is limited to the result set returned by the manual examples retrieval step, and provides a normalised score value $s(d,EV) \in [0,1]$ if $d \epsilon R(V)$, 0 otherwise. This value is finally used to re-rank the set of documents returned in the first retrieval step, using the following combined score value:

$$s(d,V,EV) = \lambda \cdot s(d,EV) + \ (1-\lambda) \cdot s(d,V)$$

where $\lambda \in [0,1]$ indicates the combination weight.

Using our development collection, we analysed the impact of using different values of λ and *N*. As we did not observe any significant impact from the optimisation of these values, we decided to leave them at neutral values of $N = 10{,}000$ and $\lambda = 0.5$. Although it is not the focus of this work, we tried a number of basic multimodal fusion techniques (see [11] for an overview) to dynamically set the λ parameter, but we did not find any significant improvement. In future work we will explore the application of more elaborated techniques, which could help in the combination of the different external KSs.

5. EXPERIMENTS

The goal of our evaluation is to analyse the impact of our external KS exploitation techniques over the two proposed retrieval strategies. We choose to perform a collection-driven experimentation, which facilitates us obtaining comparable results for our different retrieval strategies. More formally, our evaluation seeks to address the following research questions:

Q1. Can we exploit the semantics underlying external KSs in order to improve the retrieval of high-quality visual examples?

Q2. Which external KS is better for the retrieval of visual examples?

Q3. What is the effect of complementing user provided visual examples with examples obtained from external KSs in a video retrieval system?

5.1 Experimental Setup

In order to evaluate our retrieval strategies, we use TRECVid 2007 and TRECVid 2008 collections. TRECVid "is an international benchmarking activity to encourage research in video information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organisations interested in comparing their results" [16]. TRECVid 2007 collection provides over 100 hours of video, and 24 topic (task) descriptions, along with their respective relevance judgements. TRECVid 2008 collection provides over 200 hours of video, and 48 topic descriptions. As development collection, we use TRECVid 2006 collection, which provides 24 evaluations topics.

Each topic is represented as a short text query (e.g., "find shots of a door being opened"), and a set of visual examples: two external images and two example videos. It is worth noting that these example videos belong to the development part of the TRECVid collection, and come from the same content provider. This plays in favour of the available visual examples as they have the same content format, increasing the probability of matching relevant results. Each topic text query is used as input of our external KS exploitation techniques. The visual examples provided on each topic are considered as a hypothetical set of visual examples a user could have provided to the retrieval system.

We implemented a retrieval system based solely on low-level features as follows. The system uses the shot boundaries provided on the TRECVid collections, and extracts one keyframe per second. This leaves over 350K keyframes on TRECVid 2007 collection, and over 700K keyframes on TRECVid 2008 collection. For each keyframe, the system extracts six low-level features: colour layout, colour histogram, edge histogram, Tamura texture feature histogram, colour and edge directivity descriptor (CEDD) [3], and fuzzy colour and texture histogram (FCTH) [4]. As a query-time fusion methodology for the different low-level features and visual examples, we use the method described in [19]. We discard the use of the ASR output and high level concepts, as it would not drive any additional conclusions to our experiments; we assume these features are complementary to the low-level features obtained from the visual examples.

We used the development collection to tune up our retrieval system. With the obtained system setting we achieved comparable results to those obtained by the low-level features runs of the systems presented in TRECVid 2006, 2007 and 2008. Our system's performance values were around the median of their overall performance values.

6. RESULTS AND ANALYSIS

This section presents and analyses the performance results for the two presented retrieval strategies. As performance measure, we use the inferred Average Precision (infAP) metric which, in this study, is equivalent to the Mean Average Precision (MAP) metric. The infAP has been adopted as a system performance comparison measure on TRECVid [20].

6.1 External Knowledge Retrieval

In order to address research question Q1, we measure the performance values when applying the different proposed KS exploitation techniques presented in Section 3. Table 1 shows the performance results of the external knowledge retrieval strategy (explained in Section 4.1) using the above techniques, for the TRECVid 2007 and 2008 collections, together with the average for all topics. The evaluated external KSs are the following: DBPedia; Flickr with "search by text" (Flickr Text); Flickr with "search by tag" (Flickr Tag); and, for comparison purposes, the results obtained with the manual visual examples provided on the TRECVid collection (Manual Example). The results given in the table do not consider the ranking heuristics presented for the DBPedia and Flickr KSs (see Sections 3.2.1 and 3.2.2, respectively).

