Chapter 12

Web-based Multimedia Information Extraction Based on Social Redundancy

Jose San Pedro
Telefonica Research
Via Augusta 177
08021 Barcelona
Spain
jsanpedro@mac.com

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

Vaiva Kalnikaite
University of Sheffield
Regent Court
211 Portobello St.
Sheffield S1 4DP, UK
vaivak@gmail.com

Steve Whittaker
UC Santa Cruz
1156 High St
Santa Cruz
CA, 95064, USA
swhittak@ucsc.edu

12.1. Introduction

Social networking sites are among the most frequently visited on the web (Cha et al. 2007) and their use has expanded into professional contexts for expertise sharing and knowledge discovery (Millen, Feinberg and Kerr 2006). These virtual communities can be enormous, with millions of users and shared resources. Social multimedia websites, such as YouTube, are particularly popular. Network traffic involving YouTube accounts for 20% of web traffic and 10% of all internet traffic (Cheng, Dale and Liu 2007). This chapter focuses on new techniques for information extraction from such social multimedia sites, to support improved search and browsing.

Social multimedia sites are organic and user-centric, with few centralised control policies or systematic ways to organise content. They rely on user annotations, feedback and access behaviours to manage and present content. The drawbacks of this user-centric focus are:

- *Sparsity and limits of tags*: Although user annotations have been shown to support various applications such as topic detection and tracking (Allan, Papka and Lavrenko 1998), information filtering (Zhang, Callan and Minka 2002), and document ranking (Zhang et al. 2005), they nevertheless have limitations. Manually annotating content is a time consuming process. And not all content receives equal attention from taggers (Sun and Datta 2009). As a consequence, metadata and annotations are often very sparse for large parts of the collection. Furthermore, keywords and community-provided tags may lack consistency and present numerous irregularities (e.g., abbreviations and typos). In addition, only a minority of users generate tags so that metadata may represent the interests of a small minority (Paolillo and Penumarthy 2007). Overall, this makes it hard to rely on tag-based techniques for automatic data organization, retrieval and knowledge extraction.
- *Lack of granular access*: The reliance on user tags and related metadata mean that the majority of the tools used to access multimedia data, are not directly content–based. Instead they apply to the entire file. For example tags or popularity scores are applied to an entire video, music track or movie. Yet for extended content, users may

want to identify specific elements within that content, e.g. highlights or favourite moments, and neither tags nor popularity applies at this level of granularity.

- *Duplicated content*. Despite the use of tools for its removal, recent studies (Cha et al. 2007; Wu, Hauptmann and Ngo 2007) report significant amounts of redundant footage in video sharing websites, with over 25% near-duplicate videos detected in search results. Such redundant content can affect retrieval performance, by increasing browsing time to skip repeated entries, or require additional processing to eliminate highly overlapping content. For this reason, the literature has considered redundancy a problem for social multimedia, and various studies have proposed techniques for its elimination (Zhang et al. 2005; Wu, Hauptmann and Ngo 2007).

In this chapter we focus on the analysis of the leading social multimedia platform, YouTube, addressing the above problems using a novel hybrid approach which combines content analysis and user-centric metadata. Rather than viewing redundancy as a *problem* to be excised from the system, we exploit it. We use state of the art content-based copy retrieval (CBCR) techniques to identify visually overlapping data – which provide us with two types of information. First we use redundancy to *detect multimedia highlights* and second we use it to *propagate tags* across sparsely tagged collections. More specifically we use content techniques to compute the network of connections between elements of a video collection in an unsupervised way. We call this the *Visual Affinity Graph*. It conveys information about the data set provided manually by the community, but extracted automatically by the content-analysis technique. This implicit user behaviour can be used to generate high level semantic cues for multimedia content in an unsupervised way, helping to bridge the so called 'Semantic Gap', i.e. the problem of inferring the high-level concepts by which humans perceive the world, from the set of low-level features which can be automatically extracted from multimedia data (Enser and Sandom 2003). In this chapter, we show how to harness this content-derived social knowledge to solve two video-related problems:

- *Summarization of video content:* Summarizing multimedia is known to be an extremely difficult problem (Yang and Hauptmann 2008). Automatic methods rely on low level features that do not tend to map well to aspects of content that users find interesting or useful (Christel et al. 1998). The analysis of redundancy in YouTube enables us to study uploaders' behaviour. When several users have independently selected and uploaded the same video, a consensus about its importance can be inferred. We use this idea to derive an importance metric for uploaded video sequences. Such a metric enables us to derive highlights: regions of socially agreed interest within the timeline of the video.

