Analyzing evolving stories in news articles

Roberto Camacho Barranco · Arnold P. Boedihardjo · M. Shahriar Hossain

Received: 27 May 2017 / Accepted: 12 December 2017 / Published online: 21 December 2017
© Springer International Publishing AG, part of Springer Nature 2017

Abstract
There is an overwhelming number of news articles published every day around the globe. Following the evolution of a news story is a difficult task given that there is no such mechanism available to track back in time to discover and study the hidden relationships between relevant events in digital news feeds. The techniques developed so far to extract meaningful information from a massive corpus rely on similarity search, which results in a myopic loopback to the same topic without providing the needed insights to hypothesize the origin of a story that may be completely different than the news today. In this paper, we present an algorithm that mines historical news data to detect the origin of an event, segments the timeline into disjoint groups of coherent news articles, and outlines the most important documents in a timeline with a soft probability to provide a better understanding of the evolution of a story. Qualitative and quantitative evaluations of our framework demonstrate that our algorithm discovers statistically significant and meaningful stories in reasonable time. Additionally, a relevant case study on a set of news articles demonstrates that the generated output of the algorithm holds the promise to aid prediction of future entities (e.g., actors) in a story.

Keywords Entity evolution · Connecting the dots · Storytelling · Entity prediction

1 Introduction

The pervasiveness of the Internet has greatly facilitated editors to publish a large number of news articles everyday in any topic, practically without any restrictions of page-limits. Thus, the number of articles available to avid readers increases every day at a very fast pace. Many business and government organizations track these news articles to study public sentiment, business directions, progression of events, and many other aspects of economic and sociopolitical issues. However, tracking topics from different perspectives is a difficult task given that one news story of today’s interest could have evolved from another story of the past, thus forming a chain of relevant events. This leads to the concept of diffusion theory [4] which refers to the change of the distributional patterns of a phenomenon over time.

While capturing such diffusions over a timeline is still a challenge, researchers have targeted the problem of tracking stories in different guises, e.g., storytelling [15,21], storyboarding [25], connecting the dots [16,33], and metro maps [34]. Most of these methods find underlying connections between articles using a similarity-based network of documents. Their objective is mainly focused on generating a cohesive thread of a story. An extensive usage of cohesion may result in stories that revolve around a single chapter of the evolving thread of the story.

Figure 1a illustrates a scenario generated using our diffusion-based approach, which captures the articles published in the past reflecting the evolution of the story. The chain starts with an article that describes the increase in drone strikes from the U.S. against Al Qaeda. The next article describes a decrease in this type of strikes. The chain continues with an article about the possible use of drones in Africa to fight Al Qaeda and other extremist groups such as ISIS.
followed by an article that discusses ISIS recruiting. Finally, the chain ends with an article that describes the use of drones in the fight against ISIS.

Figure 1b presents the results obtained, for the same seed document, using a similarity-based approach to track past documents similar to the article of interest. While our approach brings out the underlying diffusion of concepts and their progression over time, the similarity-based approach provides a story that is highly specific to a single event—a drone strike in Kunduz. Unlike the similarity-based model, our approach is able to include dissimilar documents from the distant past which add critical context to the document of interest over time. An additional limitation of the similarity-based approach is that it has a tendency to track back only to the recent past because similar documents to the current event of interest become much more rare as their historical time span is increased.

This paper presents a novel framework that takes as input a large corpus of documents, along with a user-specified set of seed documents. The framework runs through a palette of natural language processing (NLP) and other preprocessor-

\[\text{Evolution} \quad \text{Diffusion and similarity} \quad \text{Prediction} \quad \text{Evaluation}\]

\(\text{Related work}\)

Several problems and tasks related to our work have been well studied. These problems include summarization, event detection, event extraction, event evolution, event prediction and storytelling, also known as connecting the dots. Summarization consists of extracting the most important sentences from a text corpus to help the user understand the main idea of the set of documents [3,6,28,30,39].

Event detection consists of detecting temporal bursts of highly correlated documents [19,40,41], while event extraction aims at obtaining useful knowledge regarding these events [22,32]. Event evolution [38,42] and topic evolution [17,18,36] study how these abstract concepts (events and topics) evolve over time, based on latent relations within the corpus and temporal information encoded in each document.

The event prediction through text mining problem has also been addressed in the literature. Radinsky et al. [31] described a system that uses causality extraction to obtain pairs of terms that have a causal relation and that are later used to train a prediction algorithm. Luo et al. [24] also address this problem by introducing the concept of semantic uncertainty, which is used to estimate the most certain next state based on the current state, or event. This approach is particularly useful when there is limited historical data available.

The main component of our framework is focused on the problem of storytelling, or connecting the dots. There are
several approaches that aim to solve this problem for different types of datasets, such as scientific articles [16], entity networks [8,9,14], image collections [13,35] and document collections [1,11,12,21,27,33,34,38,45]. Most of the work on this problem uses graph-based representations of a document or entity set [8,11,15,27,45]. Storytelling has several different applications, such as intelligence analysis [14,15], news recommendation [12], search engines [9,45], and social network analysis [8,29,35].

Solutions to the connecting the dots problem include the use of probabilistic approaches, such as random walks [12,33,34,38,45], Monte Carlo simulations [1] and determinantal point processes (DPP) [11]. Shahaf et al. [33] introduced the concept of coherence and coverage to assess the quality of a chain limited by two user-specified endpoints, and then extended this work by building metro maps [34] which are formed by several coherent chains that intersect at some points, forming a map-like structure. Gillenwater et al. also obtain several coherent chains using a DPP-based model.

Our work has several differences with respect to the previously mentioned methods. First, we use an optimization-based approach similar to the one presented by Shahaf et al. [33], but we introduce the concept of diffusion to identify the evolution of the concepts in the seed documents. The use of diffusion allows our framework to create stories based on a combination of different events, which is not possible using previously proposed methods. Second, our work discovers the most relevant documents associated with each turning point to provide a concise summary for each turning point event. Third, we leverage the topic distribution of each document to reduce our search space while keeping important documents for our optimization. Finally, we suggest the possibility of using the resulting relevant documents to build a regression framework for entity prediction.

3 Problem description

This work focuses on news articles and the entities within the news. The entities detected are persons, organizations, and locations. Let $D = \{d_1, d_2, \ldots, d_\|D\|$} be the set of documents and $E = \{e_1, e_2, \ldots, e_\|E\|$} be the set of entities in the news corpus. Each document $d \in D$ contains a set of entities $E_d \subseteq E$. Additionally, each document $d$ has a publication date $t_d$. As part of the preprocessing steps, we apply topic modeling using Latent Dirichlet Allocation (LDA) [7] on $D$, obtaining a topic distribution $T_d$ for each document $d$.

