Conference paper
Siersdorfer, S., San Pedro, J. and Sanderson, M. (2009) *Automatic Video Tagging using Content Redundancy.* In: The 32nd Annual ACM SIGIR Conference, July 19-23 2009, Boston, Massachusetts, USA.

To appear in Proceedings of ACM SIGIR, 2009.

Automatic Video Tagging using Content Redundancy

Stefan Siersdorfer
L3S Research Center
Appelstr. 9a
30167 Hannover, Germany
siersdorfer@L3S.de

Jose San Pedro
University of Sheffield
211 Portobello Street
Sheffield S1 4DP, UK
jsanpedro@mac.com

Mark Sanderson
University of Sheffield
211 Portobello Street
Sheffield S1 4DP, UK
m.sanderson@sheffield.ac.uk

ABSTRACT

The analysis of the leading social video sharing platform YouTube reveals a high amount of redundancy, in the form of videos with overlapping or duplicated content. In this paper, we show that this redundancy can provide useful information about connections between videos. We reveal these links using robust content-based video analysis techniques and exploit them for generating new tag assignments. To this end, we propose different tag propagation methods for automatically obtaining richer video annotations. Our techniques provide the user with additional information about videos, and lead to enhanced feature representations for applications such as automatic data organization and search. Experiments on video clustering and classification as well as a user evaluation demonstrate the viability of our approach.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.5 [**Information Systems**]: INFORMATION STORAGE AND RETRIEVAL—*On-line Information Services*

General Terms

Algorithms

Keywords

video duplicates, content-based links, automatic tagging, neighbor-based tagging, tag propagation, data organization

1. INTRODUCTION

The rapidly increasing popularity and data volume of modern Web 2.0 content sharing applications originate in their ease of operation for even unexperienced users, suitable mechanisms for supporting collaboration, and attractiveness of shared annotated material (images in Flickr, bookmarks in del.icio.us, etc.). One of the primary sites for video sharing is YouTube[1]. Recent studies have shown that traffic to/from this site accounts for over 20% of the web total and 10% of the whole internet [5], comprising 60% of the videos watched on-line [9].

The growing size of folksonomies has motivated a real necessity to provide effective knowledge mining and retrieval tools. Current solutions are primarily focused on the analysis of user generated text; folksonomies have shown to provide relevant results at a relatively low cost, as they are contributed by the community and text-based tools have been used for topic detection and tracking [1], information filtering [30], or document ranking [29].

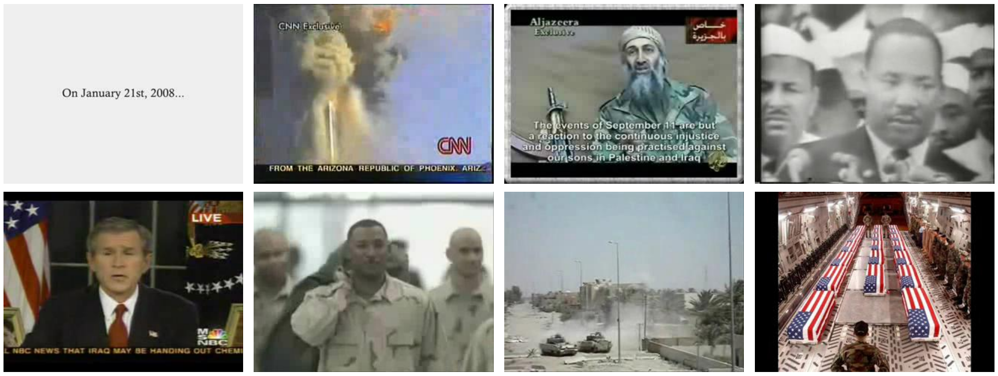
However, manually annotating content such as videos in YouTube, is an intellectually expensive and time consuming process, and, as a consequence, annotations such as tags are often very sparse. Furthermore, keywords and community-provided tags lack consistency and present numerous irregularities (e.g. abbreviations and mistypes). This can lead to a decreased quality of the information presented to users of the Web 2.0 environment; moreover, it makes the utilization of techniques for automatic data organization using classification or clustering, retrieval and knowledge extraction relying on textual features a difficult task. Content-based techniques, however, are not yet mature to outperform text-based methods for these purposes [11].

Recent studies [3, 27] show evidence of a significant amount of visually redundant footage (i.e. near-duplicate videos) in video sharing websites, with a reported average of over 25% near-duplicate videos detected in search results. Many works have already started to consider this redundancy a problem for content retrieval, and propose ways to remove it from the system. In this paper, we look at duplication from a different perspective and demonstrate that content redundancy in Web 2.0 environments such as Youtube can be seen as a feature rather than a problem. We automatically analyze the dataset to find duplicates and exploit them to obtain richer video annotations as well as additional features for automatic data organization.

Previous work has focused on metadata as the way to get to the content. In our work we proceed differently, using content to improve the overall quality of annotations. We present a novel hybrid approach combining video content analysis and network algorithms to improve the quality of annotations associated to videos in community websites. In contrast to existing approaches, we adapt robust video duplicate detection methods to the Web 2.0 context, enabling us to effectively use content features to link video assets and

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...$5.00.