- *Improvement of annotations:* We also take advantage of the *Visual Affinity Graph* to *propagate annotations* (i.e. tags) between related videos, utilizing the graph edges to spread community knowledge into the network. Uploaders of overlapping sequences of the same video provide their personal perspective on its content in the form of different annotations. This propagation addresses the *tag sparsity* problem; by combining tags for a related resource to achieve a more comprehensive description.

In both cases, we describe our algorithms and present an evaluation showing the utility of our new approach.

The approach differs from other chapters in this book that address video analysis. Several chapters point to the need to use contextually available metadata (Chapter 7). Others analyse the speech contained in the video using NLP methods (Chapters 10 and 11), or combine metadata and linguistic content. Our approach is perhaps closest to that of

Chapter 8 which combines visual content analysis with speech analysis. However one important contrast is that in our approach we use content analysis to analyse people's social behaviours to determine important regions, as well as to propagate end user tags across videos. Although various later chapters address human social behaviour (Chapters 16 and 18), in our approach we are not interested in social behaviour per se, but rather in how we can apply this to solve summarisation or tagging problems.

This chapter is organised as follows. In Section 12.2 we first describe the CBCR techniques that allow us to detect redundancy in video collections and to generate the network of content-based connections. Section 12.3 describes the related work, algorithm and evaluation for a social summarization tool for video. Section 12.4 describes a method to improve annotations of elements in community websites, again presenting related work, algorithm and evaluation. We conclude in Section 12.5, with a discussion of future work.

12.2. Redundancy Detection and Generation of the Visual Affinity Graph

Here we introduce the tools used for the automatic detection of redundant content in video-based Web 2.0 sites, commonly referred to as Content-based Copy Retrieval (CBCR) tools. These techniques achieve very high accuracy in the detection of exact and near duplicates, providing reliable insights into the properties of the video collection. CBCR analysis provides a set of connected video pairs which we represent, for convenience, as a graph.

12.2.1. CBCR Methods

There is a plethora of literature about near-duplicate detection of video content. CBCR can be considered a particular case of Query-By-Example Content-Based Information Retrieval (Joly, Buisson and Frelicot 2007), where the result set only includes duplicates of the target sample. Two main problems need to be tackled by CBCR systems. On the one hand, the computational complexity associated to the comparison process requires sophisticated representation (Liu et al. 2006) and indexing (Joly, Buisson and Frelicot 2007) techniques. On the other, the concept of "identity" has to be carefully dealt with, as videos may experience transformations of visual content during different stages of their life-cycle. The detection of near-duplicate text documents presents analogous problems (Huffman et al. 2007). Many of the principles used by text-based duplicate detection techniques can be adapted to the video domain. Fingerprints are the most commonly used detection tool; they are generated using specific features of visual content, such as temporal video structure (San Pedro, Denis and Dominguez 2005) or time-sampled frame invariants (Joly, Buisson and Frelicot 2007) .

CBCR's main application is to Digital Rights Management (DRM), where it is used to detect unauthorized usage of video footage. The proliferation of multimedia content broadcasting channels, from current standards of TV (DVB-T/S) to the latest trends in mobile and wireless communications (UMTS), produce a massive amount of broadcast multimedia content that requires advanced content-based analysis algorithms to guarantee copyright agreements. CBCR is also used in many other applications. The ability to obtain airtime statistics for specific clips can be exploited in many different ways, for instance to monitor advertisement campaigns, an application of fundamental importance in the advertising industry.

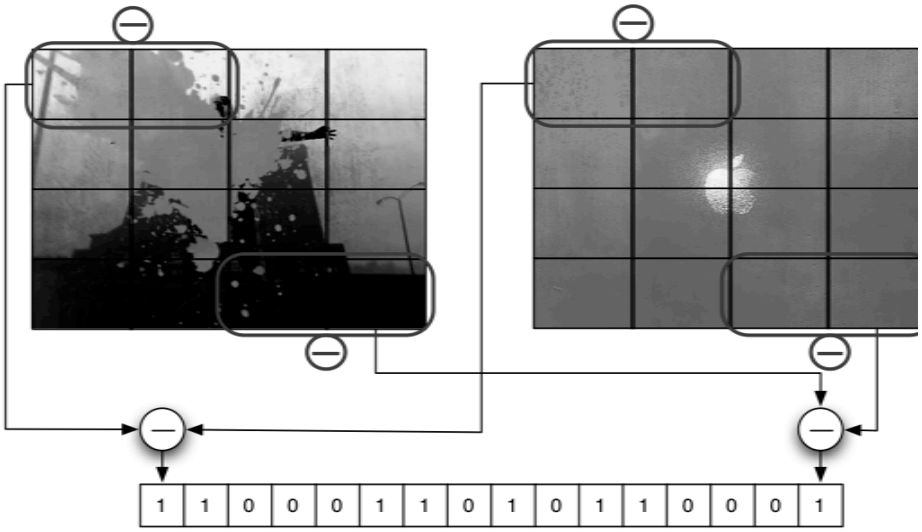

Figure 12.1 Hash value computation, using a 2x2 spatio-temporal Haar filter. A 16 bit hash code is produced for every pair of selected frames.