3.1 What is the expected outcome?

Let the user input be a set of seed documents $D_{seed} \subseteq D$, where $\|D_{seed}\| \geq 1$. We define a turning point in the story $\tau \in T$ as a specific date in which the story under analysis has a significant change. Let $S$ be a vector where each element, defined as segment, is a pair of consecutive turning points such that $|S| = |T| - 1$, and $S = [(\tau_1, \tau_2), (\tau_2, \tau_3), \ldots, (\tau_{|T|-1}, \tau_T)]$.

Our main goal is to split the set of documents in $|S|$ segments by finding $|T| - 2$ turning points, since the first and last turning points are fixed. For each segment $s \in S$ we want to find a subset of documents $D^{s} \subseteq D$ that help the user study the evolution of the prominent entities in $D_{seed}$ through various different events. The value of $|S|$ can be chosen by the user or set to different values automatically to obtain the optimal result.

4 Methodology

Our framework comprises two main stages: (1) preprocessing, which creates document and topic models from a text corpus and (2) story generation, which takes a set of seed documents and other constraints as inputs and, via an optimization routine, outputs a story formed by relevant documents that have a common thread, but that belong to different events. Figure 2 illustrates this framework.

4.1 Preprocessing

In the preprocessing stage, our framework: (1) extracts entities (e.g., person, location, organization) from the documents, (2) represents each document as a vector of entities, and (3) obtains a topic distribution for each document in the corpus. We extract entities from the text corpus using standard
Named Entity Recognizers [2,10]. Our framework leverages a tf-idf model with cosine normalization [15] to generate weights \( w(e, d) \) for each entity \( e \in E \) in each document \( d \in D \).

4.2 Story generation

In this stage, our framework furnishes a candidate set of documents \( D_c \subset D \) from the entire dataset that satisfy some temporal and topical constraints with respect to the seed documents \( D_{seed} \). The temporal and topical constraints are driven by user input regarding how far back in time the algorithm should track to detect an origin, or, how much deviation the algorithm should allow in terms of topic distribution as compared with the seed set.

The temporal criteria establishes that all of the candidate documents must have been published before the most recent seed document and within a certain maximum threshold \( t_{max} \), i.e., \( d_i \) can only be part of \( D_c \) if \( 0 < \min(t_{seed}) - t_i < t_{max} \), where \( t_i \) is the publication date of document \( d_i \) and \( \min(t_{seed}) \) is the publication date of the oldest article.

The topical criteria expresses, in terms of topic distributions, how much deviation the candidate documents can have from the seed documents when optimizing for a diffusion and cohesion objective. For each document \( d_{seed} \in D_{seed} \), we compute the KL-divergence [20], or topical divergence, between the topic distribution of the seed document \( T_{d_{seed}} \) and the \( T_d \) of all the documents \( d \in D \). Document \( d \) is included in \( D_c \), only if \( \forall d_{seed} \in D_{seed} (\text{KL-divergence}(T_d, T_{d_{seed}}) \leq \alpha) \), where \( T_d \) is the topic distribution of document \( D \), obtained using the LDA [7] and \( \alpha \) is a user-defined parameter. In Sect. 5, we will refer to \( D_c \) as \( D \), for brevity.

Finally, the framework optimizes fitment of a story model that identifies turning points of the story over a timeline and provides relevant documents within each segment. The outcome of the optimization routine helps understand the evolution of the news story described in the seed document(s). This essential step is further described in Sect. 5.

5 Formalizing our approach

In this section, we gradually introduce the concepts and formulations that give shape to our model, which has the goal of obtaining a coherent story from separate events. We present our ideas iteratively and explain why and how new penalty terms were developed to achieve our goal.

5.1 Defining incoherence

The objective is to ensure that the documents within each time segment of \( S \) are coherent. A simplistic approach is to minimize the distance of all pairs of documents within each time segment. For simplicity, we use the notation \( d^s \), to indicate the segment \( s \) to which document \( d \) belongs to, i.e., \( d^1 \) is defined as a document in the segment between \( \tau_1 \) and \( \tau_2 \). We define this distance as incoherence:

\[
incoherence_1(s) = \frac{\sum_{i \neq j} ||d^s_i - d^s_j||_\rho}{|P^s|} \quad (1)
\]

where \( S \) is the set of segments, \( d^s \) is a document s.t. \( d^s \in D^s \), where \( D^s \) is the set of candidate documents that are within segment \( s \). \( P^s \) is a set of pairs \((i, j)\) that represent the different 2-permutations of the indices of pairs of documents \( d^s_i \) and \( d^s_j \), computed using the following equation.

\[
|P^s| = \frac{|D^s|!}{2(|D^s| - 2)!} = \frac{|D^s|2 - |D^s|}{2} \quad (2)
\]

5.2 Defining a distance metric and forming an initial objective function

In Eq. 1, the distance computation between the weight vectors of two documents is represented as \( || \cdot ||_\rho \), where \( \rho \) is substituted by the distance measure chosen. We prefer the Soergel distance between a pair of documents \( d_i \) and \( d_j \), defined in Eq. 3.

\[
soergel(d_i, d_j) = \frac{\sum_{e \in E} |w(d_i, e) - w(d_j, e)|}{\sum_{e \in E} \max (w(d_i, e), w(d_j, e))} \quad (3)
\]

where \( E \) is the set of all the entities that appear in the dataset, and \( w(d, e) \) is the tf-idf weight of entity \( e \) for document \( d \).

The initial objective function to minimize is thus defined as Eq. 4, where \( s \) represents the current segment.

\[
\mathcal{F}_1(T) = \sum_{s=1}^{|S|} \text{incoherence}_1(s) \quad (4)
\]

This initial objective function is not enough to serve the purpose of capturing evolution in terms of diffusion. Equation 4 tends to generate one very large segment and many small ones, which is undesirable. Ideally, two documents that were published at distant dates should not belong to the same segment. Another limitation of the approach in Eq. 4 is that it uses hard assignments of documents to a single segment, while we would prefer a soft membership of documents to time segments for better analytic flexibility.

5.3 Including penalty based on time segmentation

To tackle the problem of generating small segments, we include a new term in our definition of incoherence. The
new term helps by avoiding segments containing document pairs with distant publication dates. Therefore, a solution with many small segments or a solution with a very large segment will not be the optimal. This term is simply a date difference penalty \( \text{date}_\Delta(1) \).

\[
\text{date}_\Delta(i, j) = |t_i - t_j|
\]  

(5)

where \( t_i \) is the publication date of document \( d_i \). Although Eq. 5 provides date differences, in practice, we normalize the values to a smaller range and the user can flexibly modify the normalization range. With this new term, we can redefine incoherence as Eq. 6.

\[
\text{incoherence}_2(s) = \frac{\sum_{i < j} \text{soergel}(d_i, d_j) * \text{date}_\Delta(i, j)}{|P^s|}
\]  

(6)

The objective function has the same form of Eq. 4, but uses the new incoherence equation. The advantage of this new formulation is that it reduces the incidence of very small segments significantly. However, the current objective function is not capable of capturing the essence of diffusion or turning point because it only takes into account coherence of each segment. This can result in very similar segments that do not show the evolution of a story. Another disadvantage is that a document can only be assigned to a single segment, and thus only discrete optimization routines can be used.

