Visual Memes in Social Media

Tracking Real-World News in YouTube Videos

Lexing Xie
Australian National University
lexing.xie@anu.edu.au

Apostol Natsev
IBM Research
natsev@us.ibm.com

John R. Kender
Columbia University
jrk@cs.columbia.edu

Matthew Hill
IBM Research
mh@us.ibm.com

John R. Smith
IBM Research
jsmith@us.ibm.com

ABSTRACT

We propose visual memes, or frequently reposted short video segments, for tracking large-scale video remix in social media. Visual memes are extracted by novel and highly scalable detection algorithms that we develop, with over 96% precision and 80% recall. We monitor real-world events on YouTube, and we model interactions using a graph model over memes, with people and content as nodes and meme postings as links. This allows us to define several measures of influence. These abstractions, using more than two million video shots from several large-scale event datasets, enable us to quantify and efficiently extract several important observations: over half of the videos contain re-mixed content, which appears rapidly; video view counts, particularly high ones, are poorly correlated with the virality of content; the influence of traditional news media versus citizen journalists varies from event to event; iconic single images of an event are easily extracted; and content that will have long lifespan can be predicted within a day after it first appears. Visual memes can be applied to a number of social media scenarios: brand monitoring, social buzz tracking, ranking content and users, among others.

Categories and Subject Descriptors: J.4 [Social and Behavioral Sciences]: Sociology, I.4.9 [Image Processing and Computer Vision]: Applications

General Terms: Algorithms, Measurement, Experimentation.

1. INTRODUCTION

Important happenings from around the world are increasingly captured on video and uploaded to news and social media sites. The ease of publishing and sharing videos has outpaced the progress of modern search engines, collaborative tagging sites, and content aggregation services—leaving users to deal with a deluge of content [2]. This information overload problem is particularly prominent for linear media (such as audio, video, animations), where at-a-glance impressions are hard to develop and are often unreliable. While text-based information networks such as Twitter rely on retweets [5, 18], hashtags, mentions, or trackbacks to identify influence and trending topics [1], similar functions for large video-sharing repository is lacking. A reliable video-based "quote" tracking and popularity analysis system would find immediate practical applications in many domains—e.g., selecting of the "most typical" video for a given topic or collection; measuring influence and ranking people in news events; improving targeted advertising based on page/author influence; and denoising video search and query expansion results, to name a few.

We propose to use visual memes for making sense of video "buzz". A *meme*[1] is defined as a cultural unit (e.g., an idea, value, or pattern of behavior) that is passed from one person to another in social settings. We define a *visual meme* as a short segment of video that is frequently remixed and reposted by more than one author. Video-making requires significant effort and time, so we regard reposting a video meme as a deeper stamp of approval or awareness than simply viewing a video, leaving a comment, giving a rating, or sending a tweet. Example video memes are shown in Figures 1, 2 and 3, represented in a static keyframe format. We can see that each meme instance is semantically consistent, despite many variations in the videos that contain them, such as size, coloring, captions, editing, and so on.

Figure 1 summarizes the approach proposed in this paper. We develop a large-scale event monitoring system for video content, using generic text queries as a pre-filter for content collection on a given topic (Box D). We deploy this system for YouTube, and collect large video datasets over a range of topics. We then perform fast visual meme detection on tens of thousands of videos and millions of video shots (Box A). We showcase the potential applications of visual memes using a network model over the meme videos and authors (Box B). Using this model, we derive graph metrics that capture content influence and user roles. Using such visual meme extraction and exploitation strategies, we have made several observations on real-world news event collections (Box C), such as: over half of the event videos contain remixed content, and about 70% of authors participate in video remixing; video view counts are a poor proxy for the likelihood of a video being reposted; over 50% of memes are discovered and re-posted within 3 hours after their first appearance; meme influence indices can be used to delineate the roles of different user groups, such as *mavens* or *connec-*

*Area chair: Mor Naaman

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

[1] http://wordnetweb.princeton.edu/perl/webwn?s=meme

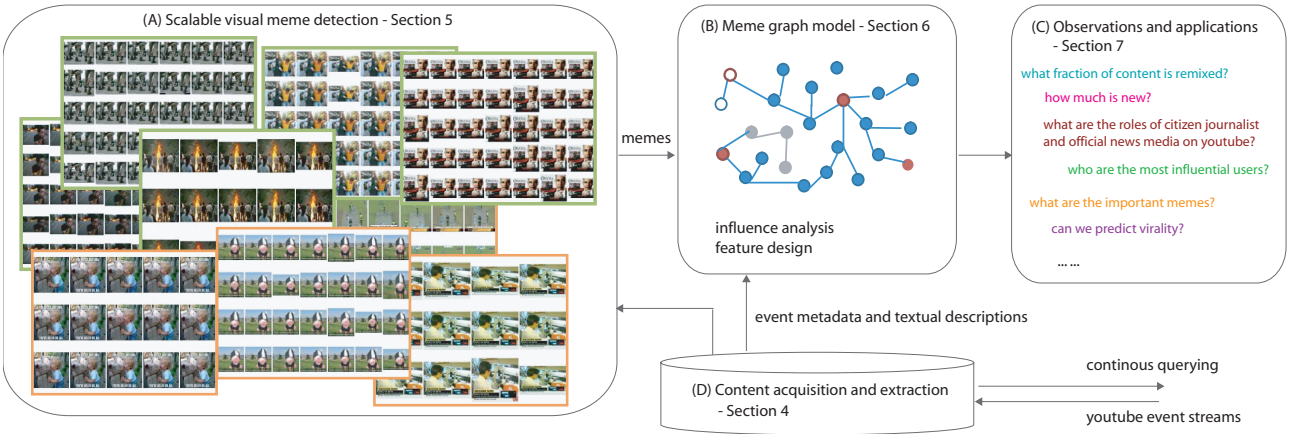

Figure 1: Overview of visual meme tracking and analysis in social event streams. In Box A, the border color of meme clusters denotes the event they are from: green (top): Iran; orange (bottom): SwineFlu.

tors who play notable roles in social changes [14]. We use features derived from the meme network model to predict the lifespan of memes, achieving an area-under-ROC-curve (AUC) measure of 0.78.

The main contributions of this work are as follows:

- We propose *visual memes* as a novel tool to track large-scale video remixing in social media. We implement a scalable system that can extract all memes from over a million video shots, in a few hours on a single CPU.
- We design and implement the first large-scale event-based social video monitoring and visual content analysis system.
- We propose an application for visual memes by building a network model on videos and authors, which can in turn be used to characterize user roles and predict meme lifespan.
- We conduct empirical evaluations with several large event datasets, producing observations about percentage of video remix, user participation, timing of video meme production, meme popularity against traditional metrics, and different user group roles.

2. RELATED WORK

This work relates to active research areas in both multimedia analysis and social media mining.

