Computational Visual Media
https://doi.org/10.1007/s41095-020-0178-4 Vol. 6, No. 3, September 2020, 333–347
Research Article

Visual exploration of Internet news via sentiment score and topic models

Songye Han, Shaojie Ye, and Hongxin Zhang[1] (✉)

Abstract Analyzing and understanding Internet news are important for many applications, such as market sentiment investigation and crisis management. However, it is challenging for users to interpret a massive amount of unstructured text, to dig out its accurate meaning, and to spot noteworthy news events. To overcome these challenges, we propose a novel visualization-driven approach for analyzing news text. We first collect Internet news from different sources and encode sentences into a vector representation suitable for input to a neural network, which calculates a sentiment score, to help detect news event patterns. A subsequent interactive visualization framework allows the user to explore the development of and relationships between Internet news topics. In addition, a method for detecting news events enables users and domain experts to interactively explore the correlations between market sentiment, topic distribution, and event patterns. We use this framework to provide a web-based interactive visualization system. We demonstrate the applicability and effectiveness of our proposed system using case studies involving blockchain news.

Keywords Internet news visualization; sentiment score; topic models; event detection

1 Introduction

Available information is growing ever more rapidly, and there is a particular requirement to interpret and extract key information from large quantities of Internet news data. This has many applications, such as market sentiment investigation and crisis management. Event detection, as well as other visualization tools, are effective in helping users to understand news. Moreover, grasping general trends of sentiment in Internet news requires exploration through techniques such as pattern analysis.

In our work, we utilize neural network models to extract latent semantic information from Internet news. Previous work like Opinionflow [1] has considered extracting public opinion from social media and interpreting its development and patterns. Many factors affect stock market sentiment, for example, so automated extraction of semantic information from multiple media source is important. Therefore, our work aims to compute a sentiment score that summarises meaningful information in Internet news; we use it as a basis for event detection and pattern analysis. We also provide a novel interactive visualization framework to help user such as investors and domain experts to better understand changes in sentiment from news. With this tool, they can explore patterns in sentiment polarity and better predict trends.

In general, finding useful news indicators is difficult and also task-based. In our analysis, we use news semantics as a key indicator to detect events of interest in news streams. Specifically, we utilize the state-of-the-art *bidirectional encoder representations from transformers* (BERT) [2] to encode text for input to a *long short term memory* (LSTM) network [3] for sentiment classification. Our case studies show that using a sentiment score in this way can efficiently detect news events.

In particular, we show that our proposed sentiment score can effectively reflect general market sentiment, and the the LSTM network has suitable

∗ Songye Han and Shaojie Ye contributed equally to this work.

1 Zhejiang University, Hangzhou, China. E-mail: S. Han, 3160102019@zju.edu.cn; S. Ye, JaYE@zju.edu.cn; H. Zhang, zhx@cad.zju.edu.cn (✉).

Manuscript received: 2020-03-14; accepted: 2020-04-30

characteristics for extracting long-term sentiment. However, directly interpreting and visualizing the output of deep learning models is still challenging. Instead, we introduce topic models to help better capture semantic features in Internet news. An online learning version of *latent Dirichlet allocation* (LDA) [4] is applied to bring out the multi-dimensional semantic topics behind the corpora.

Our main contributions are as follows:

- a novel method computing a news sentiment score, suitable as a key indicator for event detection;
- a visualization framework for exploring Internet news from multiple perspectives. It integrates topic modeling visualization, pattern analysis, and sentiment line graphs to help users comprehensively understand text semantics;
- a system combining sentiment score and topic models to fully reveal sentiment surrounding news events.

2 Related work

This section reviews various areas closely related to our problem of abstracting, visualizing, and exploring Internet news.

2.1 Text sentiment analysis

The rapid development of the Internet has brought about many websites. Researchers can automatically obtain overall opinions from a large number of text documents without human intervention, thanks to the development of automatic opinion mining techniques. Determining sentiment in text data is one of the main focuses of our work; various methods have been used. The *data cube* provides one approach to organizing data in a multi-dimensional way, allowing interactive and intuitive queries and exploration by slicing, dicing, and drilling through cube cells. Ref. [5] utilizes a text cube to analyze and visualize social media semantics. It visualizes heat maps to extract points of interest (POI) as hot spots. Other works focus on embedding unstructured text data as structured vectors. Zhu et al. introduce a vector embedding technique for urban location data based on situation awareness [6]. They emphasize the continuity of trajectory data and regard trajectories as sequential data, like words in a sentence. Topic models are widely used for extracting semantics from unstructured data such as urban trajectories [7] and news text [8].