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


On January 21st, 2008...
CNN Exclusive
CNN
FROM THE ARIZONA REPUBLIC OF PHOENIX, ARIZ.
Aljazeera Exclusive
خاص بالجزيرة
The events of September 11 are but a reaction to the continuous injustice and oppression being practised against our sons in Palestine and Iraq
LIVE
NBC NEWS THAT IRAQ MAY BE HANDING OUT CHEMI
MSNBC

original	***autotags***
Ron	2008
Paul	president
Barack	bush
Obama	martin
Kanye	luther
West	king
jay-z	protest
Giuliani	terror
CNN	rights
Politics	war
News	washington
Hilary	iraq
Clinton	speech

Figure 1: Tags in duplicated or visually overlapping videos may convey additional interpretations and complementary information which can help to create a richer description.

build a new graph structure connecting videos. In what is a novel approach to folksonomy analysis, we use these new content-based links to propagate tags to adjacent videos, utilizing visual affinity to spread community knowledge in the network.

Figure 1 shows an example of automatically generated tags using our proposed methods. Uploaders of overlapping sequences of the same video provide their personal perspective on its content, in the form of different keywords. Individually, each description lacks part of the semantics conveyed by the asset; when considering them together, we achieve a more comprehensive description. By uploading videos containing redundant sequences, different users have contributed to complete the information about the original source video, including characters (e.g. Bush), dates (e.g. 2008), and other more general concepts (e.g. protest, rights).

The rest of this paper is organized as follows: In Section 2 we discuss related work on folksonomies and sharing environments, tagging, and duplicate detection. We provide a short overview of duplicate and overlap detection of videos and a graph formalization of these relationships in Section 3. Based on this formalization, we describe several techniques for automatic tagging in Section 4, both neighbor and context-based (TagRank). In Section 5 we provide the results of the evaluation of our automatic tagging methods for YouTube video clustering and classification as well as a user study. We conclude and show directions of our future work in Section 6.

2. RELATED WORK

Mining links between resources has received growing research interest, especially on the web, the most popular and referenced example being Google's PageRank [21], where the graph of web document nodes is formed by the hyperlinks contained in them. We can also find examples of algorithms taking advantage of "implicit" links [29], where links are inferred from user actions (e.g. access patterns). In this later category we can include near-duplicate document detection techniques [28]. Tools, such as Charikar's fingerprint [4], have shown success in achieving near-duplicate identification in massive collections of web pages [20]. The graph of inter-document links is used, in this case, to perform many common procedures in web processes optimization, such as web crawling filtering [20], enhanced ranking schemes [29], first story detection [24], or plagiarism detection [8]. These techniques rely on the textual nature of documents. In this paper, we focus on exploiting *visual* relationships available in richer multimedia scenarios.

Visual content analysis enhances knowledge about video collections. Many previous works use this information to improve retrieval results. Re-ranking is perhaps the most common application. Hauptmann *et al.* [27] use a Content-based copy retrieval (CBCR) approach to detect near-duplicate videos in order to promote diversity on search results by removing redundant entries. Liu *et al.* [19] proceed similarly, but consider also text to establish links between videos, to enable covering different interpretations of queries at the top of the results list. In contrast to previous approaches, we exploit visual links between videos to improve the quality and homogeneity of *annotations*.

Various approaches have been exploiting graph structures in Folksonomies. A node ranking procedure for folksonomies, the FolkRank algorithm, has been proposed in [12]. FolkRank operates on a tripartite graph of users, resources and items, and generates a ranking of tags for a given user. Another procedure is the Markov Clustering algorithm (MCL) in which a renormalization-like scheme is used in order to detect communities of nodes in weighted networks [26]. A PageRank-like algorithm based on visual links between images is used to improve the ranking function for photo *search* in [14]; in contrast we are using weight propagation for automatic video *tagging*. In another paper on image search Craswell and Szummer [6] introduce a random walk algorithm operating on a bipartite click graph with edges connecting queries and clicked images. The term "TagRank" occurs in another context in the preliminary work [25] where a general ranking of tags for whole folksonomies is generated, rather than the relevancy of tags for individual objects as described in our work for the specific case of videos.

To the best of our knowledge, our paper is the first to propose a hybrid approach using content-based copy retrieval techniques in combination with novel tag propagation methods to automatically annotate videos in folksonomies.

3. CONTENT LINKS BETWEEN VIDEOS

In this section, we provide a short overview on automatic video duplicate and overlap detection. We then formalize the output of overlap detection methods as graph structures. In the next section, we describe our methods for automatic tag assignment and propagation based on this formalization.

3.1 Duplication and Overlap Detection on Videos

The identification of near-duplicate video content has received significant attention by the multimedia research community for a number of applications, e.g. copyright infrigement detection. These techniques are commonly referred as Content-based Copy Retrieval (CBCR). Current works feature recall and precission values close to 100%, and support the detection of noisy and degraded copies, which may include compression artifacts, analog conversion distortions, overlaid content such as subtitles or chromatic changes among others [15]. Many of the principles used by text-based duplicate detection techniques can be adapted to the video domain [13]. Fingerprints are the most commonly used detection tool; they are generated using specific features of moving visual content, such as temporal video structure or time-sampled local or global image invariants.

We built a video copy detection system to detect redundancy in the YouTube scenario based on the work originally presented in [23]. This fingerprint-based method relies on robust hash functions, which take an input message (video frames in our case) and generate a compact output hash value, with the condition that similar input messages generate similar output values. All videos involved in the detection are converted into hash values, and detection is performed as a search problem in the hash space.