12.2.2. System Description

We use a copy detection system based on robust hash fingerprint techniques. Robust hash functions generate similar hash values for similar input messages, in contrast to standard hash functions, which try to reduce collisions and specifically create very different values for similar messages (useful in cryptographic applications). All videos used by the system are pre-analyzed and transformed into strings of hash values, i.e. fingerprints, which represent the evolution of their visual features in time.

Robust hash fingerprints achieve very high comparison effectiveness, and allow us to handle significant changes in visual quality, including frame size reduction, bitrate reduction and other artifacts generated in re-encoding processes (Oostveen, Kalker and Haitsma 2001). On the other hand, the string-based nature of the comparison makes these systems very sensitive to changes in the temporal features. Hash signatures are sequentially created from video frames; this may create problems as frame rate may differ between videos being compared: larger frame sequences will create larger hash strings.

To handle these difficulties, we base our system on (San Pedro and Dominguez 2007a). The chosen robust hash function works in the luminance colour space and uses a 2x2 spatio-temporal Haar filter to compute values using pairs of video frames, as illustrated by Figure 12.1. Frame pairs are selected using an entropy-based sub-sampling approach. Shannon's entropy is computed for the luminance distribution of each frame. The time series generated by computing this value for every time t of the video is analyzed using a sliding window. For each window W_k, a pair of frames is selected using points of maximum, t_M, and minimum, t_m, entropy in this local scope.

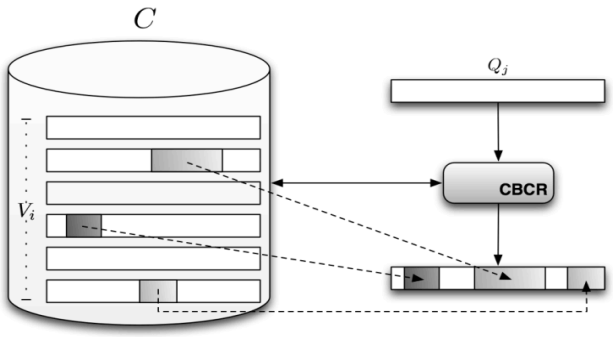

Figure 12.2. Video identification scenario. A video collection, C, is queried via the CBCR system using Q_j as input query.

This content-based sub-sampling approach aims to repeatedly select the same time points for different variations of the input video. The resulting sub-sampled set of frames can therefore be used to generate a temporal resolution independent fingerprint for video *v*, that we denote *h(v)*. Video comparison is then performed in the hash space, using standard string matching algorithms. Such algorithms allow the duplicate detection both at the string level (near-duplicate videos) and at the substring level (overlapping videos) (San Pedro and Dominguez 2007b).

At the application level, our CBCR system maintains a collection of searchable content:

$C = \{v_i : 1 \le i \le |C|\}$, where |C| denotes the cardinality of the collection.

This video collection, C, can be queried using input video streams, which we denote as Q_j. The system is able to identify if a given incoming stream, Q_j, is a near-duplicate or overlaps with one or more elements in collection *C*. This process is depicted in Fig. 12.2.

12.2.3. Building the Visual Affinity Graph

In our social website application, we want to analyze the whole collection of resources, C, for redundancy. Note that every video $v_i \subset C$ may potentially include sequences from every other v_j. To perform a comprehensive identification, we need to consider the set $C'=C-\{v_i\}$ and the input query $Q_i = v_i$ for every element $i \in [1, |C|]$.

We transform this set of connected video pairs into a weighted undirected graph, *the Visual Affinity Graph*, *G=(V,E)*. In this graph, nodes (*V*) represent individual videos of collection C. On the other hand, edges (*E*) link together nodes when the corresponding videos overlap. Formally, these two sets are defined as:

$$V = \{v_i \in C : \exists v_j \rightarrow v_i \cap v_j \neq \varnothing\}$$

$$E = \{\{v_i, v_j\} : v_i \cap v_j \neq \varnothing\},\ \ v_i, v_j \in V$$

Every edge is then weighted as a function of the duration of the overlap between the pair of videos it connects

$$w(v_i, v_j) = \left| v_i \cap v_j \right|$$

A graph representation enables us to take advantage of the numerous mathematical tools available for these structures, and provides an intuitive and organized view of the derived relationships. This graph formalizes all the visual connections in the video set.

