Visualizing Timelines: Evolutionary Summarization via Iterative Reinforcement between Text and Image Streams

Rui Yan [‡,†]
[†]Dept. of Computer Science and Information Engineering
National Taiwan University
Taipei 10617, Taiwan
r.yan@pku.edu.cn

Xiaojun Wan
Institute of Computer Science and Technology
Peking University
Beijing 100871, P. R. China
wanxiaojun@pku.edu.cn

Mirella Lapata
Institute for Language, Cognition and Computation
University of Edinburgh
Edinburgh EH8 9AB, UK
mlap@inf.ed.ac.uk

Wayne Xin Zhao
[‡]Dept. of Computer Science and Technology
Peking University
Beijing 100871, P. R. China
zhaoxin@net.pku.edu.cn

Pu-Jen Cheng
Dept. of Computer Science and Information Engineering
National Taiwan University
Taipei 10617, Taiwan
pjcheng@csie.ntu.edu.tw

Xiaoming Li [♮,‡]
[♮]State Key Lab of Virtual Reality Tech. and Systems
Beihang University
Beijing 100083, P. R. China
lxm@pku.edu.cn

ABSTRACT

We present a novel graph-based framework for timeline summarization, the task of creating different summaries for different timestamps but for the same topic. Our work extends timeline summarization to a multimodal setting and creates timelines that are both textual and visual. Our approach exploits the fact that news documents are often accompanied by pictures and the two share some common content. Our model optimizes local summary creation and global timeline generation jointly following an iterative approach based on mutual reinforcement and co-ranking. In our algorithm, individual summaries are generated by taking into account the mutual dependencies between sentences and images, and are iteratively refined by considering how they contribute to the global timeline and its coherence. Experiments on real-world datasets show that the timelines produced by our model outperform several competitive baselines both in terms of ROUGE and when assessed by human evaluators.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*; I.4.9 [**Image Processing and Computer Vision**]: Application

General Terms

Algorithms, Experimentation, Performance

Keywords

Evolutionary summarization, iterative reinforcement, visual timeline, text-to-image translation

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

1. INTRODUCTION

With the rapidly growing amounts of information now being available over the Internet, it is becoming increasingly important to develop improved computer-based methods for document access, filtering, and content extraction. Document summarization is one of the oldest applications of information gathering. A summary can be loosely defined as a text that is derived from one or more original texts, conveys the most important information in them, and is substantially shorter in length.

Of the many types of summaries that have been identified over the years (see [20] and [16] for comprehensive overviews), perhaps the most important distinctions concern whether the summary is being created over a *single* or *multiple* documents, whether it is an *extract* (portions of the original text appear in the summary verbatim) or an *abstract* (the extracted content is reformulated in novel terms), and whether it is *indicative* (it merely provides an indication of the subject matter of the input text) or *informative* (it is a shortened version of the content of the original one).

In this paper we address a relatively new summarization task, namely *timeline summarization*. The idea behind timeline summarization is to capture how a news topic evolves over time. Timelines consist of several component summaries that are linked to different timestamps. Each component summary provides a *local view* on the news story for a specific point in time, whereas the timeline provides a *global view* on the development of the story across time. Timeline summarization is related to but different from update summarization [25]. Update summaries focus on what is new relative to a previous body of information, whereas timelines involve not a single but several steps of updates. Both tasks produce multi-document, typically extractive, summaries for a set of articles on the same topic.

Our work extends timeline summarization to a multimodal setting. Most online news articles contain images whose role is to complement or emphasize what is described in the text. An example is shown in Table 1; the article describes a devastating earthquake in Japan and the picture accentuates the extent of damage it caused. We argue that incorporating visual information into timelines is desirable for several reasons. Firstly, images will supplement the textual summaries by providing additional information. Secondly, visual timelines can be seen as an extreme form of sum-

marizing the content of very large document collections. Users need not read the textual summaries; instead, they can only look at the images to get an impression of how a story developed over time. Thirdly, by exploiting the natural dependence between articles and their images during modeling, it is possible to improve the output of the textual summaries as the images provide cues about important content.

Our approach exploits the fact that news documents are often accompanied by pictures and the two share some common content. Our summarization model formalizes the following intuitions. Component summaries must capture *local* importance in relation to specific timestamps (e.g., corresponding to specific dates). They consist of two heterogeneous streams, i.e., images and text which are correlated and thematically matched. Component summaries must combine to create a *globally* coherent timeline across timestamps. Our model optimizes local summary creation and global timeline generation jointly following an iterative approach based on mutual reinforcement and co-ranking (see Figure 1 for an illustration). Our algorithm operates over sentences and images whose dependencies are captured in a heterogeneous network consisting of three types of graphs. One graph represents how individual sentences (and analogously images) relate to each other, a bipartite graph represents how sentences relate to images, and a third graph captures global dependencies between local sentences (and images) and the timeline created at each time step. The main idea behind co-ranking is that there is a mutually reinforcing relationship between sentences and images which influences their ranking and whether they should be included in the timeline or not.

An important component of the framework sketched above is to be able to express and quantify the meaning shared between images and the documents that contain them. With the help of image annotation techniques [6, 7, 8], we create a translation model that bridges the gap between textual and visual information. We preprocess images so that they resemble word-like units and define a probabilistic model that translates a visual word into a textual word and vice versa.

Experiments on real-world datasets show that the timelines generated by our model outperform several competitive baselines both in terms of ROUGE and when assessed by humans. Interestingly, our results also indicate that the model selects images and sentences that are thematically related and that the visual information helps create better textual summaries in addition to improving the timelines altogether. The remainder of this paper is structured as follows. In Section 2 we present an overview of previous work and then move on to formalize our summarization model (Section 3). We describe our experimental setup in Section 4 and present our results in Section 5. Section 6 concludes the paper.

A massive earthquake has hit off north-east of Japan today, triggering a tsunami that has caused extensive damage. Japan's TV showed cars, ships and even buildings being swept away in the Fukushima prefecture, after the 8.9 magnitude earthquake.

Officials said a wave as high as 6 m (20 ft) could strike the coast. The quake struck about 250 miles (400km) from Tokyo at a depth of 20 miles, shaking buildings in the capital for several minutes.

Table 1: An excerpt of a news report from the BBC website and its accompanying image.

2. RELATED WORK

Most work to date focuses on extractive summarization. The idea is to create a summary automatically simply by identifying and subsequently concatenating the most important sentences in a document. The approach is robust (it can be easily ported to different languages and domains) and produces grammatical output. One of the most popular extractive methods that have been proposed for multi-document summarization is centroid-based. It operates on a document collection with a common subject. A cluster centroid is built representing the most important words from the whole collection. The centroid is then used to determine which sentences from the individual documents are most representative of the collection. MEAD [19] is a publicly available implementation of this approach. NeATS [11] adds topic signatures and term clustering to sentence selection. Both MEAD and NeATS use MMR [9] to remove redundancy. Information based on themes in documents has been also used for sentence selection [22, 26].

