Socially-Informed Timeline Generation for Complex Events

Lu Wang Claire Cardie Galen Marchetti
Department of Computer Science
Cornell University
Ithaca, NY 14853
{luwang, cardie}@cs.cornell.edu gjm97@cornell.edu

Abstract

Existing timeline generation systems for complex events consider only information from traditional media, ignoring the rich social context provided by user-generated content that reveals representative public interests or insightful opinions. We instead aim to generate socially-informed timelines that contain both news article summaries and selected user comments. We present an optimization framework designed to balance topical cohesion between the article and comment summaries along with their informativeness and coverage of the event. Automatic evaluations on real-world datasets that cover four complex events show that our system produces more informative timelines than state-of-the-art systems. In human evaluation, the associated comment summaries are furthermore rated more insightful than editor's picks and comments ranked highly by users.

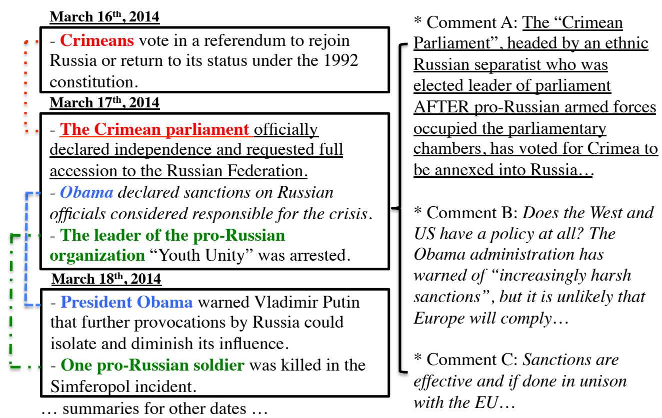

Figure 1: A snippet of the event timeline on Ukraine Crisis is displayed on the left. On the right, we display a set of representative comments addressing the article summary of March 17^{th}. Comment A (underlined) brings a perspective on "Crimean parliament passes declaration of independence" (the article sentence is also underlined on the left). Comments B and C focus on Obama's sanctions on Ukrainian and Russian officials. Sentences linked by edges belong to the same event thread, which is centered on the entities with the same color.

1 Introduction

Social media sites on the Internet provide increasingly more, and increasingly popular, means for people to voice their opinions on trending events. Traditional news media — the New York Times and CNN, for example — now provide online mechanisms that allow and encourage readers to share reactions, opinions, and personal experiences relevant to a news story. For complex emerging events, in particular, user comments can provide relevant, interesting and insightful information beyond the facts reported in the news. But their large volume and tremendous variation in quality make it impossible for readers to efficiently digest the user-generated content, much less integrate it with reported facts from the dozens or hundreds of news reports produced on the event each day.

In this work, we present a *socially-informed timeline generation system* that jointly generates a news article summary and a user comment summary for each day of an ongoing complex event. A sample (gold standard) timeline snippet for Ukraine Crisis is shown in Figure 1. The event timeline is on the left; the comment summary for March 17^{th} is on the right.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1055–1065,
Denver, Colorado, May 31 – June 5, 2015.

While generating timelines from news articles and summarizing user comments have been studied as separate problems (Yan et al., 2011; Ma et al., 2012), their joint summarization for timeline generation raises new challenges. Firstly, there should be a tight connection between the article and comment portion of the timeline. By definition, users comment on socially relevant events. So the important part of articles and insightful comments should both cover these events. Moreover, good reading experience requires that the article summary and comment summary demonstrate evident connectivity. For example, Comment C in Figure 1 ("Sanctions are effective and if done in unison with the EU") is obscure without knowing the context that "sanctions are imposed by U.S". Simply combining the outputs from a timeline generation system and a comment summarization system may lead to timelines that lack cohesion. On the other hand, articles and comments are from intrinsically different genres of text: articles emphasize facts and are written in a professional style; comments reflect opinions in a less formal way. Thus, it could be difficult to recognize the connections between articles and comments. Finally, it is also challenging to enforce continuity in timelines with many entities and events.

To address the challenges mentioned above, we formulate the timeline generation task as an optimization problem, where we maximize topic cohesion between the article and comment summaries while preserving their ability to reflect *important* concepts and subevents, adequate *coverage* of mentioned topics, and *continuity* of the timeline as it is updated with new material each day. We design a novel alternating optimizing algorithm that allows the generation of a high quality article summary and comment summary via mutual reinforcement. We demonstrate the effectiveness of our algorithm on four disparate complex event datasets collected over months from the New York Times, CNN, and BBC. Automatic evaluation using ROUGE (Lin and Hovy, 2003) and gold standard timelines indicates that our system can effectively leverage user comments to outperform state-of-the-art approaches on timeline generation. In a human evaluation via Amazon Mechanical Turk, the comment summaries generated by our method were selected as the best in terms of informativeness and insightfulness in 66.7% and 51.7% of the evaluations (vs. 26.7% and 30.0% for randomly selected editor's-picks).

Especially, our optimization framework relies on two scoring functions that estimate the importance of including individual article sentences and user comments in the timeline. Based on the observation that entities or events frequently discussed in the user comments can help with identify summary-worthy content, we show that the scoring functions can be learned jointly by utilizing graph-based regularization. Experiments show that our joint learning model outperforms state-of-the-art ranking algorithms and other joint learning based methods when evaluated on sentence ranking and comment ranking. For example, we achieve an NDCG@3 of 0.88 on the Ukraine crisis dataset, compared to 0.77 from Yang et al. (2011) which also conducts joint learning between articles and social context using factor graphs.

Finally, to encourage continuity in the generated timeline, we propose an entity-centered event threading algorithm. Human evaluation demonstrates that users who read timelines with event threads write more informative answers than users who do not see the threads while answering the same questions. This implies that our system constructed threads can help users better navigate the timelines and collect relevant information in a short time.