12.3. Social Summarization

12.3.1. Description

Summarization of content is crucial, as it provides a compact, rapidly analysable, representation of a larger resource. However, it is hard for the following two reasons:

- Summarization requires a high level of content understanding to enable selection of the scenes that best represent the entire content. For video, this understanding is limited by the *Semantic Gap*; audio-visual content summarization techniques are currently restricted to the use of low level features to infer the semantic high-level features that users demand. The inference process achieves poor precision, and results are often far from being semantically meaningful (Yang and Hauptmann 2008).

- Summarization is known to be a highly subjective task (Jones 1993). The set of summary scenes selected by different users are often very different. But standard machine learning/test collection methods that depend on having multiple human judges to rate video content are known to be time-consuming and costly.

In this section, we present a summarization technique that circumvents these limitations by obtaining *implicit* ratings inferred from user behaviour on Web 2.0 sites. This community-based approach is important because it provides an alternative way to bridge the *Semantic Gap*, exploiting social consensus to reveal video highlights.

Our algorithm harnesses users' redundant uploading patterns from the Visual Affinity Graph, to obtain human-generated semantic cues about content in an unsupervised fashion. We exploit the idea that the intentional uploading by multiple users of the same clip from within a given video indicates the importance of that particular video clip. This intentional clip selection is analogous to users' active website linking behaviour: popularity information that is exploited by major search engines in algorithms such as PageRank (Page et al. 1999). If such a clip is a part of a larger original video (normally the case given the duration limits imposed on uploading content to Web 2.0 sites), such duplication provides information about regions of importance within the larger event. We exploit this information about important regions to build a summary of the entire video.

12.3.2. Methodology

The first stage in our social video summarization technique involves locating all relevant video resources from a social sharing site, such as YouTube. In our scenario, we work externally to the social website, so we need to recreate the topology of the relevant subnetwork in our local context, enabling us to restrict our study to this subset. We are looking to extract highlights, i.e. clips of the highest interest, of a specific video, v, for which we already know some meta-information (e.g. title, director, publisher, etc). We

use these metadata to construct a query to identify relevant content using the social website search engine. The returned results constitute our subset of relevant items. Optionally, we can supplement the results set by using 'related content' recommended lists, provided by many of these portals.

We construct the *Visual Affinity Graph* using all collected results, along with the original video footage we are trying to summarize, v. Following the notation introduced in Section 12.2, we consider our collection C to contain just our target video v. Each of the results obtained from the social website will constitute our input queries, Q_j. The CBCR system will then detect all the timepoints where results collected overlap with v. Discovering these overlaps between content enables us to define an importance metric. This metric is based on the *frequency* with which different scenes of the target video v have been uploaded into the social sharing engine. We choose frequency as a simple and intuitive metric although of course other metrics are possible. We can express this importance value for any given time, t, of the target video as

$$\upsilon_t = \frac{\left|\{Q_j : Q_j \cap v^{(t)} \neq \varnothing\}\right|}{\left|\{Q_j : Q_j \cap v \neq \varnothing\}\right|}$$

where $v^{(t)}$ denotes the frames at time t of video v, and the set intersection represents the visual overlap, which we find automatically using our CBCR algorithm. The time series obtained by the computation of υ_t for all possible values of t we call the "*importance timeline*" (see Figure 12.3). The importance timeline conveys information about the video scenes that are most frequently uploaded, and therefore, most important according to our definition. Peaks indicate potential highlights of V.

12.3.3. Highlights Selection

The importance timeline is a continuous function, but for our analysis we need to identify discrete regions to present to users. In this last stage of the analysis, we identify specific highlighted regions, by analysing the neighbourhood around maximal points υ_k in the series to find the appropriate boundaries of each region. To select the region boundaries, we compute a threshold $\theta(\upsilon_k, D)$ dynamically as a function of two variables: the importance value at the current maximal point being considered, υ_k, and an optional maximum duration for the selected region, D:

$$\theta(\upsilon_k, D) = \omega \cdot \upsilon_k \cdot \varphi(l, D)$$

In the previous expression, ω is a fixed weight factor, l denotes the current length of the region selected so far, and $\phi(l,D)$ is a weighting function so that

$$\lim_{l \to D} \varphi(l, D) = \frac{1}{\omega}, \;\; i.e. \;\; \varphi(l.D) = \frac{l}{\omega \cdot D}$$

To enhance the boundaries for our detected regions, we proceed to analyse the video to identify its shot structure. A shot is a continuous set of frames captured in a single camera operation. It is the lowest level meaningful unit into which videos can be divided. We use a colour histogram difference-based algorithm (San Pedro, Denis and Dominguez 2005) to detect abrupt transitions between consecutive shots and take advantage of these partitions to constrain the time points which will be later used as region boundaries. Figure 12.3 shows an example of the result obtained with this procedure.