YouTube has been the focal platform for many social network monitoring studies. The first large-scale YouTube measurement study [6] characterized content category distributions, and tracked exact duplicates of popular videos. Benevenuto et al. studied video response actions on YouTube using metadata [3], and De Choudhury et al. monitored user comments to determine interesting conversations [10]. Recently, early views of YouTube videos have been used to predict ultimate popularity, characterized by view counts [25].

Quoting, duplication, and reposting are popular phenomena in online information networks. One well-known example is retweeting on micro-blogs [5, 18], where users often quote the original text message verbatim, having little freedom for remixing and context changes within the 140 character limit. Another example is MemeTracker [19], which tracks the lifecycles of popular phrases among blogs and news websites. Prior studies have shown that the frequency of video reuse can be used as an implicit video quality indicator [23]. However, none of the prior work has defined the unit for *retweet* or *meme* on a video sharing network.

Tracking near-duplicates in images and video has been a problem of interest since the early years of content-based retrieval. Recent focus of this problem has been on user-dependent definitions of duplicates [8], speeding up detection on image sequence, frame, or local image points [26], and scaling out to web-scale computations using large compute clusters [20]. We note, however, that most prior work in this area is concerned with optimizing retrieval accuracy of detecting near-duplicate frames or sequences, rather than tracking large-scale duplication behavior. Kennedy and Chang [17] tracked editing and provenance of images on the web, with a focus on distinguishing different types of image edits and their ideological perspective. Our work, in comparison, tracks large-scale video remixes using both content and metadata such as authorship and creation time, and focuses on inferring social roles in video propagation.

Several recent works have looked at YouTube phenomena—Biel and Gatica-Perez [4] focused on individual social behavior such as non-verbal cues, while Hong et al. [15] presented content summarization by monitoring a query over time. In comparison, we use visual memes to capture the behavior of large groups and to track information dissemination.

3. VISUAL MEMES AND VIDEO REMIXES

Visual memes are defined as frequently reposted video segments or images. It has been observed that users tend to create "curated selections based on what they liked or thought was important" ([24], page 270). News event collections are particularly suited for studying large-scale user curation, since remixing is more prevalent here than on video genres designed for self-expression, such as video blogs. The unit of interaction appears to be video segments, consisting of one or a few contiguous shots. The remixed shots typically contain minor modifications that include video formatting changes (such as aspect ratio, color, contrast, gamma), and video production edits (such as the superimposition of text, captions, borders, transition effects). Most of these are well-known as the targets of visual copy detection benchmarks [22]. In this paper, *meme* refers both to individual

Figure 2: Visual meme shots and meme clusters. (Left) Two YouTube videos that share multiple different memes. Note that it is impossible to tell from metadata or the YouTube video page that they shared content, and that the appearance of the remixed shots (bottom row) has large variations. (Right) A sample of other meme keyframes corresponding to one of the meme shots, and the number of videos containing this meme over time – 193 videos in total between June 13 and August 11, 2009.

instances, visualized as representative icons (as in Figure 2 Left and Figure 3), and to the entire equivalence class of re-posted near-duplicate video segments, visualized as clusters of keyframes (as in Figure 1 and Figure 2 Right).

Intuitively, re-posting is a stronger endorsement, requiring much more effort than simply viewing, commenting on, or linking to the video content. A re-posted visual meme is an explicit statement of mutual awareness, or a relevance statement on a subject of mutual interest. Hence, memes can be used to study virality, lifetimes and timeliness, influential originators, and (in)equality of reference.

4. MONITORING EVENTS ON YOUTUBE

YouTube has become a virtual worldwide bazaar for video content of almost every type. With more than 48 hours of video being added every minute [2], it is a living marketplace of ideas and a vibrant recorder of current events.

We use text queries to pre-filter content, thus making the scale of monitoring feasible. We use a few generic, time-insensitive text queries as content pre-filters. The queries are manually designed to capture the topic theme, as well as the generally understood cause, phenomena, and consequences of the topic. For example, our queries on the "global warming" topic consist of *global warming, climate change, green house gas, CO_2 emission*, whereas the "swine flu" topic expands into *swine flu, H1N1, H1N1 travel advisory, swine flu vaccination*, and so on. We aim to create queries covering the main invariant aspects of a topic, but automatic time-varying query expansion is open for future work. We use the YouTube API to extract video entries for each query, sorted by relevance and recency. The API will return up to 1000 entries per query, so varying the sorting criteria helps to increase content coverage and diversity. The retrieved video entries are those responding to keyword queries based on YouTube's proprietary algorithm, and often contain entries not directly relevant to the event being monitored. We filter the results to restrict the video database to unique videos, removing redundant entries that responded to multiple queries or whose YouTube identifier matched one that had previously been gathered. Then, for each unique video, we segment it into shots using thresholded color histogram differences. For each shot we randomly select and extract a frame as keyframe, and extract visual features from each keyframe. We process the XML metadata associated with each video, and extract information such as author, publish date, view counts, and free-text title and descriptions.

We use the term *buzz* to refer to all the videos that respond to keyword queries on YouTube, although their content may not be directly related to the target event or topic of interest. We use the term *meme videos* to refer to videos containing one or more memes. The volume of buzz and memes are telling indicators of event evolution in the real world, and we present a few examples in Figure 3.

Figure 3(a) graphs the volume of all unique videos acquired according to their upload date. There are local peaks on the Swine Flu topic during April-May 2009 when new cases were spreading over the globe, and in October-November 2009 when vaccination first became available in the US, following with a steady volume decrease into 2010. For the 21-month period shown for Pakistan politics, there are two notable peaks: in December 2007, at the time of assassination of Benazir Bhutto; and in February-May 2009, during a series of crises, including serial bombings, an attack on the Sri-Lanka cricket team, and nation-wide protests.

Figure 3(b) tracks and illustrates the volume of meme videos for the Iranian Politics topic (dataset Iran3 in Table 1). The number of meme videos is significant—hundreds to thousands per day. There are three prominent peaks in June-August 2009 corresponding to important events in the real world[2]. The first mid-June peak reflects a highly controversial election prompting massive protests and violent clashes. A second mid-June peak captures a viral amateur video on the shooting of Neda Soltan, which became the symbol for the whole event. A third peak in mid-July corresponds to a Friday prayer sermon which drew over two million people, an event described as "the most critical and turbulent Friday prayer in the history of contemporary Iran"[2].

5. SCALABLE VISUAL MEME DETECTION

Detecting visual memes in a large video collection is a non-trivial problem. There are two main challenges. First, remixing online video segments changes their visual appearance, adding noise as the video is edited and re-compressed.