The detailed description of the CBCR system used is beyond the scope of this paper and can be consulted in the original publication. However, it is necessary to introduce a global overview of how the system works as well as some terminology which we will use in the following sections. We consider a video collection $C = \{V_i : 1 \leq i \leq N\}$ of N elements. Each video $V_i = \{f_j^i : 1 \leq j \leq |V_i|\}$ of the collection is composed of a number, $|V_i|$, of frames, f_j^i. We also consider a set of video queries, denoted by $Q_k = \{f_j^k : 1 \leq j \leq |Q_k|\}$. The system uses the robust hash function and search procedure described in [23] to identify in the incoming stream Q_k any occurrences of a complete video V_i, or a fragment $V_i^{(m,n)} = \{f_j^i : m \leq j \leq n\}$ of it. The ability to detect video fragments enables us to obtain a more comprehensive visual affinity graph, comprising not just near-duplicates, but other forms of redundancy which can also be exploited in the context of this paper. The computational complexity of this process is linear with the database size, but can be reduced using Locality Sensitive Hashing (LSH [16]) techniques to improve performance in more demanding scenarios. We conducted a pilot experiment to validate the viability of this detector using 150 hours obtained by querying for popular music video clips in YouTube, including numerous duplicates and common distortions (different frame rates, resolutions, subtitles, etc). We used this as the query video collection, Q_k, and compared them to a database C of DVD quality versions of the queried songs. The precision-recall break-even point was approximately 0.8.

3.2 Relationships between Videos

If we consider our content database $C = \{V_i : 1 \leq i \leq N\}$ to be a set of YouTube search results, every item V_i can potentially include content from every other. To perform a comprehensive identification, we need to consider the set $C' = C - V_i$ and the input query $Q_i = V_i$ for $i \in [1, N]$, where N denotes the size of the database.

Visual connections between each pair of videos in the analyzed set are expressed in the form $V_i^{(m,n)} \leftrightarrow V_j^{(p,q)}$. A subsequent processing stage is performed, in which connections found for each pair of videos (V_i, V_j) are closely analyzed to classify their relationship. We consider the duration of each pair of videos, $|V_i|$ and $|V_j|$. We also consider the visual overlap between them, $O(V_i, V_j)$, i.e. the video resulting of the frames present in both V_i and V_j. According to these parameters we classify videos as:

- Duplicates: when $|V_i| \approx |V_j|$ and $|O(V_i, V_j)| \approx |V_i|$, both videos are said to be *duplicates*, formally $V_i \equiv V_j$
- Part-of: when $|V_i| > |V_j|$ and $|O(V_i, V_j)| \approx |V_j|$, V_j is said to be *part-of* V_i, formally $V_j \subset V_i$
- Overlap: if $|O(V_i, V_j)| > 0$, both videos are said to *overlap*, formally $V_i \cap V_j \neq \emptyset$. We consider overlaps as a super-set of *duplicate* and *part-of* relationships:

$$V_i \equiv V_j \longrightarrow V_i \cap V_j \neq \emptyset$$

$$V_i \subset V_j \longrightarrow V_i \cap V_j \neq \emptyset.$$

These relationships can be formalized in a visual affinity graph $G_O = (V_O, E_O)$ with undirected weighted edges denoting overlaps. In this abstraction, videos can be considered as single elements instead of frame sets; we will refer to them as v_i in the rest of the paper.

The set of nodes $V_O \subseteq C$ is the subset of videos in the collection having one or more relationships of any kind to others. The set of edges E_O links together visually related videos:

$$E_O = \{\{v_i, v_j\} : v_i \cap v_j \neq \emptyset\},\ v_i, v_j \in V_O \subset C$$

4. AUTOMATIC TAGGING

In this section, we describe two classes of novel methods for automatic tag assignment using content overlap. These are 1) *neighbor-based* methods, which take just immediately overlapping videos, i.e. direct neighbors in the overlap graph into account, and 2) the *TagRank* method based on propagation of tag weights within the visual overlap graph.

4.1 Neighbor-based Tagging

For neighbor-based tagging we consider relationships in the overlap graph $G_O = (V_O, E_O)$ in order to transfer tags between adjacent videos.

4.1.1 Simple Neighbor-based Tagging

For simple neighbor-based tagging, we transform the undirected overlap graph into a directed and weighted graph $G'_O = (V_O, E'_O)$, with (v_i, v_j) and $(v_j, v_i) \in E'_O$ iff $\{v_i, v_j\} \in E_0$. The weight $w(v_i, v_j)$ assigned to an edge (v_i, v_j) reflects the influence of video v_i on video v_j for tag assignment. In this paper we are using the heuristic weighting function

$$w(v_i, v_j) = \frac{|v_i \cap v_j|}{|v_j|} \tag{1}$$

where $|v_j|$ is the (temporal) length of video v_j, and $|v_i \cap v_j|$ denotes the length of the intersection between v_i and v_j. This weighting function describes to what degree video v_j is covered by video v_i. Note that in case where v_i and v_j are duplicates, if v_i is a parent of v_j (meaning that v_i is the more general video, and v_j can be considered as a specific

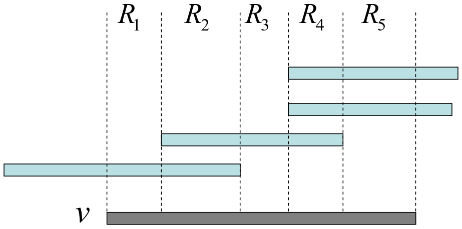


Figure 2: Overlap regions $R_1, \ldots, R_5$ of a video v covered by four other videos

scene from this video) then the weighting function w assigns the maximum value 1 to (v_i, v_j).