More recently, graph-based methods have been proposed to rank sentences based on "votes" or "recommendations" between them. TextRank [17] and LexPageRank [5] use algorithms similar to PageRank and HITS to compute sentence importance. These methods first construct a graph representing the relationships between sentences and then evaluate their importance or salience based on the topology of the graph. Wan et al. [23, 22] present an improved graph-ranking algorithm which differentiates *intra*- and *inter*-document linkage between sentences [24] and incorporates topic cluster information in manifold-ranking.

A few methods have been developed specifically for timeline summarization For example, Swan et al. [21] construct timelines by extracting clusters of noun phrases and named entities. Later they build a system to provide timelines which consist of one sentence per date, based on their usefulness and novelty. Chieu et al. [3] present a system that extracts events (i.e., sentences) relevant to a query from a collection of documents and places such events along a timeline. Events are considered important if they are widely cited in many documents for a period of time. Yan et al. [27, 28] improve timeline summarization by modeling the evolutionary nature of news articles. Specifically, they model correlations among component summaries using inter-date and intra-date sentence dependencies.

To the best of our knowledge, the use of visual information in timeline summarization is unexplored in previous work. We propose a novel framework for this task which is based on iterative reinforcement: text-to-image correlations and local-to-global dependencies are taken into account simultaneously in order to create component summaries which are locally informative and globally coherent.

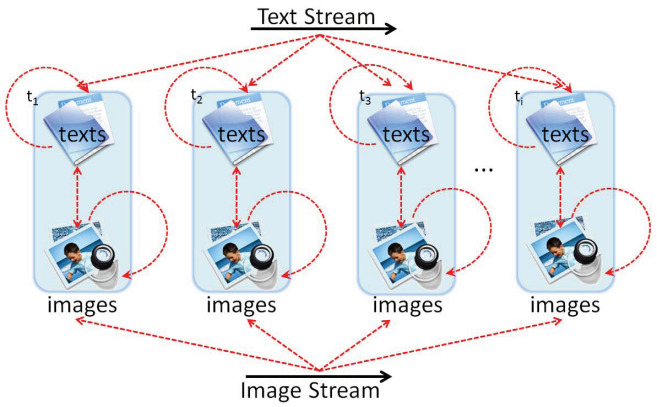


Figure 1: Iterative reinforcement for VTS. Shaded areas denote texts and images published on the same date; arrows between them indicate mutual reinforcement.

3. PROBLEM FORMULATION

The input to our visual timeline summarizer consists of a document collection C relating to a specific news subject (e.g., an earthquake or a disease). C is partitioned into sentences S and images M, i.e., C=$\{S, M\}$. Each sentence $s \in S$ and each image $m \in M$ is associated with a timestamp, i.e., the published date of the source document where the sentence and image appeared. The sentence collection S is further partitioned into S=$\{S_1, S_2, \ldots, S_{|T|}\}$, according to timestamps T=$\{t_1, t_2, \ldots, t_{|T|}\}$. Analogously, the image collection M is partitioned into M=$\{M_1, M_2, \ldots, M_{|T|}\}$. Let C_i=$\{S_i, M_i\}$ denote the collection consisting of sentence set S_i and image set M_i published on the same date t_i.[1] The output of our system is a series of individual but correlated component summaries I=$\{I_1, I_2, \ldots, I_{|T|}\}$. Each component I_i=$\{I_{s_i}, I_{m_i}\}$ on t_i is a subset of C_i, ($I_i \subseteq C_i$) with I_{s_i} being extracted from S_i ($I_{s_i} \subseteq S_i$) and I_{m_i} from M_i ($I_{m_i} \subseteq M_i$), respectively.

We conceptualize VTS in terms of three components, namely sentence selection, image selection, and sentence-to-image matching. Rather than optimizing each component in isolation, we introduce a global framework that performs the optimization task jointly, and thus exploits inter- and intra-component dependencies. For example, ranking individual sentences depends on image selection, and the selection of images should also relate to the sentence ranking.

We propose an iterative reinforcement framework for *co-ranking* sentences and images. The framework is illustrated in Figure 1 and formalized in Section 3.1. As can be seen, sentences and images (and their ranking) are coupled together and dependent on each other.

3.1 Iterative Reinforcement Framework

The mutual reinforcement chain shown in Figure 1 captures the following intuitions behind the VTS problem. A *local sentence* is important if (1) it associates to other important local sentences; (2) it associates to important *local images*; and (3) associates to other important sentences selected for neighboring timestamps (we refer to these as *global neighbors* of sentences). Analogously, a *local image* is important if (1) it associates to other important local images; (2) it associates to important *local sentences*; and (3) associates to important images selected for neighboring timestamps (called *global neighbors* of images).

[1]We use day-based timestamps [1, 3, 28, 27].

For each timestamp $t \in T$ we aim to find the most important or salient local sentences I_{s_t} and the most important local images I_{m_t}, so that they semantically related. Selected sentences should explain the images and selected images ought to depict some of the sentences' content. We derive the ranking of local sentences and local images *iteratively* from the mutual reinforcement chain across different timestamps. To simplify notation, we remove the subscript t from all local component choices when there is no ambiguity. We use two vectors $\mathbf{s}$=$[\pi(s_i)]_{1\times|\mathbf{s}|}$ and $\mathbf{m}$=$[\pi(m_i)]_{1\times|\mathbf{m}|}$ to denote the saliency scores $\pi(.)$ of local sentences and local images from timestamp t. We use vectors $\mathbf{u}$=$[\phi(s_i)]_{1\times|I_s|}$ and $\mathbf{v}$=$[\phi(m_i)]_{1\times|I_m|}$ to denote the candidate sentences and images in each iteration; the discrimination function $\phi(.)$ records the saliency scores of candidate sentences (and images) that are *selected* for timelines (see Equation (4)).