Strategy Topics	DBPedia	Flickr Text	Flickr Tag	Google	Manual Example
2007	0.0076	0.0155	0.0134	0.0130	0.0180
2008	0.0064	0.0039	0.0017	0.0039	0.0139

Table 1. Inferred Average Precision (infAP) performance values for the different external KS retrieval strategies

Table 2 shows the performance values when using the ranking heuristics proposed for DBPedia and Flickr. The goal of these heuristics is to retrieve more suitable visual examples. The results are encouraging, as they show that exploiting the semantics available in these KSs leads to sensible performance improvements compared to the basic approach results presented in Table 1. The DBPedia ranking heuristic leads to a 34.25% and 17.21% performance increase on TRECVid 2007 and 2008 collections, respectively, which is statistically significant (Wilcoxon test, $p < 0.05$). The heuristics applied on the Flickr KS result on a ~65% performance increase over the 2008 collection, which is statistically significant but a decrease on the 2007 collection, which is not statistically significant. Regarding research question Q1, the increase of performance with the ranking approaches presented for DBpedia and Flickr suggest that more KSs with more formal semantic structures allow the implementation of ranking heuristics to provide higher quality visual examples.

To address research question Q2, and based on the results given in Table 2, we conduct a comparison of our different approaches. DBPedia and Flickr "search by text" seem to have in overall the highest performance. Exploiting the title and text description of visual examples on Flickr seems to be a good complement to the tag metadata. The results also show that even a low structured KS such as Google can be exploited with acceptable results (although these are lower than the ones obtained with DBPedia and Flickr "search by text").

The external knowledge retrieval strategy has a lower performance than using manually provided examples. However, the results indicate that, in absence of such examples, the strategy can be a good alternative. The user does not have to provide manual examples to enrich the query, since it is done automatically by analysing a textual query. Moreover, there is a decrease in the performance values of Flickr and Google retrieval strategies in TRECVid 2008 with respect to TRECVid 2007. This decay is not proportional to the performance decay of the manual examples results. DBPedia KS exploitation approach, however, seems to give more consistent results.

Strategy Topics	DBPedia	Flickr Text	Flickr Tag	Google	Manual Example
2007	0.0102	0.0123	0.0127	0.0130	0.0180
2008	0.0075	0.0063	0.0029	0.0039	0.0139
Δ 2007	+34.25%	-20.86%	-5.55%		
Δ 2008	+17.21%	+61.30%	+68.86%		

Table 2: Final InfAP performance values including the ranking heuristics applicable to DBPedia and Flickr KSs. The last two rows show the improvement of performance over the basic approaches presented in Table 1. Starred results indicate statistically significant results

Table A, included as an appendix of this paper, presents the performance results per each evaluated topic. On a per-topic analysis, DBPedia KS seems to be more prone to not finding relevant visual examples, as on 13.9% of the topics no visual examples were found. This is due to the fact that DBPedia is a far more restricted KS for visual examples than Flickr or Google. One of our concerns was about topics with more than one concept, e.g., "find shots of one or more people at a table or desk, with a computer visible" as when exploiting DBPedia, we are only able to find visual examples that are related to a single concept ("table", "desk" and "computer"), whereas with KSs such as Flickr and Google, we can retrieve visual examples related to all concepts. The results show no evidence against the one-concept-per-image approach of DBPedia, compared to the multiple-concepts-per-image approach of Flickr and Google. To investigate further on this, we also evaluated the one-concept-per-image approach on Flickr and Google, and results were also similar to our original approaches. All of our approaches had low performance results on topics emphasising on semantics, e.g., "find shots of a road taken from a moving vehicle, looking to the side", as these are harder to analyse and exploit. More topic examples are presented in Table B, at the end of the paper.

6.2 External Knowledge Applied to Manual Query Examples

To address research question Q3, we measure the performance of the retrieval strategy explained in Section 4.2, which complements a set of manually provided visual examples with examples provided by our external KS exploitation techniques. Table 3 shows the performance results for the above retrieval strategy. In addition to infAP, we show P@15 values, as the retrieval strategy is based on a re-ranking approach, and thus is more inclined to improve precision, rather than recall values. The last two rows of the table show the overall performance variation compared with the retrieval performance using only manual examples (Manual Example). Starred values indicate a statistical significance (paired t-test, $p<0.03$). Values in bold indicate the best performing approach for each collection and metric.

