# Content Redundancy in YouTube and Its Application to Video Tagging

JOSE SAN PEDRO, Telefonica Research, and Penn State University
STEFAN SIERSDORFER, L3S Research Center
MARK SANDERSON, RMIT University

The emergence of large-scale social Web communities has enabled users to share online vast amounts of multimedia content. An analysis of YouTube reveals a high amount of redundancy, in the form of videos with overlapping or duplicated content. We use robust content-based video analysis techniques to detect overlapping sequences between videos. Based on the output of these techniques, we present an in-depth study of duplication and content overlap in YouTube, and analyze various dependencies between content overlap and meta data such as video titles, views, video ratings, and tags. As an application, we show that content-based links provide useful information for generating new tag assignments. We propose different tag propagation methods for automatically obtaining richer video annotations. Experiments on video clustering and classification as well as a user evaluation demonstrate the viability of our approach.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia

General Terms: Algorithms

Additional Key Words and Phrases: Video duplicates, content-based links, automatic tagging, neighbor-based tagging, tag propagation, data organization

**ACM Reference Format:**
San Pedro, J., Siersdorfer, S., and Sanderson, M. 2011. Content redundancy in YouTube and its application to video tagging. ACM Trans. Inf. Syst. 29, 3, Article 13 (July 2011), 31 pages.
DOI = 10.1145/1993036.1993037 http://doi.acm.org/10.1145/1993036.1993037

## 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 inexperienced users, suitable mechanisms for supporting collaboration, and attractiveness of shared annotated material (images in Flickr, bookmarks in del.icio.us, etc.). Recent studies have shown that traffic to/from YouTube accounts for over 20% of the Web total and 10% of the whole internet [Cheng et al. 2007], comprising 60% of the videos watched online [Gill et al. 2007]. It has also been shown that a noticeable amount of visually redundant footage (i.e. near-duplicate videos) exist in such Websites [Cha et al. 2007; Wu et al. 2007], with a reported average of >25% near-duplicate videos detected in search results. Previous literature developed technology to dynamically filter redundancy aiming at enhancing content retrieval. However, no previous works considered methods to leverage redundancy for knowledge extraction purposes.

This work was supported by the European Union FP7 integration project LivingKnowledge (Contract No. 231126) and the European Union FP7 Marie Curie International Outgoing Fellowships project for career development Mieson (PIOF-GA-2009-254370).
Author's address: email: jsanpedro@mac.com.

DOI 10.1145/1993036.1993037 http://doi.acm.org/10.1145/1993036.1993037

We show that, by the use of appropriate content-based analysis tools, duplication can be used to understand collective behavior, and point to ways of improving video Websites. Specifically, we use content-based copy detection (CBCR) tools to establish connections between videos that share duplicate scenes. This analysis determines the types of redundancy present, and examines properties of redundant content.

Social sharing systems are primarily supported by user generated metadata, in particular tags. Users tend to tag proactively [Lipczak and Milios 2010; Ames and Naaman 2007; Marlow et al. 2006]. In contrast to ontologies, tagging-based systems impose no restrictions on user annotations. However, manually annotating content is an intellectually expensive and time consuming process. Consequently, annotations are often sparse [Abbasi and Staab 2009; Golder and Huberman 2006]. Furthermore, keywords and community-provided tags lack consistency and present numerous irregularities (e.g., abbreviations and mistypes) [Krestel et al. 2009].

The research community is actively working on overcoming the difficulties posed by tag sparseness. Automatic tagging is the prevalent line of work in this field, with many techniques successfully proposed for improving the quality of tag assignments in an unsupervised fashion [Krestel et al. 2009].

In this article, we leverage robust Content-Based Copy Detection (CBCR) techniques to establish a set of graphs where edges connect highly related videos. We show that we can improve the overall quality of annotations. We present a hybrid approach combining content-based video duplication analysis and graph algorithms to generate tag assignments in community websites. We use these new content-based links to propagate tags to adjacent videos, utilizing visual affinity to spread community knowledge in the network.

*Outline.* The remainder of this article is structured as follows. Section 2 discusses related work; Section 3 presents an overview on duplicate and overlap detection for videos, evaluating these techniques in the YouTube context, and providing a graph formalization of relationships found between videos. An extensive empirical analysis of duplication and dependencies between content overlap and video metadata (tags, titles, views, ratings, etc.) is discussed in Section 4. We describe several techniques for automatic tagging in Section 5. Section 6 provides the results of the evaluation of our automatic tagging methods for YouTube video clustering and classification as well as a user study. We conclude in Section 7.

## 2. RELATED WORK

In the context of multimedia social Websites, specifically YouTube, we can classify previous works into two groups. Firstly, studies centered on the the analysis of YouTube network traffic: Holding an estimated 83.4 million videos, YouTube requires around 1 Petabyte of storage capacity [Gill et al. 2007]. Comprehensive traffic characterizations have been provided, and guidelines for handling the new generation of content have been established, both for Networks and Service Providers [Gill et al. 2007; Cha et al. 2007]. Secondly, some studies focused on social aspects of YouTube: Halvey and Keane [2007] showed that YouTube users do not make extensive use of the community interaction tools provided. They also revealed the existence of a correlation between descriptiveness of the metadata (e.g., tags, title) and number of views of a video. The popularity of videos was shown to decrease after an initial boost, in the general case. Cheng et al. [2007] contrasted YouTube's short video sharing approach with other video repositories, finding noticeable differences in access patterns and life span of videos.

Recently, a number of works have analyzed YouTube content, focusing on the enhancement of retrieval and user experience. Wu et al. [2007] analyzed YouTube search results and found a large amount of duplication (25%) at the top ranks. An approach based on video copy detection was proposed to detect near-duplicate videos in order to promote diversity of search results. Search diversification was also proposed by Liu et al. [2006] using a different strategy that considered also text to establish links between videos. However, no important research efforts were directed towards the characterization of video sharing in this vast community. In this article we cover such a characterization of the YouTube video collection.

A large amount of literature considered user-generated data for different applications. The community interacts with a system by accessing certain resources, manually ranking them, commenting on them, making them favorites, etc. The result of this interaction is a complex set of links between resources, content, and annotations. Complex networks are common in related fields, such as Social Network Analysis. A well-known application of network analysis is the PageRank algorithm [Page et al. 1998]. A node ranking procedure for folksonomies, FolkRank was proposed in Hotho et al. [2006]. It operates on a tripartite graph of users, resources and tags, 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 to detect communities of nodes in weighted networks [van Dongen 2000]. A PageRank-like algorithm based on visual links between images is used to improve the ranking function for photo *search* in Jing and Baluja [2008].

We can also find examples of algorithms leveraging "implicit" links [Zhang et al. 2005], where connections are inferred from user actions (e.g., access patterns), such as near duplicate document detection [Yang and Callan 2006; Charikar 2002; Manku et al. 2007]. A graph of interdocument links can be used to perform common procedures in web process optimization, such as web crawl filtering [Manku et al. 2007], enhanced ranking schemes [Zhang et al. 2005], first story detection [Stokes and Carthy 2001], or plagiarism detection [Shivakumar and Garcia-Molina 1995]. These techniques rely on the textual nature of documents. In this paper, we focus on exploiting *visual* relationships available in richer multimedia scenarios.

We consider the specific scenario of improving tag quality in YouTube. Tags are a critical part of YouTube, but tag sparseness affects retrieval qualtiy, and automatic tagging techniques have been proposed to enhance quality of annotations in multimedia sharing communities. Automatic tagging of multimedia resources is closely related to content-based image retrieval (CBIR), for which comprehensive literature reviews exist [Smeulders et al. 2000; Datta et al. 2005].

In general, automatic tagging methods use content and text features to derive models for predicting tag assignments. Generative mixture models have been widely used to learn joint probability distributions of visual features and vocabulary subsets [Li and Wang 2006]. In contrast, discriminative models learn independent models for each target vocabulary term [Lindstaedt et al. 2009; Grangier and Bengio 2008], and are commonly used to create concept detectors [Snoek and Worring 2008]. The problem can also be posed as a machine translation task, where models are created to translate visual features into captions [Duygulu et al. 2006]. All of these machine learning-based methods have the common problem of not being extensible to large sets of concepts. In uncontrolled visual content scenarios, diversity of visual appearance, even from the same concept as a result of the sensory gap, has a direct impact on tag generation performance [Smeulders et al. 2000]. Using a similar idea, other works related to automatic tagging consider the scenario of neighbor-based tag assignments. Guillaumin et al. [2009] propose a method to predict tags by analyzing the presence or absence of them in near neighbors. A similar method is proposed by Li et al. [2008], but in this case neighbors are used to compute relevance values for the original set of tags without creating new tag assignments. In contrast to these works, we investigate neighbor-based tag propagation strategies for videos where neighbors are established by visual overlap.

A different strategy is to generate new tag assignments based mainly on viewing a set of tagged resources as text corpus, and expanding existing annotations. Krestel et al. [2009] use Latent Dirichlet Allocation to learn a concept space from the original tag-document matrix. Resources are then expressed as a sum of weighted concepts. Each concept contributes a set of tags to the image, and concept weights are used to compute relevance values for the new tag assignment. The method has been shown to outperform the approach described in Heymann et al. [2008], which makes more direct use of tag cooccurrences by mining association rules. A similar idea as in Heymann et al. [2008] is proposed by Sigurbjörnsson and van Zwol [2008], in this case using global tag cooccurrence metrics in the massive Flickr dataset, which provides a set of recommended tags based on the original annotations.

Video content-based copy detection (CBCR) has received noticeable attention from the multimedia research community for a number of applications, for instance, copyright infringement detection or database purge. The main difficulty of copy detection is dealing with the so-called *sensory gap* [Smeulders et al. 2000], that is, the differences between the original video footage and its final presentation. Tolerance to common artifacts and noise is, therefore, required. Fingerprints are the most commonly used detection tool; they are generated using specific features of moving visual content, such as temporal video structure [San Pedro et al. 2005], or time-sampled frame invariants [Joly et al. 2007]. Robust hash-based fingerprints have been widely used in CBCR systems [Wu et al. 2007]: videos are transformed into a sequence of hash values, and detection is performed as a search problem in the hash space. Additionally, extensive work on audio-based duplicate detection exists with direct application to video copy detection [Liu et al. 2010; Lebossé et al. 2007; Jang et al. 2009]. However, in YouTube, two duplicate videos may be visually identical but feature completely different audio tracks (a typical example being dubbed movies); equally, visually different videos may have the same audio track (it is common to find in YouTube picture slideshow videos featuring popular songs as their audio background).

The content presented in this article is partially based on our own work [Siersdorfer et al. 2009], which mainly focuses on the aspect of automatic tagging. This article includes, amongst other things, a detailed specification of the CBCR system built, a new in-depth study of duplication and content overlap in YouTube, and new comparisons and series of experiments on automatic tagging.

## 3. CONTENT LINKS BETWEEN VIDEOS

### 3.1. Duplication and Overlap Detection on Videos

Fingerprint-based CBCR systems follow a general framework comprising the following components.

—Fingerprint generation module. Transforms video content into a sequence of points in the fingerprint feature space. All videos processed by the system are transformed before being used by any further module.

—Reference content database. Fingerprint-based systems compute signatures directly from the incoming video stream, and compare against a database of known fingerprints. We refer to the reference content database as $C = \{V_i : 1 \leq i \leq N\}$, where $N$ denotes the number of elements. Each video $V_i = \{f_j^i : 1 \leq j \leq |V_i|\}$ of $C$ is composed of $|V_i|$ frames, $f_j^i$.

—Search module. Matches a query fingerprint to the reference content database and returns a comprehensive matching report.

To create a reference content database, all videos are processed using the fingerprint generation module. The resulting signature is then stored in the database ready for matching against video queries, denoted by $Q_k = \{f_j^k : 1 \leq j \leq |Q_k|\}$. Most systems assume that queries $Q_k$ match only single elements of the reference database [Joly et al. 2007]. As we will see in Section 3.2, this restriction imposes hard limitations on the effective number of different overlapping relationships the system is able to detect in the database. In our system, we generalize the search module to identify in the incoming video stream, $Q_k$, any occurrences of one *or more* complete videos $V_i$ or fragments denoted as $V_i^{(m,n)} = \{f_j^i : 1 \leq m \leq j \leq n \leq |V_i|\}$.

*3.1.1. Implementation Details.* Our CBCR system is based on our previous work [San Pedro and Dominguez 2007]. This choice is motivated by the capabilities of the system for our application setting. It uses visual hash fingerprints and can handle identification of heavily transformed variations. Furthermore, it is designed to cope with transformations in the temporal dimension, including frame rate resampling and temporal crop. The latter of these features allows for the detection of partial matches.

Fingerprint computation is performed using a two-fold strategy. Firstly, input videos are subsampled to remove fingerprint temporal dependencies. Secondly, a robust hash function is applied on the resulting frame set to generate the final hash value string.