We use an adjacency matrix $[\hat{U}]_{|\mathbf{s}|\times|\mathbf{s}|}$ to represent the *homogeneous affinity* between local sentences, and matrix $[\bar{U}]_{|\mathbf{m}|\times|\mathbf{m}|}$ to describe the affinity between local images.

$$\mathbf{s} \propto \hat{U}^T\mathbf{s}, \qquad \mathbf{m} \propto \bar{U}^T\mathbf{m} \tag{1}$$

We use adjacency matrix $[\hat{W}]_{|\mathbf{s}|\times|\mathbf{m}|}$ to capture the local *heterogeneous affinity* between sentences and images; analogously matrix $[\bar{W}]_{|\mathbf{m}|\times|\mathbf{s}|}$ represents the heterogeneous affinity between images and sentences.

$$\mathbf{s} \propto \hat{W}^T\mathbf{m}, \qquad \mathbf{m} \propto \bar{W}^T\mathbf{s} \tag{2}$$

We use two matrices to represent neighboring sentences and images that are globally salient. Let $[\hat{N}]_{|\mathbf{s}|\times|I_s|}$ denote global neighboring sentences and $[\bar{N}]_{|\mathbf{m}|\times|I_m|}$ global neighboring images.

$$\mathbf{s} \propto \hat{N}^T\mathbf{u}, \qquad \mathbf{m} \propto \bar{N}^T\mathbf{v} \tag{3}$$

Elements u_i=$\phi(s_i)$ and v_j=$\phi(m_i)$ indicate saliency scores retained in vectors $\mathbf{u}$, $\mathbf{v}$; this is the case for sentences or images that are selected as summary candidates:

$$u_i = \begin{cases} \pi(s_i) & \text{if } s_i \in I_s \\ 0 & \text{otherwise} \end{cases}, \quad v_j = \begin{cases} \pi(m_j) & \text{if } m_j \in I_m \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We are now ready to formulate our iterative procedure for text-to-image co-ranking. The following two steps are used to select sentences and their corresponding images until convergence. We initialize the algorithm by setting the the saliency scores for all sentences and images to 1:

Step 1: compute the saliency scores of local sentences, and then normalize using ℓ-1 norm.

$$\begin{aligned} \mathbf{s}^{(n+1,k)} &= \alpha\hat{U}^T\mathbf{s}^{(n,k)} + \beta\hat{W}^T\mathbf{m}^{(n,k)} + \gamma\hat{N}^T\mathbf{u}^{(0,k-1)} \\ \mathbf{s}^{(n,k)} &= \mathbf{s}^{(n,k)}/||\mathbf{s}^{(n,k)}||_1 \end{aligned} \tag{5}$$

Step 2: compute the saliency scores of local images, and then normalize in ℓ-1 norm.

$$\begin{aligned} \mathbf{m}^{(n+1,k)} &= \alpha\bar{U}^T\mathbf{m}^{(n,k)} + \beta\bar{W}^T\mathbf{s}^{(n,k)} + \gamma\bar{N}^T\mathbf{v}^{(0,k-1)} \\ \mathbf{m}^{(n,k)} &= \mathbf{m}^{(n,k)}/||\mathbf{m}^{(n,k)}||_1 \end{aligned} \tag{6}$$

Parameters α, β and γ specify the relative contributions to the final saliency scores from 1) the affinity between homogeneous local sentences (and analogously images), 2) the affinity between heterogeneous local sentences and images, and 3) the affinity from homogeneous global salient nodes. For simplicity, we let α+β+γ=1. In

Algorithm 1 Local Text-Image Reinforced Co-Ranking

procedure L-CoRank(**u**,**v**)
  calculate $\hat{U}$, $\bar{U}$, $\hat{W}$, $\bar{W}$, $\hat{N}$, $\bar{N}$
  n←0, initialize **s**, **m**
  repeat
    $\mathbf{s}' \leftarrow \mathbf{s}$, $\mathbf{m}' \leftarrow \mathbf{m}$
    $\mathbf{s} \leftarrow \alpha\hat{U}^T\mathbf{s}' + \beta\hat{W}^T\mathbf{m}' + \gamma\hat{N}^T\mathbf{u}$
    $\mathbf{m} \leftarrow \alpha\bar{U}^T\mathbf{m}' + \beta\bar{W}^T\mathbf{s}' + \gamma\bar{N}^T\mathbf{v}$
    s ←**s**/||**s**||, **m** ←**m**/||**m**||

$$\nabla_L \leftarrow \max\begin{pmatrix} ||\mathbf{s}-\mathbf{s}'|| \\ ||\mathbf{m}-\mathbf{m}'|| \end{pmatrix}$$

    n← n+1
  until $\nabla_L < \epsilon$
  choose top ranked sentences/images by **s** and **m**
  // update the saliency scores of local sentences/images
  $\mathbf{u}' \leftarrow$ UPDATE(**u**)
  $\mathbf{v}' \leftarrow$ UPDATE(**v**)
  return $\mathbf{u}'$,$\mathbf{v}'$
end procedure

function UPDATE(**x**)
  for all $x_i \in$ **x do**
    if $x_i \in I$ **then**
      // highly-ranked ones are chosen as candidates
      $x_i \leftarrow x_i$
    else
      $x_i \leftarrow 0$
    end if
  end for
end function

Algorithm 2 Visual Timeline Summarization

k←0, initialize **u**, **v**
repeat
  $\mathbf{u}' \leftarrow \mathbf{u}$, $\mathbf{v}' \leftarrow \mathbf{v}$
  for $t \leftarrow 1$ to |T| **do**
    (u, v) ← L-CoRank($\mathbf{u}'$,$\mathbf{v}'$)
  end for
  u ←**u**/||**u**||, **v** ←**v**/||**v**||

$$\nabla_G \leftarrow \max\begin{pmatrix} ||\mathbf{u}-\mathbf{u}'|| \\ ||\mathbf{v}-\mathbf{v}'|| \end{pmatrix}$$

  k← k+1
until $\nabla_G < \epsilon$
OUTPUT(**u**,**v**) *// output non-zero elements as summaries*

function OUTPUT(**x**)
  for all $x_i \in$ **x do**
    if $x_i > 0$ **then**
      select x_i into timeline
    end if
  end for
end function

order to guarantee the convergence of the iterative form, we must force the transition matrix to be stochastic and irreducible. To this end, we must make the **s** and **m** *column stochastic* [10]. We therefore normalize **s** and **m** after each iteration in Equations (5) and (6).

When all component summaries are generated, **u** and **v** are updated and the algorithm returns to Step 1 and the same procedure is repeated until convergence. Note that n and k are different iteration indicators. n controls the iteration towards convergence among local sentences/images (see Algorithm 1). k controls the iteration towards global convergence at the timeline level, shown in Algorithm 2. Empirically, the algorithm converges when the difference between the scores computed at two successive iterations for any sentences/images falls below a small threshold ϵ (set to 0.001 in this study).

The framework just described critically explores the relationship between elements within the same modality and across modalities. In order to capture which images correspond to which summaries and vice versa we need some means of translating between visual and textual information. In the following, we explain how we model text-to-image correspondences and then move on to describe the estimation of our affinity matrices.