### 5.4 Introducing unconnectedness

While we prefer similar documents within a segment, we expect high dissimilarity between the documents of different segments. We call this property of diffusion across segments **unconnectedness**. We expect high unconnectedness as expressed by Eq. 7.

\[
\text{unconnectedness}(s) = \frac{\sum_{i=1}^{\left| D^s \right|} \sum_{j=1}^{\left| G^s \right|} \text{soergel}(d_i, g_j)}{|D^s| * |G^s|}
\]  

(7)

where \( G^s = D - D^s \) is the set of documents that are not within segment \( s \), and \( g^s \in G^s \). Combining unconnectedness with the original objective function, we obtain the following equation.

\[
\mathcal{F}_2(T) = \sum_{s=1}^{\left| S \right|} (\text{incoherence}_2(s) - \text{unconnectedness}(s))
\]  

(8)

In practice, we normalized and rescaled these two terms to adjust the importance of minimizing incoherence and maximizing unconnectedness. However, a major problem with this formulation is that because of the conflicting objectives, we cannot easily define which penalty should have the largest effect.

### 5.5 Changing to segment similarity

To eliminate the conflicting objectives issue, and to obtain a smooth formulation, we introduce the concept of similarity between segments. This penalty, formalized as Eq. 9, will measure how similar documents from one segment are with respect to the documents in the other segments.

\[
\text{similarity}_1(s) = \frac{\sum_{i=1}^{\left| D^s \right|} \sum_{j=1}^{\left| G^s \right|} e^{-\text{soergel}(d_i, g_j)}}{|D^s| * |G^s|}
\]  

(9)

We seek to minimize similarity \( s \) for each segment \( s \). The updated objective function is:

\[
\mathcal{F}_3(T) = \sum_{s=1}^{\left| S \right|} (\text{incoherence}_2(s) * \text{similarity}_1(s))
\]  

(10)

Eq. 10 allows minimization of incoherence and similarity simultaneously without keeping track of the importance of each other since they are represented as factors. However, this formulation still suffers from two issues: (1) a news article can only belong to a single segment, which means only discrete optimization routines can be used, and (2) it is possible to have several articles that are irrelevant to the story within each segment. We address these issues in the following subsections.

### 5.6 From discrete to continuous

The objective function of Eq. 10 considers discrete assignments of documents to time segments. This results in lesser flexibility in terms of smoothness and use of optimization routines. To indicate the certainty that a document with timestamp \( t \) belongs to a continuous range of time, in our case a segment, where \( t^L \) is the lower limit or turning point and \( t^H \) is the upper turning point, a membership function can be defined as:

\[
\gamma(t, t^L, t^H) = \begin{cases} 
\frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(t-t^L)^2}{2\sigma^2}} & \text{if } t \leq t^L \\
\frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(t-t^H)^2}{2\sigma^2}} & \text{if } t^L < t < t^H \\
\frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(t-t^H)^2}{2\sigma^2}} & \text{if } t^H \leq t 
\end{cases}
\]  

(11)

Although Eq. 11 is partitioned, it is continuous and smooth because the membership function is formed by two halves of
different Gaussian distributions, and a linear section that connects both, as shown in Fig. 3. The means of the distributions are the lower and higher limits, and the standard deviation of both is selected by the user to modify the shape of the membership function. This score is not a probability, but we can obtain the probability that timestamp \( t \) belongs to segment \( s \) by dividing the score over the sum of the membership values of \( t \) for all segments in \( S \).

With this new concept of membership function, we must update our previous definitions.

\[
incoherence_3(s) = \frac{\sum_{i,j}^{D|D|} \Phi * \text{soergel}(d_i, d_j) * \text{date}_\Delta(i, j)}{\sum_{i,j}^{D|D|} \Phi} \tag{12}
\]

where

\[
\Phi = \gamma(t_i, t_L^i, t_H^i) * \gamma(t_j, t_L^j, t_H^j)
\]

Segment similarity can be redefined as:

\[
similarity_2(s) = \frac{\sum_{i,j}^{D|D|} \phi * e^{-\text{soergel}(d_i, d_j)}}{\sum_{i,j}^{D|D|} \phi} \tag{14}
\]

where

\[
\phi = \gamma(t_i, t_L^i, t_H^i) * (1 - \gamma(t_j, t_L^j, t_H^j))
\]

The updated objective function has the same form of Eq. 10, but uses the new incoherence and similarity equations. The main advantage of this new formulation is that it is suitable to be used with a larger variety of optimization routines. However, a new problem is introduced with this objective function, which is that it tends to overlap the turning points. Furthermore, we still have the problem of many documents being irrelevant to a segment.

\[
\text{overlap} = \left(1 + \sum_{m,n,m<n} e^{-\frac{(t_m-t_n)^2}{2\sigma^2}}\right) \tag{16}
\]

where \( t_m \) and \( t_n \) are the publication dates of two different documents, and \( \sigma \) is a hyperparameter that allows the user to set a threshold defining the degree of the allowed overlap.

The objective function with the overlap penalty is:

\[
F_4(T) = \sum_i^{S} (\text{incoherence}_3(s) * \text{similarity}_2(s)) * \text{overlap} \tag{17}
\]

5.7 Introducing segment overlap penalty

To avoid overlap of segments (as shown in Fig. 4), we include a penalty that is high when two turning points are very close to each other. It is defined as:

\[
incoherence_4(s) = \frac{\sum_{i,j}^{D|D|} w_i * w_j * \Phi * \text{soergel}(d_i, d_j) * \text{date}_\Delta(i, j)}{\sum_{i,j}^{D|D|} w_i * w_j * \Phi} \tag{18}
\]

\[
similarity_3(s) = \frac{\sum_{i,j}^{D|D|} w_i * w_j * \phi * e^{-\text{soergel}(d_i, d_j)}}{\sum_{i,j}^{D|D|} w_i * w_j * \phi} \tag{19}
\]

5.8 Adding weights as a relevance factor

It is difficult for a user to analyze a massive amount of candidate documents given only the time segments. Our framework mitigates this problem by providing a score with each document representing how important it is for this study. This score is a variable for the optimization routine and it is used in the objective function. The variable is represented as a vector \( \mathcal{W} \) of length equal to the number of candidate documents. A weight of 1.0 indicates that the document is relevant and 0.0 indicates that it is not.

The updated definitions are Eqs. 18 and 19, where \( w_i \in \mathcal{W} \) is the weight for document \( d_i \).
The new objective function has the same form of Eq. 17, but uses the new incoherence and similarity formulations and has $\mathcal{W}$ as input. However, a new problem emerges—the optimal value is when all the weights are set to zero.