For the rest of the paper, we first describe data collection (Section 2). We then introduce the joint learning model for importance prediction (Section 3). The full timeline generation system is presented in Section 4, which is followed by evaluations (Section 5). Related work and conclusion are in Sections 6 and 7.

2 Data Collection and Preprocessing

We crawled news articles from New York Times (NYT), CNN, and BBC on four trending events: the missing Malaysia Airlines Flight MH370 (MH370), the political unrest in Ukraine (Ukraine), the Israel-Gaza conflict (Israel-Gaza), and the NSA surveillance leaks (NSA). For each event, we select a set of key words (usually entities' name), which are used to filter out irrelevant articles. We collect comments for NYT articles through NYT community API, and comments for CNN articles via Disqus

API. [1] NYT comments come with information on whether a comment is an editor's-pick. The statistics on the four datasets are displayed in Table 1.[2]

	Time Span	# Articles	# Comments
MH370	03/08 - 06/30	955	406,646
Ukraine	03/08 - 06/30	3,779	646,961
Israel-Gaza	07/20 - 09/30	909	322,244
NSA	03/23 - 06/30	145	60,481

Table 1: Statistics on the four event datasets.

We extract parse trees, dependency trees, and coreference resolution results of articles and comments with Stanford CoreNLP (Manning et al., 2014). Sentences in articles are labeled with timestamps using SUTime (Chang and Manning, 2012).

We also collect all articles with comments from NYT in 2013 (henceforth NYT2013) to form a training set for learning importance scoring functions on articles sentences and comments (see Section 3). NYT2013 contains $3,863$ articles and $833,032$ comments.

3 Joint Learning for Importance Scoring

We first introduce a joint learning method that uses *graph-based regularization* to simultaneously learn two functions — a SENTENCE scorer and a COMMENT scorer — that predict the importance of including an individual news article sentence or a particular user comment in the timeline.

We train the model on the aforementioned NYT2013 dataset, where 20% of the articles and their comments are reserved for parameter tuning. Formally, the training data consists of a set of articles $D = \{d_i\}_{i=0}^{|D|-1}$. Each article d_i contains a set of sentences $x_{s_{d_i}} = \{x_{s_{d_i},j}\}_{j=0}^{|s_{d_i}|-1}$ and a set of associated comments $x_{c_{d_i}} = \{x_{c_{d_i},k}\}_{k=0}^{|c_{d_i}|-1}$, where $|s_{d_i}|$ and $|c_{d_i}|$ are the numbers of sentences and comments for d_i. For simplicity, we use x_s or x_c to denote a sentence or a comment wherever there is no ambiguity.

In addition, each article has a human-written abstract. We use the ROUGE-2 (Lin and Hovy, 2003) score of each sentence computed against the associated abstract as its gold-standard importance score. Each comment is assigned a gold-standard value of 1.0 if it is an editor's pick, or 0.0 otherwise.

The SENTENCE and COMMENT scorers rely on two classifiers, each designed to handle the special characteristics of news and user comments, respectively; and a graph-based regularizing constraint that encourages similarity between selected sentences and comments. We describe each component below.

Article SENTENCE **Importance.** Each sentence x_s in a news article is represented as a k-dimensional feature vector $\mathbf{x_s} \in \mathbb{R}^k$, with a gold-standard label y_s. We denote the training set as a feature matrix $\mathbf{\tilde{X}_s}$, with a label vector $\mathbf{\tilde{Y}_s}$. To produce the SENTENCE scoring function $f_s(x_s) = \mathbf{x_s} \cdot \mathbf{w_s}$, we use ridge regression to learn a vector $\mathbf{w_s}$ that minimizes $||\mathbf{\tilde{X}_s w_s} - \mathbf{\tilde{Y}_s}||_2^2 + \beta_s \cdot ||\mathbf{w_s}||_2^2$. Features used in the model are listed in Table 2. We also impose the following *position-based regularizing constraint* to encode the fact that the first sentence in a news article usually conveys the most essential information: $\lambda_s \cdot \sum_{d_i} \sum_{x_{s_{d_i},j}, j\neq 0} ||(\mathbf{x_{s_{d_i},0}} - \mathbf{x_{s_{d_i},j}}) \cdot \mathbf{w_s} - (y_{s_{d_i},0} - y_{s_{d_i},j})||_2^2$, where $x_{s_{d_i},j}$ is the j-th sentence in document d_i. Term $(\mathbf{x_{s_{d_i},0}} - \mathbf{x_{s_{d_i},j}}) \cdot \mathbf{w_s}$ measures the difference in predicted scores between the first sentence and any other sentence. This value is expected be close to the true difference. We further construct $\mathbf{\tilde{X}'_s}$ to contain all difference vectors $(\mathbf{x_{s_{d_i},0}} - \mathbf{x_{s_{d_i},j}})$, with $\mathbf{\tilde{Y}'_s}$ as label difference vector. The objective function to minimize becomes

$$J_s(\mathbf{w_s}) = \\ ||\mathbf{\tilde{X}_s w_s} - \mathbf{\tilde{Y}_s}||_2^2 + \lambda_s \cdot ||\mathbf{\tilde{X}'_s w_s} - \mathbf{\tilde{Y}'_s}||_2^2 + \beta_s \cdot ||\mathbf{w_s}||_2^2 \quad (1)$$