3.2 Text-to-Image Translation

Texts and images represent distinct modalities. Images live in a continuous feature space, whereas words are discrete. We follow previous work [6, 2, 4] in converting the visual features from a continuous onto a discrete space, thereby rendering image features more like word units. In order to do this, we use the Scale Invariant Feature Transform (SIFT) algorithm [13]. The general idea behind the algorithm is to identify local image regions using a difference-of-Gaussians point detector. Importantly, this detector is, to some extent, invariant to to small shifts in position, changes in illumination, noise, and viewpoint and can be used to perform reliable matching between different views of an object or scene. Each detected region is represented with a SIFT descriptor which is a histogram of edge directions at different locations. We further quantize the SIFT descriptors using the K-means clustering algorithm to obtain a discrete set of visual words which form our visual vocabulary. Each entry in this vocabulary stands for a group of image regions which are similar in content or appearance and assumed to originate from similar objects. Formally, each image is expressed in a bag-of-visual-words format vector, $[w_{v_1}, w_{v_2}, ..., w_{v_L}]$.

Both visual and textual modalities are represented as a bag-of-units, i.e., a vector of textual or visual terms. Given this representation, we can then define a translation model between the two modalities under the assumption that they express related concepts. Our model defines the probability of translating a textual word into a visual word and vice versa. We learn image-to-text correspondences from training data consisting of documents and the images embedded in them. In this work, we make use of news articles which are often accompanied with images illustrating events, objects or people mentioned in the text.

Let images m denote a set of visual words m=$\{w_v\}$, and texts s denote words w. Let Ψ=$\{m, s\}$ denote a training pair consisting of a document and its corresponding image. The translation model can be estimated by maximizing the likelihood of images given their surrounding texts:

$$w_v^* = \underset{w_v}{\text{argmax}} \prod_{\{m,s\}\in\Psi} \Pr(m|s,w) \quad (7)$$

Although we could use EM to estimate Equation (7), we approximate it heuristically with a simpler form which is considerably more efficient to compute [14]:

Gaussian kernel

$$\Gamma(\Delta t) = \exp[\frac{-\Delta t^2}{2\sigma^2}]$$

Triangle kernel

$$\Gamma(\Delta t) = \begin{cases} 1 - \frac{\Delta t}{\sigma} & \text{if } \Delta t \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$

Window kernel

$$\Gamma(\Delta t) = \begin{cases} 1 & \text{if } \Delta t \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$

Circle kernel

$$\Gamma(\Delta t) = \begin{cases} \sqrt{1 - (\frac{\Delta t}{\sigma})^2} & \text{if } \Delta t \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$

Cosine (Hamming) kernel

$$\Gamma(\Delta t) = \begin{cases} \frac{1}{2}[1 + \cos(\frac{\Delta t \cdot \pi}{\sigma})] & \text{if } \Delta t \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$

Table 2: Kernel functions for temporal biased projection.

$$\begin{aligned} w_v^* \approx & \underset{w_v}{\text{argmax}} \Pr(w_v|w) \\ \approx & \underset{w_v}{\text{argmax}} \frac{\#(w_v, w)}{\#w} \end{aligned} \tag{8}$$

where $\#(w_v, w)$ is the number of times w_v and w co-occur in Ψ, and $\#w$ is the total number of times w occurs in the training set.

The model in Equation (8) allows us to translate a visual word into a textual word, which we denote as $w_v = f(w)$. Analogously, we use $w = f^{-1}(w_v)$ to denote the translation of a textual word into a visual word. We can also translate an image m containing a set of visual words into a set of textual words ($f^{-1}(m)$) as well as a sentence into visual words ($f(s)$).

3.3 Affinity Matrices

In this section we explain how our affinity matrices are calculated. Recall from Section 3.1 that our co-ranking framework makes use of a local homogeneous affinity matrix ($\hat{U}, \bar{U}$), a local heterogeneous affinity matrix ($\hat{W}, \bar{W}$), and a global homogeneous affinity matrix ($\hat{N}, \bar{N}$).

3.3.1 Homogeneous Affinity Matrix

The local sentence collection can be modeled as a weighted undirected graph. Nodes in the graph represent sentences, edges represent intra-sentential relatedness, and their weights are computed via cosine similarity. The adjacency matrix U describes such a graph with each entry corresponding to the weight of an edge. Analogously, we calculate the cosine similarity between images represented by visual words. The adjacency matrix for sentences $U^s = [U^s_{ij}]_{|\vec{s}|\times|\vec{s}|}$ and images $U^m = [U^m_{ij}]_{|\vec{m}|\times|\vec{m}|}$ are defined as follows:

$$U^s_{ij} = \frac{\mathbf{s_i} \cdot \mathbf{s_j}}{||\mathbf{s_i}||||\mathbf{s_j}||}, \qquad U^m_{ij} = \frac{\mathbf{m_i} \cdot \mathbf{m_j}}{||\mathbf{m_i}||||\mathbf{m_j}||} \tag{9}$$

where **s** and **m** represent vectors of textual and visual words, respectively. Vector components are set to their *tf.idf* values [18]. *tf* is the term frequency (visual or textual) and *idf* is the inverse

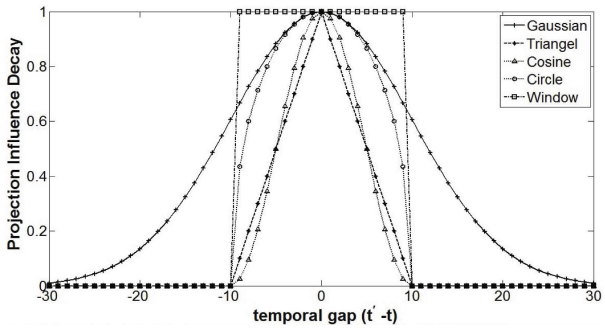


Figure 2: Proximity-based kernel functions, where σ=10.

sentence frequency in the case of textual words and inverse image frequency in the case of visual words.

Both matrices are normalized to make the sum of each row equal to 1, i.e., U^s is normalized into $\hat{U}$ and U^m into $\bar{U}$:

$$U_{ij} = \begin{cases} \frac{U_{ij}}{\sum_j U_{ij}}, & \text{if} \sum_j U_{ij} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

3.3.2 Heterogeneous Affinity Matrix

We capture the dependencies between sentences S and and images M=$\{m \in M\}$, in a bipartite graph. Graph edges are weighted according to semantic relatedness:

$$W_{ij} = \frac{1}{2}\left(\frac{\mathbf{s_i} \cdot f^{-1}(\mathbf{m_j})}{||\mathbf{s_i}||||f^{-1}(\mathbf{m_j})||} + \frac{\mathbf{m_j} \cdot f(\mathbf{s_i})}{||\mathbf{m_j}||||f(\mathbf{s_i})||}\right) \tag{11}$$

To ensure that images and texts contribute equally to the definition of the matrix, we take the average similarity of the bidirectional translation; again we calculate similarity using the cosine measure. Vector components representing sentences and images are set to their *tf.idf* values as mentioned above. W is normalized to $\hat{W}$ to make the sum of each row equal to 1. In addition, we normalize the transpose of W, i.e., W^T, to $\bar{W}$ to make the sum of each row in W^T equal to 1.

