Int J Multimed Info Retr (2013) 2:117–130
DOI 10.1007/s13735-012-0017-1

REGULAR PAPER

Exploiting semantics on external resources to gather visual examples for video retrieval

David Vallet · Iván Cantador · Joemon M. Jose

Received: 20 June 2012 / Accepted: 11 July 2012 / Published online: 2 September 2012

Abstract With the huge and ever rising amount of video content available on the Web, there is a need to facilitate video retrieval functionalities on very large collections. Most of the current Web video retrieval systems rely on manual textual annotations to provide keyword-based search interfaces. These systems have to face the problems that users are often reticent to provide annotations, and that the quality of such annotations is questionable in many cases. An alternative commonly used approach is to ask the user for an image example, and exploit the low-level features of the image to find video content whose keyframes are similar to the image. In this case, the main limitation is the so-called semantic gap, which consists of the fact that low-level image features often do not match with the real semantics of the videos. Moreover, this approach may be a burden to the user, as it requires finding and providing the system with relevant visual examples. Aiming to address this limitation, in this paper, we present a hybrid video retrieval technique that automatically obtains visual examples by performing textual searches on external knowledge sources, such as DBpedia, Flickr and Google Images, which have different coverage and structure characteristics. Our approach exploits the semantics underlying the above knowledge sources to address the semantic gap problem. We have conducted evaluations to assess the quality of visual examples retrieved from the above external knowledge sources. The obtained results suggest that the use of external knowledge can provide valid visual examples based on a keyword-based query and, in the case that visual examples are provided explicitly by the user, it can provide visual examples that complement the manually provided ones to improve video search performance.

D. Vallet (✉) · I. Cantador
Universidad Autónoma de Madrid, Madrid, Spain
e-mail: david.vallet@uam.es

I. Cantador
e-mail: ivan.cantador@uam.es

J. M. Jose
University of Glasgow, Glasgow, UK
e-mail: jj@dcs.gla.ac.uk

1 Introduction

During the last years, the amount of video content available online has increased exponentially. This is mostly due to the simplifications made in the publication process of video content on the Web. Online Web services such as YouTube[1] are nowadays hosting ever rising amount of video content, uploaded by both casual and professional users.[2] One of the main challenges of these services is how to facilitate users accessing the vast video collections. Research in the video retrieval area has been addressing this problem since the mid 1990s [2,8]. However, providing effective, generic video retrieval is far from been achieved; the multimodal nature of video content—which implies that information in video is represented in different forms, such as text, visual information, speech, and spatiotemporal data—makes video content very difficult to be indexed and retrieved effectively [3].

The keyword-based retrieval paradigm—made popular by current Web search engines—has been integrated into the most commonly used video retrieval systems, e.g. YouTube. This retrieval approach relies on having significant textual metadata, which in many cases consist of manual video annotations. Two main problems then arise. First, the users are often reticent to provide manual annotations when uploading content to community-based services. Second, the

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

[2] YouTube statistics: http://www.youtube.com/t/press_statistics.

annotations provided by the users may be of either poor quality or subjective [9].

To address the lack of textual metadata in video retrieval, the exploitation of low-level features has been widely explored. However, it suffers from the semantic gap problem, as the low-level information that the features provide is difficult to match with the real semantics of video contents [13].

An alternative approach is the extraction of high-level features, which, in practice, are usually obtained from analysis of low-level features. High-level features are mapped to a number of concepts (e.g. "car", "person") belonging to a particular thesaurus, such as the Large Scale Ontology for Multimedia (LSCOM) [18]. Based on such mapping, high-level feature extractors classify video shots into specific categories, each of them associated to one or more of the considered concepts [22]. In this context, the main drawbacks are that a classifier has to be trained for a particular concept, and that classification is limited to the selected concepts. Hence, high-level feature extraction techniques are difficult to apply in large scale and dynamic collections.

Despite the aforementioned problems and limitations, low-level features are still one of the primary sources of non-textual information exploitable for video retrieval. Even when there are high-level classifiers available, the literature has shown that combining low- and high-level features can achieve best performance [11,21]. Furthermore, low-level features are more scalable than high-level features, as they can be exploited without requiring a training collection.

A typical search process with low-level features involves the user providing a set of visual examples as a search query. The low-level features of the provided visual examples are then extracted and compared with the features of items in the collection. Subsequently, an aggregation method combines results for each type of low-level feature or results for multiple visual examples, in case more than one visual example was provided. This search scenario is exemplified in most of the research work on low-level feature extraction and video retrieval. For instance, the TRECVid (TREC Video Retrieval Evaluation) workshop proposes several search tasks, and provides sets of image and video shot examples for each task, along with textual search information. Most of the video retrieval systems that have participated in the TRECVid evaluation campaigns have [20] utilised provided visual examples to extract and exploit low-level features.