*Content-based video subsampling*. Hash fingerprint generation is sensitive to dynamic conditions of temporal video parameters. Different frames from multiple variations of the same original video could be selected to generate a fingerprint, leading to signatures different in size and/or value. We use a content-based subsampling technique to ensure identical sets of frames are chosen from any variation of the original video. Our method considers video frame entropy series to achieve such a synchronized subsampling of video sequences.

Shannon's entropy can be used to obtain a useful representation of the color distribution of frames in a video sequence. Given a random variable $X$, which takes on a finite set of values according to a probability distribution, $p(X)$, the entropy of this probability distribution is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i). \tag{1}$$

The entropy of an image can be computed to obtain a quantitative value of its color complexity by considering $X$ as the intensity of a random pixel in the image. Entropy values are higher in images with wider histograms, as colors of pixels are more uniformly distributed. Maximum entropy is therefore obtained when the color distribution is equiprobable, while minimum entropy happens in monochrome images where pixels are contained within a single histogram bin.

We compute an entropy value for each video frame to create an entropy time series. These time series are analyzed to detect salient points, which we use to determine the frames that will be selected in our subsampling strategy. It is important to note that we purposely consider a reference-less frame selection scenario from a temporal perspective. Relying, for instance, on the first frame of a video and subsequently using a time-based approach to select frames will obviously fail to generate a repeatable sequence of frames for different variations (for instance, consider the situation of a variation that lacks the first 5 frames of the original video).

In order to cope with dynamic temporal features, we based our subsampling algorithm on the content-based analysis of *delimited temporal intervals*. Videos are segmented in time intervals $I_k$, and a set of synchronization frames are selected from each interval. This process produces the aimed results if the following conditions are met.

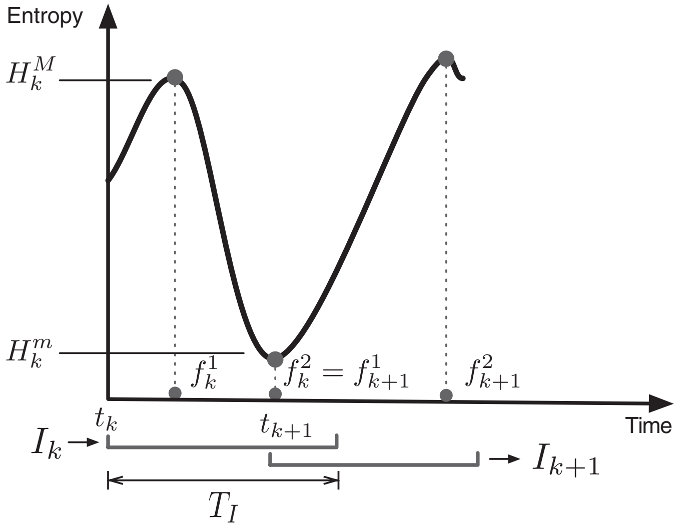


Fig. 1. Example of interval synchronization. Entropy series are analyzed in discrete intervals, $I_k$. Extrema points, $H_k^M$ and $H_k^m$, are selected from each interval and used to extract synchronization frames. This strategy allows for identical selection of frames from different video variations. On the other hand, intervals get synchronized by dynamically changing their overlap. The starting point of $I_{k+1}$ is determined by the time of the latest synchronization frame selected from interval $I_k$, denoted by $f_k^2$.

—Given the same time interval, $I_k$, of two variations of the same video, the set of chosen synchronization frames is identical.
—Intervals chosen for two variations of the same video are identical.

The remainder of this section describes the algorithm used to meet both conditions in order to produce repeatable video subsampling sets.

*Selection of synchronization frames.* In this stage, our algorithm analyzes a given interval, $I_k$, of the incoming video, which spans from $t_k^0$ to $t_k^f$, and selects a fixed number, $N$, of *synchronization* frames, that is, frames chosen by the algorithm to represent that interval in any variation of the video:

$$F_k = \{f_k^1, \ldots, f_k^N\}, \tag{2}$$

where $F_k$ is the set of synchronization frames for interval $I_k$, and selected frames are represented by their timestamp, $f_k^i$, in ascending order, i.e. $f_k^i < f_k^{i+1}$. We use a Max/Min selection approach, where frames with absolute maximum, $H_k^M$, and minimum, $H_k^m$, entropy values are selected. This approach has the following features:

—The number of chosen frames per interval, $N$, is 2.
—When there are several frames which share the absolute maximum or minimum values, the algorithm chooses $f_k^1$ and $f_k^2$ so $D_k = f_k^2 - f_k^1$ is maximized.

Figure 1 illustrates the approach. The specific selection criterion relies on the property of entropy to average out video content in a color complexity metric, which reduces sensitivity to changes in visual parameters. This allows to select nearly identical frames from the same time interval of different variations whenever the visual content of both videos is perceptually similar. Severe reduction of visual quality parameters might lead to loss of synchronization. The actual precision-recall trade-off is therefore dependent on the content features and criteria used to select the frames. Our pilot study described in Section 3.1.2 shows how this approach provides satisfactory discriminative characteristics in our specific application setting.

*Synchronization of intervals.* In this stage, our algorithm determines the next interval to be analyzed. We aim at defining identical intervals for different variations, independently of their dynamic temporal features. To this end, we treat intervals as sliding windows of fixed size and *variable overlapping area*.

Consider interval $I_k$ defined by its starting point $t_k$. This interval spans from $[t_k, t_k + T_I]$, where $T_I$ denotes the fixed interval length. The set of salient frames chosen for this interval is denoted by $\{f_k^1, f_k^2\}$, where $t_k \leq f_k^1 \leq f_k^2 \leq t_k + t_I$. The algorithm dynamically adjusts the overlapping area of the next interval, $I_{k+1}$, to start at $t_{k+1} = f_k^2$, i.e. $I_{k+1}$ starts just after $f_k^2$, the time of the latest frame selected for $I_k$. Figure 1 illustrates this procedure.

By dynamically adjusting the overlapping area of sliding windows, the algorithm successfully synchronizes between video variations with temporal crops. Many conditions in the entropy timeline cause the algorithm to effectively synchronize intervals of two variations, for instance, the presence of two local extrema within a single interval length, $T_I$. Note that disparity between intervals is limited by the interval size and is always below $\frac{T_I}{2}$. Therefore, intervals as well as synchronization frames tend to converge rapidly as they are selected in salient points of the entropy series. This fact is widely covered and illustrated in the original publication [San Pedro and Dominguez 2007].

*Visual hash function.* We use a simple visual hash function based on Oostveen et al. [2001]. This hash function uses features extracted from the luminance channel. It is straightforward to include chromatic information in the process, though at the cost of increasing the number of bits required to represent hash values. This extension increases fingerprint length and adds computational complexity to the search module. For this reason, we chose the simpler luminance-based function; our pilot experiment shows that accuracy values obtained using this strategy are satisfactory for duplicate detection in the YouTube scenario.

The chosen hash function divides frames into a $4 \times 4$ rectangular grid. Each block contributes a bit to the final signature, for a length of 16 bit per frame pair. Bit values are obtained by first computing the difference between adjacent blocks in space, and then computing a second difference between corresponding block differences in sequential frames. Bits are set to 1 if this difference is positive, and 0 in any other case. A spatio-temporal Haar filter is used to compute bit values, as illustrated by Figure 2. Frame pairs are selected using the entropy-based subsampling strategy presented above. This fingerprint generation technique performs extremely fast and produces very compact signatures. Fingerprint words condense layout and luminance information, and provide high tolerance to visual artifacts produced by common video processing stages (e.g. bitrate reduction, transcoding, digital/analog noise, etc.). On the other hand, the compactness of these words requires grouping them into larger substrings to ensure satisfactory discriminative characteristics during search.

*3.1.2. Evaluation of CBCR Effectiveness.* In this section we analyze the effectiveness of the CBCR method. We conducted a pilot experiment to validate the viability of the detector to identify duplicated YouTube video content. For this purpose, we created a reference database of over 2000 music video clips, featuring 150 hours of high quality video material. Two main sources were used to build this collection.

—Music DVDs from the libraries of research groups members.

—DVB-T: thematic digital TV music channels recorded during several weeks and at different day times to increase diversity of captured footage.

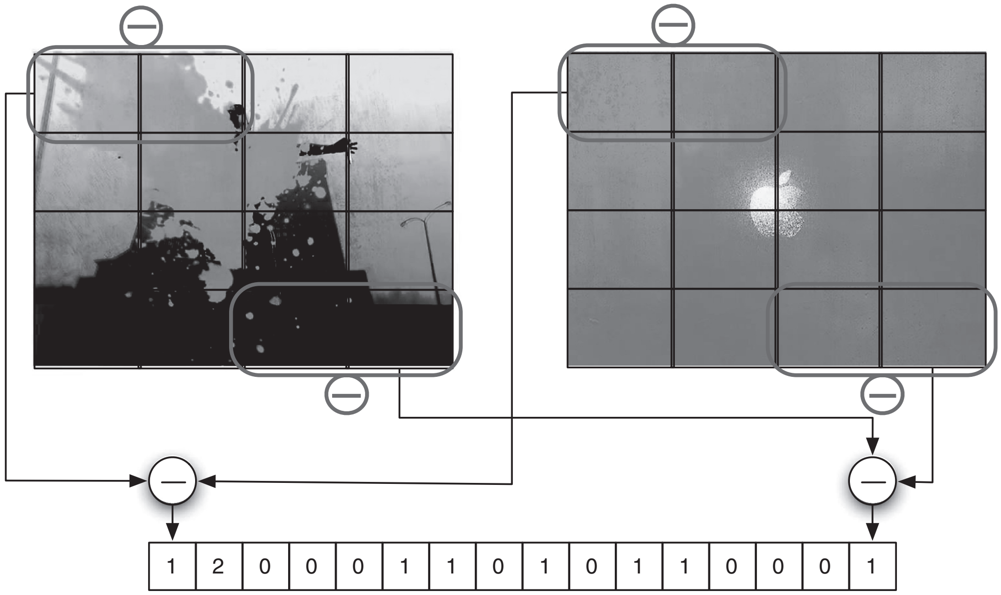


Fig. 2. Hash value computation, using a spatio-temporal Haar filter. A 16 bit hash code is produced for every pair of selected frames.

The selection of songs did not follow any particular criterion, and was entirely based on the musical preferences of private contributors and TV programme producers. The material was manually revised to ensure nonduplication of items in the reference database, and annotated to include artist and song names. We then crawled YouTube to search for duplicates of the videos in our reference set. We used artist and song names of 200 uniformly picked videos to generate YouTube queries. The top 10 results from each query were retrieved and manually screened to exclude videos not present in the reference set. This set of excluded videos comprised different versions of the songs in the reference set, either coming from live performances (instead of the official video clip) or from alternative visual compilations built by creative artists' supporters.

After the manual screening, 550 near-duplicates of items in the reference set were found, and used to create our test set. On average, the test set contained an average of 2.75 (std dev $= 1.73$) variations for each of the 200 preselected elements, and comprised a total of over 25 hours of input video featuring:

—low compression bit-rates, which cause characteristic artifacts;
—smaller frame sizes, compared to videos in the reference set;
—varying frame rates, in the range $[10, 29.97]$;
—overlaid content, such as subtitles, channel logos, etc.

Our system needs to deal with lack of knowledge about the timepoints where video clips start and finish. In order to eliminate temporal references from our test set, we concatenated all selected videos into a single 25 hour stream. Our algorithm handles these multiple matching settings by segmenting the input stream into windows of 4 hash words (with an overlap of 1 hash word between consecutive windows) and processing each window separately. We computed global precision and recall by inter-averaging individual window result values. Also, we varied thresholds for hash word identity and hash substring identity to allow for different visual discrepancy tolerance levels, leading to different precision-recall value pairs.

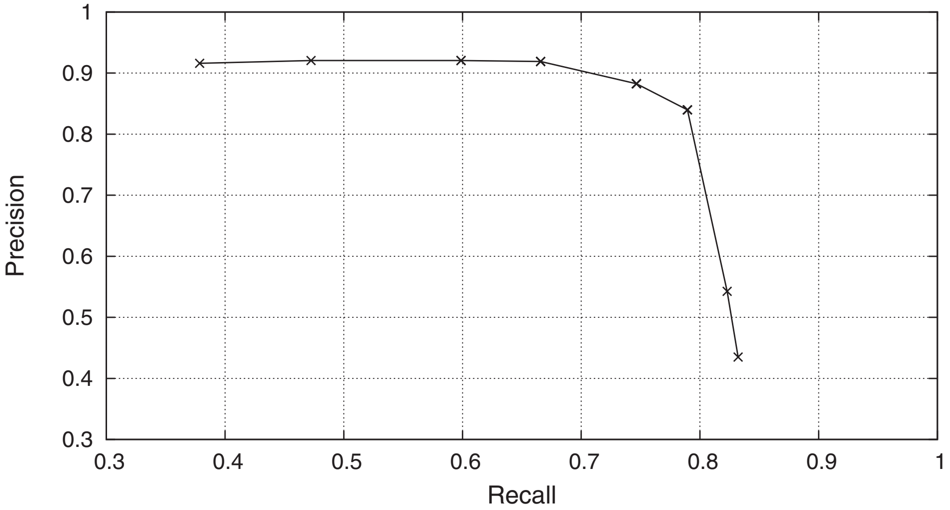


Fig. 3. Recall vs. Precision plot showing results obtained by our proposed CBCR system for the detection of YouTube variations.