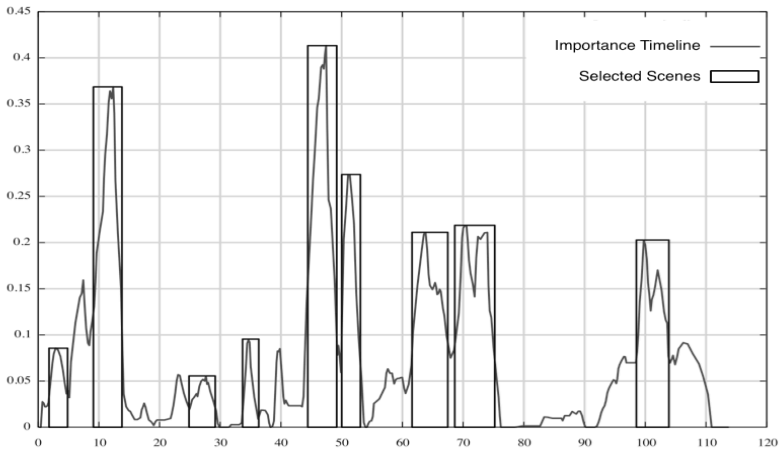

Figure 12.3. Importance timeline. The time series depicts the values of v_t found for each time point of a given video. Bounding boxes delimit highlights selected.

12.3.3. Evaluation

We evaluated the quality of automatically extracted highlights by comparing them to subjective judgements by film experts. For the experiment, we selected 7 popular films: The Godfather (1972), Star Wars (1977), Forrest Gump (1994), The Matrix (1999), Lord of the Rings (2001), Pirates of the Caribbean (2003) and 300 (2006). First, we crawled YouTube to retrieve all the uploaded videos related to these films, and applied our algorithm to derive highlights. We then presented film experts with a series of these highlights and asked them to rate their significance compared with other clips selected from the film. We wanted to see whether film experts' judgements replicated our automatic analysis of region significance.

We built a web-based application to collect experts' importance ratings. For each film, participants were asked to rate a set of 18 clips presented 3 clips at a time: one clip our algorithm rated as of *high* importance, another of *intermediate* and a final one of *low* importance. Each triplet was selected from a random pool of clips (containing 6 top, 6 medium and 6 low importance clips), and order of presentation was randomised with respect to algorithm importance. Detected highlights, *i.e.* peaks in Figure 12.3, were classed as *high* importance. Clips below this but with more than 0 uploads were categorized as *medium*, and clips with no uploads were classed as *low*. Each clip represented a full scene, ranging from 1-3 minutes. Clip cuttings were adjusted using shot boundaries to preserve the integrity of self-contained individual regions.

We recruited 45 participants by posting to special interest film sites and social networking sites. They generated a total of 86 sessions, and 945 clip judgements, with 15 users rating multiple films. We asked our participants to select a film and provide their knowledge level about it (*passing knowledge*, *knowledgeable*, *expert*, *world expert*). Users were only allowed to rate the same film once, although they could rate multiple films. Users could choose different knowledge levels for each film, although we asked them to focus on films they were familiar with. Users were asked to decide on their 'favourite', 'not so

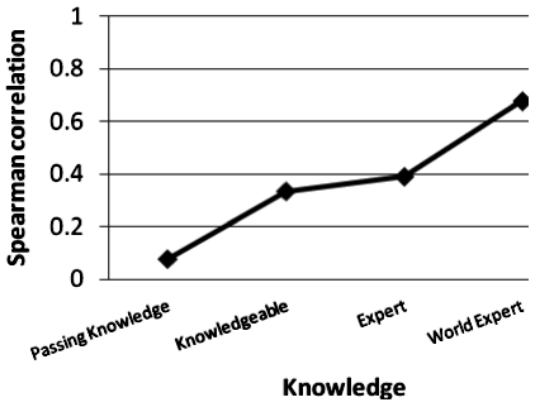

Figure 12.4. Correlation as a function of the knowledge of participants.

favourite', and 'least favourite' clip in each triplet. We also recorded how long users played each clip and collected qualitative comments about their ratings. They had no prior knowledge of this project.

12.3.3.1. Experimental Results

For each clip we analysed the relation between (a) the ratings provided by participants and (b) the importance coefficient generated by the algorithm. Both metrics ranged in value between 1 (for important regions), 2 (for intermediate regions) and 3 (for unimportant regions). We first correlated *all* user and algorithm importance metrics. This was significant (Spearman's rho = 0.351, $p < 0.01$, 944 df) showing a clear relation between automatic methods and user judgements.