Although effective, it is difficult to fit the above scenario into a real setting, as users are forced to provide visual examples to initiate a query. This can really be a burden to a user, who may not have a visual example to provide, or simply does not want to be bothered with the task of finding a suitable example. To alleviate the user's effort, some systems follow an interactive setup [17], where the user performs a keyword-based search, selects relevant visual examples from retrieved results, and launches image-based searches in an iterative fashion. Other approaches allow the user to provide a sketch as a visual example, which is analysed with specialised algorithms to let low-level feature search over the video collection [6]. These two approaches, however, still require more effort to the user than the keyword-based paradigm. A pseudo-relevance feedback approach can be followed to automate the visual example retrieval process using information from the same collection. In this case, an initial keyword-based query is used to retrieve visual examples from the collection, and the top N retrieved results are considered relevant as visual examples [1]. This last retrieval approach requires the collection to contain additional features (e.g. textual annotations) or high-level feature classifiers. And these requirements are not usually satisfied by online collections.

A possible solution to the above problem is to exploit an external media collection that meets the requirements to perform an automatic retrieval of visual examples. In this case, the user's query is used to retrieve relevant visual examples from the external collection, which are then analysed and exploited for searching in the video collection. This approach has been followed by various video retrieval systems in the TRECVid workshop [7,16]. However, it has been applied informally in the context of that workshop, and has been mostly focused on the Flickr[3] media service, using image titles and user tags. In this paper, on the contrary, we present a formal study on the effect of different external collections, and propose automatic visual example retrieval approaches that make use of structured metadata available in external online collections.

The aim of our work is to empirically validate the feasibility of exploiting external Knowledge Sources (KSs) to provide relevant visual examples related to a user's search query, without the need of asking the user to provide visual examples. More specifically, we study if external KSs do provide visual examples that can substitute or complement those that a user would have provided manually with additional effort. Thus, we explore the use case that better fits the current search paradigm: the user issuing a keyword-based query to a collection that lacks of textual annotations. Our search system exploits external KSs to retrieve visual examples that are used as search inputs for low-level feature retrieval models. In this paper, we use three different external KSs, which have different characteristics:

1. DBpedia[4]: a highly structured, collaboratively built KS, with a relatively low amount of multimedia files, i.e., with low coverage of visual examples related to a query.
2. Flickr: a semi-structured KS, freely defined by users using folksonomies, but with a great coverage and a high quality of visual examples related to a search query.

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

[4] http://dbpedia.org/.

3. Google[5] and Yahoo![6] image search services: KSs with almost no metadata structure and variable quality, built from the crawling of images on the Web, and providing the greatest coverage of visual examples.

The remainder of the paper is organised as follows. Section 2 discusses the research hypotheses of our study. Section 3 presents our framework for external KS exploitation, and the different visual example retrieval methodologies adopted for the used KSs. Section 4 describes the different video retrieval strategies that make use of the visual examples obtained by our framework. Sections 5 and 6 present our experimental setup and obtained results, respectively. Section 7 provides an overview of related works on the use of external KSs for video retrieval. Finally, Sect. 8 concludes with some discussion and future research lines.

2 Research hypotheses

Our research is based on the following hypotheses.

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

To test this hypothesis, we propose to use the TRECVid 2007 and TRECVid 2008 collections. We shall use the textual representation of the search topics in the collections to automatically retrieve visual examples from a KS. The retrieved visual examples will then be used as a source of low-level features to be exploited by a video retrieval process. Results will be evaluated using TRECVid's evaluation assessments, as to analyse the role of each external KS on providing relevant visual examples. We shall also compare the use of the automatically obtained visual examples with the use of those examples provided with the TRECVid topic descriptions, which will be considered as visual examples manually provided by the users. Furthermore, we shall analyse whether the external resources are a good complement to the manual visual examples.

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

To test this hypothesis, we shall exploit three different KSs with various degrees of structure in their metadata. Our goal will be to test if more structured metadata, such as the one provided by DBpedia, which includes relations as generalisation and specification, can be exploited successfully to find relevant visual examples. The obtained results will be compared with KSs with lower degrees of semantic structure, namely a semi-structured KS by means of folksonomies (Flickr) and KSs based solely in related textual annotation, such as text anchors (Google and Yahoo!).