Figure 3 shows the recall vs. precision plot obtained for this experiment, with a break-even point of approximately 0.8. This high BEP value supports the viability of the detector in our application scenario. Furthermore, given the vast size of the YouTube collection, it seems sensible to tune the algorithm to trade recall for precision. As illustrated by Figure 3, precision values of 90% are obtained for recall values of approximately 70%. Note also that precision saturates at around 92%, even for low recall values. This evidences that even for very restrictive values of the identity threshold, similar (although not identical) videos are categorized as duplicates due to the compactness of the descriptor used by the algorithm.

### 3.2. Relationships between Videos

This section will first present the requirements needed for a CBCR system to provide insightful information about overlaps in a video collection. We analyze the features and capabilities of such a CBCR system to define a set of relationships that can be effectively derived from analyzing its results. Finally, we formalize this information into a set of graphs for further analysis and use in specific applications.

*3.2.1. CBCR Requirements for Maximizing Knowledge Discovery.* To perform a comprehensive discovery of overlaps, or visual affinity links, in a video collection, $C = \{V_i : 1 \leq i \leq N\}$, we need to consider the scenario where any video element, $V_i$, can potentially include content from every other element of the set. Formally, for every $i \in [1, N]$ we consider a reference set $C' = C - \{V_i\}$ and an input query $Q_i = V_i$. There are two specific requirements in this process.

(1) The reference set, $C$, may contain duplicated content. A comprehensive overlap detection needs to ensure that all matches of a specific video sequence in $C$ are reported. Therefore, multiple concurrent matching is required.
(2) No temporal references can be assumed. Any video $V_i$ is subject to containing sequences from other videos in $C$ starting at any point of their timelines and perhaps following a different chronological order. Formally, given a sequential segmentation

of a video

$$V_i = \bigcup_{k=1}^{|V_i|} V_i^{(k,k+l)},$$

where $l$ denotes a minimum sequence size, each segment is assumed to have equal probability to match $V_j^{(m,m+l)}$ for any $j \neq i$, $1 \leq j \leq N$ and any $1 \leq m \leq |V_j|$. Therefore, independence for every sequence $V_i^{(k,k+l)}$ in a video is assumed. In addition, the presence of a sequence $V_i^{(m,n)}$ before another sequence $V_i^{(p,q)}$, where $m < p$, cannot be taken as a guarantee about the absolute order of those two sequences in other videos.

*3.2.2. Definition of Visual Affinity Relationships.* As a result of the CBCR stage, we obtain a list of visual connections between sequences of different videos, expressed in the form $V_i^{(m,n)} \leftrightarrow V_j^{(p,q)}$: the sequence between frames $m$ and $n$ in $V_i$ overlaps sequence $p$ to $q$ in $V_j$ (with $i \neq j$). Note that the list of connections can contain multiple entries for the same video pair $(V_i, V_j)$ and matching sequences are not guaranteed to preserve temporal coherence, i.e. we might find that $V_i^{(m,n)} \leftrightarrow V_j^{(p,q)}$ and $V_i^{(m',n')} \leftrightarrow V_j^{(p',q')}$, given $m' > m$ and $p' < p$.

Representation of temporal knowledge for reasoning and problem solving purposes was examined by Allen [1983] who provided a comprehensive study of temporal pairwise relationships between intervals. Allen considered a scenario where absolute interval ordering is known, and defined a set of relationships from this assumption. In our context, with no a-priori knowledge of the video set, the information extracted by the CBCR stage was not sufficient to determine the order between video pairs. Therefore, our set of connections can only be based on content overlap. The following list enumerates Allen's relationships.

(1) *Before/After*. One video precedes another in an absolute timeline, and a gap exists between both. This nonoverlapping relationship cannot be derived from the results of CBCR framework described given our assumption of lack of temporal references.
(2) *Equal*. Two videos of equal size completely overlap one another.
(3) *Meets*. One video precedes another in an absolute timeline, and a gap does not exist between both. (Same case as *Before/After*.)
(4) *Overlaps*. Two videos overlap with each other for only part of their timeline.
(5) *During*. One video is completely contained within another.
(6) *Starts/Finishes*. Particular case of the "During" case, and therefore derivable from the CBCR results, where the overlap happens at the beginning or end of the larger video.

We use Allen's set of relationships as our starting point. Particular cases of the *During* relationship (i.e., *Starts* and *Finishes*) are merged into the general case. We also introduce the *Sibling* relationship established between two videos having a common ancestor. This transitive relationship does not consider order, but conveys that both were used in the production of their common ancestor. The inverse relationship, *Spouse*, where two videos have a common descendant is not considered in our set. Two spouses share, at least, the common descendant sequence and, therefore, they are already connected by another relationship of our set.

Let $|V_i|$ be the duration of video $V_i$. Let $O(V_i, V_j)$ be the visual overlap between two different videos $V_i$ and $V_j$, i.e. the video resulting from the frames present in both $V_i$ and $V_j$. Our set of visual affinity relationships includes the following types.

—Duplicates. If $|V_i| \approx |V_j|$ and $|O(V_i, V_j)| \approx |V_i|$, both videos are said to be *duplicates*, formally $V_i \equiv V_j$. The inequality in this expression serves the purpose of illustrating the inclusion of near-duplicates within this category. Duplication is then computed using a tolerance threshold as $|O(V_i, V_j)| > \theta \ \max(|V_i|, |V_j|)$.

—Part-of. If $|V_i| >> |V_j|$ and $|O(V_i, V_j)| \approx |V_j|$, $V_j$ is said to be *part-of* $V_i$ (we also refer to $V_j$ as a *child* of the *parent* $V_i$), formally $V_j \subset V_i$. Size and overlap comparisons of Part-of relationships are also computed using a threshold $\theta = 0.9$ as:

$$|V_i|\,\theta \ > |V_j|, \ \ |O(V_i, V_j)| > \theta\ |V_j|.$$

—Siblings. If $V_i \subset V_k$, and $V_j \subset V_k$, both videos share a parent and are referred to as *siblings*.

—Overlap. If $|O(V_i, V_j)| > 0$, both videos are said to *overlap*, formally $V_i \cap V_j \neq \emptyset$. *Duplicates* and *part-of* relationships are special cases of *overlaps*.

The use of thresholds can introduce irregularities because of the directionality of these relationships. Thus we consider only *duplicate* and *part-of* video pairs, and apply simple heuristics to find and correct the label for the incoherent direction. To this end, conditions were relaxed for incongruent connections (mainly by decreasing the threshold to $\theta = 0.8$) and links were reevaluated under this new setup. For links positively evaluated, we updated their type to the more restrictive class (e.g., from *overlap* to *duplicate*). For those negatively evaluated, the corresponding mismatched link was demoted to reach congruency.

*3.2.3. Creating the Visual Affinity Graphs.* The relationships can be formalized as a set of "Visual Affinity Graphs" (VAG) where videos can be considered as single elements instead of frame sets. Therefore, we will denote videos with lowercase notation (e.g., $v_i$) in the remainder of the paper.

Given a video collection, $C$, we define the following graphs derived directly from our set of visual affinity relationships.

—*Duplicates*. $G_D = (V_D, E_D)$, with undirected edges

$$E_D = \{\{v_i, v_j\} : i \neq j,\ v_i \equiv v_j,\ v_i, v_j \in V_D \subset C\}.$$

—*Part-of*. $G_P = (V_P, E_P)$, with directed edges

$$E_P = \{(v_i, v_j) : i \neq j,\ v_i \subset v_j,\ v_i, v_j \in V_P \subset C\}.$$

—*Siblings* as $G_S = (V_S, E_S)$, with undirected edges

$$E_S = \{\{v_i, v_j\} : i \neq j \neq k \neq i,\ (v_i, v_k) \in E_P, (v_j, v_k) \in E_P, v_k \in V_P\}.$$

—*Overlap*. $G_O = (V_O, E_O)$, with undirected edges

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

$V_D$ denotes the set of videos with one or more duplicates. $V_P$, $V_S$ and $V_O$ are defined analogously for the other relationships. We will also consider the study of the graph of related videos, $G_R = (V_R, E_R)$, a super-graph of all relationships defined as:

$$V_R \ = \ V_D \cup V_P \cup V_S \cup V_O \tag{3}$$

$$E_R \ = \ E_D \cup E_P \cup E_S \cup E_O. \tag{4}$$

Note that we can use $V_R$ to define a partition of a video collection, $C = V_R \cup V_U$, separating visually related ($V_R$) and unrelated ($V_U$) videos. $E_R$ is formed solely by undirected edges; to this end, we removed directionality from $E_P$ links before the union.

Table I.
Properties of test collection $C$, and its partitions $C_A$ and $C_B$

| | N. Queries | N. Vids | Duration | Size |
|---|---|---|---|---|
| $C_A$ | 579 | $28,216$ | $2055h$ | 263 GB |
| $C_B$ | 267 | $10,067$ | $751h$ | 91 GB |
| $C$ | 846 | $38,283$ | $2806h$ | 354 GB |

## 4. ANALYSIS OF CONTENT REDUNDANCY IN YOUTUBE

This section studies the redundancy characteristics of YouTube. The objective is to determine the proportion of duplicated footage and the different relationship types present. We then try to understand the main reasons driving the redundant upload of duplicated footage, and potential applications.

### 4.1. Data

We created a test collection by formulating queries and subsequent searches for “related videos” in YouTube. We collected the “Top 10 Gaining Weekly Queries” from Google Zeitgeist,[1] June 2001 to May 2007, which accounted for 703 unique queries. While the collected set is mainly composed of English terms, it also includes a few non-English queries. Around 50% were language-independent concepts, mainly names and numbers (e.g., Radiohead, 300, Zidane). Some queries in the collection were strongly tied to the original purpose of web search, and were not suitable for video search (e.g., “windows update”). We chose to remove these by filtering queries returning less than 100 YouTube results. In total, 579 queries were accepted, for which the top 50 results were retrieved. Altogether, we collected $28,216$ videos using those queries. We refer to this subset as $C_A$.

A random sample of $C_A$ was used to extend the collection: 1) a query from the original set was chosen uniformly at random; 2) a subset of 5 results was uniformly selected; 3) a maximum of 50 related videos (obtained from YouTube suggestions) for each of the 5 videos was collected. We fixed a target size of 10,000 and iterated to reach that cardinality. In total, 58 queries from the set were used to generate 267 related video requests, which generated a collection of 10,067 additional unique elements, verified by their YouTube video id. We refer to this subset as $C_B$.

Our crawling strategy considers popular topics and selects the top ranked videos for each. This produces a collection biased towards more popular videos. However, popular videos attract the majority of views, and therefore, their analysis is expected to have a much higher impact on the system than the proportion of the collection they represent. Features of the test collection are summarized in Table I. We used the Fobs project [San Pedro 2008] as the programming interface to access content and characteristics of downloaded videos.

### 4.2. Topological Study

*4.2.1. Graph Properties.* For any graph, $G = (V, E)$, we can find a partition of maximal connected subgraphs $G^i$ so $G = \bigcup G^i$. Each of these subgraphs is called a connected component, and any two nodes in them are connected. We can compute the distance between two nodes, $d(u, v)$, as the shortest path between them. The *diameter* of a graph is determined as $\mathrm{diam}(G) = \max_{u,v} d(u, v)$. We can also compute the *characteristic path length*, $D(G)$, as the average $d(u, v)$ between every two nodes [Lovejoy and Loch 2003].

[1] http://www.google.com/intl/en/press/zeitgeist.

Table II.
Redundancy found in subcollections $C_A$ and $C_B$ as well as in the complete collection $C$, expressed as the proportion of visual affinity relationships established by our analysis methodology

| | $\lvert V_R \rvert$ | % | Dup | % | Parent | % | Child | % | Sibling | % |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_A$ | 9926 | 35.17 | 4238 | 15.01 | 2372 | 8.41 | 2686 | 9.52 | 1707 | 6.05 |
| $C_B$ | 3746 | 37.22 | 1813 | 18.00 | 827 | 8.22 | 886 | 8.81 | 533 | 5.30 |
| $C$ | 13672 | 35.71 | 6051 | 15.80 | 3199 | 8.35 | 3572 | 9.33 | 2240 | 5.85 |

Table III.
Visual Affinity Graphs properties. This table gives values for: the number of connected components ($G^i$) per query; their average size ($\overline{\lvert G^i \rvert}$); the average clustering coefficient ($\gamma_G$); the average degree of nodes ($k_G$); the average diameter ($\mathrm{diam}(G)$); and the characteristic path length ($D(G)$). We considered duplicates ($G_D$); part-of ($G_P$); siblings ($G_S$); and related videos ($G_R$)

| | $G^i$/query | $\overline{\lvert G^i \rvert}$ | $\gamma_G$ | $k_G$ | diam($G$) | $D(G)$ |
|---|---|---|---|---|---|---|
| $G_D$ | 2.71 | 3.13 | 1 | 2.13 | 1 | 1 |
| $G'_P$ | 1.36 | 4.05 | 0.44 | 2.05 | 1.82 | 1.69 |
| $G'_S$ | 0.60 | 4.32 | 0.91 | 4.14 | 1.38 | 1.24 |
| $G'_R$ | 3.07 | 5.56 | 0.51 | 2.75 | 2.30 | 2.14 |

The *clustering coefficient* [Nakano et al. 2007] provides a metric for the connectivity between neighbors of a common node, $u$

$$\gamma_u = \frac{|\{e_{vw} : v, w \in \Gamma_u\}|}{k_u(k_u - 1)}, e_{vw} \in E,$$

where $\Gamma_u = \{v : e_{uv} \vee e_{vu} \in E\}$ denotes the neighborhood of node $u$ and $k_u$ its degree. Both the clustering coefficient and the degree can be averaged for all of the nodes in $G$, denoted by $\gamma_G$ and $k_G$ respectively.