[2]See timeline: http://en.wikipedia.org/wiki/Timeline_of_the_2009_Iranian_election_protests

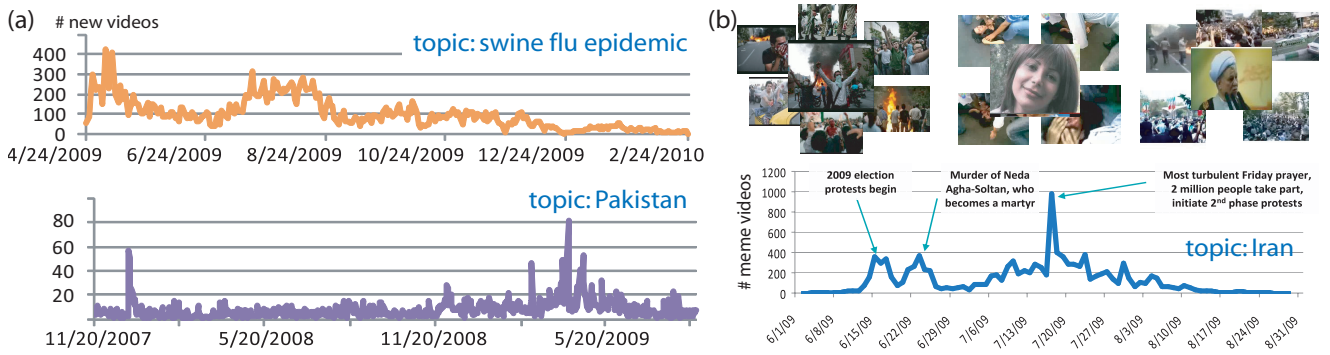

Figure 3: Volume of event buzz and visual memes. (a) Event buzz: number of new videos uploaded daily for two topics. (b) Number of videos containing visual meme on the Iran3 topic, illustrated with the representative memes on a timeline, June-August 2009.

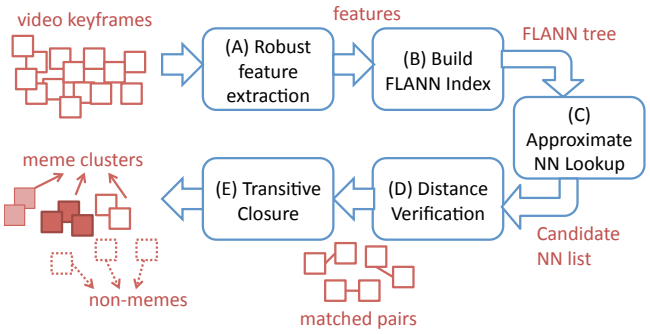

Figure 4: Flow diagram for visual meme detection.

Second, finding all pairs of near-duplicates by matching all N shots against each other has a complexity of $O(N^2)$, which is infeasible for collections containing millions of shots.

Our operational definition of a meme is a reposted video segment that starts and ends at shot boundaries. This definition motivates our processing pipeline of using a single keyframe to represent a video shot (Section 4) without sacrificing matching quality, as the feature-based shot detector is generally robust to intra-shot changes but sensitive to large inter-shot variations in visual appearance.

Our process for detecting video memes is outlined in Figure 4. The input to this system is a set of video frames, and the output splits this set into two parts. The first part consists of a collection of meme clusters, where frames in the same cluster are considered near-duplicates with each other. The second part consists of the rest of the frames, which are not considered near-duplicates with any other frame. Blocks A and D address the robust matching challenge using color correlogram features and query-adaptive thresholding, and blocks B, C and E address the scalability challenge using approximate nearest-neighbor (ANN) indexing and an efficient set transitive closure algorithm.

5.1 Robust keyframe matching

Our solution to the visual appearance challenge is to normalize frames visually, use robust features, and tune the frame matching methods. Before feature extraction, we perform a series of pre-processing steps to normalize the image and reduce noise. These include removing blank or uninformative frames (based on entropy threhsolding); detecting and removing frame borders of uniform colors; normalizing the aspect ratio; performing de-noising; and applying contrast-limited histogram equalization to correct for contrast and gamma differences. We use a frame similarity metric based on the *color correlogram* [16] that captures the local spatial correlation of pairs of colors. The color correlogram is rotation-, scale-, and to some extent, viewpoint-invariant. It was designed to tolerate moderate changes in appearance and shape that are largely color-preserving, e.g., viewpoint changes, camera zoom, noise, compression, and to a smaller degree, shifts, crops, and aspect ratio changes. We also use a "cross"-layout that extracts the descriptor only from horizontal and vertical central image stripes, thereby emphasizing the center portion of the image, while disregarding the corners. This layout improves robustness with respect to text and logo overlay, borders, crops, and shifts. It is also invariant to horizontal or vertical flips, while capturing some spatial layout information. We extract the auto correlogram in a 166-dimensional perceptually quantized HSV color space, and the resulting descriptor with a "cross" layout has 332 dimensions. The result of the above processing (Figure 4 Box A) is a set of features, one per input frame.

Furthermore, we use query-adaptive thresholding on the L_2 distance of the correlogram features to generate a binary judgement for each candidate pair of frames as to whether they are a near-duplicate pair. This corresponds to Figure 4 Box D. The main purpose of this threshold-tuning step is to relax the match threshold for complex query frames and to tighten the threshold for visually simple frames (e.g., blank frames at the extreme). For a given video keyframe q and its correlogram feature $\mathbf{f}_q$, the threshold for determining matches is parameterized as $T_q = \tau \frac{|\mathbf{f}_q|_2}{|\mathbf{f}_{max}|_2}$. Here $|.|_2$ is the L_2 vector norm. $\mathbf{f}_{\max}$ is the collection max feature vector, composed of the largest observed coefficients for each dimension. τ is a global distance threshold tuned on an independent validation dataset. The $|\mathbf{f}_q|_2$ term scales τ based on the information content of q: it lowers the effective threshold for those frames that are visually simple, such as frames with uniform colors or simple charts, which can otherwise lead to false or trivial matches. At the same time, it increases the threshold for highly complex query frames.

5.2 Scaling up

Our solution to the complexity challenge is to use an indexing scheme for fast approximate nearest neighbor (ANN)

look-up. Exhaustively finding all pairs of frames that are within a given distance threshold has complexity $O(N^2)$, while ANN indexing can speed this up significantly. We use the Fast Library for Approximate Nearest Neighbor (FLANN) [21] to implement the indexing structure and ANN lookup. FLANN automatically selects the best indexing structure between k-means tree and kd-tree, and chooses the appropriate tree parameters for a given dataset that optimize the trade off between running time and query approximation error. This corresponds to Figure 4 Box B. FLANN allows us to set the maximum number of candidate nodes m to check in a search so that each query runtime is bound to $O(m)$. Executing N queries against the entire set of N keyframes therefore takes $O(Nm)$ time. We have found that values of $m \sim \sqrt{N}$ can approximate k-NN results with over 0.95 precision. Running in $O(N\sqrt{N})$ time, this implementation achieves three decimal orders of magnitude speed-up over exact nearest neighbor search for 1M frames.