3.3.3 Neighboring Affinity Matrix

We represent the relationship between local sentences (or images) and the summary candidates $\vec{u}$ (or $\vec{v}$) as a weighted graph:

$$\begin{aligned} N^s_{ij} &= \frac{\mathbf{s_i} \cdot \mathbf{u_j}}{||\mathbf{s_i}||||\mathbf{u_j}||} \cdot \Gamma(|t(\mathbf{s_i}) - t(\mathbf{u_j})|), \\ N^m_{ij} &= \frac{\mathbf{m_i} \cdot \mathbf{v_j}}{||\mathbf{m_i}||||\mathbf{v_j}||} \cdot \Gamma(|t(\mathbf{m_i}) - t(\mathbf{v_j})|) \end{aligned} \tag{12}$$

Function $t(.)$ denotes the timestamp of the sentence or image. The adjacency matrix N^s describes the affinity between local sentences and candidate timeline sentences. Each entry of the matrix corresponds to the linkage weight and is normalized to $\hat{N}$ so that the sum of each row equals 1. Similarly, the image matrix N^m is normalized to $\bar{N}$.

Function $\Gamma(\Delta t)$ is a temporal decay function denoting the time distance between two component summaries. Intuitively, component summaries with a temporal gap will have less influence on each other [28, 27]. Following [15], we experiment with five representative kernel functions expressing temporal bias: Gaussian, Triangle, Cosine, Circle, and Window. The functions are defined in Table 2 and illustrated in Figure 2. Different kernels lead to different projections. For instance, the Window kernel treats neighboring component summaries equally within a certain scope while the Gaussian kernel models a gradient decaying relationship.

All kernels are parameterized by σ which controls the spread of the kernel curve, i.e., it restricts the projection scope of each sentence and image. In general, the optimal setting for σ may vary across datasets. For example, some news subjects may evolve quickly and require a small value of σ, whereas others may evolve slowly and require a higher σ value.

4. EXPERIMENTAL SETUP

4.1 Datasets

Our experiments used the data provided in [28], a collection of articles gathered from British (e.g., BBC), American (e.g., CNN, ABC, Reuters), and Chinese (e.g., Xinhua) news sources. We created four datasets, each covering a different news topic (i.e., science, disasters, accidents, and politics). Table 3 provides basic statistics for each dataset. Aside from documents and images, the datasets are accompanied with timeline summaries (text-based, or image-based, or both) written by professional editors. We treat these as gold standard and use them for the system evaluation.

News Subjects	#Doc	#Sent	#Img	#T	#RT
Influenza A	2557	115026	952	331	11
BP Oil Spill	1468	63021	337	135	9
Haiti Quake	247	12073	175	83	5
Obama Presidency	2160	79761	813	349	16

Table 3: Datasets used in our experiments (#Doc: number of documents, #Sent: number of sentences, #Img: number of images, #T: number of timestamps, #RT: number of reference timelines).

4.2 Model Training

In order to train our model and estimate the affinity matrices from Section 3.3, we decomposed news articles into sentences. We also removed stop-words and performed stemming. We trained a translation model based on pairs of images and their surrounding text found in our datasets. Each pair comes from the same web document. We generate a component summary for each timestamp according to a user specified *compression rate*. All component summaries constitute the global timeline.

Source documents bear temporal tags, i.e., the time of their publication. Sentences inherit temporal tags from their documents, as well as images. The sentence and image collections S and M are further partitioned according to their timestamps (e.g., $S = S_1 \cup S_2 \cup \cdots \cup S_{|T|}$). As mentioned previously, I_{s_i} is generated from sub-collection S_i, and I_{m_i} from M_i. The sizes of component summaries are not necessarily equal. Users specify the overall compression rate ϕ, and we extract more content (either texts or images) for important dates and less for other dates. The importance of dates is measured by their *burstiness* with probable significant occurrences [3]. The compression rate on t_i is set as $\phi_{s_i} = \frac{|S_i|}{|S|}\phi$ for sentences and $\phi_{m_i} = \frac{|M_i|}{|M|}\phi$ for images.

4.3 System Comparison

We compared our approach against a wide range of well-known summarization methods. All comparison systems were subject to the same preprocessing procedures as our own algorithm, including the image-to-text translation model. As these methods have not been developed with images in mind, we selected images by translating them into pseudo-sentences which we then subsequently ranked. However, note that none of the comparison systems take the mutual dependence of images and sentences into account. Instead, they select sentences and images in parallel.

Our first baseline selects sentences or images randomly for each document collection (Random). Our second method uses MEAD algorithm [19] to extract sentences and images according to centroid value and positional value (Centroid). The third baseline applies the graph-based summarization model proposed by Wan et al. [22]. It first constructs a sentence (or image) connectivity graph based on cosine similarity and then selects important sentences or images based on eigenvector centrality (GBS). We also compared our method against the evolutionary timeline summarization algorithm proposed by Yan et al. [28] which is the state of the art but ignores the image stream (ETS).

4.4 Evaluation

We evaluated the summaries produced by our system and the baselines automatically using the ROUGE evaluation metric [12]. ROUGE counts the number of overlapping units such as N-grams, word sequences, and word pairs between a candidate summary and reference summaries. There are several variants of ROUGE, all aiming at measuring similarity between system and reference summaries. We formally describe ROUGE-N below, one of the most widely used variants. We first define ROUGE-N-R and ROUGE-N-P, two N-gram metrics based on recall and precision, respectively.

$$\text{ROUGE-N-R} = \frac{\sum_{\text{I}\in\text{GT}} \sum_{\text{N-gram}\in\text{I}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{I}\in\text{GT}} \sum_{\text{N-gram}\in\text{I}} \text{Count (N-gram)}} \tag{13}$$

$$\text{ROUGE-N-P} = \frac{\sum_{\text{I}\in\text{CT}} \sum_{\text{N-gram}\in\text{I}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{I}\in\text{CT}} \sum_{\text{N-gram}\in\text{I}} \text{Count (N-gram)}} \tag{14}$$

Here N is the length of the N-gram and N-gram$\in$GT denotes the N-grams in the reference timeline GT, while N-gram$\in$CT denotes the N-grams in the system timeline CT. $\text{Count}_{\text{match}}$(N-gram) is the maximum number of N-grams in the candidate summary and in the set of reference summaries. $\text{Count}_{\text{(N-gram)}}$ is the number of N-grams in the reference summaries or system summary. Rouge-N is the harmonic mean of ROUGE-N-R and ROUGE-N-P:

$$\text{ROUGE-N} = \frac{2 \times \text{ROUGE-N-P} \times \text{ROUGE-N-R}}{\text{ROUGE-N-P} + \text{ROUGE-N-R}} \tag{15}$$

We evaluated our textual summaries using all variants provided by the ROUGE package (version 1.55) and obtained similar results across the board. For the sake of brevity we only report ROUGE-1, ROUGE-2, and the weighted longest common subsequence ROUGE-W (with W set to 1.2). We evaluated our visual summaries in a similar fashion. Recall that we represent images as a bag of visual words. We can therefore use ROUGE to measure the visual word overlap between the images selected by our system and those found in the reference summaries. Since visual words do not have sequential dependencies, we only report ROUGE-1.