DBPedia, which was the best KS for the external knowledge strategy (Section 6.1), results overall on the best performing values in terms of P@15, compared to the manual example approach. The improvement on precision is notable, achieving around a 40% increase over the manual examples on the two test collections. This improvement is statistically significant when compared not only to the manual examples, but also to the other external KSs. The DBPedia approach also has the highest values on other P@N values, reaching similar improvements on P@5 (overall 36.74%*) and P@10 (overall 41.54%*). The other external KSs approaches have a more moderate improvement of precision over the manual examples, although improvement is still statistically significant. This suggests that DBPedia would be able to provide more diverse visual examples that are better for discerning the relevant documents retrieved using the manual visual examples. As expected, infAP values do not vary significantly, although it is worth noting that this retrieval strategy does not affect negatively the overall performance of the results based on manual examples, and at the same time improves sensibly the precision values.

Regarding research question Q3, we can conclude that the obtained increments in performance are significant, and demonstrate that external KSs can be successfully exploited to complement visual examples provided by a user.

Strategy Metrics	DBPedia	Flickr Text	Flickr Tag	Google	Manual Example
2007 infAP	0.0175	0.0176	0.0174	**0.0181**	0.0180
2007 P@15	**0.0861**	0.0778	0.0722	0.0750	0.0611
2008 infAP	**0.0144**	0.0137	0.0133	0.0132	0.0139
2008 P@15	**0.0570***	0.0486	0.0486	0.0528*	0.0417
Δ infAP 2007	-2.89%	-2.48%	-3.37%	+0.51%	
Δ P@15 2007	**+40.90%***	+27.27%*	+18.18%*	+22.72%*	
Δ infAP 2008	+3.58%	-1.11%	-3.91%	-5.06%	
Δ P@15 2008	**+36.68%***	+16.69%*	+16.68%*	+26.68%*	

Table 3: Performance values for the external KS approaches applied to manual examples

7. DISCUSSION AND CONCLUSSIONS

With the goal of obtaining high quality visual examples for a content-based video retrieval system, and based on an input text query, this work presents several techniques that automatically retrieve visual examples by exploiting the semantics of three external Knowledge Sources (KS): DBPedia, Flickr and Google Images.

We stated two hypotheses: 1) visual examples obtained from external KSs can complement or mitigate the lack of visual examples manually provided by users; and 2) the exploitation of the semantics available in such KSs can help to better discern which of their visual examples are more relevant to the input text query.

To validate these hypotheses we introduced and evaluated two retrieval strategies that make use of the external visual examples. The first strategy uses these examples alone, while the second strategy uses them to complement a set of visual examples provided by users.

The first hypothesis was validated as follows. Although the conducted evaluations showed that using only external visual examples provides lower performance than using manual visual examples, the performance values obtained with the former arise the 55% of the performance values obtained with the latter, indicating that in absence of manually selected examples, our retrieval strategies might represent good alternatives. Moreover, we think that releasing the user from the burden of providing relevant visual examples is a great benefit. In addition, we showed that the visual examples from external KSs successfully complement manual examples, achieving improvements of around 40% for precision measures on the two test collections.

Regarding our second hypothesis, our evaluation results showed that the exploitation of the semantic structure available on some of the studied external KSs improves the quality of the retrieved visual examples. We also showed that the more structured the KS is, the more benefit can be obtained from its exploitation.

After analysing the performance of our external KS exploitation techniques, our intuition was that these approaches can complement each other. We tested some basic ranking aggregation techniques, but we did not obtain significant results. This suggests that integrating multiple external KSs might require more sophisticated techniques, such as e.g. those related to query performance prediction [6].

We have used a video retrieval framework to evaluate the external KS exploitation techniques. They could also be incorporated into a content-based image retrieval system. A proper evaluation would have to be conducted to determine this. In this context, a comparison with state of the art approaches, such as [15], could be possible.

Acknowledgements. This research was supported by the European Commission under contract FP6-027122-SALERO and by the Spanish Ministry of Science and Education (TIN2008-06566-C04-02).

8. REFERENCES

[1] Amir, A., Berg, M., Permuter, H. 2005. Mutual Relevance Feedback for Multimodal Query Formulation in Video Retrieval. In *MIR'05*, ACM Press, 17–24.