Effects of Knowledge Level:

Despite our instructions, some users rated films that they were less familiar with. Such non-expert ratings are likely to correlate less well with our algorithm. If we consider just those judgements where participants rated themselves as 'expert' or 'world expert', The correlation increases to 0.467 ($p<0.01$, 320 df). As Figure 12.4 shows, higher correlations are obtained for higher user knowledge levels. The contingency table is shown in Table 12.1. The main diagonal represents the frequency of agreement, while the values off the diagonal represent disagreements between automatic and user provided judgements. The table shows high agreement, with the main diagonal being more highly populated than off diagonal scores.

Film experts also spent less time re-playing the clips before judging them. A one-way ANOVA with play time as dependent variable and knowledge level as independent variable showed a significant effect ($F(3, 944)=7.53$, $p<0.0001$). These findings validate our method, offering behavioural support for participants' self-evaluations of expertise, as experts should have less need to access films to make their judgements.

Count		Algorithm Importance Rating			Total
		1	2	3	
User Importance Rating	1	**72**	21	14	107
	2	18	**55**	34	107
	3	17	31	**59**	107
Total		107	107	107	321

Table 12.1. Contingency table for participants of *two* top knowledge levels: a) Expert and b) World expert.

There were also differences between films. Figure 12.5 suggests that action films led to better correlated ratings. For slower pace films, it is somewhat more difficult to agree on key moments, as there may be fewer visually salient regions, and subjective rating factors become more significant.

Number of Uploads and Relation between User Judgements and Algorithm Scores:

We then applied a linear regression test to determine the extent to which our algorithm's accuracy was affected by the number of overlapping uploads. We found that the number of uploads was unrelated to correlations between algorithm and user judgements (R squared = 0.0001, $p > 0.10$). This suggests we had collected enough overlapping videos to generate sufficient data to obtain accurate algorithm scores.

12.3.4. Discussion

We proposed a social video summarisation method based on frequency of content redundancy detected and formalized on the Visual Affinity Graph. An extensive user evaluation revealed that our algorithm is able to derive highlights judged by experts to be important. Further tests confirmed that the correlation between algorithmic and user judgments tend to increase when experts are more knowledgeable. Last, we found no relation between the number of redundant videos found and the user-algorithm correlation, indicating that YouTube contains enough overlaps to systematically produce dense Visual Affinity Graphs, and therefore accurate highlight detection. However, this social summarisation technique can only be applied to popular media streams where there is enough content overlap.

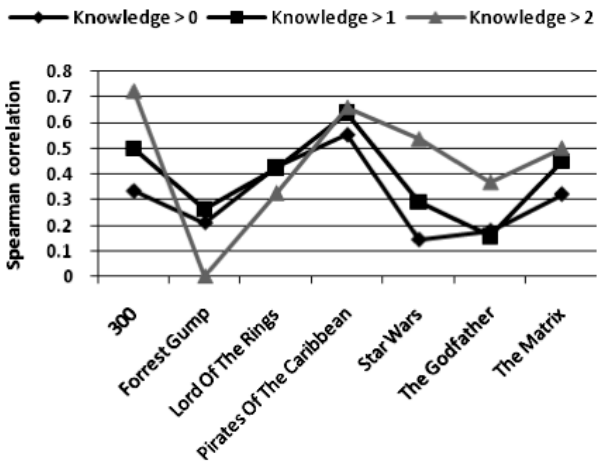

Figure 12.5. Correlation as a function of knowledge for different films.

12.4. Improving Annotations

12.4.1. Description

Metadata allow us to attach structured information to data objects, videos in our case, with the purpose of providing easily parseable meta-information about them. These annotations can be automatic or manual in origin. Automatic annotations are normally restricted to technical and low level aspects of the content. For example, DV video capturing devices encode in the stream meta-information about time/date, zoom and white balance, among other things. On the other hand, manual annotations such as tags, titles and descriptions are provided by users and normally describe the data object at a much higher, semantic level. For this reason, manual annotations are commonly exploited for the effective retrieval of complex kinds of information contained in multimedia data. However, content annotation in social multimedia sites requires active intellectual effort and is very time consuming. In consequence, community-provided annotations can lack consistency and present numerous irregularities (e.g. abbreviations and typos) that degrade the performance of applications such as automatic data organization and retrieval. Furthermore users tend to tag content of interest so that tags can be unevenly distributed in a collection, again potentially compromising retrieval.

In this section we analyse content to achieve more comprehensive and accurate annotations of video content. We use the Visual Affinity Graph to locate groups of videos with duplicated content. Each uploader provides annotations in the form of different tags expressing their personal perspective on a given clip including characters, dates, and other more general concepts. We use the new content-based links provided by the Visual Affinity Graph to propagate tags between related videos, utilizing visual affinity to extend community knowledge in the network. With this approach, we exploit visual redundancy to combine information about related clips extending the (currently limited) set of metadata currently available for that content,. We propose and evaluate different tag propagation techniques, and show systematic improvements in quality and applicability of the resulting annotations.