3 Exploiting 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 issues, on formulating a clear and meaningful query.

Since content-based information retrieval systems deal with the search of visual objects, it seems natural to conduct search processes using (visual) examples of such objects. In fact, many content-based information retrieval systems follow query-by-example (QbE) approaches, in which a user is required to pick one or more video examples beforehand. When the user does not exactly know which video shots she is looking for, or the dimension of the search space is very large—as is often the case—this approach may 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 that contain [keywords]", and videos have to be annotated with semantic concepts corresponding to all possible keywords the user may introduce.

Then, textual annotation of videos represents 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 shown recently in Web 2.0 applications, such as YouTube, Yahoo Videos,[7] Metacafe,[8] Revver,[9] and Daily Motion,[10] the community can play an important role to annotate on-line videos, letting multimedia content retrieval based on collaborative social tagging to be extremely successful.

Hence, it turns out that both QbE and QbT strategies are needed. In the approach we propose in this paper, the user provides a textual query to describe the semantic concepts that should appear in the videos she is interested in. Instead of looking for these concepts by directly analysing video content, we propose to explore external collaboratively annotated

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

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

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

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

[9] http://revver.com/.

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

image repositories 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 approach, and the external KSs with which we have empirically evaluated it.

3.1 Architecture

In this paper, we study the exploitation of two public available collaborative KSs with large image collections, namely DBpedia and Flickr.

DBpedia is a Semantic Web gateway that collects data from Wikipedia[11] encyclopaedia. Wikipedia articles mostly consist 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 June 2012, the DBpedia dataset describes more than 3.6 million "things", including people, places, companies, etc. These descriptions are completed with more than 2.7M 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 shall also obtain images of related concepts.

Flickr, on the other hand, is an image-hosting website that 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 photograph owners. As of June 2012, Flickr claims to host more than 6 billion images. Given a certain concept, we propose to match it to one or more social tags 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 and Yahoo! Images—two well-known QbT-based image search services. The details of this comparison are described in Sect. 6.

The general architecture of our approach is shown in Fig. 1. The user provides the system a natural language query describing the contents of the videos she wants to retrieve, and the system returns a ranked list of videos, in which ranking scores are similarity values between the video contents and the given input query. In our experiments, the user's input is simulated through a subset of natural language queries extracted from TRECVid collections. The whole video retrieval process is divided into five steps, numbered in the figure.

1. The extracted concepts are passed to a module that matches them with semantic entities (i.e., DBpedia entities and Flickr social tags) belonging to the external KSs.
2. Once the semantic entities are identified, several heuristics, which depend on the KSs, are performed by an image retriever to return ranked lists of images that are annotated with the above entities.
3. The gathered images are analysed, and some of their low-level features (e.g. colour, shape and texture) are obtained.
4. 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 to filter and sort them for the user.

In the remainder 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 Sect. 4.

3.2 Knowledge sources

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

First, each query concept label has to be matched with semantic entities existing in a 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 may 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 that are annotated with the final entities have to be ranked. Again, a customised ranking strategy may be followed.