Finally, as the timeline consists of a series of individual summaries I which are not equally significant, we compute ROUGE-N scores for timelines as the weighted average ROUGE-N of all summaries:

$$\text{ROUGE-N(I)} = \frac{1}{|\text{I}|} \frac{\sum_{I_i \in I} \phi_i \cdot \text{ROUGE-N}(\text{I}_\text{i})}{\sum_{I_i \in I} \sum_{I_k \in I} \phi_k \cdot \text{ROUGE-N}(\text{I}_\text{k})} \tag{16}$$

System	R_s-1	R_s-2	R_s-W	R_m-1	A(s,m)	H(s,m)
Random	0.317	0.039	0.081	0.126	0.079	0.784
Centroid	0.331	0.050	0.114	0.267	0.223	1.25
GBS	0.364	0.062	0.130	0.283	0.259	1.512
ETS	0.396	0.085	**0.139**	0.297	0.285	2.016
VTS	**0.402**	**0.087**	0.138	**0.320**	**0.376**	**3.380**
INFLUENZA A (ROI* category: Science)						

System	R_s-1	R_s-2	R_s-W	R_m-1	A(s,m)	H(s,m)
Random	0.266	0.043	0.093	0.114	0.139	0.806
Centroid	0.362	0.060	0.129	0.266	0.175	1.651
GBS	0.380	0.106	0.137	0.237	0.198	1.871
ETS	0.481	0.123	0.160	0.316	0.259	2.509
VTS	**0.488**	**0.128**	**0.163**	**0.339**	**0.387**	**3.595**
HAITI EARTHQUAKE (ROI* category: Disasters)						

System	R_s-1	R_s-2	R_s-W	R_m-1	A(s,m)	H(s,m)
Random	0.262	0.041	0.096	0.317	0.157	0.512
Centroid	0.369	0.062	0.128	0.317	0.196	1.482
GBS	0.389	0.084	0.139	0.317	0.217	1.782
ETS	0.483	0.119	0.163	0.369	0.298	2.248
VTS	**0.486**	**0.121**	**0.168**	**0.394**	**0.402**	**3.630**
BP OIL (ROI* category: Accidents)						

System	R_s-1	R_s-2	R_s-W	R_m-1	A(s,m)	H(s,m)
Random	0.254	0.039	0.084	0.085	0.039	0.206
Centroid	0.325	0.053	0.111	0.117	0.099	1.089
GBS	0.359	0.061	0.129	0.128	0.124	0.091
ETS	0.388	0.083	0.134	**0.233**	0.179	1.780
VTS	**0.393**	**0.103**	**0.138**	0.228	**0.339**	**3.085**
OBAMA PRESIDENCY (ROI* category: Politics)						

Table 4: System comparison on four datasets (Influenza A, BP Oil Spill, Haiti Quake, and Presidency).

We set the weight ϕ_i to the compression rate for a sentence component summary or an image component summary.

We also assessed the semantic fit between the selected images and sentences directly. We used Equation (11) to measure the correlation between images and texts, under the assumption that higher values indicate higher similarity between sentences and images, and should thus correspond to better timelines.

We also obtained human judgements for the same task. We asked 15 participants to read the output summaries based on texts and images and rate how well the two correlate. Participants used a 5-point rating scale and were advised to use high numbers in cases where texts and images were a good match and low numbers otherwise (0-terrible, 1-bad, 2-normal, 3-good, 4-excellent).

Components	R_s-1	R_s-2	R_s-W	R_m-1	A(s,m)
LOA	**0.373**	**0.079**	**0.134**	**0.268**	0.244
LEA	0.335	0.078	0.133	0.252	**0.399**
GHA	0.334	0.078	**0.134**	0.241	0.201
−LOA	**0.309**	**0.077**	**0.132**	**0.241**	0.365
−LEA	0.381	0.080	0.143	0.254	**0.244**
−GHA	0.428	0.100	0.151	0.343	0.400
VTS	0.442	0.109	0.152	0.321	0.376

Table 5: Performance of individual components and component combinations for the VTS system.

5. RESULTS

5.1 System Performance

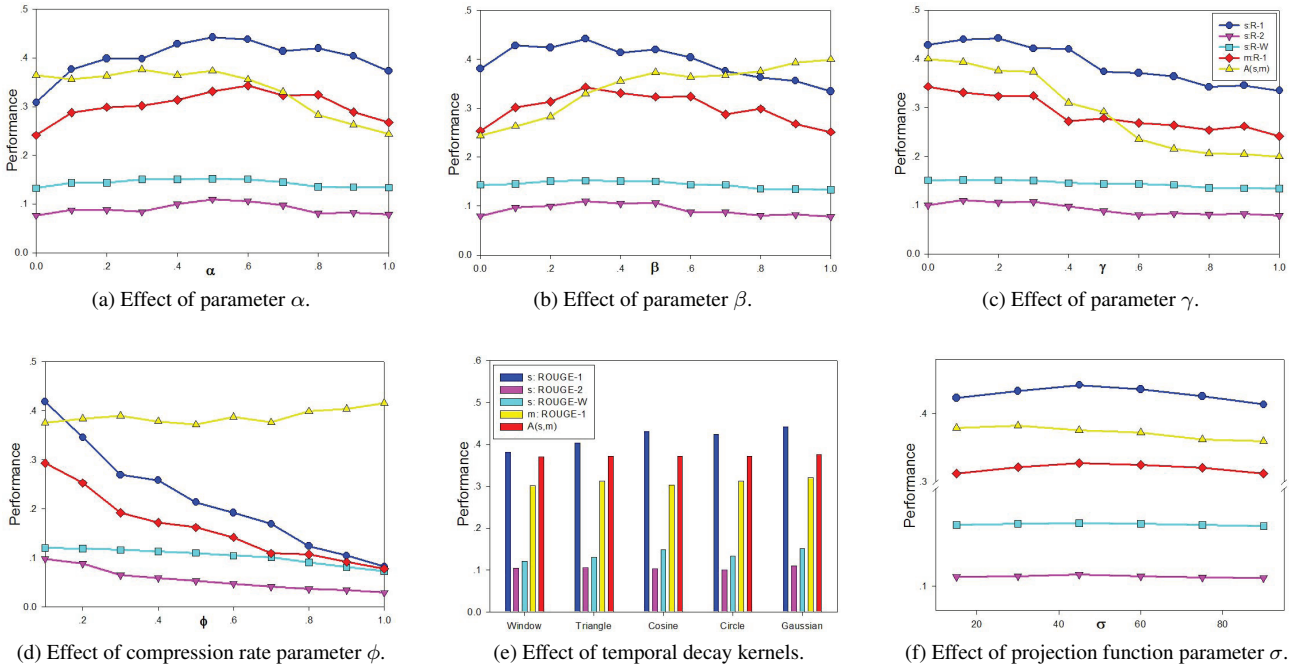
Our results are summarized in Table 4. We report results on each dataset using the following cross-validation scheme: we train parameters on one news set and examine the performance on the others. After 4 training-testing iterations, we take the average performance on all sets. We report results on textual summaries using ROUGE-1 (denoted as R_s-1 in the table), ROUGE-2 (denoted as R_s-2) and ROUGE-W (denoted as R_s-W). We use ROUGE-1 to evaluate the similarity between reference visual summaries and system summaries (denoted as R_m-1). We also show results using an automatic measure of the semantic fit between textual and visual summaries (denoted as A(s,m)) and the human judgments (denoted as H(s,m)).