*4.2.2. Presence of Redundancy.* Tables I and II show general properties of created VAGs, including cardinalities for the different subsets of interest. Table III shows topological properties of VAGs, focusing on the analysis of connected components in the $V_R$ partition, that is, videos visually connected to at least one other video. Note also that Table III presents some values averaged across queries, referring to those used to generate the collection as described in Section 4.1.

The size of $V_R$ was over 35% of $C$, that is, YouTube contains a high level of visual redundancy. *Duplicate* videos comprised almost 16%, making duplication the prevalent visual relationship found in our set. The proportion of duplicates was higher in $C_B$, as YouTube generates restrictive queries when searching for related videos. Groups of duplicate videos, $G^i_D$, form *cliques*, that is, complete subgraphs. On average, we found 2.71 *duplicate cliques* per query (std dev = 1.63), with an average size of 3.13 (std dev = 2.55). For the analysis of other VAGs, these cliques are treated as single elements, that is, supernodes, avoiding the study of redundant relationships. We refer to these duplicate-free graphs as $G'_X$.

Figure 4 shows the distribution of duplicate videos. The distribution exhibits a mean value of 6.98 (std dev 4.83). This indicates an important dependency of the number of duplicates on the specific query. Returning many identical search results, out of the top 50, is expected to have degrading effects in terms of user experience.

*Part-of* were present in the set, though considerably less than duplicates. The number of *parents* and *children* remains balanced for every set, as illustrated in Table II. In $G'_P$, we observe an average of 4.05 (std dev = 4.15) videos per connected component, that is, more than duplicate cliques. However, the average number of cliques per query was lower with a value of 1.36 (std dev = 0.97). $G'_P$ subgraphs feature relatively low

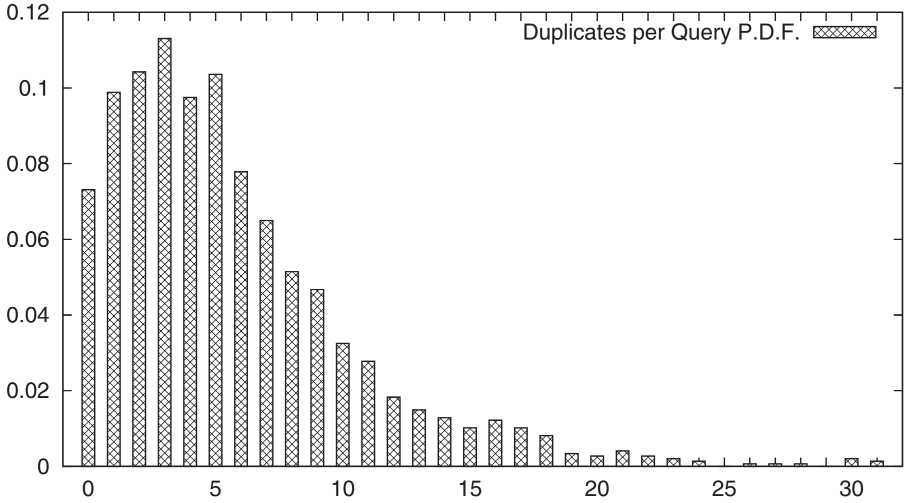


Fig. 4. Distribution of duplicates per query for the top 50 search results.

connectivity, with a clustering coefficient $\gamma_{G_P} = 0.44$, revealing the presence of indirect paths between nodes, which we categorized as *siblings*.

*Sibling* is the least common form of relationship. Counterintuitively, we found $\gamma_{G_S} < 1$, indicating *siblings* do not organize in cliques. This observation is supported by the diameter ($\text{diam}(G_P) > 1$) and characteristic path length ($D(G_P) > 1$) values. The fact that two *siblings* may have different neighbors is explained by the existence of *children* related to more than one *parent*, as revealed by the average degree of children ($k_{chld} = 1.67$) and parent nodes ($k_{prnt} = 1.84$).

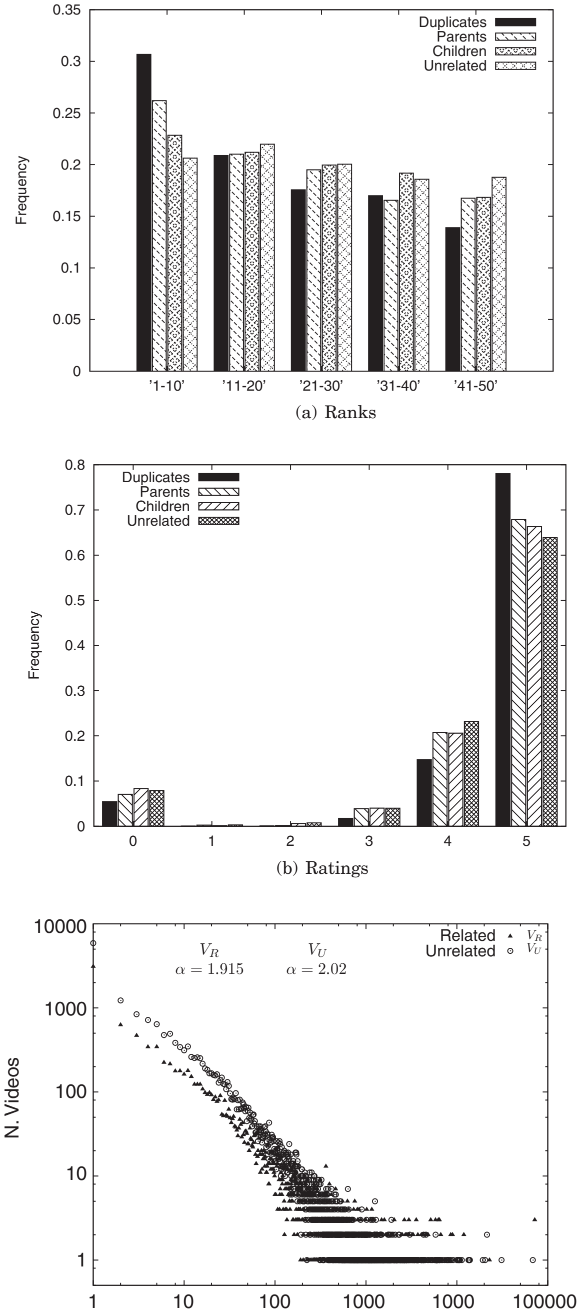
When considering $G'_O$, we observe that the number of connected components increases noticeably. Rarely *overlaps* edges, $E_O$, are able to bridge previously existing connected components, $G'_P$, mainly adding new ones. These newly created components are sparse, as indicated by high values $\text{diam}(G'_O)$ and $D(G'_O)$. They denote video connections with lower exploitation potential.

The figures obtained in this analysis illustrate the presence of redundancy in the YouTube database. Aside from the usability implications this may have for users of the system, we have been able to identify different subgraphs of video elements with noticeable average presence in all of the most popular queries during a long period of time. Each type of subgraph has a different nature and conveys specific semantics that can be exploited for different purposes.

### 4.3. Reasons for Duplication

In this section we discuss the possible reasons behind the common presence of near-duplicate videos. We perform this analysis focusing on three aspects: video popularity, metadata diversity and multilingualism.

*4.3.1. Video Popularity.* Because it is a social-oriented Website, many contributors to YouTube search for recognition. It is not surprising to find users re-uploading popular content in order to boost their popularity. We provide evidence of this in Figure 5. First, we consider the probability density function of the video ranks in the search results for different categories in Figure 5(a). This figure illustrates how connected videos, $V_R$, tend to have higher ranks; it is twice as likely to find a duplicate in the top 5 videos than it is to find unrelated videos, $V_U$.



(a) Ranks

(b) Ratings

(c) N. Views

Fig. 5. Metrics of video popularity as a function of their visual relationships. Popularity is established by: (a) rank of video in search results; (b) ratings received by other users of the community; (c) number of views. Videos having one or more visual connections to others tend to be more popular by these three metrics.

Second, we also consider ratings for each different overlapping relationship. Ratings, up to 5 stars, are assigned by website users after watching a video, to express their feeling about it. Figure 5(b) shows consistently higher ratings for related videos, especially duplicates. This histogram also illustrates how the search engine promotes ranks of highly rated videos: the probability density for the top 50 results, as shown in Figure 5(b), is mainly concentrated in the highest rating values (4 and 5).

Finally, in Figure 5(c) we show a log scaled representation of the frequency of videos per numbers of views. A power-law pattern, $p(x) \propto x^{-\alpha}$, is shown featuring sparse tails due to the relatively small sample used (see Section 4.1). In this figure we grouped together related videos, $V_R$, for the sake of clarity. To fit our data to a power-law distribution, we used the approach from Clauset et al. [2009]. Related videos resulted in $\alpha = 1.915 \pm 0.032$ while for unrelated videos we obtained $\alpha = 2.029 \pm 0.023$. This difference illustrates a visible divergence in the distribution of views. Unrelated videos have lower frequency values when considering very high number of views, indicating that related content tends to be more popular.

*4.3.2. Metadata Diversity.* YouTube offers a somewhat limited community interaction interface, restricted to ratings and comments only. Only the uploader can add tags to a video, or edit its title and description. These impediments for collaborative tagging might in addition encourage redundantly uploading videos in order to apply a personal selection of tags.

We conducted experiments to establish the correlation between tags and visual similarity of video pairs to study annotation as a potential reason for duplication.

We found metadata similarity by computing pairwise differences between:

—duplicates: $\mathcal{T}_D = \{(v_i, v_j) \in V_D^2 : i \neq j,\ v_i \equiv v_j\}$ , i.e. pairs of *duplicate* videos;
—parent-children: $\mathcal{T}_P = \{(v_i, v_j) \in V_P^2 : i \neq j,\ v_j \subset v_i\}$;
—siblings: $\mathcal{T}_S = \{(v_i, v_j) \in V_S^2 : i \neq j \neq k \neq i, \exists\ v_k | v_i, v_j \subset v_k\}$;
—baseline: Uniformly selected pairs of videos, $\mathcal{T}_{\text{rnd}}$, restricted to being generated by the same query.

All metadata was preprocessed using a set of common English stop words as well as Porter Stemming.[2] The symmetric Jaccard index [Jaccard 1901] was used as the metric of metadata similarity, and Levenshtein distance was used to compute string similarity.

We consider $X_R^{\text{tag}}$, the random variable "tag-similarity of $(v_i, v_j)$", for all $(v_i, v_j) \in R$, $R \in \{\mathcal{T}_D, \mathcal{T}_P, \mathcal{T}_S, \mathcal{T}_{\text{rnd}}\}$. We also consider $X_R^{\text{title}}$, defined analogously. Figure 6 shows the probability density function for the series of random variables $X_R^{\text{tag}}$ and $X_R^{\text{title}}$. These figures illustrate the probability of finding two related videos with a specific value of metadata similarity. For instance, we can see that the probability of two duplicate videos having 40% tag similarity is roughly 0.1 (e.g., $P(35 < X_{\text{dup}}^{\text{tag}} \leq 45) \approx 0.1$). We consider only tags and title of videos; results obtained for other metadata fields (e.g., description) do not show the degree of correlation found for these two.

The baseline gives a measurement of the metadata similarity of pairs of videos chosen uniformly at random. The energy is mainly concentrated below 30% similarity, a somewhat high number explained by the fact that the collection is full of related material. When we consider pairs of related videos, we expect the average similarity to grow if a correlation exists between visual and metadata similarity. This is the case, especially for pairs of *Duplicates* and *Parent-children*; the difference is, however, small. In duplicate detection, we find many pairs below the 30% similarity threshold

[2] http://tartarus.org/ martin/PorterStemmer/index.html.