How can we use the described relationships and weights for automatic tagging? Let $T = \{t_1, \ldots, t_n\}$ be the set of tags originally (manually) assigned to the videos in V_O and let $I(t, v_i)$ be an indicator function for original tags $t \in T$, with $I(t, v_i) = 1$ iff v_i was manually tagged by a user with tag t, $I(t, v_i) = 0$ otherwise. We compute the relevance $rel(t, v_i)$ of a tag t from adjacent videos as follows:

$$rel(t, v_i) = \sum_{(v_j, v_i) \in E'_O} I(t, v_j) w(v_j, v_i) \tag{2}$$

i.e., we compute a weighted sum of influences of the overlapping videos containing tag t. In this way, we obtain relevance values for all tags from the overlapping videos, and can generate $autotags(v_i)$ of automatically assigned *new* tags for each video $v_i \in V$ using a threshold δ for tag relevancy:

$$autotags(v_i) = \{t \in T | I(t, v_i) = 0 \wedge rel(v_i, t) > \delta\} \tag{3}$$

In order to compute feature vectors (e.g. for clustering or classification) for videos v_i, we use the relevance values $rel(v_i, t)$ of tags t as features weights. Enhanced feature vectors can be constructed as a combination of the original tag weights ($I(t, v_i)$ normalized by the number of tags) and the relevance weights for new, automatically added tags (normalized by the number of tags).

4.1.2 Overlap Redundancy Aware Tagging

For a set of overlapping videos, i.e. neighbors in the graph, situations with multiple redundant overlaps as shown in Figure 2 can occur. In order to avoid a too high increase of the relevance values for automatically generated tags in comparison to original tags, we propose a relaxation method for regions with redundant overlap.

Let $N(v) = \{v_i | (v_i, v) \in E'_O\}$ be the set of overlapping videos for video v. An *overlap region* R_i of v can be defined as a video subsequence $s \subseteq v, |s| > 0$, of maximum length contained in a maximum subset $Y \subset N(v)$, i.e. with none of the remaining videos $N(v) - Y$ overlapping with s. Figure 2 shows an example of a video with 5 overlap regions. For the $k = \sum_{y \in Y} I(t, y)$ videos from Y containing tag t, the contribution of t to an overlap region s is computed by

$$\sum_{i=0}^{k-1} \alpha^i \cdot \frac{|s|}{|v|} \tag{4}$$

i.e., for each additional video contributing the same tag t, this contribution is reduced by a relaxation parameter $0 < \alpha \leq 1$. In order to obtain all contributions to the relevance of tag t to video v, we sum up the contributions for the (disjoint) overlap regions. Putting all pieces together we obtain the following equation for the relevance of tag t for video v:

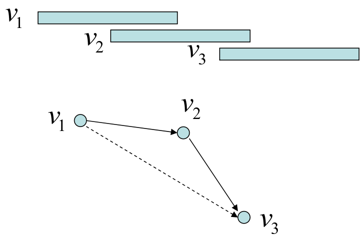


Figure 3: Although there is no overlap between videos v_1 and v_3, a context relationship (dotted line) can be established via the overlap graph.

$$rel(t, v) = \sum_{X \in \mathcal{P}(N(v))} \sum_{i=0}^{k(X)-1} \alpha^i \cdot \frac{\left| v \cap \bigcap_{x \in X} x - \bigcup_{u \in N(v) - X} u \right|}{|v|} \tag{5}$$

where

$$k(X) = \sum_{x \in X} I(t, x) \tag{6}$$

is the number of videos in subset X containing tag t. Thresholds can be applied and feature vectors constructed as described above for the simple case.

4.2 TagRank: Context-based Tag Propagation in Video Graphs

Up to now, we have just taken the direct neighbors of videos in the overlap graph into account, neglecting context relationships as shown in Figure 3. In this subsection we describe a tag weight propagation method which allows for the iterative transfer of tags along paths of arbitrary length. Due to its similarity to PageRank-like weight propagation for web pages, we call the method *TagRank*. However, the objective of TagRank is *not to assign relevance values for the videos* in the overlap graph; instead, TagRank is an alternative method for computing relevance values $rel(t, v)$ of a *tag* t for a given video v.