We set FLANN to return up to 50 likely neighbors for any query frame q, as the output of Figure 4 Box C. T_q is used to filter out the false neighbors and those that are too far to be declared a match. This filtering results in an incomplete set of matched pairs, depicted as the output of Figure 4 Box D. Therefore, we perform transitive closure on the neighbor relationship to find full equivalence classes of near-duplicate sets. This is done with an efficient set union-find algorithm [12] that runs in amortized time of $O(E)$, where E is the number of matched pairs.

5.3 Discussion

Our design choices for the visual meme detection system aim to find the best combination of accuracy and speed that is feasible to implement in a single PC. Note that using local features such as SIFT can arguably give more accurate near-duplicate detection results but such approaches require larger storage and memory for storing the features, and are less feasible for computing all pairs of near-duplicates in millions of frames on a single node. Most edits are also done at the granularity of entire frames, or even shots, and transformations such as picture-in-picture or significant crops do not not appear to be frequent in casual user remixes. We have found that shot-level matching by keyframes is suitable for capturing community video remixing and provides a good balance of speed and accuracy in public evaluations on video copy detection tasks [22]. Hashing-based techniques [27] are another alternative for speeding up ANN queries but their precision is typically not more than 50%, which is too low for reliably tracking all sets of near-duplicates in large collections. The ANN indexing scheme we adopt scales to several million video shots. On collections consisting of tens of millions to billions of video shots, we expect that the computation infrastructure will need to change—e.g., by implementing a massively distributed tree index [20] and/or hybrid tree-hashing techniques.

A few examples of identified near-duplicate sets are shown in Figures 1 and 2. The performance of the meme detection algorithm is evaluated in Section 7.1.

6. VISUAL MEME NETWORK

Visual memes can be viewed as *links* between videos and also between authors that share the same unit of visual expression. We therefore propose a network model linking visual memes and their authors. This enables us to quantify influence and the importance of visual memes in video-publishing information networks.

Denote a video (or any multimedia document) as d_i in event collection $\mathcal{D}$, with $i = 1, \ldots, N$. Each video is authored (i.e., uploaded) by author $a(d_i)$ at time $t(d_i)$, with $a(d_i)$ taking its value from a set of authors $\mathcal{A} = \{a_r, r = 1, \ldots, R\}$. Each document d_i can contain a set of memes, $\{v_1, v_2, \ldots, v_{K_i}\}$ from a meme dictionary $\mathcal{V}$. In this network model, each meme induces a time-sensitive edge e_{ij} with creation time $t(e_{ij})$, where i, j are over video documents.

6.1 Meme video graph

Let $G = \{\mathcal{D}, \mathcal{E}_G\}$ be a video graph with nodes $d \in \mathcal{D}$. There is a directed edge $e_{ij} \in \mathcal{E}_G$ if documents d_i and d_j share at least one visual meme and if d_i precedes d_j in time: $t(d_i) < t(d_j)$. The presence of e_{ij} represents a possibility that d_j was derived from d_i, even though there is no conclusive evidence within the video collection alone whether or not this is true. We denote the number of shared visual memes as $\nu_{ij} = | d_i \cap d_j |$, and the time elapsed between the posting time of the two videos as $\triangle t_{ji} = t(d_j) - t(d_i)$.

We use two recipes for computing the edge weight ω_{ij}. Equation 1 uses a weight proportional to the number of common memes ν_{ij}, and Equation 2 scales this weight by a power-law memory factor related to the time difference $\triangle t_{ji}$. The first model is insensitive to $\triangle t_{ji}$, so it can accommodate the resurgence of popular memes, as seen in textual memes [19]. The power law decay comes from known behaviors on YouTube [9], and it also agrees with our observations on the recency of the content returned by the YouTube search API.

$$\omega^*_{ij} = \quad \nu_{ij} \qquad (i,j) \in \mathcal{E}_G \tag{1}$$

$$\omega'_{ij} = \quad \nu_{ij} \triangle t_{ji}^{-\eta} \tag{2}$$

The unit for time t is in days. We estimate the exponent η to be 0.7654, using a process described in Section 7.4. Other edge-weighting factors incorporating the number of views or the rating scores could also be used, although our observations (Figure 6) suggest that the number of views is a poor indicator of the number of memes.

6.2 Meme author graph

Similarly, let us define an author graph $H = \{\mathcal{A}, \mathcal{E}_H\}$, with author nodes $a \in \mathcal{A}$. There is an undirected edge $e_{rs} \in \mathcal{E}_H$ between authors a_r and a_s if they share at least one visual meme in any of their videos.

We compute the edge weights θ_{rs} on edge e_{rs} as the aggregation of the weights on all the edges in the video graph G connecting documents authored by a_r and a_s.

$$\theta_{rs} \quad = \quad \Sigma_{\{i,a(d_i)=a_r\}} \Sigma_{\{j,a(d_j)=a_s\}} \omega_{ij} \tag{3}$$

with $r, s \in \mathcal{A},\ i, j \in \mathcal{D}$. We adopt two simplifying assumptions in this definition. The set of edges $\mathcal{E}_H$ are bidirectional, as authors often repost memes from each other at different times. The edge weights are cumulative over time, because in our datasets most authors post no more than a handful of videos (Figure 10), and there is rarely enough data to estimate instantaneous activities.

6.3 Meme influence indices

We define three indices based on meme graphs, which capture the influence on information diffusion among memes, and thereby quantify the impact of content and of authors within the video sharing network.

First, for each visual meme v, we extract from the event collection $\mathcal{D}$ the subcollection containing all videos that have at least one shot matching meme v, denoted as $\mathcal{D}_v = \{d_j \in \mathcal{D}, \text{s.t. } v \in d_j\}$. We extract the video document subgraph G_v corresponding to $\mathcal{D}_v$, setting all edge weights $\nu \in G_v$ to 1 since only a single meme is involved. We compute the in-degree and out-degree of every video d_i in $\mathcal{D}_v$ as the number of videos preceding and following d_i in time:

$$\zeta_{i,v}^{in} = \Sigma_j I\{d_i, d_j \in \mathcal{D}_v, \quad t(d_j) < t(d_i)\} \tag{4}$$
$$\zeta_{i,v}^{out} = \Sigma_j I\{d_i, d_j \in \mathcal{D}_v, \quad t(d_j) > t(d_i)\}$$

where $I\{\cdot\}$ is the indicator function that takes a value of 1 when its argument is true, and 0 otherwise. Intuitively, ζ_i^{in} is the number of videos with meme v that precede video d_i (potential sources), and ζ_i^{out} is the number of videos that succeed meme v after video d_i (potential followers).