12.4.2. Methodology

Our algorithm uses edges of the Visual Affinity Graph to propagate tags between neighbours. Neighbours of this graph are known to share the same visual content, indicating joint semantic content which is often reflected by corresponding annotations. For each connected video in the Visual Affinity Graph, we compute a set of *autotags*, i.e. tags imported from visually overlapping videos. Every element of autotags is assigned a weight which determines the importance of the new tag for the video. These weights are determined by the amount of overlaps occurring between the source and the target videos. Different propagation strategies can be defined, affecting the set of autotags and their associated weights.

For *neighbour-based tagging*, we transform the undirected overlap graph into a directed and weighted graph $G'(V, E')$, with (v_i, v_j) and $(v_j, v_i) \in E'$ iff $\{v_i, v_j\} \in E$. The weight $\omega(v_i, v_j)$ 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 use the heuristic weighting function

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

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 overlaps with video v_i. Note that in case v_i and v_j are duplicates, if v_i is a parent of v_j (meaning that v_i is the more general video, and v_j can be considered as a specific region from this video) then the weighting function ω assigns the maximum value 1 to (v_i, v_j).

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

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

i.e., we compute a weighted sum of influences of the overlapping videos containing tag t. For a set of overlapping videos, i.e. neighbours in the graph, situations with multiple redundant overlaps as shown in Figure 12.6 can occur. In order to avoid a too high increase of the relevance values for automatically generated tags in comparison to original tags, we propose a relaxation method for regions with redundant overlap, reducing the contribution of each additional video in each region by a factor α ($0 < \alpha \leq 1$). We call this variant *overlap redundancy aware tagging*.

We use the obtained relevance values for all tags from the overlapping videos, to generate sets $autotags(v_i)$ of automatically assigned new tags for each video $v_i \in V$ applying a threshold δ for tag relevancy:

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

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

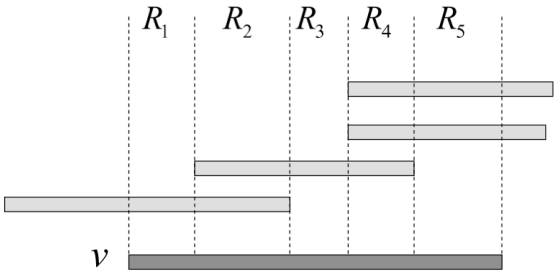

Figure 12.6. Overlap regions $R_1, \ldots, R_5$ for a video *v* covered by 4 other videos.

12.4.3. Evaluation

We have presented different methods for automatically generating tags, resulting in richer feature representations of videos. Machine learning algorithms can make use of this feature information to generate models, and to automatically organise the data. We hypothesise that these extended feature representations of videos lead to the generation of better models improving the quality of automatic data organisation. We test this by performing a quantitative evaluation of the new set of tags, by examining the influence of enhanced tag annotations on automatic video classification.

12.4.3.1. Test Collection

We created our test collection from YouTube by formulating queries and subsequent searches for "related videos", analogous to the typical user interaction with Web 2.0 sites. Note that, in our scenario, we require access not just to metadata, but to the actual video content. Therefore, crawling and storage requirements are much higher, imposing a limit on the subset size we could gather. Given that an archive of most common queries does not exist for YouTube, we selected our set of queries from Google's Zeitgeist archive from 2001 to 2007. These are generic queries, used to search for web pages and not videos. Therefore, some of them might not be suitable for video search (e.g. "windows update"). We set a threshold on the number of search results returned by queries. Those for which YouTube returned less than 100 results were considered not suitable for video search, and ignored. In total, 579 queries were accepted, for which the top 50 results were retrieved. Altogether, we collected 28,216 videos using those queries (some of the results were not accessible during the crawling because they were removed by the system or by the owner). A random sample of these videos was used to extend the test collection by gathering related videos, as offered by the YouTube API. In total, 267 queries for related videos were performed, generating 10,067 additional elements. The complete test collection included 38,283 video files for a total of over 2900 hours.

12.4.3.2 Classification

Classifying data into thematic categories usually follows a supervised learning paradigm and is based on training items that need to be provided for each topic. Linear support vector machines (SVMs) construct a hyperplane $\vec{w} \cdot \vec{x} + b = 0$ that separates the set of positive training examples from a set of negative examples with maximum margin. We

	BaseOrig	NTag	RedNTag
T=10	0.5794	0.6341	0.6345
T=25	0.6357	0.7203	0.7247
T=50	0.7045	0.7615	0.7646
T=100	0.7507	0.7896	0.7907
T=200	0.7906	0.8162	0.8176
T=400	0.8286	0.8398	0.8417

Table 12.2. Classification accuracy with T=10, 25, 50, 100, 400 training videos using different tag representations for videos.

used the SVMlight (Joachims 1999) implementation of support vector machines (SVMs) with linear kernel and standard parameterization in our experiments; SVMs have been shown to perform very well for text-based classification tasks (Dumais et al. 2006).