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

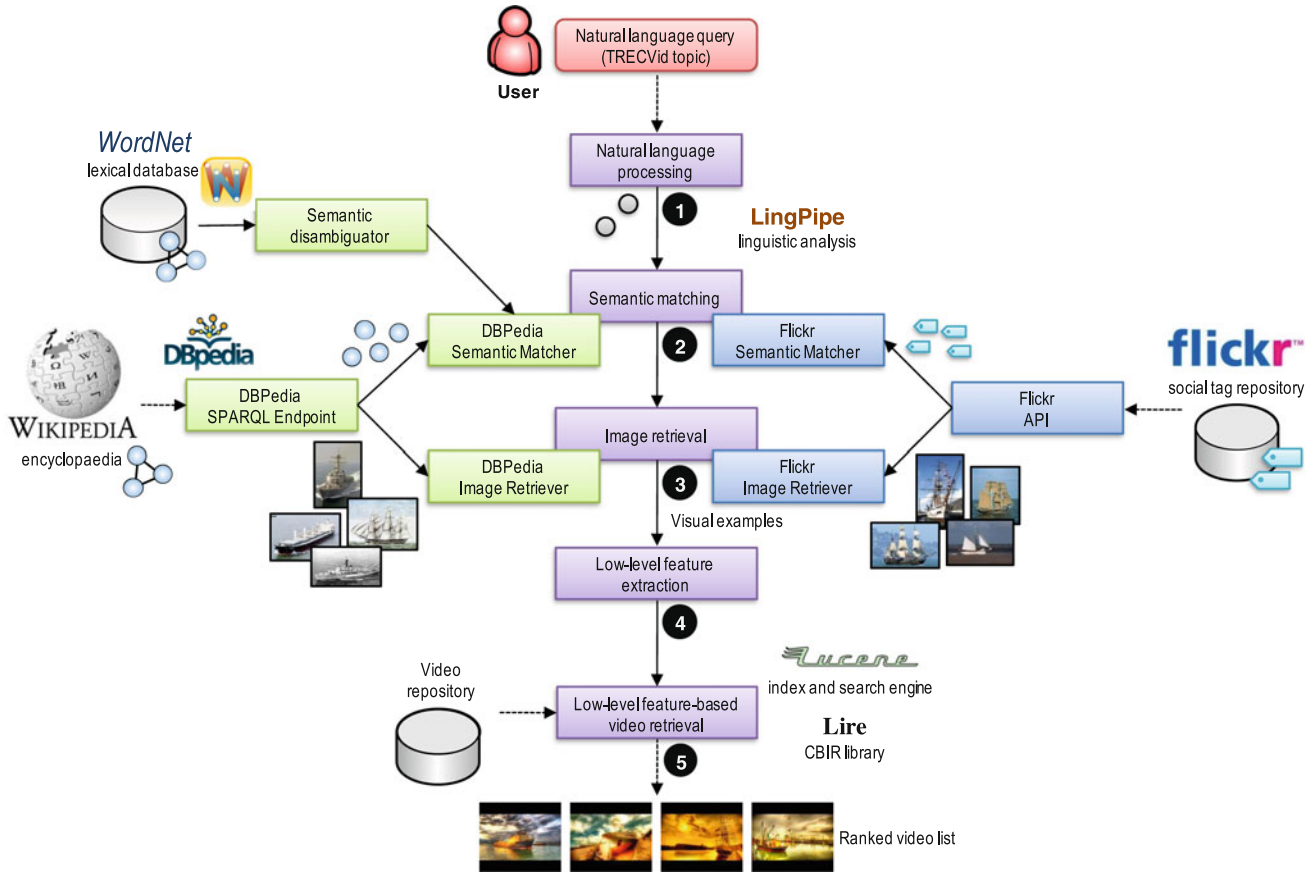

Fig. 1 General architecture of the proposal

The subsequent subsections describe how we have accomplished the previous tasks for DBpedia, Flickr, and Google and Yahoo! Images KSs.

3.2.1 DBpedia

DBpedia is an ontology that stores structured information obtained from Wikipedia, and, making use of Semantic Web technologies, links that information with other KSs, such as OpenCyc, WordNet and DBLP, among others.[12]

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" whose value would be different for each city.

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 approach, 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. 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:

[12] http://wiki.DBpedia.org/Interlinking.

- 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 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. To choose one of them, we make use of WordNet [17]. 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" may 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. To obtain more related images, we exploit the semantic relations available in DBpedia. We explain our approach with an example, shown in Fig. 2.

Let us suppose that the user has entered the query "find shots of a *building*". Let us focus on the concept "building", and assume that DBpedia contains information about the concept "building" in the way depicted in the figure. The entity "Building" has an image in Wikipedia (linked by the property *foaf:depiction*), and belongs to the category (class) "Buildings and structures", as declared by the property *dbpedia:ontology/category* (equivalent to the general property "subClassOf"). To obtain more images of buildings, we extract all the subcategories of the class "Buildings and structures" following the property *skos:broader*, which can be understood as the inverse relation of "subClassOf". In the example, we find the subcategories "Tower", "Church", and "Skyscraper". 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.

With the entities related to "Building", the query has been extended in such a way that the system takes into consideration different types of building structures, thus returning images that contain different types of buildings, such as towers, churches and skyscrapers, even though they were not explicitly annotated with the concept "building".

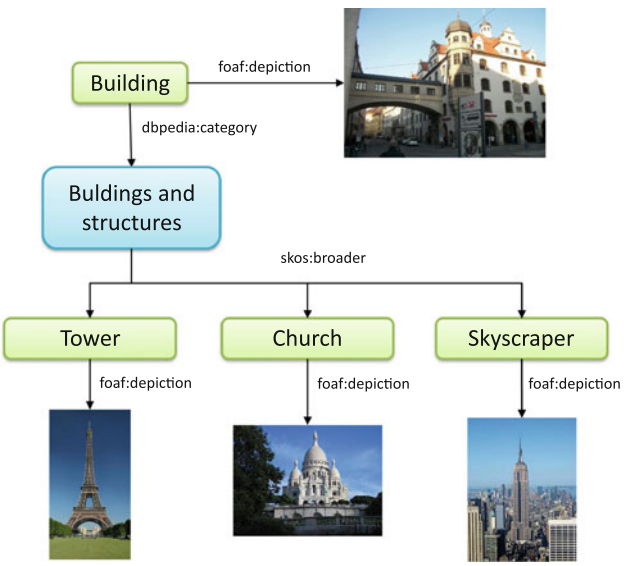

Fig. 2 Relations of concept "Building" extracted from DBpedia

It is important to notice 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 *church* during the *sunset*". This situation does not occur in other KSs such as Flickr or Google and Yahoo! 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 "Building" should have a higher score than the images retrieved from the entities "Tower", "Church" and "Skyscraper." 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:

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

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, its images 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 Sect. 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 a *church* during the *sunset*" is transformed into "church sunset", and not into the two independent queries concepts "church" and "sunset."

The transformed query is provided to Flickr's 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:

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

where T is the set of tags that are annotations of the M retrieved images, n_t is the number of times the tag t appears in the annotations of the retrieved images, and $\text{avg}(n_t)$ 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 and Yahoo! image search services

Similar to Flickr, Google's and Yahoo!'s images search services allow 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 the image search APIs. We thus consider these services as highly unstructured KSs. Our hypothesis is that by 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 than those from Google and Yahoo! Although in Sect. 5 we compare the quality of each KS, our understanding is that the presented approaches are complementary.

4 Retrieval strategies

In this section, we briefly present the two video retrieval strategies analysed in this work. These strategies follow a QbE strategy 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 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 Sect. 5.1

One of the faced problems was to set up a limit on the number of 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.

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 may be appropriate when the manually provided examples are not sufficient for a query, or not 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 that scores each document $d \in D$ with a normalised score value $s(d, V) \in [0, 1]$. We then create a final result set $R(V) = \{d_1, d_2, \ldots, d_N\}$ 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 $s(d, \text{EV}) \in [0, 1]$ if $d \in R(V)$, and 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:

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

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 tested a number of basic multimodal fusion techniques (see [14] for an overview) to dynamically set the λ parameter, but we did not find any significant improvement. In future work, we shall explore the application of more elaborated techniques, which could help on 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. Formally, our evaluation aims to address the following research questions:

Q1. Can we exploit the semantics underlying external KSs 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

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" [20]. TRECVid 2007 collection provides over 100 h of video, and 24 topic (task) descriptions, along with their respective relevance judgements. TRECVid 2008 collection provides over 200 h 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 350 K keyframes on TRECVid 2007 collection, and over 700 K 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) [4], and fuzzy colour and texture histogram (FCTH) [5]. As a query-time fusion methodology for the different low-level features and visual examples, we use the method described in [24]. 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 feature 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

In this section, we present and analyse 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 [27].

6.1 External knowledge retrieval

To address research question Q1, we measure the performance values obtained when applying the different proposed KS exploitation techniques presented in Sect. 3. Table 1 shows the performance results of the external knowledge retrieval strategy (explained in Sect. 4.1) using the above techniques on 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 Examples). The results given in the first two rows of the table do not consider the ranking heuristics presented for the DBpedia and Flickr KSs (see Sects. 3.2.1 and 3.2.2, respectively).

In addition, Table 1 shows the performance values obtained 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. The DBpedia ranking heuristic leads to a 34.25 and 17.21 % performance increases on TRECVid 2007 and 2008 collections, respectively, which are statistically significant (Wilcoxon test, $p < 0.05$). The heuristics applied on the Flickr KS result on a $\sim$ 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 suggests that KSs with more formal semantic structures allow the implementation of ranking heuristics to provide higher-quality visual examples. It is worth noting that there is a decrease in the performance val-

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

Topics	Strategy					
	DBpedia	Flickr tag	Flickr text	Yahoo!	Google	Manual example
2007	0.0076	0.0134	0.0155	0.0077	0.0130	0.0180
2008	0.0064	0.0017	0.0039	0.0019	0.0039	0.0139
2007 (heuristic)	0.0102	0.0127	0.0123			
2008 (heuristic)	0.0075	0.0029	0.0063			
Δ 2007 (%)	+34.25	−5.55	−20.86			
Δ 2008 (%)	+17.21	+68.86	+61.30			

Rows tagged with (heuristic) indicate values obtained from the ranking heuristics proposed in previous sections. The difference between the basic and ranking heuristic results are shown in the last two rows, when applicable

ues of Flickr and Google retrieval strategies on TRECVid 2008 with respect to TRECVid 2007. This decay is not proportional to the performance decay of the results of the manual examples. DBpedia KS exploitation approach, however, seems to give more consistent results.

To address research question Q2, and based on the results given in Table 1, 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").