Let $w(v_i, v_j)$ be the edge weight corresponding to the influence of video v_i to an overlapping video v_j. Then we define the TagRank $\mathrm{TR}(t, v_i)$ for a video v by the following recursive relationship:

$$rel(t, v_i) = \mathrm{TR}(v_i, t) = \sum_{(v_j, v_i) \in E'_O} \mathrm{TR}(v_j, t) w(v_j, v_i) \tag{7}$$

For all videos v_i this computation can be expressed in matrix form as:

$$\mathbf{TR}(t) = \begin{pmatrix} w(v_1, v_1) & w(v_1, v_2) & \cdots & w(v_1, v_n) \\ w(v_2, v_1) & w(v_2, v_2) & \cdots & w(v_2, v_n) \\ \vdots & \vdots & \ddots & \vdots \\ w(v_n, v_1) & w(v_n, v_2) & \cdots & w(v_n, v_n) \end{pmatrix}^{\mathrm{T}} \cdot \begin{pmatrix} \mathrm{TR}(v_1, t) \\ \mathrm{TR}(v_2, t) \\ \vdots \\ \mathrm{TR}(v_n, t) \end{pmatrix} \tag{8}$$

This Eigenvector equation can be solved using power iteration. Similar to Kleinbergs HITS [17] the rows are not guaranteed to sum up to 1, and re-normalizations of the rank vector are required. In contrast to the Random Surfer Model for PageRank, we don't consider the possibility of random jumps within the video graph. For the TagRank method to converge, this is not necessary because sinks as in the Web graph are impossible due to the reflexivity of the overlap relationship. Furthermore, this enables us to perform the TagRank computation separately, and thus more efficiently, for each connected component, which is crucial because of the high number of tags.

Another aspect is that we want to take the original (manually generated) tag assignments into account. If we would simply consider the solution for Equation 8 we would lose this information because for a given node v in a connected component the solution would eventually converge to the same value $\mathrm{TR}(v, t)$ for each tag t. This is not intuitive and instead we perform a limited number Γ of iterations (where Γ is a tuning parameter) using the original tag assignment in the form

$$\mathbf{TR}(t) = \Big(I(t, v_1), \ldots, I(t, v_n)\Big)^{\mathrm{T}} \tag{9}$$

as start vector for the TagRank iterations. Limiting the number of iterations results in higher weights for tags from videos in the closer neighborhood, and is similar to the strategy deployed in [6] for random walks in click graphs.

5. EVALUATION

In this section we present the results of our evaluation for automatic tagging. We first describe our strategy for gathering a video collection from YouTube, and elaborate on the characteristics of our data set. Then, we present the outcome of our two-fold evaluation methodology: 1) we examine the influence of the enhanced tag annotations on automatic video classification and clustering; 2) we provide the results of a user evaluation by direct relevance assessment of the automatically generated tags for a set of videos.

5.1 Data

5.1.1 Data Collection

We created our test collection by formulating queries and subsequent searches for "related videos", analogously to the typical user interaction with the YouTube system. Given that an archive of most common queries does not exist for YouTube, we selected our set of queries from Google's Zeitgeist archive from 2001 to 2007. These are generic queries, used to search for web pages and not videos. In order to remove queries not appropriate to video search (e.g. "windows update") we removed those for which YouTube returned less than 100 results. In total, 579 queries were accepted, for which the top 50 results were retrieved. Altogether, we collected $28,216$ videos using those queries (some of the results were not accessible during the crawling because they were removed by the system or by the owner). A random sample of these videos was used to extend the test collection by gathering related videos, as supported by the YouTube API. In total, 267 queries for related videos were performed, generating $10,067$ additional elements.

The complete collection, C, used for the evaluation had a final size of $38,283$ videos, 390 GB of information comprising over 2900 hours of video with an average duration of 4:15 min per video. We used the 'Fobs project' [22] as the programming interface to access content and characteristics of downloaded videos.

Table 1: Results of CBCR analysis for collection C.

Set	Size	Proportion
$\|C\|$	38,283	100%
$\|V_O\|$	13,672	35.71%
Duplicates, $\{v_i : \exists v_j, v_i \equiv v_j\}$	6,051	15.80%
Children, $\{v_i : \exists v_j, v_i \subset v_j\}$	3,570	9.32%
Parents, $\{v_i : \exists v_j, v_j \subset v_i\}$	3,200	8.35%

5.1.2 Visual Redundancy Analysis

We analyzed the collected set C to quantify the presence of visual connections in them, and studied the distribution of the different relationships. We considered the study of $G_O = (V_O, E_O)$, the graph of related videos, as well as the relevance of each kind of relationship, computing the subset of nodes having each of the defined visual links.

Table 1 summarizes the results. A significant proportion of videos in the test collections, over 35%, feature one or more connections to other elements. These videos, denoted by the set V_O, represent the subset of the collection that we used to evaluate our tag propagation methods in Section 5.2. They are organized in 3779 connected components, with an average size of 3.61 videos per component. Note that any video can be linked by more than one relationship, contributing to the different categories that we analyze below.

Duplication is the most common form of visual relationship, accounting for over 15% of the videos. Note that duplicated videos share the exact same visual content, and their tags should be intuitively valid in any element of the clique they belong to. The high presence of this relationship suggests the occurrence of many useful links which can be exploited to propagate relevant tags. *Part-of* relationships have also noticeable presence in the set, though their amount is considerably lower than duplicates, with an average of around 9%.

5.2 Data Organization using Automatically Generated Tags

In Section 4, we have presented different methods for automatically generating tags, resulting in richer feature representations of videos. Machine learning algorithms make use of this feature information to generate models, and to automatically organize the data. In this section, we will show results for classification (supervised learning) as well as clustering (unsupervised learning) of YouTube videos using feature vectors obtained by automatic tagging.