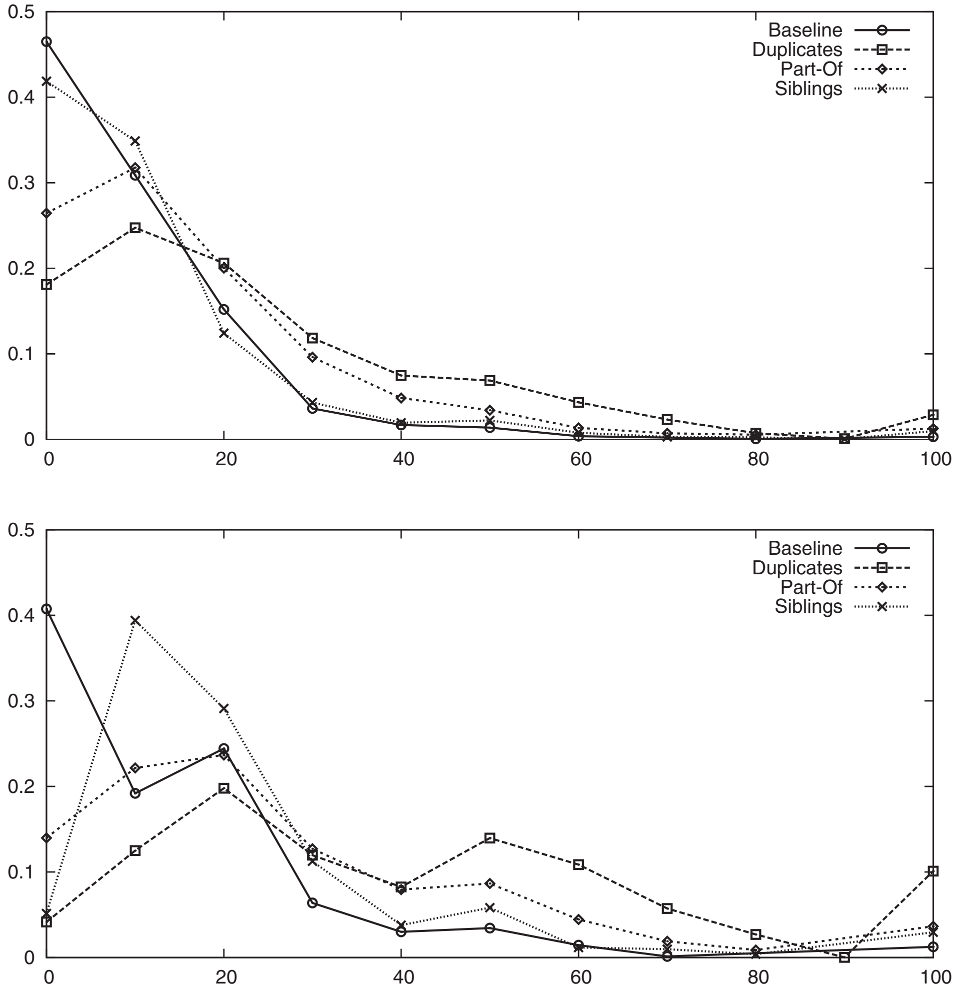


Fig. 6. Inter-video tag (top) and title (bottom) similarity, as a probability density function. These illustrate the probability of finding two related videos with a specific value of metadata similarity. Metadata similarity for identical videos do not differ noticeably from unrelated videos, indicating lack of user agreement for tag and title assignments.

($P(X^{\text{tag}}_{\text{dup}} < 30) = 0.64$, $P(X^{\text{title}}_{\text{dup}} < 30) = 0.37$), denoting a high amount of different tags appearing in duplicate videos. Our interpretation is that redundant uploads are to some degree motivated by personal annotation needs.

Another goal of diversifying metadata would be for uploaders to hinder the removal of copyrighted content. While this explanation serves to explain differences in metadata it has two main drawbacks: 1) by obscuring metadata uploaders demote content, reducing its visibility in the system for interested viewers; 2) YouTube has introduced monetization options[3] to allow copyright holders to profit from the presence of their videos on their site.

[3] http://googleblog.blogspot.com/2009/07/i-now-pronounce-you-monetized-youtube.html.

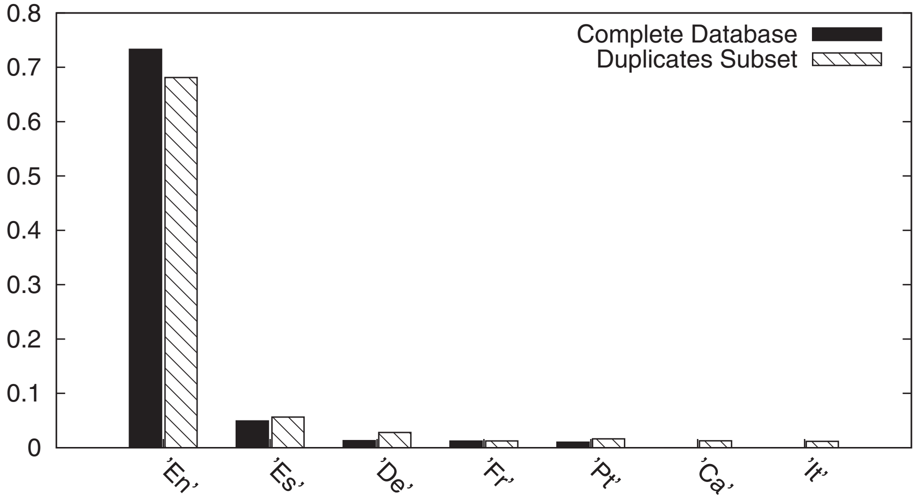


Fig. 7. Distribution of languages for our test collection, $C$, and the subset of duplicate videos $V_D$. Duplicate videos represent a slightly lower proportion of English content, indicating potential multilingual tagging needs for the YouTube user community.

*4.3.3. Multilingualism within the Community.* A plausible reason for uploading redundant content is multilingualism: users in each country will upload these sequences either dubbed or subtitled in their own language. We conducted an experiment to reveal the distribution of languages. We studied the differences in distribution for the collection, $C$, and for the duplicates subset, $V_D$. We used an automatic text-based language identification detector based on n-grams [Artemenko et al. 2006] on selected metadata fields. We discarded noisy metadata fields, specifically “Title” and “Tags,” as they are relatively short, do not normally form complete setences, and it is common to include names which are language independent. “Descriptions” have normally enough words to allow high precision detection. When “comments” were available, they were used to expand the detection text.

Figure 7 shows these two distributions. We expected the clear predominance of English language to decrease for the $V_D$ set, and converge towards a more uniform distribution among all languages. Though this pattern is shown in Figure 7, the difference is not big enough to consider it a main motivation for redundant uploading. However, it is difficult to establish the actual relevance of multilingualism quantitatively. Because English videos are viewed more frequently, they tend to be ranked higher, biasing the distribution found for the considered subset of top 50 results. Furthermore, our test collection $C$ was gathered from Google’s Zeitgeist Archive, which included mostly English queries, contributing to accentuate this bias. These two facts further support the significance of the distribution shift towards other languages when considering the subset of duplicated footage.

In this section we studied redundancy in the video sharing Website YouTube, which revealed a high amount of redundant content in the collection. Different types of visual relationship between elements, including *near-duplication* or *inclusion* were found. Each type of overlap conveyed different semantics, allowing for multiple knowledge extraction applications based on them. For instance, we have used *part-of* relationships for automatic video summarization leading to high quality summaries [San Pedro et al. 2009], and for video tagging described in the following.

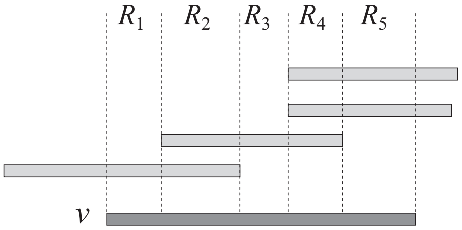


Fig. 8. Overlap regions $R_1, \ldots, R_5$ of a video $v$ covered by four other videos.

## 5. AUTOMATIC VIDEO TAGGING

In this section, we exploit the video overlap relationships for deriving two methods of tag assignment: 1) *neighbor-based* methods, which take just immediately overlapping videos; 2) a method based on propagation of tag weights within the visual overlap graph.

### 5.1. Neighbor-Based Tagging

These tagging methods consider relationships in the overlap graph $G_R$ to transfer tags between adjacent videos.

*5.1.1. Simple Neighbor-Based Tagging.* We transform the undirected graph of related videos, $G_R$, into a directed and weighted graph $G'_R = (V_R, E'_R)$, with $(v_i, v_j)$ and $(v_j, v_i) \in E'_R$ iff $\{v_i, v_j\} \in E_R$. 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

$$w(v_i, v_j) = \frac{|v_i \cap v_j|}{|v_j|}, \tag{5}$$

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 $v_i$.

Let $T = \{t_1, \ldots, t_n\}$ be the set of tags originally (manually) assigned to the videos in $V_R$ and let $I(t, v_i) = 1$ iff $v_i$ was manually tagged by a user with tag $t$, $I(t, v_i) = 0$ otherwise. We compute the relevance $\text{rel}(t, v_i)$ of a tag $t$ from adjacent videos as follows:

$$\text{rel}(t, v_i) = \sum_{(v_j, v_i) \in E'_R} I(t, v_j) w(v_j, v_i), \tag{6}$$

that is, we compute a weighted sum of influences of the overlapping videos containing tag $t$. Thus we generate autotags$(v_i)$ of automatically assigned *new* tags for each video $v_i \in V$ using a threshold $\delta$ for tag relevancy:

$$\text{autotags}(v_i) = \{t \in T \,|\, I(t, v_i) = 0 \wedge \text{rel}(t, v_i) > \delta\} \tag{7}$$

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

*5.1.2. Overlap Redundancy Aware Tagging.* For a set of overlapping videos with multiple redundant overlaps (see Figure 8) the relevance values for automatically generated tags can be too high compared to original tags, we propose a relaxation method for this case.

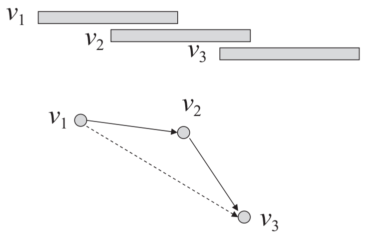


Fig. 9. Although there is no overlap between videos $v_1$ and $v_3$, a context relationship (dotted line) can be established via the overlap graph.

Let $N(v) = \{v_i | (v_i, v) \in E'_R\}$ be the set of overlapping videos for video $v$. An *overlap region* $R_i$ can be defined as a video subsequence $s \subseteq v, |s| > 0$, of maximum length contained in a maximum subset $Y \subset N(v)$, that is, with none of the remaining videos $N(v) \setminus Y$ overlapping with $s$. 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

$$\sum_{i=0}^{k-1} \alpha^i \cdot \frac{|s|}{|v|}, \tag{8}$$

that is, 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 $v$:

$$\mathrm{rel}(t, v) = \sum_{X \in \mathcal{P}(N(v))} \sum_{i=0}^{k(X)-1} \alpha^i \cdot \frac{\left| v \cap \bigcap_{x \in X} x - \bigcup_{u \in N(v) \setminus X} u \right|}{|v|}, \tag{9}$$

where

$$k(X) = \sum_{x \in X} I(t, x) \tag{10}$$

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.

### 5.2. TagRank: Context-Based Tag Propagation in Video Graphs

In this subsection we describe a tag weight propagation method which allows for the iterative transfer of tags along paths of arbitrary length; see Figure 9. We call the method *TagRank*,[4] 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 TR$(t, v_i)$ for a video $v$ by the following recursive relationship:

$$\mathrm{rel}(t, v_i) = \mathrm{TR}(t, v_i) = \sum_{(v_j, v_i) \in E'_R} \mathrm{TR}(t, v_j) w(v_j, v_i). \tag{11}$$

[4] The term “TagRank” occurs in another context in the preliminary work [Szekely and Torres 2005] and Ling et al. [2008] 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.

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

$$\mathbf{TR}(t) = \begin{pmatrix} w(v_1, v_1) & w(v_1, v_2) & \cdots & w(v_1, v_n) \\ w(v_2, v_1) & w(v_2, v_2) & \cdots & w(v_2, v_n) \\ \vdots & \vdots & \ddots & \vdots \\ w(v_n, v_1) & w(v_n, v_2) & \cdots & w(v_n, v_n) \end{pmatrix}^{\mathrm{T}} \cdot \begin{pmatrix} \mathrm{TR}(t, v_1) \\ \mathrm{TR}(t, v_2) \\ \vdots \\ \mathrm{TR}(t, v_n) \end{pmatrix}. \tag{12}$$

This eigenvector equation can be solved using power iteration. Similar to Kleinbergs HITS [Kleinberg 1999] 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 simply considered the solution for Equation (12) we would lose this information because for a given node $v$ in a connected component the solution would eventually converge to the same value $\mathrm{TR}(t, v)$ for each tag $t$. This is not intuitive and instead we perform a limited number $\Gamma$ of iterations (where $\Gamma$ is a tuning parameter) using the original tag assignment in the form

$$\mathbf{TR}(t) = (I(t, v_1), \ldots, I(t, v_n))^{\mathrm{T}}, \tag{13}$$

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 Craswell and Szummer [2007] for random walks in click graphs.