User COMMENT **Importance.** Similarly, each comment x_c is represented as an $l-$dimensional feature vector $\mathbf{x_c} \in \mathbb{R}^l$, with label y_c. Comments in the training data are denoted with a feature matrix $\mathbf{\tilde{X}_c}$ with a label vector $\mathbf{\tilde{Y}_c}$. Likewise, we learn $f_c(x_c) = \mathbf{x_c} \cdot \mathbf{w_c}$ by minimizing $||\mathbf{\tilde{X}_c w_c} - \mathbf{\tilde{Y}_c}||_2^2 + \beta_c \cdot ||\mathbf{w_c}||_2^2$. Features are listed in Table 3. We apply a *pairwise preference-based regularizing constraint* (Joachims, 2002) to incorporate a bias toward editor's picks: $\lambda_c \cdot \sum_{d_i} \sum_{x_{c_{d_i},j} \in \mathbf{E_{d_i}}, x_{c_{d_i},k} \notin \mathbf{E_{d_i}}} ||(\mathbf{x_{c_{d_i},j}} - \mathbf{x_{c_{d_i},k}}) \cdot \mathbf{w_c} - 1||_2^2$, where $\mathbf{E_{d_i}}$ are the editor's picks for d_i. Term $(\mathbf{x_{c_{d_i},j}} - \mathbf{x_{c_{d_i},k}}) \cdot \mathbf{w_c}$ enforces the separation of editor's picks from regular comments. We further construct $\mathbf{\tilde{X}'_c}$ to contain all the pairwise differences

[1]BBC comment volume is low, so we do not collect it.

[2]The datasets are available at `http://www.cs.cornell.edu/~luwang/data.html`.

$(\mathbf{x_{c_{d_i},j}} - \mathbf{x_{c_{d_i},k}})$. $\mathbf{\tilde{Y}'_c}$ is a vector of same size as $\mathbf{\tilde{X}'_c}$ with each element as 1. Thus, the objective function to minimize is:

$$J_c(\mathbf{w_c}) = ||\mathbf{\tilde{X}_c w_c} - \mathbf{\tilde{Y}_c}||_2^2 + \lambda_c \cdot ||\mathbf{\tilde{X}'_c w_c} - \mathbf{\tilde{Y}'_c}||_2^2 + \beta_c \cdot ||\mathbf{w_c}||_2^2 \quad (2)$$

Graph-Based Regularization. The regularizing constraint is based on two mutually reinforcing hypotheses: (1) the importance of a sentence depends partially on the availability of sufficient insightful comments that touch on topics in the sentence; (2) the importance of a comment depends partially on whether it addresses notable events reported in the sentences. For example, we want our model to bias $\mathbf{w_s}$ to predict a high score for a sentence with high similarity to numerous insightful comments.

We first create a bipartite graph from sentences and comments on the same articles, where edge *weights* are based on the content similarity between a sentence and a comment (TF-IDF similarity is used). Let $\mathbf{\tilde{R}}$ be an $N \times M$ adjacency matrix, where N and M are the numbers of sentences and comments. R_{sc} is the similarity between sentence x_s and comment x_c. We normalize $\mathbf{\tilde{R}}$ by $\mathbf{\tilde{Q}} = \mathbf{\tilde{D}}^{-\frac{1}{2}}\mathbf{\tilde{R}}\mathbf{\tilde{D}'}^{-\frac{1}{2}}$, where $\mathbf{\tilde{D}}$ and $\mathbf{\tilde{D}'}$ are diagonal matrices: $\mathbf{\tilde{D}} \in \mathbb{R}^{N\times N}$, $D_{i,i} = \sum_{j=1}^{M} R_{i,j}$; $\mathbf{\tilde{D}'} \in \mathbb{R}^{M\times M}$, $D'_{j,j} = \sum_{i=1}^{N} R_{i,j}$. The interplay between the two types of data is encoded in the following regularizing constraint:

$$J_{s,c}(\mathbf{w_s}, \mathbf{w_c}) = \lambda_{sc} \cdot \sum_{d_i} \sum_{x_s \in x_{s_{d_i}}, x_c \in x_{c_{d_i}}} Q_{x_s,x_c} \cdot (\mathbf{x_s} \cdot \mathbf{w_s} - \mathbf{x_c} \cdot \mathbf{w_c})^2 \quad (3)$$

Full Objective Function. Thus, the full objective function consists of the three parts discussed above:

$$J(\mathbf{w_s}, \mathbf{w_c}) = J_s(\mathbf{w_s}) + J_c(\mathbf{w_c}) + J_{s,c}(\mathbf{w_s}, \mathbf{w_c}) \quad (4)$$

Furthermore, using the following notation,

$$\mathbf{\tilde{X}} = \begin{bmatrix} \mathbf{\tilde{X}_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{\tilde{X}_c} \end{bmatrix} \; \mathbf{\tilde{Y}} = \begin{bmatrix} \mathbf{\tilde{Y}_s} \\ \mathbf{\tilde{Y}_c} \end{bmatrix} \; \mathbf{\tilde{X}'} = \begin{bmatrix} \mathbf{\tilde{X}'_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{\tilde{X}'_c} \end{bmatrix} \; \mathbf{\tilde{Y}'} = \begin{bmatrix} \mathbf{\tilde{Y}'_s} \\ \mathbf{\tilde{Y}'_c} \end{bmatrix}$$

$$\tilde{\boldsymbol{\beta}} = \begin{bmatrix} \beta_s \mathbf{I_k} & \mathbf{0} \\ \mathbf{0} & \beta_c \mathbf{I_l} \end{bmatrix} \quad \tilde{\boldsymbol{\lambda}} = \begin{bmatrix} \lambda_s \mathbf{I}_{|\mathbf{X'_s}|} & \mathbf{0} \\ \mathbf{0} & \lambda_c \mathbf{I}_{|\mathbf{X'_c}|} \end{bmatrix}$$

$$\mathbf{\tilde{L}} = \begin{bmatrix} \lambda_{sc} \mathbf{I}_{|\mathbf{X_s}|} & -\lambda_{sc} \mathbf{\tilde{Q}} \\ -\lambda_{sc} \mathbf{\tilde{Q}^T} & \lambda_{sc} \mathbf{I}_{|\mathbf{X_c}|} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} \mathbf{w_s} \\ \mathbf{w_c} \end{bmatrix}$$

we can show a **closed form solution** to Equation 4 as follows:

$$\mathbf{\hat{w}} = (\mathbf{\tilde{X}^T \tilde{L} \tilde{X}} + \mathbf{\tilde{X}^T \tilde{X}} + \mathbf{\tilde{X}'^T} \tilde{\boldsymbol{\lambda}} \mathbf{\tilde{X}'} + \tilde{\boldsymbol{\beta}})^{-1} (\mathbf{\tilde{X}^T \tilde{Y}} + \mathbf{\tilde{X}'^T} \tilde{\boldsymbol{\lambda}} \mathbf{\tilde{Y}'}) \quad (5)$$