5.2.1 Classification Experiments

Classifying data into thematic categories usually follows a supervised learning paradigm and is based on training items that need to be provided for each topic. We used linear support vector machines (SVMs) in our experiments, as they have been shown to perform very well for text-based classification tasks (see, e.g.,[7]).

As classes for our classification experiments, we chose the YouTube categories containing at least 900 videos in our data set. These were the 7 categories "Comedy", "Entertainment", "Film & Animation", "News & Politics", "Sports", "People & Blogs", and "Music". We did this in order to obtain equal numbers of training/test videos per category, omitting videos without category label as well as categories containing fewer videos. We performed binary classification experiments for all $\binom{7}{2} = 21$ combinations of these class pairs (e.g. "Music" vs. "Sports"). Settings with more than two classes can be reduced to multiple binary classification problems that can be solved separately [2]. For each category, we randomly selected 400 videos for training the classification model and a disjoint set of 500 videos for testing the model. We trained different models based on T=10,25,50,100, 200, and all 400 training videos per class.

We compared the following methods for constructing feature vectors from video tags:

1. **BaseOrig**: Vectors based on the original tags of the videos (i.e. tags manually assigned by the owner of the video in YouTube). This serves as the baseline for the comparison with our the vector representations based on automatic tagging.
2. **NTag**: Vectors constructed based on the tags and their relevance values produced by simple neighbor-based tagging described in Section 4.1.1 in addition to the original tags.
3. **RedNTag**: Vectors using tags generated by overlap redundancy aware neighbor-based tagging plus the original tags as described in Section 4.1.2. We did not pursue any extensive parameter tuning and chose $\alpha = 0.5$ for the relaxation parameter.
4. **TagRank**Γ (with Γ = 2,4,8 iteration steps): Vectors using, in addition to the original tags, new tags produced by the TagRank algorithm described in Section 4.2.

Our quality measure is the fraction of correctly classified videos (*accuracy*). Finally, we computed micro-averaged results for all topic pairs. The results of the comparison are shown in Table 2. The main observations are:

- Classification taking automatically generated tagging into account clearly outperforms classification using just the original tags. This holds for all of the three introduced tagging methods. For classification using 50 training documents per class, for example, we increased the accuracy from approximately 70% to 76%.

- Overlap redundancy aware neighbor-based tagging provides slightly but consistently more accurate results than the simple neighbor-based tagging variant.

- Both of the Neighborhood-based methods outperform TagRank, although the later might seem the more appealing method from a theoretical point of view. We observed that the accuracy decreased with a higher number of iterations. Context relationships link together videos with no overlapping content by means of a common neighbor (see Figure 3). With each additional iteration, tags are pushed farther in the link graph. The relationship between source and targets of the propagated tags becomes increasingly weak, introducing increasing amounts of noise and which explains the systematic decrease of accuracy.

5.2.2 *Clustering Experiments*

Clustering algorithms partition a set of objects, YouTube videos in our case, into groups called *clusters*. For our experiments we choose *k-Means* [10], a simple but still very popular and efficient clustering method. Let k be the number of classes and clusters, N_i the total number of clustered documents in $class_i$, N_{ij} the number of documents contained in $class_i$ and having cluster label j. Unlike classification results, the clusters do not have explicit topic labels. We define the clustering accuracy as follows:

$$accuracy = \max_{(j_1,\ldots,j_k)\in perm((1,\ldots,k))} \frac{\sum_{i=1}^{k} N_{i,j_i}}{\sum_{i=1}^{k} N_i} \qquad (10)$$

Accuracy for clustering can be seen as a measure of how good the partitioning produced by a clustering method reflects the actual class structure. Note that, similar to some other measures of cluster validity known in the literature, the minimum value for clustering accuracy is larger than 0 ($1/k$ for the case of equal number of items in each of the k classes).

For a number of clusters $k = 2, 3, 4, 5$ we considered all possible $\binom{7}{k}$ combinations of tuples of classes (e.g. the triple "Music" vs. "Sports" vs. "Film & Animation") for the 7 YouTube categories mentioned in Section 5.2.1. For each tuple, we randomly selected 500 videos per class, performed the k-means algorithm and computed the macro-averaged accuracy for the k-tuples.

We constructed feature vectors based on the original tags (**BaseOrig**), simple neighbor-based tagging (**NTag**), overlap redundancy aware neighbor-based tagging (**RedNTag**), and TagRank (**TagRank**Γ with number of iterations Γ = 2,4,8) in the same way as for the classification experiments described in the previous Section 5.2.1. The results of the comparison are shown in Table 3. The main observations are very similar to the supervised scenario:

- Clustering using automatically generated tags clearly outperforms clustering with features just obtained from the original tags. For example, for clustering with $k = 3$ we increased the accuracy from approximately 40% to 55%.

- Overlap redundancy aware neighbor-based tagging outperforms simple neighbor-based tagging; both of the neighbor-based techniques outperform TagRank.

5.3 User-based Evaluation

To support the results obtained for automatic data organization, we conducted an additional user-based experiment. Three assessors provided relevance judgments for the automatically generated tags. For this experiment we chose the neighbor based methods (**NTag** and **RedNTag**) as they provided the best results for both classification and clustering.