### 5.9 Uniformity penalty

To ensure that only a subset of $\mathcal{D}$ is truly relevant, and to avoid the degenerate case where the optimal value is when all the weights are set to zero, we need to minimize the uniformity of the weight vector $\mathcal{W}$. A completely uniform vector $\mathcal{W}$, i.e., all zeros or all ones, will result in a high penalty, while a completely non-uniform weight vector will result in no penalty. The uniformity penalty is:

$$
\text{uniformity} = \left( 1 + \sum_{s=1}^{\lvert \mathcal{S} \rvert} \left( 1 - \frac{\left( \sum_{w \in \mathcal{W}_s} \Gamma_w \right) \sqrt{\lvert \mathcal{W}_s \rvert} - 1}{\sqrt{\lvert \mathcal{W}_s \rvert} - 1} \right) \right)
$$

where $\Gamma_w$ is a vector of values returned by the membership function (Eq. 11 for the documents that fall in segment $s$).

### 5.10 The final objective function

The final objective function is shown below.

$$
\mathcal{F}_3(T, \mathcal{W}) = \sum_{s=1}^{\lvert \mathcal{S} \rvert} (\text{incoherence}_4(s) \ast \text{similarity}_3(s)) \ast \text{overlap} \ast \text{uniformity}
$$

The objective function is minimized for two vectors: $\mathcal{S}$ and $\mathcal{W}$. The elements $s \in \mathcal{S}$ are bounded between $[0, \text{date}_{\text{max}}]$ where $\text{date}_{\text{max}}$ is the value representing the publication date of the most recent news article while 0 represents the earliest document on the candidate dataset. The elements $w \in \mathcal{W}$ are bounded between $[0, 1]$, as stated earlier. We used the quasi-newton limited memory algorithm for bound constrained optimization (L-BFGS-B) [44] to minimize our objective function.

### 6 Experimental results

For our experiments, we use a corpus of 400,842 New York Times articles published between January 2000 and June 2016 on the U.S. and World news sections. The corpus contains 3,320,886 unique entities. To the best of our knowledge, there are no publicly available labeled benchmark datasets that can be used to perform a supervised evaluation of our framework. Our evaluations are based on a combination of statistics- and human participant-based studies. We seek to answer the following questions.

1. How well does our approach reflect the evolution of a story? (Sect. 6.1)
2. How does our method compare quantitatively to other methods such as clustering and similarity-based storytelling? (Sect. 6.2)
3. How does our method compare to a sophisticated storytelling algorithm? (Sect. 6.3)
4. How do the various proposed regularizations impact story quality? (Sect. 6.4)
5. How does the statistical significance for the position of the turning points change while modifying the different hyperparameters of our optimization? (Sect. 6.5)
6. How does the dispersion coefficient change while modifying the number of segments $\lvert \mathcal{S} \rvert$? (Sect. 6.6)
7. What is the repeatability of the optimization? (Sect. 6.7)
8. Can we use the set of highly relevant documents with respect to a seed document to predict which entities are expected to appear in a future document? (Sect. 6.8)

#### 6.1 Evaluation of chain continuity

The lack of a benchmark dataset makes it difficult to evaluate the performance of our optimization framework. To address this issue, we designed a user study motivated by the work of Shahaf et al. [33]. We picked four different topics and two different random documents related to each topic as seed documents. The topics selected were aviation, Brexit, ISIS, and tensions between North and South Korea. Ten evaluators were asked to score the resulting chain for each seed document, so that each topic had a total of twenty evaluations.\(^1\)

The sequence of documents presented to the evaluators consisted of one document per segment, which was selected based on the relevance weights. The number of segments $\lvert \mathcal{S} \rvert$ was set to five when running the optimization algorithm. Users evaluated four criteria on a scale of 1–5, with five being the best, for each sequence:

- **Familiarity** How familiar is the evaluator with the topic?
- **Relevance** Do the documents seem relevant to the topic?
- **Coherence** Is the chain coherent? A lower score is provided if there is a document in the chain that does not belong to the topic.
- **Broadness** Does each document provide new information guaranteeing a slight shift in topic over time?

The average scores per criterion are shown/presented in Fig. 5. The results indicate that our algorithm performs, on

\(^1\) Evaluation questions available at: [https://storyeval.herokuapp.com/](https://storyeval.herokuapp.com/).
average, above a score of 3.5, which indicates that the chains are coherent, relevant, and broad enough to capture evolution. An interesting observation is that the Brexit topic, which had the lowest average familiarity score, also has the lowest scores for coherence and relevance, and the topic with the highest average familiarity score Korea, had the highest scores for coherence and relevance, indicating that such a study with human participants might be biased toward familiarity with the topic.

6.2 Quantitative comparison with other methods

We quantitatively compare the outcomes of our framework with those of several other methods:
- similarity-based approach
- k-means clustering
- agglomerative clustering (using average, complete and Ward linkage)
- spectral clustering (with Gaussian kernel (RBF) and nearest neighbors affinity)

To evaluate our method, we selected the most relevant document per time segment. For the similarity-based approach, the consecutive nearest neighbors were selected as the chain starting with the seed document. For all of the clustering-based experiments, we added time as a feature. For the k-means clustering-based approach, we selected the document closest to the centroid of each cluster as a relevant document. For the rest of the clustering algorithms, we selected the document in the middle of the cluster (in terms of publication date). Notice that these alternative approaches focus on distance or similarity, while our objective function is designed to capture diffusion. Hossain et al. introduced the concept of dispersion coefficient in evaluating the Storytelling algorithm [15]. We use the same concept to evaluate the quality of a chain of documents \( \{d_0, d_1, \ldots, d_{n-1}\} \), containing \( n \) articles.

\[
\psi = 1 - \frac{1}{n-2} \sum_{i=0}^{n-3} \sum_{j=i+2}^{n-1} \text{disp}(d_i, d_j) \tag{22}
\]

where

\[
\text{disp}(d_i, d_j) = \begin{cases} 
\frac{1}{n+i-j} & \text{if } \text{soergel}(d_i, d_j) < \theta \\
0 & \text{otherwise}
\end{cases}
\tag{23}
\]

The dispersion coefficient of a chain of documents is 1.0 only if consecutive pairs of documents meet a distance threshold \( \theta \). The coefficient \( \psi \) is 0 when every pair of documents in the chain satisfies the distance threshold.

We generated chains for 32 different seed documents using all seven approaches under consideration. Figure 6 compares the average dispersion coefficient versus the distance threshold. It shows that the average dispersion of our diffusion-based method provides the highest dispersion coefficients for any distance threshold. This indicates that our method generates chains with smooth transition of topics, which is one of our goals.

6.3 Comparison with metro maps [34]

There are several algorithms and frameworks that attempt to solve the connecting the dots problem, as described in Sect. 2. It is difficult to perform a quantitative, or in many cases qualitative, comparison of these methods with our algorithm because most of these frameworks are not publicly available or have a restricted dataset. Moreover, the objectives, scopes, and context of the algorithms vary widely. The SNAP library from Stanford University [23] provides a demo of the metro maps framework introduced by Shahaf et al. [34]. We conduct a comparative study between the results of metro maps and our framework.