As can be seen from Table 4 the VTS approach outperforms the comparison systems on almost all datasets, and evaluation measures. The random baseline (Random) performs worst across the board. This is not entirely surprising as it does not take into account the importance of texts or images. The centroid-based system (Centroid) performs better than Random as it tries to identify important sentences and images by taking into account positional information as well as content overlap. The GBS system outperforms the Centroid method in terms of ROUGE and the correlation evaluation introduced in Section 4.4. This is due to the fact that the PageRank-based framework ranks sentences (and images) using eigenvector centrality which implicitly accounts for information subsumption among all sentences or images. ETS produces better timelines than more traditional methods since it has explicit mechanisms for capturing how information evolves over time. A major difference between ETS and VTS is that the former does not take the correlation between images and text into account. Visual and textual timelines are generated independently and as result the fit between the two is not perfect. Taking into account the dependency between the two modalities, improves textual and visual summarization across the board and also produces thematically coherent timelines with the textual and visual components being properly matched. Notice that VTS outperforms all comparison methods on A(s,m) and H(s,m) scores by a large margin on all datasets.

The results in Table 4 have been produced with optimal parameter values. We explore the influence of different parameters in the following sections. Examples of system output are given in Tables 6 and 7 using a compression rate of 5%. Our implementation allows users to have access to the source documents by clicking on the extracted sentences that make up the output summary.

5.2 Component Analysis

We next examine the relative contribution of the individual components of our algorithm. Specifically, we assess the algorithm's performance using only one of the three affinity matrices: Local Homogeneous Affinity (LOA), Local Heterogeneous Affinity (LEA) and Global Homogeneous Affinity (GHA). We also perform ablation studies where one component is removed at a time. Table 5 shows the performance of the individual components (upper half) and the results of our ablation studies (lower half; we use the symbol '−' to indicate which component has been removed). For comparison, we also show the performance of our VTS system. Scores in the table are averages across all four datasets. We use boldface to indicate which components incur remarkable performance changes.



(a) Effect of parameter α.

(b) Effect of parameter β.

(c) Effect of parameter γ.

(d) Effect of compression rate parameter ϕ.

(e) Effect of temporal decay kernels.

(f) Effect of projection function parameter σ.

Figure 3: Parameter effect on system performance using R_s-1, R_s-2, R_s-W, R_m-1 and A(s,m).

As can be seen, LOA has the highest effect on system performance when the latter is measured in terms of ROUGE. LEA on the other hand is important for achieving a good semantic fit between visual and textual summaries. This component, on its own, achieves the highest A(s,m) score. Its removal from the VTS system results in the lowest A(s,m) score.

5.3 Parameter Settings

Recall that our co-ranking framework is parameterized with respect to the relative influence of three different affinities. Parameters α, β, and γ (see equations (5) and (6)) control the contribution of the local homogeneous, local heterogeneous, and global homogeneous affinities. We performed a series of experiments to determine the optimal parameter values for our VTS framework. We varied one parameter at a time, keeping all other parameters fixed as illustrated in Figure 3. Our results indicate that the local homogeneous affinity (parameter α) has the greatest effect on system performance. High values of the γ parameter (global homogeneous affinity) have a negative influence on performance as do very low or very high values of the β parameter (local heterogeneous affinity). Optimal parameter values were set to α=0.6, β=0.3, and γ=0.1.

We also experimented with the effect of the compression rate ϕ. Note that typically ϕ is regulated by users. For example, users who want to read more content, might favor a larger ϕ. We varied ϕ from 0.1 to 1 with a step of 0.1. Generally, the ROUGE lines are down-sloping (see Figure 3d) as our ground truth timelines are small compared to the large source collection our system has access to. Interestingly, the correlation line (see A(s,m) in Figure 3d) is more stable, indicating the intrinsic dependence between texts and images within the news documents.

Finally, an important parameter in timeline summarization is σ which controls the influence of the temporal projection for sentences/images from different dates and thus influences the weights of the neighboring affinity. Changes in σ values do not incur large differences in performance (see Figure 3f). This is partly due to the small value of γ=0.1. We experimented with $\sigma \in [20, 80]$ and empirically set the parameter to σ=35. Given this value, we next examined the effect of different projections. Generally, the Gaussian kernel is the best performing projection and the window kernel the worst performing one. We attribute this to the fact that the Gaussian kernel provides the best smoothing effect without imposing any arbitrary cutoffs.

6. CONCLUSIONS

In this paper we introduced visual timeline summarization, a novel summarization task that creates visual and textual timeline summaries for news topics. We proposed a framework that ranks images and sentences jointly whilst taking into account how a sentence (or an image) associates to other sentences (or images), how sentences and images related to each other, and how the two relate to the overall textual and visual summaries being created. Our model explores the relationship between elements within the same modality and across modalities. We proposed an algorithm that selects images and sentences through mutual reinforcement: it uses the global timeline summary to iteratively refine the local component summaries.

Experimental results on four datasets show that our system outperforms previously proposed competitive baselines. Analysis of the components of our system and its parameters indicates that the semantic fit between images and sentences is important for VTS (β=0.3) as well as taking auxiliary global information into account (γ=0.1). The local homogeneous affinity would perform best in isolation and its absence would lead to the most significant performance drop.