A web service was implemented to help the assessors to provide their judgments. The interface included a playable version of the video, automatically extracted key-frames, the title and the description to help them understand the content, so they could provide more accurate judgments. The videos presented to the evaluators were randomly selected and the tags were displayed in random order. The evaluators were asked to rate each new tag using a five-level Likert scale [18]. This interface is depicted in Figure 4.

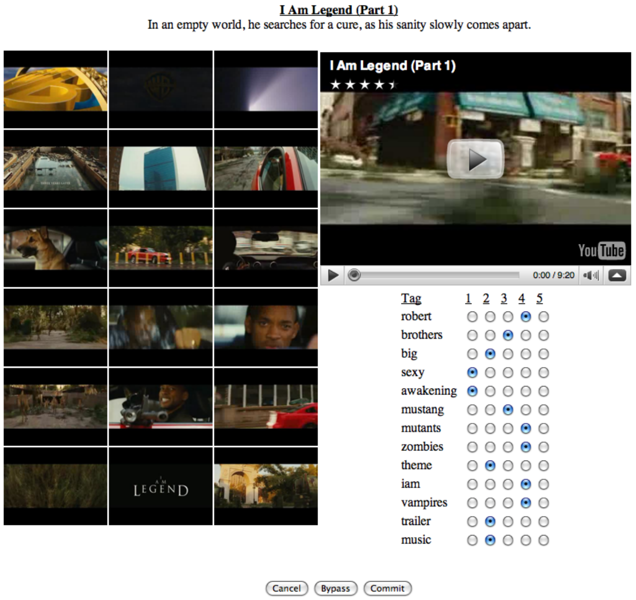
A total of 3578 tags, for 300 different videos, were manually assessed using the described interface. (Manual evaluation of video annotations can require a substantial amount of time including the inquiry of background information.) We studied the average relevance for the set of generated tags, $autotags(v_i)$, for different values of threshold δ (see

Table 2: Classification with T=10,25,50,100,400 training videos using different tag representations for videos

	BaseOrig	NTag	RedNTag	TagRank2	TagRank4	TagRank8
T = 10	0.5794	0.6341	0.6345	0.6024	0.5950	0.5951
T = 25	0.6357	0.7203	0.7247	0.6821	0.6646	0.6544
T = 50	0.7045	0.7615	0.7646	0.7317	0.7228	0.7165
T = 100	0.7507	0.7896	0.7907	0.7673	0.7630	0.7598
T = 200	0.7906	0.8162	0.8176	0.8001	0.7977	0.7955
T = 400	0.8286	0.8398	0.8417	0.8363	0.8322	0.8300

Table 3: Clustering with k=2,3,4,5 video clusters using different tag representations for videos

	BaseOrig	NTag	RedNTag	TagRank2	TagRank4	TagRank8
k = 2	0.5487	0.6361	0.6257	0.6156	0.6164	0.6003
k = 3	0.4028	0.5183	0.5460	0.5016	0.4729	0.4673
k = 4	0.3388	0.4411	0.4917	0.4019	0.4066	0.4040
k = 5	0.3031	0.3971	0.4280	0.3608	0.3597	0.3538



I Am Legend (Part 1)

In an empty world, he searches for a cure, as his sanity slowly comes apart.

Tag	1	2	3	4	5
robert	○	○	○	●	○
brothers	○	○	●	○	○
big	○	●	○	○	○
sexy	●	○	○	○	○
awakening	●	○	○	○	○
mustang	○	○	●	○	○
mutants	○	○	○	●	○
zombies	○	○	○	●	○
theme	○	●	○	○	○
iam	○	○	○	●	○
vampires	○	○	○	●	○
trailer	○	●	○	○	○
music	○	●	○	○	○

Figure 4: Web interface for the collection of relevance judgments

equation 3). For this purpose, we sorted the list of tags in decreasing order of $rel(t, v_i)$ value, and selected δ values at different levels of that list, in increments of 10% of the full $autotags(v_i)$ size. The results are shown in Figure 5.

The figure reveals an expected decreasing relevance pattern for growing values of $autotags(v_i)$ set sizes. By raising the threshold δ, we can obtain increasingly higher average relevance values for the sets of new tags. When considering the 10% best rated $autotags(v_i)$ for each method, **RedNTag** achieves higher relevance in comparison with **NTag**. On average, the difference between both methods is small, in consonance to results obtained for automatic data organization.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown that content redundancy in social sharing systems can be used to obtain richer annotations for shared objects. More specifically, in what is a novel hybrid approach, we have used content overlap in the video sharing environment YouTube to establish new connections between videos forming a basis for our automatic tagging methods. Classification and clustering experiments show that the additional information obtained by automatic tagging can largely improve automatic structuring and organization of content; our preliminary user evaluation indicates an information gain for viewers of the videos.