Analysing the obtained results one can observe that the manual examples provide significantly better results in terms of infAP, with respect to the techniques that retrieve results automatically from external KSs. This may be expected as the manual examples are selected specifically to describe the search topic, and belong to the same search collection, which in TRECVid has specific features: all images have the same resolution, and are keyframes extracted from video content. The manual examples can thus be considered as an upper bound for our evaluation. The obtained results come close to that upper bound, with Flickr Text performing 14 % lower on the 2007 topics and DBpedia 46 % lower on the 2008 topics. These values indicate that, in absence of such examples, the external strategy can be a good alternative.

6.2 External knowledge applied to manual query examples

To address research question Q3 we measure the performance of the retrieval strategy explained in Sect. 4.2, which complements a set of manually provided visual examples with examples provided by our external KS exploitation techniques.

Table 2 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 Examples). 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 (Sect. 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 %) with statistically significant results. The other external KSs approaches have a more moderate improvement of precision over the manual examples, although the 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 show that external KSs can be successfully exploited to complement visual examples provided by a user.

6.3 Per-topic analysis

In this section, we perform a per-topic analysis of the results obtained with the different external KSs. Table 3 shows a summary of the number of topics in which each KS performs best when using its provided examples in isolation. From the results in the table, it is clear that the manually pro-

Table 2 Performance values for the external KS approaches applied to manual examples

Metrics	Strategy					
	DBpedia	Flickr tag	Flickr text	Yahoo!	Google	Manual examples
2007 infAP	0.0175	0.0174	0.0176	0.0174	**0.0181**	0.0180
2007 P@15	**0.0861**	0.0722	0.0778	0.0713	0.0750	0.0611
2008 infAP	**0.0144**	0.0133	0.0137	0.0134	0.0132	0.0139
2008 P@15	**0.0570***	0.0486	0.0486	0.0473	0.0528*	0.0417
Δ infAP 2007 (%)	−2.89	−3.37	−2.48	−3.33	+0.51	
Δ P@15 2007 (%)	**+40.90***	+18.18*	+27.27*	+16.89	+22.72*	
Δ infAP 2008(%)	+3.58	−3.91	−1.11	−3.60	−5.06	
Δ P@15 2008(%)	**+36.68***	+16.68*	+16.69*	+13.43	+26.68*	

Bold values indicate the best performing approach for each collection and metric
Asterisks indicate a statistical significance (paired t-test $P < 0.03$)

Table 3 Number of topics on which each KS is best (# best), percentage of these topics over all evaluated topics (% best), percentage of topics which return no visual examples for each KS (% no examples), and percentage of topics which return no relevant results (% no relevant)

	DBpedia	Flickr tag	Flickr text	Yahoo!	Google	Manual examples
# Best	7	4	13	5	5	39
% Best	9.86	5.63	18.31	7.04	7.04	54.93
% No examples	14.29	12.86	4.29	8.57	0.00	N/A
% No relevant	23.94	25.35	7.04	18.31	9.86	2.82

Table 4 Examples of best performing TRECVid topics for each KS

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
202	Find shots of a person talking on a telephone	Yahoo!
203	Find shots of a street market scene	Google

vided examples achieve the best performance in a significant number of topics. However, external KSs achieve a better performance in almost half of the topics. This indicates that external KSs should not be disregarded for certain topics even in the case that manual visual examples are provided. DBpedia KS seems to be more prone to not finding relevant visual examples, as on 14.29 % of the topics no visual examples were found.

This may be due to the fact that DBpedia is a far more restricted KS for visual examples than Flickr or Google. One of our concerns was that certain search topics involved more than one concept, e.g. "find shots of one or more people at a table or desk, with a computer visible". In these cases, 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, which could be of advantage to the latter KSs. However, the obtained results show no evidence against the one-concept-per-image approach of DBpedia, compared to the multiple-concepts-per-image approach of Flickr, Google, and Yahoo! images. 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. Table 4 shows a selection of examples in which each of the analysed KSs performs best.

Table 5 Average number of DBpedia categories and retrieved images for topics that resulted on an improvement over the manual examples or resulted on worse performance values

Metric	Avg. no. of categories	Avg. no. of found images
2007 improvement	1.857	232.4
2007 worsening	1.588	153.8
2008 improvement	1.600	372.3
2008 worsening	1.395	120.7

We now analyse the results obtained by DBpedia, which was the best performing external KS, and the one that best complemented the manual visual examples. We study whether there is a correlation between the improvement on the manual examples using DBpedia as an additional source of visual examples, and the retrieval technique presented in Sect. 3.2. Table 5 shows the results of this analysis.