## 6. EVALUATION OF AUTOMATIC VIDEO TAGGING

In this section, we present the results of our threefold evaluation methodology for automatic tagging.[5]

### 6.1. Classification Experiments

*6.1.1. Setup for Classification Experiments.* For classifying data into thematic categories we used linear support vector machines (SVMs), as they have been shown to perform well for text-based classification tasks [Dumais et al. 1998]. More specifically, we used the SVMlight [Joachims 1999] implementation with default parameterization.

As categories for our classification experiments, we chose the 7 YouTube categories containing at least 900 videos in our dataset. These were "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. We performed binary classification experiments for all $\binom{7}{2} = 21$ combinations of these category pairs. We trained different models based on T = 10, 25, 50, 100, 200, and 400 training videos per category to study the influence of the training set size. To this end, we randomly split the 900 videos per category into $\lfloor 900/T \rfloor$ sets of size T. Then, we performed cross-validation by taking the $i$th subset for each category as training set, and testing on the remaining videos for a category pair.

Our quality measure is the fraction of correctly classified videos (*accuracy*); we computed microaveraged results (averaging across individual classification decisions) for

[5] The dataset used in this evaluation is accessible from: http://www.jsanpedro.es/datasets/TOIS2010.zip.

all cross-validation runs and topic pairs along with their 99% confidence intervals.[6] We also computed the Area under the ROC curve values (AUC) [Fawcett 2003] for the different cross-validation runs. ROC (Receiver Operating Characteristics) curves depict the true positive with respect to the false positive rate of classifiers.

We compared the following methods for constructing tag feature vectors.

(1) *BaseOrig*. Vectors based on the original tags of the videos. This serves as the baseline.
(2) *NTag*. Vectors constructed based on the tags simple neighbor-based tagging described in Section 5.1.1 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 5.1.2. We did not pursue any extensive parameter tuning and chose $\alpha = 0.5$ for the relaxation parameter.
(4) *TagRank*$\Gamma$ (with $\Gamma = 2, 4, 8$ iteration steps). Vectors using, in addition to the original tags, new tags produced by the TagRank algorithm described in Section 5.2.
(5) *LdaTopics*$\Delta$. The original tags sets for the videos were mapped to feature vectors of latent topics (with a feature space of $\Delta = 50$, 100, 200, 500 and 1000 latent topics) using Latent Dirichlet Allocation (LDA) [Blei et al. 2003]. Vector components correspond to probabilities for the topics. We used the Mallet implementation [McCallum 2002] of LDA.
(6) *LdaTagRec*$\Delta$. We applied the algorithm proposed in Krestel et al. [2009][7] for tag recommendation based on LDA (with parameter $\Delta$ defined as for *LdaTopics*). The underlying idea is to consider videos as mixtures of latent topics, and to recommend additional characteristic tags for these topics. The method was shown to outperform the approach described in Heymann et al. [2008]. In addition to the original tags, we used the top-25 recommended tags per video which is comparable to the average number of tags added by the method *RedNTag*. Note that the method returns probabilities for recommended tags; thus, less likely tags carry less weight.
(7) *LdaTagRecComb*$\Delta$. We combined *RedNTag*, and the LDA-based recommender algorithm *LdaTagRec*$\Delta$ by linearly combining feature vectors from the original video tags with tags obtained from the two recommender models.

For combining tag vectors we doubled the weight of the original tags (compared to automatically added ones) as they were directly assigned by YouTube users. For all experiments we set threshold $\delta$ to 0, i.e. we took all automatically assigned tags with their respective weight into account. The average number of originally contained in the videos (*BaseOrig*) was 8.7 (std dev = 5.0). The average number of *additionally* generated tags per video were 22.1 (std dev = 23.5) for the approaches *NTag* and *RedNTag* that take direct neighbors into account (both producing the same tags with different weights), and 44.6 (std dev = 49.8), 71.7 (std dev = 93.0) and 94.1 (std dev = 159.0) for *TagRank2*, *TagRank4* and *TagRank8* respectively.

Note that tag recommendations based on video links, and based on LDA applied on videos interpreted as text corpus can be considered as orthogonal techniques as they make use of different types of information. Approaches based on video links can, for instance, be applied to recommend tags to videos without any existing annotation if there exist connections to other videos in the overlap based affinity graph. In contrast, LDA based recommendations or other approaches based on term cooccurrences require the existence of original tags to determine topics. On the other hand, LDA works also for

[6] Computed as $\pm 2.58 \cdot \sqrt{\frac{p \cdot (1-p)}{n}}$, where $p$ corresponds to the accuracy, and $n$ is the number of individual classification decisions.

[7] We thank the author Ralf Krestel for assisting us with implementation details.

Table IV. Classification Accuracy for Different Video Representations using Automatic Tagging

(a) Classification accuracy and 99% confidence intervals for *BaseOrig*, *NTag*, *RedNTag*, and *TagRank*Γ; T = 10, 25, 50, 100, 400 training videos per category

| | BaseOrig | NTag | RedNTag | TagRank2 | TagRank4 | TagRank8 |
|---|---|---|---|---|---|---|
| T = 10 | 0.561 ± 0.0007 | 0.605 ± 0.0007 | 0.614 ± 0.0007 | 0.596 ± 0.0007 | 0.592 ± 0.0007 | 0.589 ± 0.0007 |
| T = 25 | 0.634 ± 0.0011 | 0.688 ± 0.001 | 0.695 ± 0.001 | 0.673 ± 0.001 | 0.667 ± 0.001 | 0.664 ± 0.001 |
| T = 50 | 0.700 ± 0.0014 | 0.742 ± 0.0014 | 0.747 ± 0.0014 | 0.727 ± 0.0014 | 0.721 ± 0.0014 | 0.718 ± 0.0014 |
| T = 100 | 0.753 ± 0.002 | 0.778 ± 0.0019 | 0.781 ± 0.0019 | 0.766 ± 0.0019 | 0.761 ± 0.002 | 0.758 ± 0.002 |
| T = 200 | 0.794 ± 0.003 | 0.805 ± 0.0029 | 0.808 ± 0.0029 | 0.797 ± 0.003 | 0.792 ± 0.003 | 0.790 ± 0.003 |
| T = 400 | 0.827 ± 0.0047 | 0.830 ± 0.0046 | 0.832 ± 0.0046 | 0.828 ± 0.0047 | 0.825 ± 0.0047 | 0.824 ± 0.0047 |

(b) Classification accuracy and 99% confidence intervals *LdaTopics*, *LdaTagRec*, and *LdaTagRecComb*; T = 10, 25, 50, 100, 400 training videos per category, and 50, 100, 200, 500, and 1000 latent topics

| LdaTopics | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| T = 10 | 0.645 ± 0.0007 | 0.629 ± 0.0007 | 0.595 ± 0.0007 | 0.567 ± 0.0007 | 0.552 ± 0.0007 |
| T = 25 | 0.693 ± 0.001 | 0.689 ± 0.001 | 0.664 ± 0.001 | 0.634 ± 0.0011 | 0.612 ± 0.0011 |
| T = 50 | 0.717 ± 0.0014 | 0.723 ± 0.0014 | 0.710 ± 0.0014 | 0.694 ± 0.0015 | 0.676 ± 0.0015 |
| T = 100 | 0.735 ± 0.002 | 0.745 ± 0.002 | 0.746 ± 0.002 | 0.745 ± 0.002 | 0.736 ± 0.002 |
| T = 200 | 0.749 ± 0.0032 | 0.762 ± 0.0031 | 0.766 ± 0.0031 | 0.780 ± 0.0031 | 0.781 ± 0.0031 |
| T = 400 | 0.760 ± 0.0053 | 0.778 ± 0.0051 | 0.786 ± 0.0051 | 0.805 ± 0.0049 | 0.814 ± 0.0048 |

| LdaTagRec | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| T = 10 | 0.645 ± 0.0007 | 0.626 ± 0.0007 | 0.605 ± 0.0007 | 0.596 ± 0.0007 | 0.593 ± 0.0007 |
| T = 25 | 0.700 ± 0.001 | 0.691 ± 0.001 | 0.678 ± 0.001 | 0.673 ± 0.001 | 0.668 ± 0.001 |
| T = 50 | 0.733 ± 0.0014 | 0.730 ± 0.0014 | 0.726 ± 0.0014 | 0.728 ± 0.0014 | 0.725 ± 0.0014 |
| T = 100 | 0.759 ± 0.002 | 0.762 ± 0.002 | 0.765 ± 0.002 | 0.771 ± 0.0019 | 0.768 ± 0.0019 |
| T = 200 | 0.781 ± 0.0031 | 0.786 ± 0.003 | 0.791 ± 0.003 | 0.802 ± 0.0029 | 0.802 ± 0.0029 |
| T = 400 | 0.806 ± 0.0049 | 0.811 ± 0.0048 | 0.818 ± 0.0048 | 0.831 ± 0.0046 | 0.832 ± 0.0046 |

| LdaTagRecComb | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| T = 10 | 0.655 ± 0.0007 | 0.641 ± 0.0007 | 0.627 ± 0.0007 | 0.621 ± 0.0007 | 0.619 ± 0.0007 |
| T = 25 | 0.715 ± 0.001 | 0.712 ± 0.001 | 0.705 ± 0.001 | 0.704 ± 0.001 | 0.702 ± 0.001 |
| T = 50 | 0.750 ± 0.0014 | 0.751 ± 0.0014 | 0.750 ± 0.0014 | 0.754 ± 0.0014 | 0.753 ± 0.0014 |
| T = 100 | 0.775 ± 0.0019 | 0.780 ± 0.0019 | 0.783 ± 0.0019 | 0.789 ± 0.0019 | 0.788 ± 0.0019 |
| T = 200 | 0.798 ± 0.003 | 0.803 ± 0.0029 | 0.807 ± 0.0029 | 0.815 ± 0.0029 | 0.815 ± 0.0029 |
| T = 400 | 0.821 ± 0.0047 | 0.826 ± 0.0047 | 0.830 ± 0.0046 | 0.840 ± 0.0045 | 0.839 ± 0.0045 |

videos without connections in the affinity graph, provided these videos contain suitable tags. This was our rationale for combining both approaches in *LdaTagRecComb*.

*6.1.2. Results of Classification Experiments.* The results of the comparison are shown in Tables III(a) and IV(b) (accuracy) as well as Tables IV(a) and V(b) (AUC). The main observations are as follows.

—Classification taking automatically generated tagging into account outperforms classification using just the original tags (*BaseOrig*). This holds for all of the three introduced tagging methods. For classification with 50 training documents per category, for example, we increased the accuracy from approximately 70% to 75% for *RedNTag*.

—Overlap redundancy aware neighbor-based tagging (*RedNTag*) provides slightly but consistently more accurate results than the simple neighbor-based tagging variant (*NTag*).

—Both of the neighbor-based methods outperform *TagRank*.

—Classification in the LDA topic space (*LdaTopics*) provides good results for certain configurations (e.g., for $T = 10$ training documents, we observe an increase in accuracy from 56% to 66% with 50 latent topics). However, results are highly dependent on the number of topics. Overall, for smaller training sets, a smaller number of latent topics seems preferable, whereas for larger training sets a higher number of latent topics provides better classification performance. Similar observations can be made for *LdaTagRec*, although results are overall better and more stable, possibly also due to the direct inclusion of the original video tags.

—The combination *LdaTagRecComb* of LDA based recommended tags with the neighbor-based tagging approaches performs significantly better than *LdaTagRec*, confirming that neighbor-based methods can provide useful and complementary annotations.

The AUC values reveal the same trends described for accuracy results. For larger training sets, we can observe saturation effects as training sets contain already more information in that case, resulting in quite accurate classifications (e.g., accuracy values over 80% and AUC values around 90% for 400 training videos per category); thus, additional information added by recommended tags just leads to marginal improvements. Note, however, that although large training samples for the considered YouTube categories are available, manually labeling data for other taxonomies can be tedious; therefore, training on small samples is often of practical importance. The size of the confidence intervals increases with increasing training set size as less cross-validation tests are possible in that case.

### 6.2. Clustering Experiments

*6.2.1. Setup for Clustering Experiments.* Clustering algorithms partition a set of objects, YouTube videos in our case, into groups called *clusters*. For our experiments we chose *k-Means* [Han and Kamber 2001].

For a number of clusters $k = 2, 3, 4, 5$ we considered all possible $\binom{7}{k}$ combinations of tuples of categories for the 7 YouTube categories. For each tuple, we selected uniformly at random 900 videos per category, performed the k-means algorithm and computed the macro-averaged accuracy for the $k$-tuples (averaging across accuracies for each tuple).

Let $k$ be the number of categories and clusters, $N_i$ the total number of clustered documents in $category_i$, $N_{ij}$ the number of videos contained in $category_i$ and having cluster label $j$. Unlike classification results, the clusters do not have explicit topic labels. We define the clustering accuracy as follows:

$$\text{accuracy} = \max_{(j_1,\ldots,j_k)\in perm((1,\ldots,k))} \frac{\sum_{i=1}^{k} N_{i,j_i}}{\sum_{i=1}^{k} N_i}. \tag{14}$$

Accuracy for clustering can be seen as a measure of how well the partitioning produced by a clustering method reflects the actual category 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 $k$ categories).