Visualization of opinions extracted from unstructured texts can be done in three ways: at document level, at feature level, and a combination of both. Feature-level visualization concentrates on details: e.g., Ref. [9] proposes a method to extract customer opinions, using augmented bar charts to facilitate visual comparison of extracted feature-level data. Oelke et al. [10] introduce visual summary reports, cluster analysis, and circular correlation maps to facilitate visual analysis of customer feedback data at the feature level. Document-level visualization focuses on visualizing opinion data at the document level. Morinaga et al. [11] suggest a 2D scatter plot called a *positioning map* which groups positive or negative sentences. Chen et al. [12] present a visual analysis system with multiple coordinated views to help users understand the nature and dynamics of conflicting opinions. OpinionSeer [13] combines these two perspectives, allowing analysis of relationships between multiple data dimensions and the comparison of opinions of different groups.

2.2 Media event detection

Visual analysis of news events in Internet media has gained increasing attention from industry. Our case studies focus on blockchain news event detection, while previous studies have mainly considered social event detection. StreamExplorer [14] specialized in interactively tracking streaming social data. They used current tweet volumes as the only feature for detecting sub-events; this proved effective. Other works adopted more complex methods for detecting events. Ref. [15] utilized Bayesian location inference to reveal events, while Ref. [16] proposed a novel method using attention and an LSTM network to detect abnormal events, etc. Visual Backchannel used a similar approach to represent dynamically changing keywords in tweets, reflecting evolving topics in social media texts [17]. Ref. [18] developed the FluxFlow system for detecting and visualizing anomalous information propagation processes in Twitter. Opinionflow [1] conducted a time-oriented visual analysis to track diffusion of opinions among social media users. Our work follows the ideas of Opinionflow but focuses on a different aspect of the subject.

2.3 Blockchain pattern analysis

In our case studies, we mainly focus on news acquired from the blockchain domain. Blockchain [19] allows all network participants to reach a consensus. All data stored on a blockchain network is recorded digitally and has a common history available to all participants. This eliminates the possibilities for certain kinds of fraudulent activity without the need for a third-party. Nowadays, the blockchain industry is rapidly growing, and research on blockchain covers a wide range of areas including security, privacy, smart contracts, cryptocurrencies, and P2P broadcasting [20]. Most blockchain analysis research considers trading patterns for cryptocurrencies, e.g., Refs. [21–24]. Our analysis does not consider such direct market data, but public blockchain news.

3 Data modeling and processing

This section presents our analysis procedure and its four components, namely, a preprocessing step of text data collection, an algorithm for semantic quantization of the news corpus, an analysis tool for topic models, and an event detection method.

3.1 Data collection and feature interpretation

To avoid bias, we crawl plain text Internet news articles from multiple websites instead of one single source. Also, to illustrate the general applicability of our model, we crawl both Chinese [25] and English Internet news websites [26]. We clean the collected data by discarding useless articles containing fewer than 20 words. We also sort the text data chronologically. The main features of the items in the collected corpus are as follows:

- Time stamp: when was the article published.
- Article title: in plain text.
- Article content: plain text as several paragraphs, each being a sequence of sentences, which in turn are sequences of words.
- Reads: the number of page views for the article (used in event detection).
- Sentiment label: 0 (negative) or 1 (positive).
- Followers: number of followers of the article's author, indicating the author's influence.

3.2 Semantic quantization of news corpus

Quantizing text semantics and finding important events can greatly reduce the effort expended in tracking Internet news. In order to transform the text into a new representation easier for a neural network to process, one approach is to represent each sentence in the news corpus by a point in a vector space, such that sentences with similar meanings are close together in the vector space. As our goal is to find a computational model for sentiment score, each acquired article in the news corpus is labeled as having positive or negative sentiment for model training and testing.

Therefore, in our method, the semantic quantization of the news corpus mainly occurs into two steps. Firstly, as an upstream task, we apply the state-of-the-art method BERT [2] to embed Chinese and English sentences extracted from the news corpus into a tensor space. Secondly, as a downstream task, we represent each paragraph in the news corpus as 2-dimensional tensor and use deep learning methods to perform text sentiment classification.

3.2.1 Tensor embedding

In natural language processing tasks, one of the biggest challenges is the lack of annotated data. Thus, a variety of techniques have been developed for training general-purpose language representation models from the huge resources of unlabeled text from the Internet. Such pre-trained upstream models can be fine-tuned for various downstream natural language processing (NLP) tasks such as question answering and text semantic classification. Among those models, some are context-independent, such as word2vec [27]. However, in our framework, we apply BERT [2], which is a context-dependent model. BERT provides state-of-the-art performance for various classical NLP tasks, so it is reasonable to choose it as a key part of our semantic quantization model, to provide a firm basis for our downstream analysis.