Metro maps chains are pre-generated. To generate chains under the same topic, we searched our database for similar news articles published around the same dates. These docu-
ments were used as seeds for our diffusion-based framework. We selected the document with the largest weight per segment.

We obtained two different chains of documents for two different queries: Syria and Iran. Figure 7a shows the results for the topic Syria. The row with prefix A was created with metro maps and the second row prefixed by B shows the result of our framework. Fig. 7b depicts the results of the query Iran, where prefix C is used to identify documents in the metro maps chain and prefix D for the documents in the chain generated by our framework.

Figure 7c shows the actual output from metro maps for the Iran query. As can be seen, the format of the output from metro maps differs greatly from the output of our framework, with many different chains that intersect at several points, but not a clear definition of segments or turning points. This, along with the differences between the datasets used, makes a comparison between both methods almost impossible. It is unclear if the metro maps approach can handle longer time frames, since the available demo only uses news articles from the prior 3 months. However, looking at Fig. 7c, it seems that extending the time frame would significantly increase the number of articles presented, making it harder for the user to obtain useful information. As opposed to metro maps, our approach is capable of building stories from articles published in any arbitrary time frame.

6.3.1 Evaluation

Twelve evaluators were asked to score the resulting pair of chains for each topic. The chains did not have any information on which method was used to obtain them. The order in which the chains were presented to the evaluator was randomized across topics. Users compared each pair of chains and evaluated four criteria:

- **Relevance** Which chain is more relevant to the topic?
- **Coherence** Which chain is more coherent?
- **Broadness** Which chain is more broad?
- **Number of coherent and relevant documents** How many documents from each chain form a coherent and relevant story?

For the first three criteria, each evaluator chose one of five options in terms of each criteria: one chain is better than the other (two options), one chain is slightly better than the other (two options), and both chains are similar. The fourth criterion was evaluated for each chain.

Table 1 presents the user study results for the first three criteria: coherence, relevance and broadness. In this case, the results for each topic present a different picture of the quality of the stories generated by metro maps and our approach. Based on these results, we can observe that there is a trade-off between coherence and broadness, i.e., the more coherent a chain appears to the evaluator, the less broad it seems, since the topics do not seem to diverge significantly. Table 2 presents the percentage of users that selected a particular number of documents that are relevant and coherent w.r.t. the generated story.

![Fig. 7](image-url)
6.3.2 Analysis of results for Syria topic

For the Syria topic, the evaluators preferred the story generated by metro maps in terms of coherence with 66.7% of the users scoring this chain as slightly or more coherent, while the rest is split between a same level of coherence and our approach being slightly better. In terms of relevance, 66.7% of the evaluators selected that both chains have the same level of relevance, while the rest is split between a same level of coherence and our users scoring this chain as slightly or more coherent, while the rest is split between a same level of coherence and our users scoring this chain as slightly or more coherent.

To understand this findings, we performed an in-depth analysis of the contents of each article in both chains. For the metro maps approach, documents A1, A2 and A3 are very similar in that the main theme is the number of casualties of the conflict, depicting its various manifestations and evolution. The first three documents are very similar, but they revolve around very similar events that happened in a very short time span, which is what we are trying to avoid.

In contrast, the chain generated by our approach starts with document B1, which discusses allegations of torture against the Syrian police. Regular acts of violence by the government against civilians are considered the main reason the war started. Document B2 describes the state of the civil war at its initial stage, as well as a controversy related to the then head of the Arab League observer mission in Syria. The next document, B3, discusses the flow of refugees from Syria to Jordan, which started in the late months of 2011, once the Syrian civil war was in full swing. Document B4 describes once again actions of the government against the rebels. This article was published the same day that the Syrian government was accused of using chemical weapons. Finally, document B5 discusses the prospect of a presidential election in Syria.

The chain of documents generated by our approach presents a larger picture that starts with a possible cause of the conflict, depicts its various manifestations and evolution, and ends in the possibility of having a presidential election that could end the conflict. However, it received low scores in the user evaluation. We suspect this is the result of the users not having sufficient background knowledge of the factors and events relevant to the Syrian civil war. If the users devoted more time to fully analyze all of the provided evidentiary documents or possessed deeper contextual understanding of the Syrian civil war, then our approach would have likely obtained higher scores. Table 2 shows the percentage of scores that each chain received. The chain generated by metro maps received an average score of 4.1, while our approach had an average score of 3.0. This supports our assertion that documents B1 and B4 might not seem relevant or coherent with the story at first glance, but after further analysis we consider our story to have more value in terms of insight.

6.3.3 Analysis of results for Iran topic

For the Iran topic, the evaluators preferred the story generated by our approach in terms of coherence with 83.3% of the users scoring our chain as slightly or more coherent, while the rest is split between a same level of coherence and

| Criterion    | Topic | Metro maps is better (%) | Metro maps is slightly better (%) | Both are similar (%) | Our approach is slightly better (%) | Our approach is better (%) |
|--------------|-------|--------------------------|-----------------------------------|----------------------|-------------------------------------|---------------------------|
| Coherence    | Syria | 50.0                     | 16.7                              | 25.0                 | 8.3                                 | 0.0                       |
|              | Iran  | 0.0                      | 8.3                               | 8.3                  | 33.3                                | 50.0                      |
| Relevance    | Syria | 16.7                     | 16.7                              | 66.7                 | 0.0                                 | 0.0                       |
|              | Iran  | 0.0                      | 25.0                              | 41.7                 | 25.0                                | 8.3                       |
| Broadness    | Syria | 0.0                      | 0.0                               | 41.7                 | 16.7                                | 41.7                      |
|              | Iran  | 41.7                     | 33.3                              | 25.0                 | 0.0                                 | 0.0                       |

| Topic | Score | Syria Metro maps (%) | Our approach (%) | Iran Metro maps (%) | Our approach (%) |
|-------|-------|----------------------|------------------|---------------------|------------------|
|       | 0     | 0.0                  | 0.0              | 0.0                 | 0.0              |
|       | 1     | 0.0                  | 0.0              | 0.0                 | 0.0              |
|       | 2     | 0.0                  | 41.7             | 16.7                | 0.0              |
|       | 3     | 33.3                 | 33.3             | 66.7                | 16.7             |
|       | 4     | 25.0                 | 8.3              | 16.7                | 33.3             |
|       | 5     | 41.7                 | 16.7             | 0.0                 | 50.0             |
the metro maps approach being slightly better. In terms of relevance, 41.7% of the evaluators selected that both chains have the same level of relevance, and in this case the rest of the answers are spread among the two methods. The metro maps chain was picked as slightly or more broad by 75% of the evaluators, while the rest considered both chains to have the same broadness. However, in terms of coherent and relevant documents, the chain generated by metro maps received an average score of 3.0, while our approach had an average score of 4.3. We consider these results as evidence that our approach can provide more insightful stories.

In this case, metro maps documents C1 and C2 discuss missile drills. Document C3 describes a presidential visit to improve the relationship between Islamic countries. Document C4 discusses the willingness of the Iranian President to talk with the Western leaders about their nuclear program. Finally, document C5 is about an increase in the price of gasoline in Iran probably caused by the sanctions imposed because of their nuclear program. We believe that this chain is not very coherent and thus the chain was scored as very broad.