[2] Chang, S. F., Chen, W., Meng, H., Sundaram, H., Zhong, D. 1998. A Fully Automated Content Based Video Search Engine Supporting Spatio-Temporal Queries. In *IEEE Transaction on Circuits and Systems for Video Technology*, 8 (5), 602–615.

[3] Chatzichristofis, S. Boutalis, Y. 2008. CEDD: Color and Edge Directivity Descriptor. 2008. A Compact Descriptor for Image Indexing and Retrieval. In *ICVS'08*, Springer, 312–322.

[4] Chatzichristofis, S. A., Boutalis, Y. S. 2008. FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval. In *WIAMIS'08,* IEEE*,* 191–196.

[5] Collomosse, J. P., Mcneill, G., Watts, L. 2008. Free-hand Sketch Grouping for Video Retrieval. In *ICPR'08,* IEEE*,* 1–4.

[6] Hauff, C., Hiemstra, D., de Jong, F. 2008. A Survey of Pre-retrieval Query Performance Predictors. In *CIKM'08*, ACM Press, 1419–1420.

[7] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P. 1995. Query by Image and Video Content: The QBIC System. In *Computer*, 28 (9), 23–32.

[8] Guy, M., Tonkin, E. 2006. Folksonomies: Tidying up tags? In *D-Lib Magazine*, 12(1).

[9] Hauptmann, A. G., Christel, M. G. 2004. Successful Approaches in the TREC Video Retrieval Evaluations. In *MULTIMEDIA'04*, ACM Press, 668–675.

[10] Jaimes, A., Christel, M., Gilles, S., Ramesh, S., Ma, W. Y. 2004. Multimedia Information Retrieval: What is it, and Why isn't anyone using it? In *MIR'04*, ACM Press, 3–8.

[11] Kennedy, L., Chang, S.-F., Natsev, A. 2008. Query-adaptive fusion for multimodal search. In IEEE, 96 (4), 567–588.

[12] Miller, G. A. 1995. WordNet: A Lexical Database for English. New Horizons in Commercial and Industrial Artificial Intelligence. In *Communications of the ACM*, 38(11), 39–41.

[13] Naphade, M., Smith, J. R., Tesic, J., Chang, J. S., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J. 2006. Large-Scale Ontology for Multimedia. In *IEEE MultiMedia* 13(3), 86-91.

[14] Liu, Z., Gibbon, D., Zavesky, E., Shahraray, B., Haffner, P. 2006. AT&T Research at TRECVid 2006. In *TRECVid'06*.

[15] Olivares, X., Ciaramita, M., van Zwol, R. 2008. Boosting Image Retrieval through Aggregating Search Results based on Visual Annotations. In *MM'08*, ACM Press, 189–198.

[16] Smeaton, A. F., Over, P., Kraaij, W. 2006. Evaluation Campaigns and TRECVid. In *MIR'06*, ACM Press, 321–330.

[17] Smeaton, A. F., Wilkins, P., Worring, M., de Rooij, O., Chua, T. S., Luan, H. 2008. Content-based Video Retrieval: Three Example Systems from TRECVid. In *International Journal of Imaging Systems and Technology*, 18(2-3), 195–201.

[18] Snoek, C. G. M., Worring, M., van Gemert, J. C., Geusebroek, J. M., Smeulders, A. W. M. 2006. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In *MM'06*, ACM Press, 421–430.

[19] Wilkins, P., Ferguson, P., Smeaton, A. F. 2006. Using Score Distributions for Query-time Fusion in Multimedia Retrieval. In *MIR'06*, ACM Press, 51–60.

[20] Yilmaz, E., Aslam, J. A. 2006. Estimating Average Precision with Incomplete and Imperfect Judgments. In *CIKM'06*, 102-111.

Table A. infAP performance values for the different external KS retrieval strategies on each evaluated TRECVid topic