Basic Features	Social Features
- num of words	- avg/sum frequency of words appearing in comment
- absolute/relative position	- avg/sum frequency of dependency relations appearing in comment
- overlaps with headline	
- avg/sum TF-IDF scores	
- num of NEs	

Table 2: Features used for sentence importance scoring.

Basic Features	Readability Features
- num of words	- Flesch-Kincaid Readability
- num of sentences	- Gunning-Fog Readability
- avg num of words per sentence	**Discourse Features**
- num of NEs	- num/proportion of connectives
- num/proportion of capitalized words	- num/proportion of hedge words
- avg/sum TF-IDF	**Article Features**
- contains URL	- TF/TF-IDF simi with article
- user rating (pos/neg)	- TF/TF-IDF simi with comments
	- JS/KL divergence (div) with article
	- JS/KL div with comments
Sentiment Features	
- num /proportion of positive/negative/neutral words (MPQA (Wilson et al., 2005), General Inquirer (Stone et al., 1966))	
- num /proportion of sentiment words	

Table 3: Features used for comment importance scoring.

4 Timeline Generation

Now we present an optimization framework for timeline generation. Formally, for each day, our system takes as input a set of sentences V_s and a set of comments V_c to be summarized, and the (automatically generated) timeline $\mathcal{T}$ (represented as threads) for days prior to the current day. It then identifies a subset $S \subseteq V_s$ as the article summary and a subset $C \subseteq V_c$ as the comment summary by maximizing the following function:

$$\mathcal{Z}(S, C; \mathcal{T}) = \mathcal{S}_{qual}(S; \mathcal{T}) + \mathcal{C}_{qual}(C) + \delta \mathcal{X}(S, C) \quad (6)$$

where $\mathcal{S}_{qual}(S; \mathcal{T})$ measures the quality of the article summary S in the context of the historical timeline represented as event threads $\mathcal{T}$; $\mathcal{C}_{qual}(C)$ computes the quality of the comment summary C; and $\mathcal{X}(S, C)$ estimates the connectivity between S and C.

We solve this maximization problem using an alternating optimization algorithm which is outlined in Section 4.4. In general, we alternately search for a better article summary S with hill climbing search and a better comment summary C with Ford-Fulkerson algorithm until convergence.

In the rest of this section, we first describe an *entity-centered event threading* algorithm to construct event threads $\mathcal{T}$ which are used to boost article timeline continuity. Then we explain how to compute $\mathcal{S}_{qual}(S;\mathcal{T})$ and $\mathcal{C}_{qual}(C)$ in Section 4.2, followed by $\mathcal{X}(S, C)$ in Section 4.3.

4.1 Entity-Centered Event Threading

We present an event threading process where each thread connects sequential events centered on a set of relevant *entities*. For instance, the following thread connects events about *Obama*'s action towards the annexation of Crimea by *Russia*:

Day 1: *Obama* declared sanctions on *Russian officials*.
Day 2: *President Obama* warned *Russian*.
Day 3: *Obama* urges *Russian* to move back its troops.
Day 4: *Obama* condemns *Russian* aggression in Ukraine.

We first collect relation extractions as *(entity, relation, entity)* triples from OLLIE (Mausam et al., 2012), a dependency relation based open information extraction system. We retain extractions with confidence scores higher than 0.5. We further design syntactic patterns based on Fader et al. (2011) to identify relations expressed as a combination of a verb and nouns. Each relation contains at least one event-related word (Ritter et al., 2012).

The *entity-centered event threading* algorithm works as follows: on the first day, each sentence in the summary becomes an individual cluster; thereafter, each sentence in the current day's article summary either gets attached to an existing thread or starts a new thread. The updated threads then become the input to next day's summary generation process. On day n, we have a set of threads $\mathcal{T} = \{\tau : \mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_{n-1}}\}$ constructed from previous $n-1$ days, where $\mathbf{s_i}$ represents the set of sentences attached to thread τ from day i. The *cohesion* between a new sentence $s \in S$ and a thread τ is denoted as $cohn(s, \tau)$. s is attached to $\hat{\tau}$ if there exists $\hat{\tau} = \max_{\tau \in T} cohn(s, \tau)$ and $cohn(s, \hat{\tau}) > 0.0$. Otherwise, s becomes a new thread. We define $cohn(s, \tau) = \min_{\mathbf{s_i} \in \tau, \mathbf{s_i} \neq \emptyset} tfsimi(\mathbf{s_i}, s)$, where $tfsimi(\mathbf{s_i}, s)$ measures the TF similarity between $\mathbf{s_i}$ and s. We consider unigrams/bigrams/trigrams generated from the entities of our event extractions.