The video influence index χ_i is defined for each video document d_i as the smoothed ratio of its out-degree over its in-degree, aggregated over all meme subgraphs G_v (Equation 5, where the smoothing factor 1 in the denominator accounts for d_i itself). The author influence index $\hat{\chi}_r$ is obtained by aggregating χ_i over all videos from author a_r (Equation 6). The normalized author influence index $\bar{\chi}_r$ is its un-normalized counterpart $\hat{\chi}_r$ divided by the number of videos an author posted, which can be interpreted as the *average* influence of all videos for this author.

$$\chi_i = \Sigma_v \frac{\zeta_{i,v}^{out}}{1 + \zeta_{i,v}^{in}} \tag{5}$$

$$\hat{\chi}_r = \Sigma_{\{i, a(d_i)=a_r\}} \chi_i, \tag{6}$$

$$\bar{\chi}_r = \frac{\hat{\chi}_r}{\Sigma_i I\{a(d_i) = a_r\}}$$

The above influence indices capture two aspects in meme diffusion: the total volume of memes, as well as how *early* a video or an author is in the diffusion chain. The first aspect is similar to the *reweet* and *mention* measures recently reported for Twitter [5]. The timing aspect in diffusion is new, and it is designed to capture different roles that users play on Youtube, such as information *connectors* and *mavens* [14]. The term *connectors* refers to people who come "... with a special gift for bringing the world together, ... [an] ability to span many different worlds", and *mavens* are "people we rely upon to connect us with new information, ... [those who start] word-of-mouth epidemics".

6.4 Predicting meme importance

We use visual meme network properties to model meme importance. Just as social media influence is commonly understood as multidimensional [1], importance can also be defined in a number of ways: the number of times that a video is viewed [25], the number of times that a video meme is reposted by other YouTube users, or by the lifespan (in days) of all known instances of a meme. Note that neither audience size or the number of views are reliable indicators of influence in social media networks, as shown in Section 7.2. Furthermore, our observations in Section 7.3 show that most memes are re-posted quickly, and our pilot experiments confirm that early meme volumes (on day one or two) are the best predictors of the final meme volume. Therefore, we focus on predicting the lifespan, i.e., the longevity of a message that is kept *alive* by information propagation.

Our meme importance model is derived from three types of features. Each type intends to capture the early trend of meme propagation and author productivity and connectivity, as well as the historical influence of authors. For each visual meme v that first appeared at time $t(v)$ (called *onset* time), we compute features on the meme- and author- subgraphs up to time $t_1 = t(v) + \Delta t$, by including video nodes that appeared before t_1. The parameter Δt is set to one day in this work in order to capture early meme dynamics, as observed in Section 7.3, and similarly to what has been used for view-count prediction [25].

These features we use are as follows:

- The volume of memes up to t_1.
- Static network features of author productivity and connectivity. We use the *total number of videos* that the author has uploaded to capture author productivity. An author's connectivity includes three metrics computed over the author graph of up to time t_1: *degree centrality* is the fraction of other nodes that a node is directly connected to; *closeness centrality* is the inverse of average path length to all other nodes; and *betweenness centrality* is the fraction of all-pairs shortest paths that pass through a node.
- Dynamic features of author diffusion influence. These include the meme influence indices $\hat{\chi}_r$ and $\bar{\chi}_r$ in Equation 6 as well as the aggregate in-degree and out-degree for each author.

To compute author network features, we aggregate the author features for each meme by taking the maximum, average, and standard deviation among the group of authors who have posted or reposted the meme by t_1. We use Support Vector Machines (SVM) [7] to predict meme importance using each of the volume, static, and dynamic network features above as well as their combination.

7. EXPERIMENTS AND OBSERVATIONS

In this section, we present our experimental setup and evaluations on several YouTube event collections. We begin by evaluating the performance of meme detection (Section 7.1), and present several observations on the generation of content buzz and memes in Sections 7.2–7.4. We then build and visualize visual meme graphs (Section 7.5) using the prior observations. We present observations on influence metrics for meme authors (Section 7.6), and these metrics are finally used to predict meme lifespan in Section 7.7.

Using the targeted-querying and collection procedures described in Section 4, we downloaded video entries from about two dozen topics from May 2009 to March 2010. We used four representative sets of large volume, diversity, and change over time, to capture characteristics of events spanning the two extremes of "event urgency": "acute" news stories (Iran election) and "chronic" news stories (swine flu). These datasets are summarized in Table 1. The SwineFlu set is about the H1N1 flu epidemic. The Iran3 set is about Iranian domestic politics and related international events during the 3-month period of summer 2009. The Iran1 set is a 1-month subset of Iran3 focusing on the election in mid-June and the associated political outbreaks. The Housing set is about the housing market collapse in the 2008-09 economic crisis—this hand-annotated set was used as a validation set for tuning the visual meme detection algorithm.

We perform visual meme detection as described in Section 5. We additionally filter the meme clusters identified by the detection system, by removing singletons belonging to a single video or a single author. Moreover, meme reposting analyses are based only on memes posted by at least 10 authors. The prototype system is implemented in C++, Python, and MATLAB, and it was deployed on a single quad-core system with 8GB of memory.

Topic	#Videos	#Authors	#Shots	Upload time
SwineFlu	31,488	10,804	1,202,479	04/09∼03/10
Iran3	23,049	4,681	1,255,062	08/07∼08/09
Iran1	5,429	2,393	210,259	09/07∼07/09
Housing	2,446	654	71,872	08/07∼08/09

Table 1: Summary of YouTube event data sets.

7.1 Meme detection performance

We evaluate the visual meme detection method in Section 5 using ground-truth created from the Housing dataset. Specifically, we run multiple versions of k-means clustering with a tight cluster radius threshold; we manually go through a sample of clusters to explicitly mark correct and incorrect near-duplicates. We further augment the detected near-duplicate sets by performing visual content-based queries on the color correlogram feature, and manually mark the top returns. We specifically include many borderline pairs that were confused by the clustering and feature-similarity retrieval steps. The resulting data set contains $\sim 15,000$ near-duplicate keyframe pairs and $\sim 25,000$ non-duplicate keyframe pairs.

We compute the near-duplicate equivalence classes as described in Section 5, and calculate precision (P) and recall (R) on the labeled pairs. The results are shown on Figure 5 for varying values of the threshold parameter τ. We note that the performance is generally quite high with $P > 95\%$. There are several possible operating points, such as $P = 99.7\%$, $R = 73.5\%$ for low false alarm; or $P = 98.2\%$, $R = 80.1\%$ that produces the maximum F1 score of 0.88 (defined as $\frac{2PR}{P+R}$); or $P = 96.6\%$, $R = 80.7\%$ for the highest recall. For the rest of our analysis, we use the last, high-recall, point with $\tau = 11.5$. On the Iran3 set of over 1 million shots, feature extraction takes around 7 hours on a quad-core CPU, and indexing and querying with FLANN takes 5 to 6 CPU hours.