As classes for our classification experiments, we chose YouTube categories containing at least 900 videos in our collection. These were: "Comedy", "Entertainment", "Film & Animation", "News & Politics", "Sports", "People & Blogs", and "Music". We performed binary classification experiments for all 21 possible combinations of these class pairs using balanced training and test sets. Settings with more than two classes can be reduced to multiple binary classification problems that can be solved separately. For each category, we randomly selected 400 videos for training the classification model and a disjoint set of 500 videos for testing. We trained different models based on T=10, 25, 50, 100, 200, and all 400 training videos per class. We compared the following methods for constructing feature vectors from video tags:

- **BaseOrig**: Vectors based on the original tags of the videos (i.e. tags manually assigned by the owner of the video in YouTube). This serves as the baseline for the comparison with our vector representations based on automatic tagging.
- **NTag**: Vectors constructed based on the tags and relevance values produced by simple neighbour-based tagging (Section 12.4.2) in addition to the original tags.
- **RedNTag**: Vectors using tags generated by overlap redundancy aware neighbour-based tagging plus the original tags as described in Section 12.4.2. We did not pursue any extensive parameter tuning and chose $\alpha = 0.5$ for the relaxation parameter.

Our quality measure is the fraction of correctly classified videos (*accuracy*). Finally, we computed micro-averaged results for all topic pairs. The results of the comparison are shown in Table 12.2. We observe that classification taking automatically generated tagging into account clearly outperforms classification using just the original tags. This holds for both tag propagation methods. For classification using 50 training documents per class, for example, we increased the accuracy from approximately 70% to 76%, quite a significant gain. Overlap redundancy aware neighbour-based tagging provides slightly, but consistently more accurate results than the simple neighbour-based tagging variant.

12.4.4. Discussion

In this section, we have proposed a methodology to improve annotations of shared video resources in Web 2.0 sites. We take advantage of the existing annotations in the network, and use the edges of the Visual Affinity Graph to spread this social knowledge to other resources. Different propagation strategies are discussed and evaluated, resulting in

significant improvements on the amount and quality of video annotations. Our experiments show that with the enriched set of tags, better classification models can be generated allowing for improved automatic structuring and organization of content.

12.5. Conclusions

In this chapter we have presented a novel approach for information extraction in multimedia-enabled Web 2.0 systems. We take advantage of what is usually viewed as a undesirable feature of these websites, content redundancy, to establish connections between resources, and exploit them to increase the knowledge about the collection. Such connections are formalized into the so called 'Visual Affinity Graph'. We have presented two applications of such a graph.

Firstly, we present an approach to the unsupervised selection of highlights from video footage. Our method successfully circumvents the limitations of current content analysis methods, exploiting knowledge implicitly available in video sharing social websites. The *Visual Affinity Graph* is computed and used to identify redundant region uploads from different users. We then use this information to build importance time series that we analyze to locate the highlights of videos. The results were shown to be reliable in a subsequent user evaluation when expert users' judgements agreed with our algorithm.

The ability to detect semantically meaningful regions from video clips in an unsupervised fashion has many applications. For example, techniques that analyse users repeated access and annotation behaviours have been used to detect important regions of both recorded lectures and meetings (Kalnikaite and Whittaker 2008). A similar social summarization approach might also be used for finding favourites in collection of other kinds of shared media, such as Flickr's photo collections. However there are interesting future empirical questions about the influence of content: would there be user consensus about highlights for data such as political debates or news where content is less well structured and evaluations of it more esoteric?

Secondly, we have also shown that content redundancy in social sharing systems can be used to obtain richer annotations for shared objects. More specifically, we have used content overlap in the video sharing Web 2.0 environments to establish new connections between videos forming a basis for our automatic tagging methods. Classification experiments show that the additional information obtained by automatic tagging can largely improve automatic structuring and organization of content. We think that the proposed technique has direct application to search improvement, where augmented tag sets can reveal previously concealed resources.

This work reports another important direction for future Web 2.0 and social information processing research. So far previous work has focused on using social information (such as tags) to organize and retrieve *textual* data. Techniques for accessing text are relatively well understood however. But with the massive recent increases of multimedia data such as pictures and video, we desperately need new methods for managing multimedia information. Hybrid social and content analysis methods such as those we report here represent a promising future direction for accessing such data.