4.2 Summary Quality Measurement

Recall that we learned two separate importance scoring functions for sentences and comments, which will be denoted here as $imp_s(s)$ and $imp_c(c)$. With an article summary S and threads $\mathcal{T} = \{\tau_i\}$, the **article summary quality** function $\mathcal{S}_{qual}(S;\mathcal{T})$ has the following form:

$$\mathcal{S}_{qual}(S;\mathcal{T}) = \sum_{s \in S} imp(s)$$
$$+\theta_{cov} \sum_{s' \in V_s} \min(\sum_{s \in S} tfidf(s, s'), \alpha \sum_{\hat{s} \in V_s} tfidf(\hat{s}, s'))$$
$$+ \theta_{cont} \sum_{\tau \in \mathcal{T}} \max_{s_k \in S} cohn(s_k, \tau)$$

$tfidf(\cdot,\cdot)$ is the TF-IDF similarity function. $\mathcal{S}_{qual}(S;\mathcal{T})$ captures three desired qualities of an article summary: *importance* (first item), *coverage* (second item), and the *continuity* of the current summary to previously generated summaries. The coverage function has been used to encourage summary diversity and reduce redundancy (Lin and Bilmes, 2011; Wang et al., 2014). The continuity function considers how well article summary S can be attached to each event thread, thus favors summaries that can be connected to multiple threads.

Parameters θ_{cov} and α are tuned on multi-document summarization dataset DUC 2003 (Over and Yen, 2003). Experiments show that system performance peaks and is stable for $\theta_{cont} \in [1.0, 5.0]$. We thus fix θ_{cont} to 1.0. We discard sentences with more than 80% of content words covered by historical summaries. We use BASIC to denote a system that only optimizes on importance and coverage (i.e. first two items in $\mathcal{S}_{qual}(S;\mathcal{T})$). The system optimizing $\mathcal{S}_{qual}(S;\mathcal{T})$ is henceforth called THREAD.

The **comment summary quality** function simply takes the form $\mathcal{C}_{qual}(C) = \sum_{c \in C} imp_c(c)$.

4.3 Connectivity Measurement

We encode two objectives in the connectivity function $\mathcal{X}(S, C)$: (1) encouraging topical cohesion (i.e. connectivity) between article summary and comment summary; and (2) favoring comments that cover diversified events.

Let $conn(s, c)$ measure content similarity between a sentence $s \in S$ and a comment $c \in C$. Connectivity between article summary S and comment summary C is computed as follows. We build a bipartite graph $\mathcal{G}$ between S and C with edge weight as $conn(s, c)$.

We then find an edge set $\mathcal{M}$, the best matching of $\mathcal{G}$. $\mathcal{X}(S,C)$ is defined as the sum over edge weights in $\mathcal{M}$, i.e. $\mathcal{X}(S,C) = \sum_{e \in \mathcal{M}} weight(e)$. An example is illustrated in Figure 2.

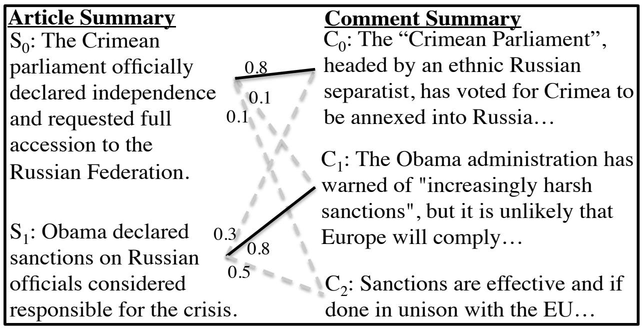

Figure 2: An example on computing the connectivity between an article summary (left) and a comment summary (right) via best matching in bipartite graph. Number on each edge indicates the content similarity between a sentence and a comment. Solid lines are edges in the best matching graph. For this example, the connectivity $\mathcal{X}(S,C)$ is $0.8 + 0.8 = 1.6$.

We consider two options for $conn(s,c)$. One is *lexical similarity* which is based on TF-IDF vectors. Another is *semantic similarity*. Let $R_s = \{(a_s, r_s, b_s)\}$ and $R_c = \{(a_c, r_c, b_c)\}$ be the sets of dependency relations in s and c. $conn(s,c)$ is calculated as:

$$\sum_{(a_s,r_s,b_s)\in R_s} \max_{\substack{(a_c,r_c,b_c)\in R_c \\ r_s=r_c}} simi(a_s,a_c) \times simi(b_s,b_c)$$

where $simi(\cdot,\cdot)$ is a word similarity function. We experiment with shortest path based similarity defined on WordNet (Miller, 1995) and Cosine similarity with word vectors trained on Google news (Mikolov et al., 2013). Systems using the three metrics that optimize $\mathcal{Z}(S,C;\mathcal{T})$ are henceforth called THREAD+OPT$_{\texttt{TFIDF}}$, THREAD+OPT$_{\texttt{WordNet}}$ and THREAD+OPT$_{\texttt{WordVec}}$.

4.4 An Alternating Optimization Algorithm

To maximize the full objective function $\mathcal{Z}(S,C;\mathcal{T})$, we design a novel alternating optimization algorithm (Alg. 1) where we alternately find better S and C.

We initialize S_0 by a greedy algorithm (Lin and Bilmes, 2011) with respect to $\mathcal{S}_{qual}(S;\mathcal{T})$. Notice that $\mathcal{S}_{qual}(S;\mathcal{T})$ is a submodular function, so that the greedy solution is a $1 - 1/e$ approximation to the optimal solution of $\mathcal{S}_{qual}(S;\mathcal{T})$. Fixing S_0, we model the problem of finding C_0 that maximizes $\mathcal{C}_{qual}(C) + \delta\mathcal{X}(S_0,C)$ as a maximum-weight bipartite graph matching problem. This problem can be reduced to a maximum network flow problem, and then be solved by Ford-Fulkerson algorithm (details are discussed in (Kleinberg and Tardos, 2005)). Thereafter, for each iteration, we alternately find a better S_t with regard to $\mathcal{S}_{qual}(S;\mathcal{T}) + \delta\mathcal{X}(S,C_{t-1})$ using hill climbing, and an exact solution C_t to $\mathcal{C}_{qual}(C) + \delta\mathcal{X}(S_t,C)$ with Ford-Fulkerson algorithm. Iteration stops when the increase of $\mathcal{Z}(S,C)$ is below threshold ϵ (set to 0.01). System performance is stable when we vary $\delta \in [1.0, 10.0]$, so we set $\delta = 1.0$.