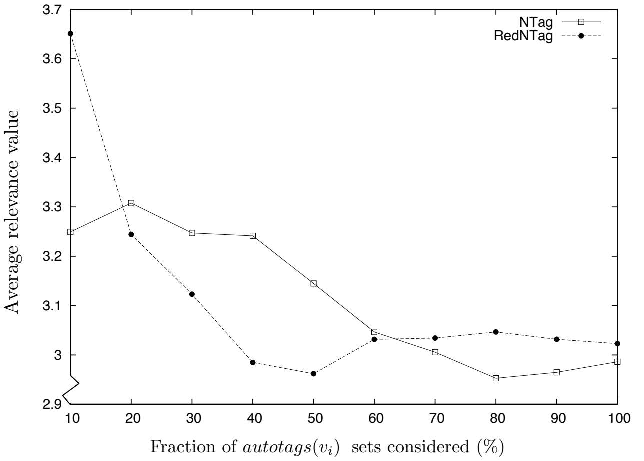


Figure 5: Average relevance judged manually by assessors for increasing sizes of $autotags(v_i)$ considered

We plan to take confidence values for visual overlap detection into account in order to generate a smoothing function to reduce the contribution of weak edeges to the overall graph. We also plan to extend and generalize this work to consider links between various kinds of resources such as videos, pictures (e.g. in Flickr) or text (e.g. in del.icio.us), and use them to extract additional information (e.g. in form of tags or other metadata) to improve knowledge management and information retrieval processes. We also plan to consider the analysis and generation of deep tags (i.e. tags linked to a small part of a larger media resource) which are starting to be supported by the main Web 2.0 sites. Furthermore, we aim to refine the defined propagation methods by introducing smoothing methods to improve context-based tag propagation.

We think that the proposed techniques have direct applications to search improvement, where augmented tag sets can reveal resources previously concealed. In this connection, integration and user evaluation within a system context and encompassing additional complementary retrieval and mining methods is of high practical importance.

7. ACKNOWLEDGEMENTS

This research was partially funded by the EU Marie Curie ToK Grant *Memoir* (MTKD-CT-2005-030008), and the Large-Scale Integrating EU Project *LivingKnowledge*.

8. REFERENCES

[1] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR '98*, pages 37–45. ACM Press, 1998.

[2] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2001.

[3] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC '07*, pages 1–14, NY, USA, 2007. ACM.

[4] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, NY, USA, 2002. ACM.

[5] X. Cheng, C. Dale, and J. Liu. Understanding the characteristics of internet short video sharing: Youtube as a case study, Technical Report arXiv:0707.3670v1 [cs.NI], Cornell University, arXiv e-prints, July 2007.

[6] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR'07*, pages 239–246, 2007.

[7] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM '98*, pages 148–155, Bethesda, Maryland, United States, 1998. ACM Press.

[8] N. Shivakumar and H. Garcia-Molina. Scam: A copy detection mechanism for digital documents. In Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries. June 1995.

[9] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *IMC '07: Proceedings of ACM SIGCOMM*, pages 15–28, New York, USA, 2007.

[10] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2001.

[11] J. S. Hare, P. H. Lewis, P. G. B. Enser, and C. J. Sandom. Mind the gap: another look at the problem of the semantic gap in image retrieval. *Multimedia Content Analysis, Management, and Retrieval 2006*, 6073(1), 2006.

[12] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information Retrieval in Folksonomies: Search and Ranking. In *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, 2006. Springer.

[13] S. Huffman, A. Lehman, A. Stolboushkin, H. Wong-Toi, F. Yang, and H. Roehrig. Multiple-signal duplicate detection for search evaluation. In *SIGIR '07*, pages 223–230, New York, USA, 2007. ACM.

[14] Y. Jing and S. Baluja. Pagerank for product image search. In *WWW '08*, pages 307–316, New York, NY, USA, 2008. ACM.

[15] A. Joly, O. Buisson, and C. Frelicot. Content-based copy retrieval using distortion-based probabilistic similarity search. *Multimedia, IEEE Transactions on*, 9(2):293–306, 2007.

[16] Y. Ke, R. Sukthankar, and L. Huston. An efficient parts-based near-duplicate and sub-image retrieval system. In *ACM Multimedia, MM'04*, pages 869–876, New York, USA, 2004. ACM Press.

[17] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[18] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, 1932.

[19] L. Liu, W. Lai, X.-S. Hua, and S.-Q. Yang. Video histogram: A novel video signature for efficient web video duplicate detection. *Advances in Multimedia Modeling*, pages 94–103, 2006.

[20] G. S. Manku, A. Jain, and A. D. Sarma. Detecting near-duplicates for web crawling. In *In ACM WWW'07*, pages 141–150, NY, USA, 2007. ACM.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[22] J. San Pedro. Fobs: an open source object-oriented library for accessing multimedia content. In *ACM Multimedia, MM '08*, pages 1097–1100, 2008.

[23] J. San Pedro and S. Dominguez. Network-aware identification of video clip fragments. In *CIVR '07*, pages 317–324, New York, USA, 2007. ACM Press.

[24] N. Stokes and J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In *SIGIR '01*, pages 424–425, New York, USA, 2001. ACM.

[25] B. Szekely and E. Torres. Ranking bookmarks and bistros: Intelligent community and folksonomy development. In *http://torrez.us/archives/2005/07/13/tagrank.pdf (unpublished)*, 2005.

[26] S. van Dongen. A cluster algorithm for graphs. *National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, Technical Report INS-R0010*, 2000.

[27] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *ACM Multimedia, MM'07*, pages 218–227, 2007.

[28] H. Yang and J. Callan. Near-duplicate detection by instance-level constrained clustering. In *SIGIR '06*, pages 421–428, New York, USA, 2006. ACM.

[29] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *SIGIR '05*, pages 504–511, New York, USA, 2005. ACM.

[30] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR '02*, pages 81–88, New York, USA, 2002. ACM.