7.2 Content views and re-posting probability

The behavior of remixing and reposting is quite dominant in the video collections we examined. Over 58% of the videos in Iran3, and 70% of the authors, contain visual memes. Likewise, 32% and 45%, respectively, for SwineFlu, as shown in Figure 6(a). These statistics suggest that, for popular topics there is much less original content than re-mixes and reprises of existing sources.

We observe that video popularity is a poor indicator of how likely a video is to be re-posted. In the Iran3 set, for example, the 4 most popular videos have no memes and have nothing to do with Iranian politics, and likewise for 7 of the first 10. One has to get beyond the first 1,600 most popular videos before the likelihood of having near-duplicates passes the average for the dataset, at about 0.58 (see Figure 6(b)). There are several reasons for this mismatch. Among the

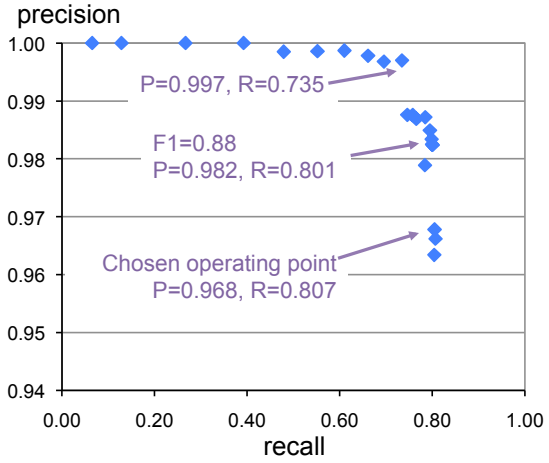

Figure 5: Performance of visual meme detection method on Housing Dataset.

video entries returned by the YouTube API using various sorting criteria[3] (Section 4), the most viewed are often not related to the query topic—for example, the one with the highest view-count is a popular music video irrelevant to Iranian politics. Moreover, view counts are highly influenced by a "rich-get-richer" effect, fueled by content recommendations, promotions, and over-zealous query expansion schemes, which tend to steer traffic towards popular clips, even if they are not topically relevant. In short, view count is a poor proxy for relevance or importance of video, and therefore is not a good predictor of overall influence.

This is an example of the very unequal distribution of views that characterizes this domain. To quantify the inequality of views-counts, we have computed the Gini coefficient [13] of this data set, and find it to have the extreme value of 0.94, both for videos with or without near-duplicates. The Gini coefficient, used in economics, ranges from 0 (each video with an equal number of views) to 1 (one video with all of the views). The value we observed far exceeds the measure of inequality for the distribution of wealth in any known country, which has its maximum at about 0.7.

7.3 Meme onset and reposting interval

We analyze the spread of visual memes by examining the interval between the onset of a meme and when it was reposted by a second user. Figure 7 contains the histogram and cumulative percentage of these interval distributions. The x-axis quantizes the reposting interval in hourly increments up to 12 hours, and then for one day, week, month, and year. The left y-axis shows the number of meme videos, and the right y-axis shows the cumulative percentage. We can see that over half of the memes are re-posted within 3 hours of initial upload, and over 70% within the same day.

7.4 Content freshness

We conduct an evaluation on the age of the returned videos from YouTube, i.e., content *freshness*. This can be used to determine the extend of influence on video remixing from past entries, and also to guide the parameter settings for meme diffusion network studies. We run the querying

[3] "...titles, keywords, descriptions, authors' usernames, and categories [are used for searching]", from http://code.google.com/apis/youtube/2.0/reference.html

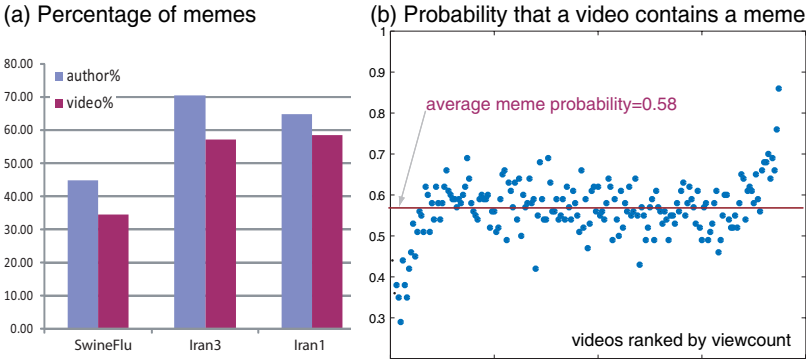

Figure 6: Video reposting probabilities. (a) Fraction of visual memes. (b) Video views vs. meme probability on Iran3 set.

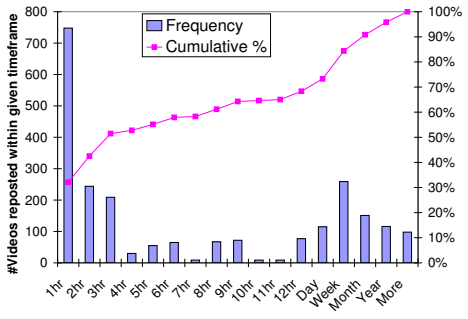

Figure 7: Distribution of meme reposting interval by a second author; see Section 7.3.

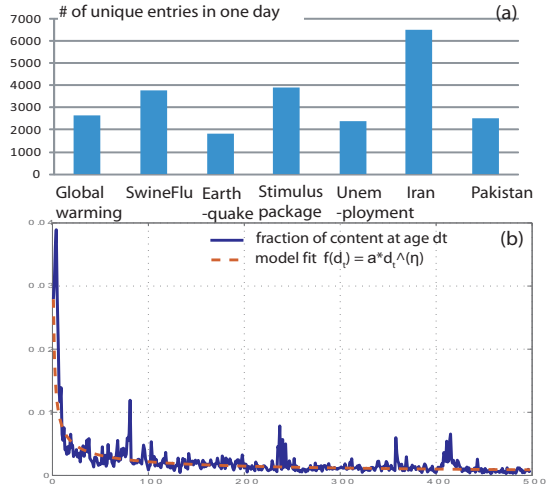

Figure 8: Content freshness experiment. (a) Number of unique entries returned for seven topics. (b) The fraction of unique entries over the age of the video (in days), averaged over the seven topics.

and content extraction during one single day, d_0='2010-04-04', across a set of seven diverse topics spanning environment, health, economics and international politics. Figure 8(a) shows the unique video entries returned for each topic, and Figure 8(b) shows the fraction of videos as a function of its age (the interval between its upload date and d_0, averaged over all seven topics). We note that the video volume is significant for any of these topics (from 1800 to 6500), and that the age distribution is approximately a power law, as observed in related studies [9]. We obtain a power-law regression fit for the content volume versus age: $f(t) = \alpha t^{\eta} \sim 0.0581 t^{-0.7654}$. Intuitively, the constant $\eta = -0.7654$ represents a YouTube "memory" factor that affects how much an early video tends to influence meme creation. It is used to scale graph weights in Equation (2).