Input : sentences V_s, comments V_c, threads $\mathcal{T}$, δ, threshold ϵ, functions $\mathcal{Z}(S,C;\mathcal{T})$, $\mathcal{S}_{qual}(S;\mathcal{T})$, $\mathcal{C}_{qual}(C)$, $\mathcal{X}(S,C)$
Output: article summary S, comment summary C

`/* Initialize S and C by greedy algorithm and Ford-Fulkerson algorithm */`
$S_0 \leftarrow \max_S \mathcal{S}_{qual}(S;\mathcal{T})$;
$C_0 \leftarrow \max_C \mathcal{C}_{qual}(C) + \delta\mathcal{X}(S_0,C)$;
$t \leftarrow 1$;
$\Delta\mathcal{Z} \leftarrow \infty$;
while $\Delta\mathcal{Z} > \epsilon$ **do**
  `/* Step 1: Hill climbing algorithm */`
  $S_t \leftarrow \max_S \mathcal{S}_{qual}(S;\mathcal{T}) + \delta\mathcal{X}(S,C_{t-1})$;
  `/* Step 2: Ford-Fulkerson algorithm */`
  $C_t \leftarrow \max_C \mathcal{C}_{qual}(C) + \delta\mathcal{X}(S_t,C)$;
  $\Delta\mathcal{Z} = \mathcal{Z}(S_t,C_t;\mathcal{T}) - \mathcal{Z}(S_{t-1},C_{t-1};\mathcal{T})$;
  $t \leftarrow t+1$;
end

Algorithm 1: Generate article summary and comment summary for a given day via alternating optimization .

Algorithm 1 is guaranteed to find a solution at least as good as S_0 and C_0. It progresses only if Step 1 finds S_t that improves upon $\mathcal{Z}(S_{t-1},C_{t-1};\mathcal{T})$, and Step 2 finds C_t where $\mathcal{Z}(S_t,C_t;\mathcal{T}) \geq \mathcal{Z}(S_t,C_{t-1};\mathcal{T})$.

5 Experimental Results

5.1 Evaluation of SENTENCE and COMMENT Importance Scorers

We test importance scorers (Section 3) on single document *sentence ranking* and *comment ranking*.

For both tasks, we compare with two previous systems on joint ranking and summarization of news articles and tweets. *Yang et al. (2011)* employ supervised learning based on factor graphs to model content similarity between the two types of data. We use the same features for this model. *Gao et al. (2012)* summarize by including the complementary information between articles and tweets, which is estimated by an unsupervised topic model.[3] We also consider two state-of-the-art rankers: *RankBoost* (Freund et al., 2003) and *LambdaMART* (Burges, 2010). Finally, we use a *position baseline* that ranks sentences based on their position in the article, and a *rating baseline* that ranks comments based on positive user ratings.

We evaluate using normalized discounted cumulative gain at top 3 returned results (NDCG@3). Sentences are considered relevant if they have ROUGE-2 scores larger than 0.0 (computed against human abstracts), and comments are considered relevant if they are editor's picks.[4] Figure 3 demonstrates that our joint learning model uniformly outperforms all the other comparisons for both ranking tasks. In general, supervised learning based approaches (e.g. our method, Yang et al. (2011), RankBoost, and LambdaMART) produce better results than unsupervised method (e.g. Gao et al. (2012)).[5]

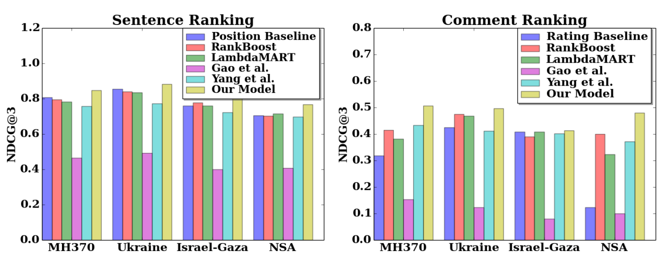

Figure 3: Evaluation of sentence and comment ranking on the four datasets by using normalized discounted cumulative gain at top 3 returned results (NDCG@3). Our joint learning based approach uniformly outperforms all the other comparisons.

5.2 Leveraging User Comments

In this section, we test if our system can leverage comments to produce better article-based summaries for event timelines. We collect **gold-standard timelines** for each of the four events from the corresponding Wikipedia page(s), NYT topic page, or BBC news page.

We consider two existing timeline creation systems that only utilize news articles, and a timeline generated from single-article human abstracts: (1) CHIEU AND LEE (2004) select sentences with high "interestingness" and "burstiness" using a likelihood ratio test to compare word distributions of sentences with articles in neighboring days. (2) YAN ET AL. (2011) design an evolutionary summarization system that selects sentences based on on coverage, coherence, and diversity. (3) We construct a timeline from the human ABSTRACTs provided with each article: we sort them chronologically according to article timestamps and add abstract sentences into each daily summary until reaching the word limit.

We test on five variations of our system. The first two systems generate article summaries with no comment information by optimizing $S_{qual}(S; \mathcal{T})$ using a greedy algorithm: BASIC ignores event threading; THREAD considers the threads. THREAD+OPT$_{\mathtt{TFIDF}}$, THREAD+OPT$_{\mathtt{WordNet}}$ and THREAD+OPT$_{\mathtt{WordVec}}$ (see Section 4.3) leverage user comments to generate article summaries as well as comment summaries based on alternating optimization of Equation 3. Although comment summaries are generated, they are not used in the evaluation.