Topic	Manual	DBPedia	Flickr Text	Flickr Tag
197	0.00040	0.00000	**0.00050**	0.00000
198	**0.01550**	0.00570	0.00530	0.00410
199	**0.04140**	0.01730	0.02830	0.02130
200	0.00060	0.00020	**0.00170**	0.00000
201	0.01360	0.03810	0.03750	**0.04250**
202	**0.00150**	0.00010	0.00080	0.00090
203	0.00100	0.00180	**0.00230**	0.00140
204	0.01340	0.00180	0.00200	0.00450
205	0.00260	0.00890	**0.01010**	0.00550
206	0.03750	0.05280	0.06220	**0.07790**
207	0.08400	0.04560	**0.12260**	0.06380
208	**0.01730**	0.00000	0.00500	0.00280
209	0.00290	0.00000	0.00200	0.00190
210	0.00120	**0.00170**	0.00100	0.00100
211	**0.00530**	0.00040	0.00090	0.00060
212	0.02260	0.03730	0.02430	**0.04430**
213	**0.02330**	0.01250	0.00860	0.00510
214	**0.04900**	0.00000	0.00420	0.00000
215	0.00050	0.00010	0.00060	0.00070
216	0.00410	0.00180	**0.00430**	0.00360
217	**0.02680**	0.00350	0.00360	0.00000
218	**0.01480**	0.00500	0.00000	0.00000
220	**0.03570**	0.00100	0.02940	0.02620
221	**0.02630**	0.00020	0.00050	0.00030
222	**0.00450**	0.00000	0.00240	0.00280
223	**0.03370**	0.00010	0.00020	0.00030
224	**0.01560**	0.00310	0.00690	0.00140
225	**0.00580**	0.00030	0.00010	0.00000
226	**0.12330**	0.05850	0.00010	0.00000
227	**0.01390**	0.00030	0.00270	0.00140
228	**0.01510**	0.01180	0.01380	0.01800
229	0.01290	0.01160	**0.01500**	0.00720
230	**0.05010**	0.00540	0.00410	0.00230
231	**0.01820**	0.00340	0.00240	0.00050
232	0.00020	**0.00050**	0.00020	0.00010
233	**0.00380**	0.00000	0.00260	0.00030

Topic	Manual	DBPedia	Flickr	Flickr Tag
234	**0.02500**	0.00130	0.00060	0.00060
235	0.00040	**0.00090**	0.00000	0.00010
236	**0.00090**	0.00040	0.00010	0.00520
237	**0.00780**	0.00770	0.00110	0.00030
238	**0.00010**	0.00000	0.00000	0.00000
239	0.00110	**0.00150**	0.00090	0.00110
240	0.00000	0.00070	**0.00110**	**0.00010**
241	**0.00550**	0.00130	0.00170	0.00000
242	**0.00110**	0.00000	0.00060	0.00000
243	0.00000	0.00000	0.00010	0.00000
244	**0.03120**	0.00120	0.00130	0.00030
245	0.00100	**0.03330**	0.00000	0.00000
246	0.00190	0.00170	**0.01210**	0.00610
247	**0.00180**	0.00100	0.00100	0.00070
248	**0.01290**	0.00140	0.00340	0.00120
249	**0.00440**	0.00000	0.00030	0.00010
250	**0.01740**	0.00680	0.00290	0.00090
251	**0.00390**	0.00000	0.00000	0.00000
252	0.00010	0.00030	**0.00050**	0.00000
253	**0.00130**	0.00000	0.00020	0.00000
254	0.00010	0.00010	0.00010	0.00000
255	**0.01210**	0.00260	0.00010	0.00000
256	**0.00030**	0.00000	0.00130	0.00120
257	0.09310	**0.15850**	0.02480	0.00300
258	0.00260	0.00000	0.00070	0.00020
259	**0.04470**	0.00000	0.00140	0.00020
260	0.00700	**0.00720**	0.00210	0.00130
261	**0.00360**	0.00150	0.00100	0.00130
262	**0.00760**	0.00000	0.00120	0.00000
263	0.01170	0.02860	**0.06020**	0.01630
264	0.00040	**0.00310**	0.00020	0.00010
265	**0.01680**	0.00120	0.00280	0.00290
266	**0.01680**	0.00000	0.00250	0.00090
267	**0.00470**	0.00200	0.00670	0.00060
268	0.00280	0.00140	**0.00310**	0.00260

Table B. Examples of TRECVid topics

Topic	Query	Best KS
198	Find shots of a door being opened	Manual
226	Find shots of one or more people with mostly trees and plants in the background; no road or building visible	Manual
232	Find shots of one or more people, each walking into a building	DBPedia
235	Find shots of a person on the street, talking to the camera	DBPedia
197	Find shots of one or more people walking up stairs	Flickr Text
268	Find shots of one or more signs with lettering	Flickr Text
201	Find shots of a canal, river, or stream with some of both banks visible	Flickr Tag
210	Find shots with hills or mountains visible	Flickr Tag