7.5 Topic hub and content islands

Figure 9 shows an example of the video and author graphs generated over the Iran1 set. Note that each meme induces a diffusion tree in the video graph G, and a clique in the author graph H. The drawing shows all connected video and author nodes, but only the subset of the edges on the minimum spanning tree, in order to avoid cluttering the display. We can easily see that there is one densely connected topic community in each graph, with a number of smaller groups in the periphery. Note that the author graph tends to have fewer isolated components, since aggregating over videos from the same authors lessens the effect of fragmented meme clusters from imperfect detection.

Memes connect content and people that contribute to the same topic in an event. In this dataset, most videos in the central topic hub are shots of post-election street protest activities. Outliers, on the other hand, contain novel views and different perspectives. Meme #052834, for example, appeared in two videos from two different authors who did not post other meme-videos. It turns out that one of the videos (the other no longer available) is an election theme-song in Persian, played over the image of a chimpanzee—a mocking caricature of a known political figure.

We will demonstrate two uses of the meme network model: distinguishing different types of authors via the meme influence index, and predicting meme lifespan.

7.6 Influence index of meme authors

We compute the diffusion index for authors with Equation 6. Figure 10 contains scatter plots of the author influence indices on the y-axis, versus number of videos produced by each author on the x-axis. For both the Iran3 topic and the SwineFlu topic, we plot the total diffusion influence $\hat{\chi}_r$ and the normalized diffusion influence $\bar{\chi}_r$.

In the Iran3 topic we can see two distinct types of contributors. We call the first contributor type *mavens* [14] (marked in red), denoting users who post only a few videos but which tend to be massively remixed and reposted. This particular maven was among the first to post the murder of Neda Soltan, who became the icon of the entire event timeline, We call the second contributor type *connectors* [14] (circled in green), denoting users who tend to produce a large number of videos, and who have high total influence factor but have low average influence per video. They aggregate notable content and serve the role of bringing this content to a broader audience. (A response metric such as view count or number of comments could further confirm this fact.) We examined the YouTube channel pages for a few authors in this group, and they seem to be voluntary political activists with screennames like "iranlover100"—we can also dub them "citizen buzz leaders". Some of their videos are slide shows of iconic images and provide good summaries of the event

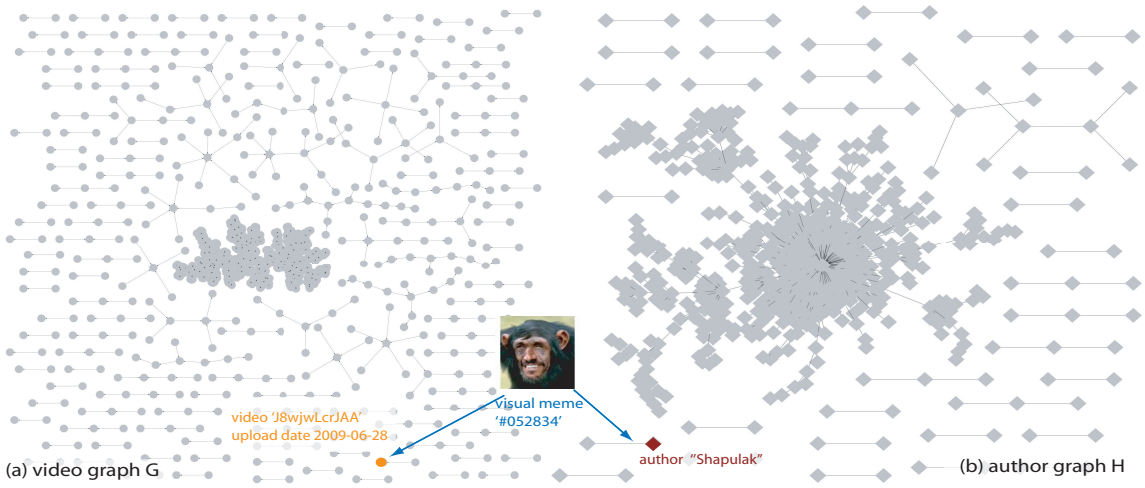

Figure 9: Graphs computed over the Iran1 data set showing (a) the video graph G, (b) the author graph H, and one example outlying meme pair.

timeline. Note that traditional news media, such as AljezeeraEnglish, AssociatedPress, and so on (circled in gray), have rather low influence metric for this topic, partially because the Iran government banned foreign journalists and severely limited international media coverage of the event.

The SwineFlu collection behaves differently in its influence index scatterplots. We can see a number of *connectors* on the upper right hand side of the total diffusion scatter. But it turns out that they are the traditional media (a few marked in gray), most of which have a large number (>40) of videos with memes. The few *mavens* in this topic (marked with green text) are less active than in the Iran topic, and notably they all reposted the identical old video containing government health propaganda for the previous outbreak of swine flu in 1976. These observations suggest that it is the traditional new media who seem to have driven most content on this topic, and, while serendipitous discovery of novel content sill exists, it has less diversity.

These visualizations can serve as a tool to observe different information dissemination patterns in different events, and henceforth characterize influential users. Such tools can identify the key influencers for each event, including both *mavens*, or early "information specialists", and *connectors*, who "bring the rest ... together" [14].

7.7 Meme lifespan prediction results

We predict the lifespan of memes as described in Section 6.4. We prune memes that appear less than 4 times, and use 2,296 memes in Iran1 for training, and 7,583 disjoint memes from Iran3 for testing. We construct the prediction task as a series of binary classification tasks (viral vs. non-viral) over progressive thresholds on meme lifespan (in days). This is because there is no clear-cut criteria about whether a meme's longevity is significant or not. Our pilot experiment also found that using regression directly on meme lifespan does not work well, yielding mean-square errors close to the average lifespan of memes. This is because most of the memes are non-viral, and predicting the lifespan values directly tends to fit the high-volume short-lifespan memes and to produce large errors on the low-volume interesting memes. As a result, we create multiple binary ground-truth by progressively thresholding the meme volume and lifespan between the 60^{th} to 90^{th} percentile on the training set.

The features we use include meme volume (one dimension), centrality measures for authors (three dimensions) and those aggregated over the videos (three dimensions), number of videos produced (one dimension), total and normalized meme influence indices (two dimensions), and the total in- and out- degrees for each author (two dimensions). All except the meme volume features are aggregated by taking the mean, maximum, median, and standard deviation over all authors of a meme. All types of features have a total of 45 dimensions. We train SVM classifiers [7] by searching over hyperparameters and different kernel types—linear, polynomial, and radial basis function. We measure the classification performance using the AUC [11] and plot the results in Figure 11 (top). We can see that author connectivity alone, or fusing all features together, are stronger features than content volume alone. Using all three types of features yields the highest AUC value of 0.779. This performance graph also shows that a suitable threshold of a long-lived meme is about 8 or 9 days (corresponding to the 78-th and 86-th percentile in Iran1), beyond which the performance of all prediction algorithms drop. We show a few top-ranked memes in the bottom half of Figure 11. We note that seven out of the top ten memes are from July 17, 2009, which coincides with the largest activity peak from Figure 3.