For all systems, we generate daily article summaries of at most 100 words, and select 5 comments for the corresponding comment summary. We employ ROUGE (Lin and Hovy, 2003) to automatically evaluate the content coverage (in terms of ngrams) of the article-based timelines vs. gold-standard timelines. ROUGE-2 (measures bigram overlap) and ROUGE-SU4 (measures unigram and skip-bigrams separated by up to four words) scores are reported in Table 4. As can be seen, under the alternating optimization framework, our systems, employing both articles and comments, consistently yield better ROUGE scores than the three baseline systems and our systems that do not leverage comments. Though constructed from single-article abstracts, baseline ABSTRACT is found to contain redundant information and thus limited in content coverage. This is due to the fact that different media tend to report on the same important events.

5.3 Evaluating Socially-Informed Timelines

We evaluate the full article+comment-based timelines on Amazon Mechanical Turk. Turkers are presented with a timeline consisting of five consecutive days' article summaries and four variations of the accompanying comment summary:

[3] We thank Zi Yang and Peng Li for providing the code.

[4] We experiment with all articles for sentence ranking, and NYT comments (with editor's picks) for comment ranking.

[5] Similar results are obtained with mean reciprocal rank.

	MH370		Ukraine		Israel-Gaza		NSA	
	R-2	*R-SU4*	*R-2*	*R-SU4*	*R-2*	*R-SU4*	*R-2*	*R-SU4*
CHIEU AND LEE	6.43	10.89	4.64	8.87	3.38	**7.32**	6.14	9.73
YAN ET AL.	6.37	10.35	4.57	8.67	2.39	5.78	3.99	7.73
ABSTRACT	6.16	10.62	3.85	8.40	2.21	5.42	7.03	8.65
- Greedy Algorithm								
BASIC	6.59	9.80	5.31	9.23	3.15	6.20	3.81	7.58
THREAD	6.55	10.86	5.73	9.75	3.16	6.16	6.29	10.09
- Alternating Optimization (leveraging comments)								
THREAD+OPT$_{\texttt{TFIDF}}$	*8.74*	11.63	*9.10*	*12.59*	3.78	6.45	*8.07*	*10.31*
THREAD+OPT$_{\texttt{WordNet}}$	*8.73*	***11.87***	*8.67*	*12.10*	**4.11**	6.64	***8.63***	***11.12***
THREAD+OPT$_{\texttt{WordVec}}$	***9.29***	11.63	***9.16***	***12.72***	3.75	6.38	*8.29*	*10.36*

Table 4: ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores (multiplied by 100) for different timeline generation approaches on four event datasets. Systems that statistically significantly outperform the three baselines ($p < 0.05$, paired $t-$test) are in *italics*. Numbers in **bold** are the highest score for each column.

RANDOMly selected comments, USER'S-PICKS (ranked by positive user ratings), *randomly* selected EDITOR'S-PICKS and timelines produced by the THREAD+OPT$_{\texttt{WordVec}}$ version of OUR SYSTEM. We also include one noisy comment summary (i.e. irrelevant to the question) to avoid spam. We display two comments per day for each system.[6]

Turkers are asked to rank the comment summary variations according to *informativeness* and *insightfulness*. For informativeness, we ask the Turkers to judge based only on knowledge displayed in the timeline, and to rate each comment summary based on how much relevant information they learn from it. For insightfulness, Turkers are required to focus on insights and valuable opinions. They are requested to leave a short explanation of their ranking.

15 five-day periods are randomly selected. We solicit four distinct Turkers located in the U.S. to evaluate each set of timelines. An inter-rater agreement of Krippendorff's α of 0.63 is achieved for informativeness ranking and α is 0.50 for insightfulness ranking.

Table 5 shows the percentage of times a particular method is selected as producing the best comment portion of the timeline, as well as the micro-average rank of each method, for both informativeness and insightfulness. Our system is selected as the best in 66.7% of the evaluations for informativeness and 51.7% for insightfulness. In both cases, we statistically significantly outperform ($p < 0.05$ using a Wilcoxon signed-rank test) the editor's-picks

[6]For our system, we select the two comments with highest importance scores from the comment summary.

	Informativeness		Insightfulness	
	% Best	Avg Rank	% Best	Avg Rank
Random	1.7%	3.67	3.3%	3.58
User's-picks	5.0%	2.83	15.0%	2.55
Editor's-picks	26.7%	2.05	30.0%	2.22
Our system	**66.7%**	**1.45**	**51.7%**	**1.65**

Table 5: Human evaluation results on the comment portion of socially-informed timelines. **Boldface** indicates statistical significance vs. other results in the same column using a Wilcoxon signed-rank test ($p < 0.05$). On average, the output from our system is ranked higher than all other alternatives.

and user's-picks. Turkers' explanations indicate that they prefer our comment summaries mainly because they are "very informative and insightful to what was happening", and "show the sharpness of the commenter". Turkers sometimes think the summaries randomly selected from editor's-picks "lack connection", and characterize user's-picks as "the information was somewhat limited".

Figure 4 shows part of the timeline generated by our system for the Ukraine crisis.

Article Summary	Comment Summary
2014-03-17 Obama administration froze the U.S. assets of seven Russian officials, while similar sanctions were imposed on four Ukrainian officials. . . .	Theodore Roosevelt said that the worst possible thing you can do in diplomacy is "soft hitting". That is what the US and the EU are doing in these timid "sanctions" against people without any overseas accounts. . .
2014-03-18 Ukraine does not recognize a treaty signed in Moscow on Tuesday making its Crimean peninsula a part of Russia. . .	Though there were many in Crimea who supported annexation, there were certainly some who did not. what about those people?. . .
2014-03-19 The head of NATO warned on Wednesday that Russian President Vladimir Putin may not stop with the annexation of Crimea . . .	If you look at a real map , Crimea is an island and has always been more connected to Russia than to Ukraine. . .
2014-03-20 The United States on Thursday expanded its sanctions on Russians. . .	The US and EU should follow up economic sanctions with concrete steps to strengthen NATO. . .