The average number of categories found per topic in Table 5 indicates that there are a higher number of DBpedia categories matched to the topics on which DBpedia can successfully complement the manual examples. This indicates that there may be more information available in DBpedia for the topics, and thus better examples could be found. The average number of found images also seems to support this hypothesis as, on average there were found more images on topics that result on an improvement over the manual examples. However, these results are not statistically significant, so that only a trend can be concluded from this analysis.

7 Exploiting external knowledge resources for multimedia information retrieval

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 [15] or a specific ontology. In this paper, we have exploited external KSs as image retrieval services, to collect relevant visual examples to be used in video retrieval tasks. Most image retrieval services accept textual keywords as input. Here, we have presented a technique that makes use of a more structured KS, DBpedia, which lets 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.

Snoek et al. [22] collect Web images to train the video search system MediaMill. Olivares et al. [19] spread manual annotations across Flickr's image collection 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 paper, we have proposed to exploit more the semantics and structure present on KSs, focusing on video retrieval.

There has been a number of works reported in the TRECVid workshop that have attempted to exploit KSs such as Google, Flickr and YouTube to retrieve further visual examples, to expand the textual query, or to provide further training examples for high-level classifiers. In the following, we summarise some of these approaches.

Xue et al. [25] obtained additional image examples using Google Images. Initial image features were extracted from this example set, which were later used in the search query, after applying a dimension reduction technique. Etter [7] also used Google Images to obtain additional visual examples, although the author does not indicate if i was manual or automatic. The author also used Wikipedia to perform an expansion over the textual query. Aly et al. [1] analysed the query using Wikipedia and WordNet, extracting related concepts to the query which are later used for query expansion. Liang et al. [15] used Flickr to complement the visual examples provided in each search topic. The reported results indicated that this complementation improved slightly the overall results of their automatic retrieval engine. However, it is not clear if they used an automatic or manual procedure to retrieve these additional image examples. Ulges et al. [23] obtained additional videos from YouTube to train a set of high-level classifiers, although the reported results were worse than the examples provided by TRECVid. Liu et al. [16] used an extra collection of ABC news as additional ground truth to re-rank video documents. They argued 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 [10]. In this paper, we have also analysed if external knowledge can enhance, or even substitute, a set of visual examples manually provided by a hypothetical user. Many of the above works do not indicate their approaches to gather visual examples from external KSs, nor provide a formal model of the exploitation of the KS. Thus, the results of these systems do not allow an in-depth analysis of the effect that the KS could have in the video retrieval process.

8 Conclusions and future work

Current Web video retrieval systems rely on manual text annotations of the video contents to provide the user with a text-based search interface. This approach is limited to the fact that users are often reticent to provide manual annotations, and the quality of such annotations is in many cases questionable. Content-based video retrieval systems, on the other hand, attempt to extract low-level features from visual query examples, and exploit such features to find related video contents in repositories. The limitation in this case is the so-called semantic gap, which consists of the fact that low-level information does not match with the real semantics associated to the videos. Moreover, this technique may be a burden to the user, as it requires finding and providing the system with relevant visual examples.

In this paper, we have presented a hybrid video retrieval approach that is based on textual queries and annotations, and exploits low-level features extracted from visual examples. In contrast to existing approaches, we propose to automatically retrieve high- quality visual examples from external knowledge sources. To address the semantic gap, we have studied different strategies that exploit the semantics underlying the above knowledge sources, reducing the ambiguity of the query, and focusing the scope of the image searches in the repositories. We analysed three different external Knowledge Sources: (1) DBpedia, a highly structured, collabora-

tive built KS, with a low semantic coverage in multimedia sources; (2) Flickr, a folksonomy-based KS, freely defined by users, with a greater coverage; and (3) Google and Yahoo! Images, two low structured KS, but with a high coverage (the Web).

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.

Regarding our first hypothesis, the conducted evaluations showed that although using only external visual examples provides lower performance than using manual visual examples, the performance values obtained with the former achieve the between 14 and 46 % of the performance values obtained with the latter. This indicates that, in absence of manually selected examples, our retrieval strategies may represent good alternatives. Moreover, we believe 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 may require more sophisticated techniques, such as e.g. those related to query performance prediction [10].

We have used a video retrieval framework to evaluate the external KS exploitation techniques. These techniques could also be incorporated into a content-based image retrieval system. A proper evaluation would have to be conducted to determine this. Hence, a comparison with state of the art approaches, such as the one presented by Olivares et al. [19], could be possible. Although the investigated basic techniques of multimodal aggregation did not improve the effectiveness of our retrieval techniques based on external KS, we will also investigate more complex multimodal models, such as manifold ranking [12] and local regression [26].