8. CONCLUSIONS

We proposed visual memes for tracking and monitoring of real-world events on YouTube, and described a large-scale event-based social video monitoring and analysis system. We proposed a scalable algorithm for extracting visual memes with high accuracy, and applied visual memes for estimating influence and predicting content popularity using network models. Using the proposed system, we have quantified the percentage of remixed content, the relationship between remix popularity and content views, and the timing of the remix. We have also shown that memes can help quantify the roles different users groups play in propagating information. A pilot evaluation showed that meme social graph features can help predict meme lifespan and volume.

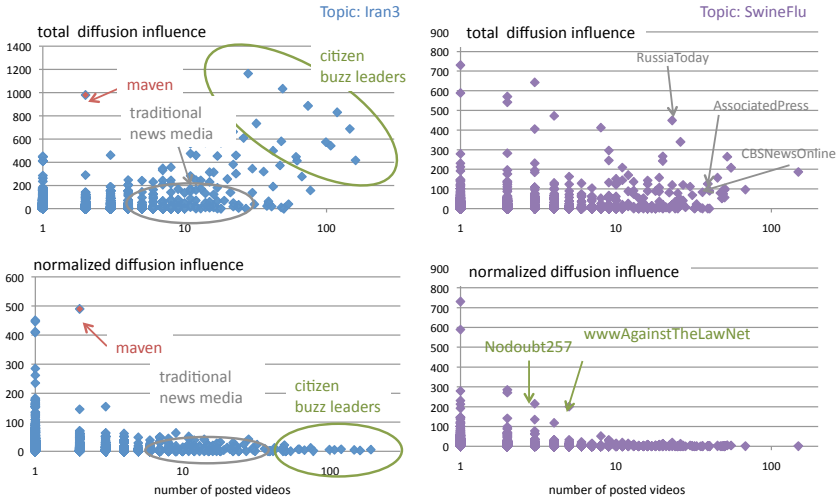

Figure 10: Meme influence indices vs author productivity on topic Iran3 (Left) and SwineFlu (Right); detailed discussions in Sec 7.6.

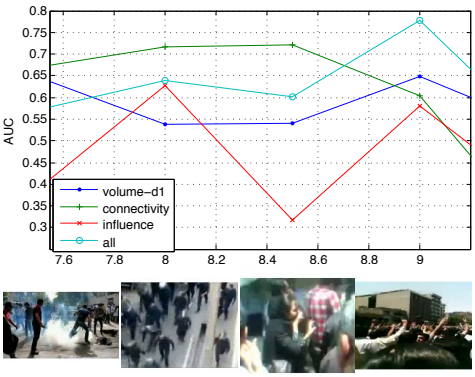

Figure 11: Predicting meme lifespan. Top: Prediction performance (AUC as the y-axis) over varying virality thresholds (as the x-axis, in days). Bottom: Icon example of top-ranked memes.

Future work includes annotating the meanings of visual memes over a course of event, extending the network and content analysis for memes, expanding the number of events and topics for evaluation, and broadening the applications of memes to video genres other than news.

9. REFERENCES

[1] What is the Klout score? Understanding the influence metric. http://klout.com/kscore , retrieved April 2011.

[2] Thanks, YouTube community, for two BIG gifts on our sixth birthday!, May 2011. The official YouTube blog, http://youtube-global.blogspot.com/2011/05/thanks-youtube-community-for-two-big.html .

[3] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and K. Ross. Video interactions in online video social networks. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 5(4):30, 2009.

[4] J.-I. Biel and D. Gatica-Perez. Voices of vlogging. In *AAAI Int. Conf. on Weblogs and Social Media (ICWSM)*, 5 2010.

[5] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th Intl. Conf. on Weblogs and Social Media (ICWSM)*, 2010.

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

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: A library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/˜cjlin/libsvm .

[8] M. Cherubini, R. de Oliveira, and N. Oliver. Understanding near-duplicate videos: A user-centric approach. In *Proc. ACM Intl. Conf. on Multimedia*, pages 35–44, 2009.

[9] R. Crane and D. Sornette. Viral, quality, and junk videos on YouTube: Separating content from noise in an information-rich environment. In *Proc. of AAAI symposium on Social Information Processing, Menlo Park, CA*, 2008.

[10] M. De Choudhury, H. Sundaram, A. John, and D. D. Seligmann. What makes conversations interesting?: Themes, participants and consequences of conversations in online social media. In *WWW*, pages 331–340, 2009.

[11] T. Fawcett. ROC graphs: Notes and practical considerations for researchers. *Machine Learning*, 31:1–38, 2004.

[12] B. A. Galler and M. J. Fisher. An improved equivalence algorithm. *Communications of ACM*, 7(5):301–303, 1964.

[13] C. Gini. Measurement of inequality of ncomes. *Economic Journal*, 31:124–126, 1921.

[14] M. Gladwell. *The tipping point: How little things can make a big difference*. Little, Brown and Co., 2000.

[15] R. Hong, J. Tang, H.-K. Tan, S. Yan, C.-W. Ngo, and T.-S. Chua. Beyond search: Event driven summarization for web videos. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2011.

[16] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3), December 1999.

[17] L. Kennedy and S.-F. Chang. Internet image archaeology: Automatically tracing the manipulation history of photographs on the web. In *Proc. ACM Multimedia*, 2008.

[18] H. Kwak, C. Lee, H. Park, , and S. Moon. What is Twitter, a Social Network or a News Media? . In *Proc. WWW*, 2010.

[19] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proc. KDD*, 2009.

[20] T. Liu, C. Rosenberg, and H. A. Rowley. Clustering billions of images with large scale nearest neighbor search. In *IEEE Workshop on Applications of Computer Vision*, 2007.

[21] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Intl. Conf. on Computer Vision Theory and Applications*, 2009.

[22] A. Natsev, M. Hill, and J. Smith. Design and evaluation of an effective and efficient video copy detection system. In *IEEE Intl. Conf. on Multimedia and Expo (ICME)*, 2010.

[23] P. Schmitz, P. Shafton, R. Shaw, S. Tripodi, B. Williams, and J. Yang. International remix: Video editing for the web. In *Proc. ACM Multimedia*, page 798. ACM, 2006.

[24] P. Snickars and P. Vonderau. *The YouTube Reader*. National Library of Sweden, 2010.

[25] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, August 2010.

[26] H.-K. Tan, X. Wu, C.-W. Ngo, and W.-L. Zhao. Accelerating near-duplicate video matching by combining visual similarity and alignment distortion. In *Proc. ACM Multimedia*, page 861, 2008.

[27] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *IEEE CVPR*, San Francisco, USA, June 2010.