The results from our diffusion-based method start with document D1 describing that Russia was considering to condemn Iran’s nuclear plan. Document D2 talks about a nuclear treaty between the United States and Russia, but also mentions the assistance that the Russian government provided to develop Iran’s nuclear program. The next article (D3) mentions that the negotiations over the nuclear program were set to resume, only to be stopped again less than a year later (D4), which resulted in several sanctions that the U.S. decided to enforce (D5).

6.4 Use case study

In this experiment, we perform a use case study to compare the results of our diffusion-based approach with two preliminary versions of our objective function, as well as with the similarity-based approach described in Sect. 6.2. We execute each of these methods for the same seed document and select the document in the middle of each segment. The number of segments is set to 5. Figure 8a shows the story generated using the formulation given in Sect. 5.2. In this case, because the date penalty was not included, one event overlaps with the other, and thus only four segments are obtained. The topics of each document are about the European Union, but lack a common thread. Figure 8b shows a story generated by optimizing the objective function given in Sect. 5.5, which is in the continuous space and includes the date and similarity penalties. An analysis of this story shows that there is no coherent thread, which is caused by the lack of weights that indicate the relevance of a document.

Figure 8c shows the story resulting from our diffusion-based approach. At first glance, some of the articles seem disconnected from a continuous thread. However, after a brief analysis, we can find a very relevant thread: the first article is about the financial assistance required by Greece and Portugal because of a severe crisis. The second article mentions how Germany made efforts to keep Greece in the Euro zone, because of fear of the consequences of Greece leaving. The next article mentions the future plans of the German chancellor with respect to the European Union and the imposed austerity, and in particular mentions that Britain’s prime minister expected the support from Ms. Merkel. The article before the seed document mentions the latest European crisis, and how several countries were “rebelling” against the German-mandated austerity. The seed document talks about immigration being one of the main motivations for Brexit. The article mentions that there was an influx of immigrants on 2015, presumably escaping from the economical crisis in their countries of origin. This is a coherent and relevant thread consisting of different events, which was our ultimate goal.

The story obtained using the similarity-based approach, shown in Fig. 8d, is also coherent and relevant. However, the story completely revolves around Brexit, and a simple keyword search about the topic would probably return similar results. Thus, we consider that our result provides a better insight into the possible paths that lead to the event described in the seed document.

6.5 Statistical significance analysis

Evaluation using precision, recall, and other common metrics does not apply in the context of our work because of the lack of labeled datasets suitable for our task. An alter-
native approach, other than the user studies and the use of unsupervised metrics such as the dispersion coefficient, is to perform a statistical analysis of the significance ($p$ value) as a sanity check to make sure that our formulation obtains results that have a very low probability of being obtained by random chance. Our objective function has several user-modifiable parameters that can change analytic viewpoints. In this section, we will explore how the significance changes with varying parameters.

For our problem, we defined $p_{TP}$ as the $p$ value for the turning points vector $T$. We generated $m$ random samples $T_S$ which are of the same size as $T$. Then, we compared each of the $m$ random samples with $T$ by computing their element-wise difference and comparing it to a tolerance $\beta$, i.e., $(T_S - T) \leq \beta$. We counted the number of times this condition was satisfied in an auxiliary variable $a$. Finally, we returned $p_{TP} = \frac{a}{m}$.

The experiments were conducted by computing the $p_{TP}$ of several different configurations. We precomputed 100,000 random samples. Then, we used a number of random seed document sets and ran the optimization routine for different configurations. We finally averaged the $p$ values across the documents.

One of the configurations relates to the publication dates of the articles. The dates are scaled to a continuous range. In Fig. 9a, it is evident that the $p_{TP}$ value significantly decreases while increasing the maximum date value. This is because the range for the turning points increases with the scaling. Figure 9a also shows that our approach generates statistically significant results with any value of the variance for the gamma function $\hat{\sigma}^2$, and that the change in significance while varying this parameter is limited.

The standard deviation parameter for overlap $\sigma$ controls the width of the bell-shaped curve shown in Fig. 4. A penalty is added if any of the previous turning points fall within this curve. Figure 9c shows that as the overlap $\sigma$ increases, the $p_{TP}$ value increases. The maximum topical divergence parameter is used during the candidate generation stage to filter documents that are above this value. Figure 9b shows that when the topical divergence increases, the $p_{TP}$ value decreases, albeit slightly. In general, our approach generates statistically significant results even with varying parameters.

### 6.6 Dispersion coefficient versus number of segments

For the analysis described in Sect. 6.5, the number of segments (or turning points) was also modified to observe the effect on the $p_{TP}$ value. However, increasing the number of segments to more than three resulted in almost all cases giving a significance value of zero, which means not a single configuration matched with the 100,000 random samples. Thus, to evaluate how our algorithm performs while varying the number of segments $|S|$, we performed a similar evaluation but measuring the dispersion coefficient detailed in Sect. 6.2.

The threshold distance $\theta$ was varied from 0.95 to 0.99 in 0.01 increments. We obtained the dispersion coefficient for these values of $\theta$ as the average for 18 different seed documents, and only using news articles from a window of 5 years from publication. Figure 10 presents the average dispersion coefficient per number of segments $|S|$. The figure indicates that an increase in the number of segments improves the dispersion coefficient, which indicates the existence of a high number of turning points, as expected in a 5 year window.

### 6.7 Local optimization: repeatability of concepts

One of the crucial elements of local optimization, as used in our framework, is that it may produce multiple results with different executions for the same set of seed documents and configurations. From an analytic perspective, multiple results for the same seed set provide a deeper understanding of the evolution. However, too much diversity in the results for the same seed set may be overwhelming for users. In this experiment, we analyze the repeatability of the results,

---

**Fig. 9** Significance of turning points vector versus (a) variance of gamma function, (b) topical divergence and (c) date overlap penalty. Each data series includes a predicted trend line as well as upper and lower 95% confidence lines.
in particular for the turning points, for the same set of seed sets. We selected random seed documents from different topics and repeated the optimization 100 times for each seed keeping track of the resulting turning point vector, each time with a different initialization for the optimization routine. The collected vectors were compared pairwise and grouped into buckets based on similarity. To define similarity in this context, we first define a match as when the distance between two turning points from different vectors is below a threshold, \( \zeta \). Two vectors are similar if the number of matches is above a minimum match parameter. Ideally, the number of buckets should be small (close to 1). If the number of buckets is 100, this means that none of the pairs of vectors fulfill the conditions mentioned above.

In Fig. 11, we observe how the average of the number of buckets for the seed documents changes with respect to the distance threshold \( \zeta \) and a minimum matches score. \( \zeta \) is varied from one to 100, since this is the range of the publication dates in this experiment, while the number of minimum matches is changed from one to four. Figure 11 shows that as the distance threshold increases, the number of buckets rapidly decreases. This indicates that, even with different initializations of the optimization routine with multiple executions for the same seed, we obtain similar chains and time segments. This helps the user by keeping the analysis focused on a relevant set of evolving concepts for the same seed document.