We constructed feature vector representations *BaseOrig*, *NTag*, *RedNTag*, *TagRank*$\Gamma$ with number of iterations ($\Gamma$ = 2,4,8), *LdaTopics* and *LdaTopics*$\Delta$ (with $\Delta$ = 50, 100, 200, 500 and 1000 latent topics), and *LdaTagRecComb*$\Delta$ as described above. For comparison, we additionally simulated a “random clustering” baseline (*Rand*) by assigning videos uniformly at random to clusters, and averaging clustering accuracies over 100,000 runs.

*6.2.2. Results of Clustering Experiments.* The results of the comparison are shown in Tables V(a) and VI(b). The main observations are the following.

—Clustering using automatically generated tags outperforms clustering based on the original tags (*BaseOrig*). For example, for clustering with $k = 3$ we increased the accuracy from approximately 43% to 56% for *RedNTag*.

Table V. Classification AUC for Different Video Representations using Automatic Tagging

(a) Classification: AUC (Area Under the ROC Curve) values for *BaseOrig*, *NTag*, *RedNTag*, and *TagRank*Γ; T = 10, 25, 50, 100, 400 training videos per category

| | BaseOrig | NTag | RedNTag | TagRank2 | TagRank4 | TagRank8 |
|---|---|---|---|---|---|---|
| T = 10 | 0.682 | 0.751 | 0.753 | 0.734 | 0.729 | 0.726 |
| T = 25 | 0.758 | 0.805 | 0.807 | 0.789 | 0.783 | 0.780 |
| T = 50 | 0.807 | 0.838 | 0.840 | 0.827 | 0.822 | 0.819 |
| T = 100 | 0.844 | 0.862 | 0.864 | 0.857 | 0.853 | 0.850 |
| T = 200 | 0.874 | 0.881 | 0.884 | 0.882 | 0.880 | 0.878 |
| T = 400 | 0.896 | 0.899 | 0.901 | 0.903 | 0.900 | 0.899 |

(b) Classification: AUC (Area Under the ROC Curve) values for *LdaTopics*, *LdaTagRec*, and *LdaTagRecComb*; T = 10, 25, 50, 100, 400 training videos per category, and 50, 100, 200, 500, and 1000 latent topics

| LdaTopics | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| T = 10 | 0.708 | 0.696 | 0.663 | 0.628 | 0.607 |
| T = 25 | 0.750 | 0.749 | 0.731 | 0.710 | 0.689 |
| T = 50 | 0.773 | 0.782 | 0.774 | 0.770 | 0.757 |
| T = 100 | 0.792 | 0.804 | 0.807 | 0.814 | 0.811 |
| T = 200 | 0.808 | 0.822 | 0.828 | 0.846 | 0.850 |
| T = 400 | 0.819 | 0.839 | 0.848 | 0.867 | 0.878 |

| LdaTagRec | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| T = 10 | 0.723 | 0.710 | 0.697 | 0.693 | 0.691 |
| T = 25 | 0.772 | 0.769 | 0.764 | 0.769 | 0.768 |
| T = 50 | 0.806 | 0.805 | 0.807 | 0.816 | 0.817 |
| T = 100 | 0.833 | 0.836 | 0.841 | 0.851 | 0.853 |
| T = 200 | 0.858 | 0.861 | 0.867 | 0.878 | 0.882 |
| T = 400 | 0.882 | 0.884 | 0.890 | 0.899 | 0.902 |

| LdaTagRecComb | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| T = 10 | 0.752 | 0.751 | 0.750 | 0.754 | 0.754 |
| T = 25 | 0.794 | 0.798 | 0.800 | 0.809 | 0.810 |
| T = 50 | 0.826 | 0.829 | 0.833 | 0.842 | 0.843 |
| T = 100 | 0.850 | 0.854 | 0.858 | 0.866 | 0.868 |
| T = 200 | 0.872 | 0.875 | 0.879 | 0.886 | 0.888 |
| T = 400 | 0.893 | 0.895 | 0.898 | 0.904 | 0.905 |

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

—Both video representation approaches based on LDA lead to a decrease in clustering performance, even when compared to the simple baseline using the original tags of the video. A possible explanation might be that clustering makes already implicit use of feature cooccurrences, and LDA introduces more noise than information.

—Combining LDA based tag recommendations with neighbor-based tagging improves clustering performance. In contrast to classification, however, using just neighbor-based information provides the best results, illustrating again quality and complementary character of annotations generated by these methods.

As expected, clustering accuracies are, in general, substantially lower than classification accuracies, as there is no information through training data available in the considered unsupervised scenario. Furthermore, clustering becomes more difficult if the data set covers a larger number of topics. Approaches like restrictive clustering [Siersdorfer and Sizov 2004], which trade data coverage for higher accuracy, can help to make unsupervised data organization more robust in practice.

Table VI. Clustering Accuracy for Different Video Representations Using Automatic Tagging

(a) Clustering accuracy for baseline video feature vectors using *BaseOrig*, *NTag*, *RedNTag*, and *TagRank*Γ; k = 2, 3, 4, 5 clusters. (*Rand*: simulated "random clustering")

| | Rand | BaseOrig | NTag | RedNTag | TagRank2 | TagRank4 | TagRank8 |
|---|---|---|---|---|---|---|---|
| k = 2 | (0.509) | 0.571 | 0.677 | 0.717 | 0.654 | 0.650 | 0.65 |
| k = 3 | (0.346) | 0.430 | 0.547 | 0.559 | 0.533 | 0.526 | 0.522 |
| k = 4 | (0.264) | 0.369 | 0.480 | 0.486 | 0.451 | 0.440 | 0.434 |
| k = 5 | (0.214) | 0.322 | 0.410 | 0.428 | 0.411 | 0.392 | 0.391 |

(b) Clustering accuracy for *LdaTopics*, *LdaTagRec*, and *LdaTagRecComb*; k = 2, 3, 4, 5 clusters, and 50, 100, 200, 500, and 1000 latent topics

| LdaTopics | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| k = 2 | 0.552 | 0.556 | 0.547 | 0.543 | 0.534 |
| k = 3 | 0.393 | 0.398 | 0.386 | 0.393 | 0.380 |
| k = 4 | 0.312 | 0.285 | 0.296 | 0.307 | 0.305 |
| k = 5 | 0.273 | 0.235 | 0.246 | 0.265 | 0.252 |

| LdaTagRec | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| k = 2 | 0.568 | 0.587 | 0.577 | 0.583 | 0.594 |
| k = 3 | 0.451 | 0.447 | 0.456 | 0.486 | 0.481 |
| k = 4 | 0.358 | 0.364 | 0.385 | 0.399 | 0.410 |
| k = 5 | 0.310 | 0.329 | 0.334 | 0.353 | 0.368 |

| LdaTagRecComb | | | | | |
|---|---|---|---|---|---|
| | 50 topics | 100 topics | 200 topics | 500 topics | 1000 topics |
| k = 2 | 0.622 | 0.613 | 0.651 | 0.650 | 0.651 |
| k = 3 | 0.488 | 0.510 | 0.551 | 0.552 | 0.548 |
| k = 4 | 0.409 | 0.426 | 0.456 | 0.472 | 0.468 |
| k = 5 | 0.360 | 0.386 | 0.388 | 0.408 | 0.414 |

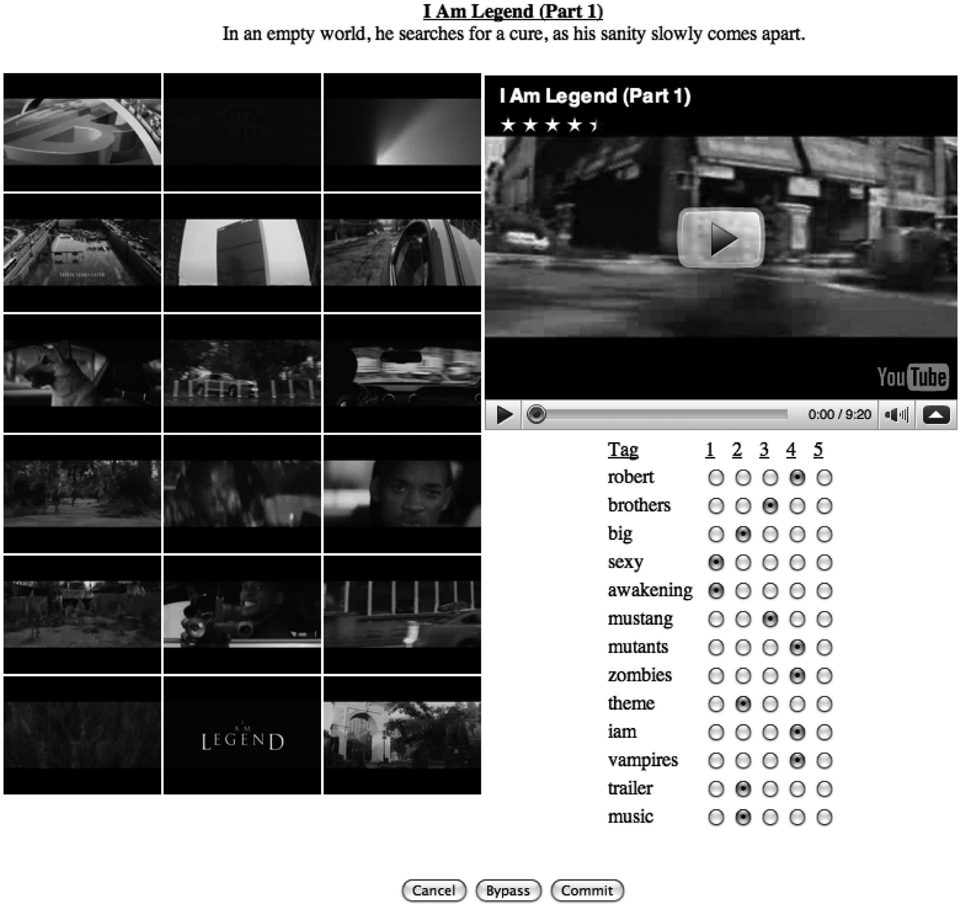
### 6.3. User-Based Evaluation

To support the results obtained for automatic data organization, we conducted an additional user-based experiment where three assessors provided relevance judgments for the automatically generated tags from *NTag* and *RedNTag*.

A web service to gather judgments was implemented, it included a playable version of the video, automatically extracted key-frames, the title and the description to help them understand the content. The assessors were instructed to judge the relevance of tags considering their suitability to describe the video at the following different levels: description (e.g., car), identification (e.g., George W. Bush), interpretation (e.g., holiday) and emotion (e.g., happiness).

The videos presented to the evaluators were selected uniformly at random and the tags were displayed in random order. The evaluators were asked to rate each new tag using a five-level Likert scale [Likert 1932] (1=*not relevant*, 2=*slightly relevant*, 3=*relevant*, 4=*very relevant*, 5=*completely relevant*). This interface is depicted in Figure 10.

A total of $3,578$ tags, for 300 different videos, were manually assessed using the described interface, for an average of 12 tags rated per video. We studied the average relevance for the set of generated tags, autotags($v_i$), for different values of threshold $\delta$. For this purpose, we sorted the list of tags in decreasing order of rel($t, v_i$) value, and selected $\delta$ values at different levels of that list, in increments of 10% of the full autotags($v_i$) size. The results are shown in Figure 11. Table VII shows $\delta$ threshold values for the different levels considered. Note that values of $\delta$ differ for each method studied; this is a consequence of the different scoring strategies of each method which result in distinct relevance values distributions.



**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 | ○ | ● | ○ | ○ | ○ |

Fig. 10. Web interface for the collection of relevance judgments.

The figure reveals decreasing relevance pattern for growing values of autotags($v_i$) set sizes. That is, by raising the threshold $\delta$, we can obtain increasingly higher average relevance values. Therefore, a correlation exists between relevance as provided manually by the assessors and automatically by our automatic tagging strategies. Depending on the specific application scenario, $\delta$ can be tuned to optimize results. 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 (note the truncated Y-axis in Figure 11), in comparison to results obtained for automatic data organization.