Figure 4: A snippet of timeline generated by our system THREAD+OPT$_{\texttt{WordVec}}$ for the Ukraine crisis. Due to space limitations, we only display partial summaries.

5.4 Human Evaluation of Event Threading

Here we evaluate on the utility of event threads for high-level information access guidance: *can event threads allow users to easily locate and absorb information with a specific interest in mind?*

We first sample a 10-day timeline for each dataset from those produced by the THREAD+OPT$_{\texttt{WordVec}}$

variation of our system. We designed one question for each timeline. Sample questions are: "describe the activities for searching for the missing flight MH370", and "describe the attitude and action of Russian Government on Eastern Ukraine". We recruited 10 undergraduate and graduate students who are native speakers of English. Each student first read one question and its corresponding timeline for 5 minutes. The timeline was then removed, and the student wrote down an answer for the question. We asked each student to answer the question for each of four timelines (one for each event dataset). Two timelines are displayed with threads, and two without threads. We presented threads by adding a thread number in front of each sentence.

We then used Amazon Mechanical Turk to evaluate the informativeness of students' answers. Turkers were asked to read all 10 answers for the same question, with five answers based on timelines with threads and five others based on timelines without threads. After that, they rated each answer with an informativeness score on a 1-to-5 rating scale (1 as "not relevant to the query", and 5 as "very informative"). We also added two quality control questions. Table 6 shows that the average rating for answers written after reading timelines *with threads* is 3.29 (43% are rated ≥ 4), higher than the 2.58 for the timelines with *no thread* exhibited (30% are rated ≥ 4).

Answer Type	Avg $\pm$ STD	Rated 5 (%)	Rated 4 (%)
No Thread	2.58 ± 1.20	7%	23%
With Threads	3.29 ± 1.28	17%	26%

Table 6: Human evaluation on the informativeness of answers written after reading timelines *with threads* vs. with *no thread*. Answers written with access to threads are rated higher (3.29) than the ones with no thread (2.58).

6 Related Work

There is a growing interest in generating article summaries informed by social context. Existing work focuses on learning users' interests from comments and incorporates the learned information into a news article summarization system (Hu et al., 2008). Zhao et al. (2013) instead estimate word distributions from tweets, and bias a Page Rank algorithm to give higher restart probability to sentences with similar distributions. Generating tweet+article summaries has been recently investigated in Yang et al. (2011). They propose a factor graph to allow sentences and tweets to mutually reinforce each other. Gao et al. (2012) exploit a co-ranking model to identify sentence-tweet pairs with complementary information estimated from a topic model. These efforts handle a small number of documents and tweets, while we target a larger scale of data.

In terms of timeline summarization, the Chieu and Lee (2004) system ranks sentences according to "burstiness" and "interestingness" estimated by a likelihood ratio test. Yan et al. (2011) explore an optimization framework that maximizes the relevance, coverage, diversity, and coherence of the timeline. Neither system has leveraged the social context. Our event threading algorithm is also inspired by work on topic detection and tracking (TDT) (Allan et al., 1998), where efforts are made for document-level link detection and topic tracking. Similarly, Nallapati et al. (2004) investigate event threading for articles, where they predict linkage based on causal and temporal dependencies. Shahaf et al. (2012) instead seek for connecting articles into one coherent graph. To the best of our knowledge, we are the first to study sentence-level event threading.

7 Conclusion

We presented a socially-informed timeline generation system, which constructs timelines consisting of article summaries and comment summaries. An alternating optimization algorithm is designed to maximize the connectivity between the two sets of summaries as well as their importance and information coverage. Automatic and human evaluations showed that our system produced more informative timelines than state-of-the-art systems. Our comment summaries were also rated as very insightful.

Acknowledgments

We thank John Hessel, Lillian Lee, Moontae Lee, David Lutz, Karthik Raman, Vikram Rao, Yiye Ruan, Xanda Schofield, Adith Swaminathan, Chenhao Tan, Bishan Yang, other members of Cornell NLP group, and the NAACL reviewers for valuable suggestions and advice on various aspects of this work. This work was supported in part by DARPA DEFT Grant FA8750-13-2-0015.

References

J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, Lansdowne, VA, USA, February. 007.

Christopher J. C. Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. Technical report, Microsoft Research.

Angel X. Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 425–432, New York, NY, USA. ACM.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, December.

Wei Gao, Peng Li, and Kareem Darwish. 2012. Joint topic modeling for event summarization across news and social media streams. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1173–1182, New York, NY, USA. ACM.

Meishan Hu, Aixin Sun, and Ee-Peng Lim. 2008. Comments-oriented document summarization: Understanding documents with readers' feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 291–298, New York, NY, USA. ACM.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA. ACM.

Jon Kleinberg and Eva Tardos. 2005. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 71–78.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 265–274, New York, NY, USA. ACM.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. 2004. Event threading within news topics. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 446–453, New York, NY, USA. ACM.

P. Over and J. Yen. 2003. An introduction to DUC 2003: Intrinsic evaluation of generic news text summarization systems.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1104–1112, New York, NY, USA. ACM.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Trains of thought: Generating information maps. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 899–908, New York, NY, USA. ACM.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

Lu Wang, Hema Raghavan, Claire Cardie, and Vittorio Castelli. 2014. Query-focused opinion summarization for user-generated content. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1660–1669, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 745–754, New York, NY, USA. ACM.

Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. 2011. Social context summarization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 255–264, New York, NY, USA. ACM.

Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. Timeline generation with social attention. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1061–1064, New York, NY, USA. ACM.