### 6.8 Prediction

Our framework provides past documents as an evolution of the story of a given seed document. All the past documents have associated weights reflecting their importance in the evolution. These relevant documents can be used to study the evolution over time of the prominent entities found in the seed document. In this experiment, we examine if we can leverage these relevant documents to not only study the evolution of the entities in the past, but also to predict, using a simple linear regression, the future evolution. A more specific question is: which entities will appear in an article published in the future given a seed document?

For this experiment, we divided the set of relevant documents based on the segments into a training dataset (four segments) and a testing dataset (one segment). Next, we generated a table that consists of all pairs of words in all pairs of the documents of the training set. This table contains the difference in publication dates, as well as a pair of entities and corresponding tf-idf weights that appear in a pair of documents. That is, the training data reflect how far in time a pair of entities may appear and what are the tf-idf scores of those entities when they appear. We built a linear regression model for each of the terms appearing in the future documents, taking into account the difference in publication dates as an extra feature.

When a new prediction is requested, we use the linear regression models that include the entities of the seed document, to predict the weight of the entities in the future document. In Fig. 12a, we show the word-cloud of a sample seed document. We then use our algorithm to predict how the word-clouds of three future documents would look like in a range between 4 and 10 days after the publication date of the original article. The actual and predicted word-clouds are shown in Fig. 12b. As can be seen, even with a simple linear regression-based prediction model, for most of the cases the predicted documents have a match of at least two of the entities from the ground truth (e.g., Paris attack, Belgium, and Paris). The advantage of using the output of our model for prediction is that we exploit both the similarity of documents within a segment, and the diffusion of a topic across segments, to obtain better results.

We expect that a more sophisticated algorithm would be able to improve these results. The main disadvantage of this method is that if no entities from the seed document appear in the training set, then no entities can be predicted, and of course, unseen entities will not be predicted.
Fig. 12 An experiment on predicting a future document. (a) A word-cloud of a seed document. (b) On the left, we show the ground truth while on the right we show the predicted word-clouds. The words are sized according to the original and predicted weights, respectively. The publication date difference between the seed document and the predicted documents ranges between 4 and 10 days.

7 Online approach

The very large number of news articles published every day means that a static (or offline) model will be out-of-date almost the moment it is uploaded for public usage. This is unacceptable for large-scale time-sensitive applications such as national-security-related data analysis, as well as systems developed for commercial data analytics solutions. Thus, in this section we describe the possible implementation of an online version of our algorithm that can take as input a data stream, such as a real-time news feed. We discuss two approaches, which differ on when the optimization is performed:

- On-request optimization (Sect. 7.2).
- Online optimization (Sect. 7.3).

Both of these approaches require an online preprocessing step, which is described in Sect. 7.1. The online approach will leverage streaming versions of several machine learning and data mining algorithms, which are implemented by distributed frameworks such as Apache Hadoop [5] and Apache Spark [43]. Using a distributed architecture enhances the performance of the framework by utilizing all of the computational resources available.

7.1 Online preprocessing

During the online preprocessing stage our framework will: (1) extract entities (e.g., person, location, organization) from the documents using the Spark version of the Stanford Named Entity Recognizer [10,37], (2) represent each document as a vector of entities using the streaming tf-idf implementation in Apache Spark’s MLlib [26], and (3) obtain a topic distribution for each document in our corpus using the online algorithm provided by MLlib [26] for running Latent Dirichlet Analysis (LDA) incrementally. This step will be performed every time a new document is added to the corpus, but due to the streaming nature of the algorithms used, as well as the distributed frameworks utilized, the impact in terms of execution time should be minimal, compared to performing this step with the complete corpus every time.

7.2 On-request optimization

This is the most direct approach to obtain a relevant news chain, but it requires a longer user wait-time. In this case, we will only perform the optimization when a user wants to obtain the set of turning points and relevant documents for a particular seed document. The main disadvantage of this method is that each optimization requires a significant amount of time, and this will only increase as the corpus grows at a fast rate.

7.3 Online optimization

The online optimization approach will have two phases: (1) offline phase, and (2) online phase. The offline phase will consist of obtaining the set of turning points and relevant documents for all of the documents that already exist in a corpus (or a subset of it, based on date constraints). Two linked lists will be kept for each document so that only the most relevant documents are stored, as well as the turning points. The offline phase will be performed only once for each of the documents in the corpus, and it can be performed periodically (e.g., every week) to obtain the linked lists of the latest documents.

The online phase will consist of finding a small number of closest candidates that are topically similar to the seed document, and relatively close in time (i.e., the top-5). The precomputed chain for each candidate will be used as the set of new relevant documents and turning points. We will obtain
the objective value using Eq. 21 adding the seed document to the chain as: (1) part of the current segment, or (2) the start of a new segment, which means placing a turning point between the candidate document and the seed document. The chain that gives the best result will be stored as the optimal set of relevant documents and turning points for the seed document.

The main advantage of this approach is that it will significantly reduce the time required to obtain an output. Furthermore, the number of segments will increase naturally. However, this approach has two main disadvantages. First, it will require high memory usage since all of the previous results must be stored. Second, the results will probably have a lower quality than if running the complete optimization every time it is needed.

8 Conclusions and future work

Our framework discovers the evolution of today’s news stories from large news archives. We have presented case studies that demonstrate the possibility of using the resulting evolution-related documents as training data to predict the relevant entities for future news articles. Based on our case studies, we plan to expand our work in the direction of prediction where our framework will be able to generate warnings regarding insurgent activities, predict a timeline for developing a new concept, and forecast emerging stories. Furthermore, we plan to implement and compare both of the streaming-based approaches presented in this paper, as well as incorporate user feedback to improve the performance of our framework by modifying the optimization hyperparameters to meet the user’s expectations.