## 7. CONCLUSIONS

In this article, we conducted an in-depth analysis on previously unexplored content-based links between videos in YouTube, the most popular video social network. We defined a methodology to generate Visual Affinity Graphs, revealing different kinds of visual relationships between elements in the network. More than 38,000 videos, comprising over 2,800 hours, were downloaded and analyzed. The resulting visual affinity graphs showed a noticeable amount of redundancy in the set, with over a third of the results being visually linked to others. The most common kind of relationship was

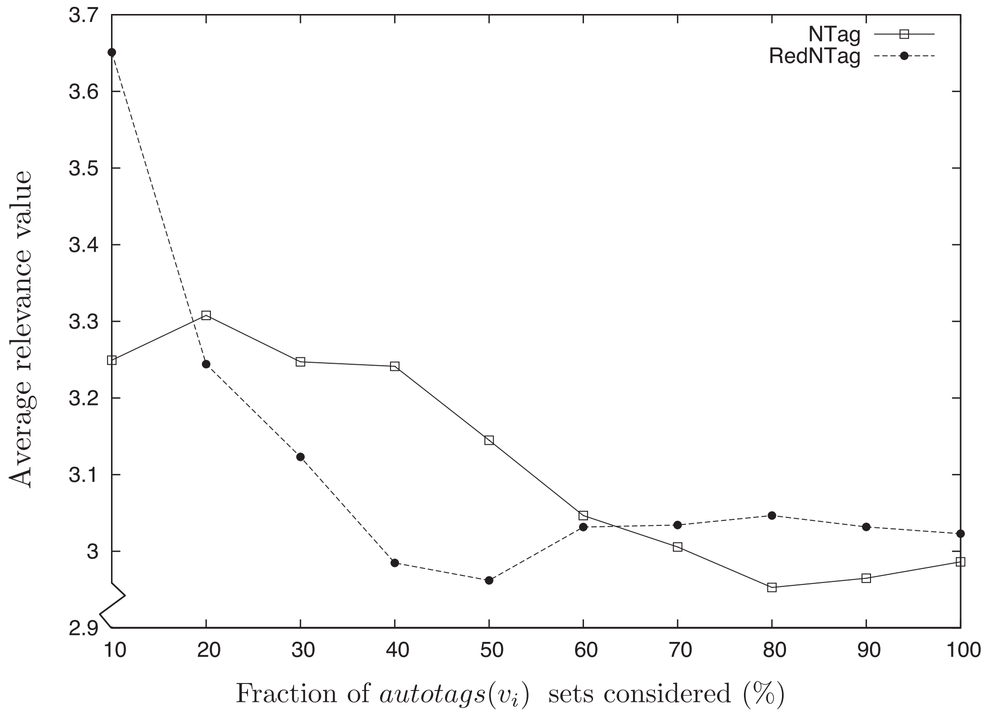


Fig. 11. Average relevance judged manually by assessors for increasing sizes of autotags($v_i$).

Table VII. Values of $\delta$ Threshold for the Different Levels Shown in Figure 11

| $\delta$ @ level | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| NTag | 1.003 | 1.000 | 0.999 | 0.738 | 0.383 | 0.183 | 0.113 | 0.075 | 0.048 | 0.001 |
| RedNTag | 1.275 | 1.000 | 1.000 | 0.730 | 0.455 | 0.231 | 0.126 | 0.081 | 0.050 | 0.001 |

duplication, accounting for 15.80% of the test collection. Additional experiments were conducted to establish the reasons supporting the large presence of redundant content. We found that gaining recognition by re-uploading popular content is an important factor. The ability to provide different annotations and language variations of the footage have also been shown to be potential reasons for duplicating visual content.

As an application, we have shown that content redundancy in social sharing systems can be used to obtain richer annotations for shared objects. More specifically, we 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 significantly improve automatic structuring and organization of content; our preliminary user evaluation indicates an information gain for viewers of the videos.

We think that the proposed techniques have direct applications to search improvement, where augmented tag sets can reveal resources previously concealed.

## ACKNOWLEDGMENTS

We thank the three anonymous reviewers and Associate Editor Stefan Rüger for their thoughtful and helpful suggestions.

## REFERENCES

ABBASI, R. AND STAAB, S. 2009. Richvsm: enriched vector space models for folksonomies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia (HT'09)*. ACM, New York, NY, 219–228.

ALLEN, J. F. 1983. Maintaining knowledge about temporal intervals. *Comm. ACM 26,* 11, 832–843.

AMES, M. AND NAAMAN, M. 2007. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*. ACM, New York, NY, 971–980.

ARTEMENKO, O., MANDL, T., SHRAMKO, M., AND WOMSER-HACKER, C. 2006. Evaluation of a language identification system for mono- and multilingual text documents. In *Proceedings of the ACM Symposium on Applied Computing (SAC'06)*. ACM, New York, NY, 859–860.

BLEI, D. M., NG, A. Y., AND JORDAN, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, 993–1022.

CHA, M., KWAK, H., RODRIGUEZ, P., AHN, Y.-Y., AND MOON, S. 2007. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC'07)*. ACM, New York, NY, 1–14.

CHARIKAR, M. S. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC'02)*. ACM, New York, NY, 380–388.

CHENG, X., DALE, C., AND LIU, J. 2007. Understanding the characteristics of internet short video sharing: Youtube as a case study. *CoRR* abs/0707.3670.

CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. J. 2009. Power-law distributions in empirical data. *SIAM Rev. 51,* 4, 661+.

CRASWELL, N. AND SZUMMER, M. 2007. Random walks on the click graph. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*. ACM, New York, NY, 239–246.

DATTA, R., LI, J., AND WANG, J. Z. 2005. Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'05)*. ACM, New York, NY, 253–262.

DUMAIS, S., PLATT, J., HECKERMAN, D., AND SAHAMI, M. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM'98)*. ACM, New York, NY, 148–155.

DUYGULU, P., BARNARD, K., DE FREITAS, J., AND FORSYTH, D. 2006. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European Conference on Computer Vision (ECCV'02)*. A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., Lecture Notes in Computer Science, vol. 2353. Springer, Berlin, 349–354.

FAWCETT, T. 2003. ROC Graphs: Notes and practical considerations for data mining researchers. Tech. rep. HPL-2003-4, HP Labs.

GILL, P., ARLITT, M., LI, Z., AND MAHANTI, A. 2007. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC'07)*. ACM, New York, NY, 15–28.

GOLDER, S. AND HUBERMAN, B. A. 2006. The structure of collaborative tagging systems. *J. Infor. Sci. 32,* 2, 198208. cite arxiv:cs/0508082.

GRANGIER, D. AND BENGIO, S. 2008. A discriminative kernel-based approach to rank images from text queries. *IEEE Trans. Patt. Anal. Mach. Intell. 30,* 8, 1371–1384.

GUILLAUMIN, M., MENSINK, T., AND VERBEEK, J. 2009. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of the 12th International Conference on Computer Vision (ICCV'09)*. IEEE Computer society, 309–316.

HALVEY, M. J. AND KEANE, M. T. 2007. Analysis of online video search and sharing. In *Proceedings of the 18th Conference on Hypertext and Hypermedia (HT'01)*. ACM, New York, 217–226.

HAN, J. AND KAMBER, M. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

HEYMANN, P., RAMAGE, D., AND GARCIA-MOLINA, H. 2008. Social tag prediction. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*. ACM, New York, NY, 531–538.

HOTHO, A., JÄSCHKE, R., SCHMITZ, C., AND STUMME, G. 2006. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*. Lecture Notes on Artifical Intelligence, vol. 4011. Springer, Berlin, 411–426.

JACCARD, P. 1901. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles 37*, 241–272.

Jang, D., Yoo, C. D., Lee, S., Kim, S., and Kalker, T. 2009. Pairwise boosted audio fingerprint. *Trans. Info. For. Sec. 4,* 4, 995–1004.

Jing, Y. and Baluja, S. 2008. Pagerank for product image search. In *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*. ACM, New York, NY, 307–316.

Joachims, T. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods–Support Vector Learning*, B. Schlkopf, C. Burges, and A. Smola, Eds., MIT Press, Cambridge, MA, Chapter 11.

Joly, A., Buisson, O., and Frelicot, C. 2007. Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Trans. Multimedia 9,* 2, 293–306.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46*, 604–632.

Krestel, R., Fankhauser, P., and Nejdl, W. 2009. Latent dirichlet allocation for tag recommendation. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys'09)*. ACM, New York, NY, 61–68.

Lebossé, J., Brun, L., and Pailles, J. C. 2007. A robust audio fingerprint extraction algorithm. In *Proceedings of the 4th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA'07)*. ACTA Press, Anaheim, CA, 269–274.

Li, J. and Wang, J. Z. 2006. Real-time computerized annotation of pictures. In *Proceedings of the 14th Annual ACM International Conference on Multimedia (MULTIMEDIA'06)*. ACM, New York, NY, 911–920.

Li, X., Snoek, C. G. M., and Worring, M. 2008. Learning tag relevance by neighbor voting for social image retrieval. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval (MIR'08)*. ACM, New York, NY, 180–187.

Likert, R. 1932. A technique for the measurement of attitudes. *Archiv. Psych. 22,* 140, 1–55.

Lindstaedt, S., Mörzinger, R., Sorschag, R., Pammer, V., and Thallinger, G. 2009. Automatic image annotation using visual content and folksonomies. *Multimedia Tools Appl. 42,* 1, 97–113.

Ling, X., Jia, J., Yu, N., and Li, M. 2008. Tagrank—measuring tag importance for image annotation. In *Proceedings of the International Conference on Multimedia and Expo*. IEEE, 109–112.

Lipczak, M. and Milios, E. 2010. The impact of resource title on tags in collaborative tagging systems. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia (HT'10)*. ACM, New York, NY, 179–188.

Liu, L., Lai, W., Hua, X.-S., and Yang, S.-Q. 2006. Video histogram: A novel video signature for efficient web video duplicate detection. *Adv. Multimedia Model.* 94–103.

Liu, Y., Zhao, W. L., Ngo, C. W., Xu, C. S., and Lu, H. Q. 2010. Coherent bag-of audio words model for efficient large-scale video copy detection. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'10)*. ACM, New York, NY, 89–96.

Lovejoy, W. S. and Loch, C. H. 2003. Minimal and maximal characteristic path lengths in connected sociomatrices. *Soc. Netw. 25,* 4, 333–347.

Manku, G. S., Jain, A., and Das Sarma, A. 2007. Detecting near-duplicates for web crawling. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM, New York, NY, 141–150.

Marlow, C., Naaman, M., Boyd, D., and Davis, M. 2006. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the 17th ACM Conference on Hypertext and Hypermedia (HT'06)*. ACM, New York, NY, USA, 31–40.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet.

Nakano, Y., Nakamura, M., and Okabe, Y. 2007. Analysis for topological properties of the network feeding usenet news. In *Proceedings of the International Symposium on Applications and the Internet (SAINT'07)*. IEEE Computer Society, Los Alamitos, CA.

Oostveen, J. C., Kalker, T., and Haitsma, J. 2001. Visual hashing of digital video: applications and techniques. In *Applications of Digital Image Processing XXIV 4472,* A. G. Tescher, Ed., 1, 121–131.

Page, L., Brin, S., Motwani, R., and Winograd, T. 1998. The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford Digital Library Technologies Project.

San Pedro, J. 2008. Fobs: An open source object-oriented library for accessing multimedia content. In *Proceedings of the 16th ACM International Conference on Multimedia (MM'08)*. ACM, New York, NY, 1097–1100.

San Pedro, J., Denis, N., and Dominguez, S. 2005. Video retrieval using an edl-based timeline. In *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis*. Lecture Notes in Computer Science, vol. 3522. Springer, Berlin, 401–408.

San Pedro, J. and Dominguez, S. 2007. Network-aware identification of video clip fragments. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR'07)*. ACM, New York, NY, 317–324.

SAN PEDRO, J., KALNIKAITE, V., AND WHITTAKER, S. 2009. You can play that again: exploring social redundancy to derive highlight regions in videos. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI'09)*. ACM, New York, NY, 469–474.

SHIVAKUMAR, N. AND GARCIA-MOLINA, H. 1995. Scam: A copy detection mechanism for digital documents. In *Proceedings of the 2nd International Conference in Theory and Practice of Digital Libraries (DL'95)*.

SIERSDORFER, S., SAN PEDRO, J., AND SANDERSON, M. 2009. Automatic video tagging using content redundancy. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, New York, NY, 395–402.

SIERSDORFER, S. AND SIZOV, S. 2004. Restrictive clustering and metaclustering for self-organizing document collections. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*. ACM, New York, NY, 226–233.

SIGURBJÖRNSSON, B. AND VAN ZWOL, R. 2008. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*. ACM, New York, NY, 327–336.

SMEULDERS, A. W. M., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. Patt. Anal. Mach. Intell. 22,* 12, 1349–1380.

SNOEK, C. G. M. AND WORRING, M. 2008. Concept-based video retrieval. *Found. Trends Inf. Retr. 2,* 4, 215–322.

STOKES, N. AND CARTHY, J. 2001. Combining semantic and syntactic document classifiers to improve first story detection. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*. ACM, New York, NY, 424–425.

SZEKELY, B. AND TORRES, E. 2005. Ranking bookmarks and bistros: Intelligent community and folksonomy development. http://torrez.us/archives/2005/07/ 13/tagrank.pdf (unpublished).

VAN DONGEN, S. 2000. A cluster algorithm for graphs. Tech. rep. INS-R0010. *National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam*.

WU, X., HAUPTMANN, A. G., AND NGO, C.-W. 2007. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA'07)*. ACM, New York, NY, 218–227.

YANG, H. AND CALLAN, J. 2006. Near-duplicate detection by instance-level constrained clustering. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. ACM, New York, NY, 421–428.

ZHANG, B., LI, H., LIU, Y., JI, L., XI, W., FAN, W., CHEN, Z., AND MA, W.-Y. 2005. Improving web search results using affinity graph. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. ACM, New York, NY, 504–511.

Received October 2009; revised October 2010, January 2011; accepted March 2011