References

1. Aly R, Hauff C, Heeren W, Hiemstra D, de Jong F, Orderlman R, Verschoor R, de Vries A (2007) The Lowlands Team at TRECVID 2007. In: TRECVid'07
2. Amir A, Berg M, Permuter H (2005) Mutual relevance feedback for multimodal query formulation in video retrieval. In MIR'05. ACM Press, London, pp 17–24
3. Chang SF, Chen W, Meng H, Sundaram H, Zhong D (1998) A fully automated content based video search engine supporting spatio-temporal queries. IEEE Trans Circuits Syst Video Technol 8(5):602–615
4. Chatzichristofis S, Boutalis Y (2008) CEDD: color and edge directivity descriptor, 2008. A compact descriptor for image indexing and retrieval. In ICVS'08. Springer, Berlin, pp 312–322
5. Chatzichristofis SA, Boutalis YS (2008) FCTH: fuzzy color and texture histogram—a low-level feature for accurate image retrieval. In WIAMIS'08. IEEE, New York, pp 191–196
6. Collomosse JP, Mcneill G, Watts L (2008) Free-hand sketch grouping for video retrieval. In ICPR'08. IEEE, New York, pp 1–4
7. Etter D (2008) Knowledge based retrieval at TRECVID 2008. In: TRECVid'08
8. 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. Computer 28(9):23–32
9. Guy M, Tonkin E (2006) Folksonomies: Tidying up tags? D-Lib Mag 12(1)
10. Hauff C, Hiemstra D, de Jong F (2008) A survey of pre-retrieval query performance predictors. In: CIKM'08. ACM Press, London, pp 1419–1420
11. Hauptmann AG, Christel MG (2004) Successful approaches in the TREC video retrieval evaluations. In: MULTIMEDIA'04. ACM Press, New York, pp 668–675
12. He J, Li M, Zhang HJ, Tong H, Zhang C (2004) Manifold-ranking based image retrieval. In: MULTIMEDIA'04. ACM Press, London, pp 9–16
13. Jaimes A, Christel M, Gilles S, Ramesh S, Ma WY (2004) Multimedia information retrieval: what is it, and why isn't anyone using it? In: MIR'04. ACM Press, London, pp 3–8
14. Kennedy L, Chang SF, Natsev A (2008) Query-adaptive fusion for multimodal search. In: Proceedings of the IEEE, vol 96(4), pp 567–588
15. Liang Y et al (2008) THU and ICRC at TRECVid 2008. In: TRECVid'08
16. Liu Z, Gibbon D, Zavesky E, Shahraray B, Haffner P (2006) AT&T research at TRECVid. In: TRECVid'06
17. Miller GA (1995) WordNet: a lexical database for English. New horizons in commercial and industrial artificial intelligence. Commun ACM 38(11):39–41
18. Naphade M, Smith JR, Tesic J, Chang JS, Hsu W, Kennedy L, Hauptmann A, Curtis J (2006) Large-scale ontology for multimedia. IEEE MultiMed 13(3):86–91
19. Olivares X, Ciaramita M, van Zwol R (2008) Boosting image retrieval through aggregating search results based on visual annotations. In: MM'08. ACM Press, London, pp 189–198
20. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and TRECVid. In: MIR'06. ACM Press, London, pp 321–330
21. Smeaton AF, Wilkins P, Worring M, de Rooij O, Chua TS, Luan H (2008) Content-based video retrieval: three example systems from TRECVid. Int J Imaging Syst Technol 18(2–3):195–201
22. Snoek CGM, Worring M, van Gemert JC, Geusebroek JM, Smeulders AWM (2006) The challenge problem for automated detection of 101 semantic concepts in multimedia. In: MM'06. ACM Press, London, pp 421–430

23. Ulges A, Koch M, Schulze C, Breuel TM (2008) Learning TRECVID'08 high-level features from YouTube. In: TRECVid'08
24. Wilkins P, Ferguson P, Smeaton AF (2006) Using score distributions for query-time fusion in multimedia retrieval. In: MIR'06. ACM Press, London, pp 51–60
25. Xue X et al (2007) Fudan University at TRECVID 2007. In: TRECVid'07
26. Yang Y, Xu D, Nie F, Luo J, Zhuang Y (2009) Ranking with local regression and global alignment for cross media retrieval. In: MM'09. ACM Press, London, pp 175–184
27. Yilmaz E, Aslam JA (2006) Estimating average precision with incomplete and imperfect judgments. In: CIKM'06. ACM Press, London, pp 102–111