Currently our model treats sentences and images equally without using any prior information with respect to how important they are. In the future, we plan to use prior knowledge to select seed sentences and images. Such knowledge can be based on the positional information of sentences in the document, their timestamps and so on. This way we can ensure better local optima and faster convergence.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. This work was partially supported by NSFC Grant No.60933004 and HGJ Grant No. 2011ZX01042-001-001; Xiaojun Wan was supported by NSFC Grant No. 61170166, Beijing Nova Program (2008B03) and the National High Technology Research and Development Program of China (2012AA011101). Rui Yan was supported by the MediaTek Fellowship.

8. REFERENCES

[1] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of new topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–18, 2001.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, Mar. 2003.

[3] H. L. Chieu and Y. K. Lee. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 425–432, 2004.

[4] P. Duygulu, K. Barnard, J. De Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of 7th European Conference on Computer Vision*, pages 349–354, 2002.

[5] G. Erkan and D. Radev. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2004.

[6] Y. Feng and M. Lapata. Automatic image annotation using auxiliary text information. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics with the Human Lauguage Technology Conference*, pages 272–280, 2008.

[7] Y. Feng and M. Lapata. How many words is a picture worth? automatic caption generation for news images. In *Proceedings of the 48th Annual Meeting on Association for Computational Linguistics*, pages 1239–1249, 2010.

[8] Y. Feng and M. Lapata. Topic models for image annotation and text illustration. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 831–839, 2010.

[9] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: sentence selection and evaluation metrics. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 121–128, 1999.

[10] A. Langville and C. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.

[11] C.-Y. Lin and E. Hovy. From single to multi-document summarization: a prototype system and its evaluation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 457–464, 2002.

[12] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, 2003.

[13] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of 7th IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

[14] Y. Lu, J. He, D. Shan, and H. Yan. Recommending citations with translation model. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2017–2020, 2011.

[15] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and development in information retrieval*, pages 299–306, 2009.

[16] I. Mani. *Automatic Summarization*. John Benjamins Pub Co, 2001.

[17] R. Mihalcea and P. Tarau. A language independent algorithm for single and multiple document summarization. In *Proceedings of the International Joint Conference on Natural Language Processing*, 2005.

[18] J. Neto, A. Santos, C. Kaestner, D. Santos, et al. Document clustering and text summarization. 2000.

[19] D. Radev, H. Jing, M. Stys, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[20] K. Sparck Jones. Automatic summarizing: Factors and directions. In I. Mani and M. T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 1–33. MIT Press, Cambridge, 1999.

[21] R. Swan and J. Allan. Automatic generation of overview timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and development in information retrieval*, pages 49–56, 2000.

[22] X. Wan and J. Yang. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, 2008.

[23] X. Wan, J. Yang, and J. Xiao. Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 2903–2908, 2007.

[24] X. Wan, J. Yang, and J. Xiao. Single document summarization with document expansion. In *Proceedings of the 21st Conference on Artificial Intelligence*, pages 931–936, 2007.

[25] D. Wang and T. Li. Document update summarization using incremental hierarchical clustering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 279–288, 2010.

[26] D. Wang, T. Li, S. Zhu, and C. Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, 2008.

[27] R. Yan, L. Kong, C. Huang, X. Wan, X. Li, and Y. Zhang. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 433–443, 2011.

[28] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of the 34th International SIGIR Conference on Research and Development in Information Retrieval*, pages 745–754, 2011.

April 23, 2009

s_1: Edgar Hernandez, known as the "patient zero" to have contracted the swine flu, lives in the southern Mexican town of La Gloria.
s_2: The WHO makes its first report about the so-called Swine Influenza (A) H1N1.

April 25, 2009

s_1: The government hands out masks at Mexico City: the outbreak of swine flu is making people cautious.
s_2: The World Health Organization has declared a global flu pandemic after holding an emergency meeting.
s_3: U.S. emergency departments step up efforts to control the virus should...
s_4: The Mexican Director-General declares Swine Influenza (A) H1N1 a public health emergency.

April 27, 2009

s_1: The World Health Organization raises the pandemic alert one level to phase 4, which is two steps short of declaring a full-blown pandemic.
s_2: Mexico city shut down schools, museums, libraries and state-run theaters across the overcrowded capital.
s_3: Massachusetts public health officials yesterday mobilized against a possible swine flu outbreak as the alert level is raised.
s_4: European Union's health commissioner warns Europeans to avoid nonessential travel to Mexico and the United States.
s_5: The number of confirmed flu infection cases in the U.S. climbs to 20 in five states.

April 29, 2009

s_1: The World Health Organization (WHO) declares A/H1N1 flu pandemic, raises alert level to highest.
s_2: The CDC said that the number of laboratory-confirmed cases of H1N1 virus infection had climbed to 91 people in 10 states.
s_3: Staff members check the temperature of passengers with the help of machines at Hong Kong International Airport in Hong Kong, south China.

October 2, 2009

s_1: The US announces implementation of a massive campaign to vaccinate millions of Americans against swine flu, with the first 600,000 doses to be distributed in coming days.
s_2: Five doses of swine flu vaccine, part of the first shipment of vaccine sent to Methodist Hospital in Omaha.
s_3: Made H1N1 vaccine is expected to go into mass production soon.

October 5, 2009

s_1: WHO said pharmaceutical firms can produce only 3 billion doses of H1N1 vaccines a year, covering less than half of the global population.
s_2: The U.S. Centers for Disease Control and Prevention says the worst could be yet to come, so Americans need to prepare for a large outbreak this fall and winter.

Table 6: Visual timeline generated by our model on Influenza A.

April 20, 2009

s_1: Explosion and fire on the BP-licensed Transocean drilling rig Deepwater Horizon in the Gulf of Mexico.
s_2: An explosion on the rig on April 20, 2010, killed 11 people working on the rig and injured 16 others.
s_3: The rig was drilling in about 5,000ft (1,525m) of water, pushing the boundaries of deepwater drilling technology.

April 22, 2009

s_1: The Deepwater Horizon sinks to the bottom of the Gulf after burning for 36 hours, raising concerns of a catastrophic...
s_2: A bird covered in oil from the BP Deepwater Horizon spill struggles to climb on to a boom in Gulf of Mexico.
s_3: Search-and-rescue operations by the US National Response Team begin.

April 23, 2010

s_1: The rig is found upside down about a quarter-mile from the blowout preventer.
s_2: Underwater robots and equipment swarm the blowout preventer, bottom left, as oil rises at the spill site.
s_3: Deepwater Horizon clean-up workers fight to prevent disaster.
s_4: The Coast Guard increases its oil spill estimate to 5,000 barrels a day, or 210,000 gallons - five times more than what was originally believed.

April 26, 2010

s_1: BP's shares fall 2% amid fears that the cost of cleanup and legal claims will hit the company hard.
s_2: BP CEO reckons the $100 million cost of drilling a well to divert the flow from a leaking oil well in the Gulf of Mexico is the biggest hit.

Table 7: Visual timeline generated by our model on BP Oil.