Compliance with ethical standards

Conflict of Interest Statement On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Ahmed, A., Ho, Q., Eisenstein, J., Xing, E., Smola, A.J., Teo, C.H.: Unified analysis of streaming news. In: WWW ’11, pp. 267–276. ACM, New York (2011)
2. Alias-i: LingPipe 4.1.0 (2008). http://alias-i.com/lingpipe/. Accessed 20 Sept 2016
3. Allan, J., Gupta, R., Kandelwal, V.: Temporal summaries of new topics. In: SIGIR ’01, pp. 10–18. ACM, New York (2001)
4. Angulo, J.J., Pedemirnas, C.A., Ehner, W., Kimura, E.M., Magale, P.: Concepts of diffusion theory and a graphic approach to the description of the epidemic flow of contagious disease. Public Health Rep. 95(5), 478–485 (1980)
5. Apache Software Foundation: Hadoop. https://hadoop.apache.org. Accessed 28 July 2017
6. Binh Tran, G.: Structured summarization for news events. In: WWW ’13 Companion, pp. 343–348. ACM, New York (2013)
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
8. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: KDD ’04, pp. 118–127. ACM, New York (2004)
9. Fang, L., Sarma, A.D., Yu, C., Bohannon, P.: REX: explaining relationships between entity pairs. Proc. VLDB Endow. 5(3), 241–252 (2011)
10. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: ACL ’05, pp. 363–370. ACL, Stroudsburg (2005)
11. Gillenwater, J., Kulesza, A., Taskar, B.: Discovering diverse and salient threads in document collections. In: EMNLP-CoNLL ’12, pp. 710–720. ACL, Stroudsburg (2012)
12. Gu, W., Dong, S., Chen, M.: Personalized news recommendation based on articles chain building. Neural Comput Appl 27(5), 1263–1272 (2016)
13. Heath, K., Gelfand, N., Ovsjanikov, M., Anjaneyya, M., Guibas, L.J.: Image webs: computing and exploiting connectivity in image collections. In: CVPR ’10, pp. 3432–3439 (2010)
14. Hossain, M.S., Andrews, C., Ramakrishnan, N., North, C.: Helping intelligence analysts make connections. In: AAAI’11, pp. 22–31. AAAI Press, Menlo Park (2011)
15. Hossain, M.S., Butler, P., Boedihardjo, A.P., Ramakrishnan, N.: Storytelling in entity networks to support intelligence analysts. In: KDD ’12, pp. 1375–1383. ACM, New York (2012)
16. Hossain, M.S., Gresock, J., Edmonds, Y., Helm, R., Potts, M., Ramakrishnan, N.: Connecting the dots between PubMed abstracts. PloS ONE 7(1), e29509 (2012)
17. Jo, Y., Hopcroft, J.E., Lagoze, C.: The web of topics: discovering the topology of topic evolution in a corpus. In: WWW ’11, pp. 257–266. ACM, New York (2011)
18. Kim, D., Oh, A.: Topic chains for understanding a news corpus. In: CICling’11, pp. 163–176. Springer, Berlin (2011)
19. Kleinberg, J.: Bursty and hierarchical structure in streams. In: KDD ’02, pp. 91–101. ACM, New York (2002)
20. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. 22(1), 79–86 (1951)
21. Kumar, D., Ramakrishnan, N., Helm, R.F., Potts, M.: Algorithms for storytelling. IEEE Trans. Knowl. Data Eng. 20(6), 736–751 (2008)
22. Kuzey, E., Vreeken, J., Weikum, G.: A fresh look on knowledge bases: distilling named events from news. In: CIKM ’14, pp. 1689–1698. ACM, New York (2014)
23. Leskovcec, J., Sosić, R.: Snap: a general-purpose network analysis and graph-mining library. ACM Trans. Intell. Syst. Technol. (TIST) 8(1), 1 (2016)
24. Luo, X., Xuan, J., Lu, J., Zhang, G.: Measuring the semantic uncertainty of news events for event potential estimation. ACM Trans. Inf. Syst. 34(4), 24:1–24:25 (2016)
25. Kader, M. A., Naim, S. M., Boedihardjo, A. P., Hossain, M. S.: Connecting the dots using contextual information hidden in text and images. In: AAAI Conference on Artificial Intelligence (2016)
26. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M.J., Zadeh, R., Zaharia, M., Talwalkar, A.: Mllib: machine learning in apache spark. J. Mach. Learn. Res. 17(1), 1235–1241 (2016)
27. Nallapati, R., Feng, A., Peng, F., Allan, J.: Event threading within news topics. In: CIKM ’04, pp. 446–453. ACM, New York (2004)
28. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 117–147. Springer, New York (2012)
29. Ning, Y., Muthiah, S., Tandon, R., Ramakrishnan, N.: Uncovering news-twitter reciprocity via interaction patterns. In: ASONAM ’15, pp. 1–8. ACM, New York (2015)
30. Radev, D., Otterbacher, J., Winkel, A., Blair-Goldensohn, S.: NewsInEssence: summarizing online news topics. Commun. ACM 48(10), 95–98 (2005)
31. Radinsky, K., Davidovich, S., Markovitch, S.: Learning causality for news events prediction. In: WWW ’12, pp. 909–918. ACM, New York (2012)
32. Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Alstäbe, I., Rigau, G., Soroa, A., Ploeger, T., Bogaard, T.: Building event-centric knowledge graphs from news. Web Semant. 37(C), 132–151 (2016)
33. Shahaf, D., Guestrin, C.: Connecting the dots between news articles. In: KDD ’10, pp. 623–632. ACM, New York (2010)
34. Shahaf, D., Guestrin, C., Horvitz, E., Leskovec, J.: Information cartography. Commun. ACM 58(11), 62–73 (2015)
35. Suen, C., Huang, S., Eksombatchai, C., Sosic, R., Leskovec, J.: NIFTY: a system for large scale information flow tracking and clustering. In: WWW ’13, pp. 1237–1248. ACM, New York (2013)
36. Wang, X., Zhai, C., Roth, D.: Understanding evolution of research themes: a probabilistic generative model for citations. In: KDD ’13, pp. 1115–1123. ACM, New York (2013)
37. Warcbase: Named Entity Recognition (2016). https://lin tool.github.io/warcbase-docs/Spark-Named-Entity-Recognition/. Accessed 20 Sept 2016
38. Wu, C., Wu, B., Wang, B.: Event evolution model based on random walk model with hot topic extraction. In: Li, J., Li, X., Wang, S., Li, J., Sheng, Q.Z. (eds.) Proceedings of advanced data mining and applications: 12th international conference, ADMA 2016, Gold Coast, QLD, Australia, 12–15 Dec 2016, pp. 591–603. Springer, Cham. https://doi.org/10.1007/978-3-319-49586-6_42 (2016)
39. Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., Zhang, Y.: Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In: SIGIR ’11, pp. 745–754. ACM, New York (2011)
40. Yang, Y., Ault, T., Pierce, T., Lattimer, C.W.: Improving text categorization methods for event tracking. In: SIGIR ’00, pp. 65–72. ACM, New York (2000)
41. Yang, Y., Carbonell, J.G., Brown, R.D., Pierce, T., Archibald, B.T., Liu, X.: Learning approaches for detecting and tracking news events. IEEE Intell. Syst. Appl. 14(4), 32–43 (1999)
42. Yu, S., Li, X., Zhao, X., Zhang, Z., Wu, F.: Tracking news article evolution by dense subgraph learning. Neurocomputing 168(C), 1076–1084 (2015)
43. Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., Ghodsi, A., Gonzalez, J., Shenker, S., Stoica, I.: Apache spark: a unified engine for big data processing. Commun. ACM 59(11), 56–65 (2016). https://doi.org/10.1145/2934664
44. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound-constrained optimization. ACM Trans. Math. Softw. 23(4), 550–560 (1997)
45. Zhu, X., Oates, T.: Finding story chains in newswire articles using random walks. Inf. Syst. Front. 16(5), 753–769 (2014)