In our framework, we do not use specific task-based variants of BERT. We only use BERT-as-a-service to obtain static representations of sentences, since it is time-consuming to fine-tune parameters in the BERT model. We divide each corpus into sentences and send each sentence to BERT-as-a-service for encoding. The result for each sentence is an $L \times 768$ matrix, where L is the sentence length; the BERT model then reduces the matrix to a 768-D vector. To unify the representation of each corpus, we count the number of sentences N in each article. News items with a

small number of sentences are padded, whereas longer ones are truncated.

3.2.2 Sentiment score

Next, we perform text semantic analysis using the tensor determined above. Since the article content is already encoded at sentence level, an ideal downstream model should focus on extracting the relationships between sentences in a corpus. Our solution is based on bidirectional LSTM (BiLSTM) [28]. The bidirectional property allows us to learn both leftward and rightward contextual information and then concatenate the leftward and rightward hidden state as $h_{\text{output}} = [h_{\text{L}}, h_{\text{R}}]$. We use a network structure with two stacked BiLSTM modules in our analysis framework for precision purposes.

Assuming the hidden layer of our LSTM network has dimensionality H, and the current paragraph has L sentences. Sentence encoding by BERT gives an $L \times 768$ tensor. We use padding and packing to unify the input data dimensionality for batch training. Applying our 2-layer stacked bidirectional LSTM gives an $L \times 2H$ tensor for each article. We then use a downstream pooling layer and full connection layer, followed by a softmax layer, to give the sentiment score. Figure 1 shows our whole downstream network.

The output of the softmax layer provides each article's sentiment label. The output of the softmax layer is the value of its largest element, which is always a positive real number, so we must multiply it by -1 for negative sentiment, and $+1$ for positive sentiment.

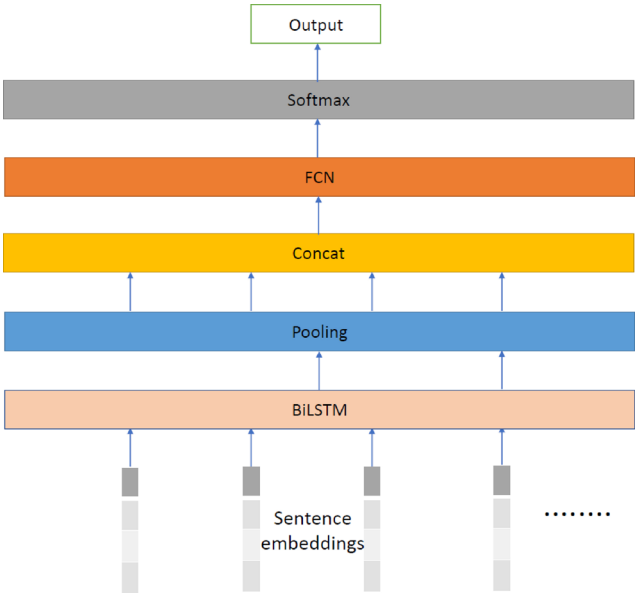


Fig. 1 Neural network overview. The input is a sequence of sentences extracted from an article, indicated by the grey columns at the bottom. The output is its sentiment score.

Finally, we apply z-score normalization to smooth the distribution of our raw sentiment and obtain the *sentiment score*: $S = (S_{\text{r}} - \mu)/\sigma$, where S_{r} is the raw sentiment value, and μ and σ denote its mean and average values.

To assess our BiLSTM-based downstream model for computing sentiment score, we also evaluated other classification models including Adaboost, a support vector machine (SVM), a random forest (RF), and a convolutional neural network (CNN). As there is no public benchmark, we labeled 2052 articles as showing positive or negative sentiment. We used accuracy and F_1 score, defined as $F_1 = 2pr/(p + r)$ where p is precision and r is recall, to evaluate performance of these models. Results are shown in Table 1. Our dataset was divided in the proportions 7:2:1 for training, validation, and testing, to optimize model hyperparameters by grid search, including learning rate, batch size, epoch, the maximum number of leaves, and regularization penalty coefficients. After doing so, we randomly split the dataset 7:3 for training and testing. The accuracy and F_1 scores given are averages over 50 such random splits.

In the CNN-based downstream model, inspired by Ref. [29], we utilized 2D convolution and multiple kernel sizes to extract local sequence information. Different kernel sizes can be viewed as different time steps, so giving a list of models generated from different time steps. We concatenated the outputs to feed them back to the full connection layer. For the CNN model, we also set the hidden layer size using a further experiment, which showed that a network with a hidden size of 2 far outperformed networks with other numbers of layers. A CNN is not suitable for dealing with sequence data as it has to pad each article to the same dimension, and so many short articles will present useless messages to the network, which probably accounts for its poor performance.

Table 1 Classification model evaluation

Model	Accuracy	F_1
CNN	0.85	0.84
SVM	0.82	0.86
Random Forest	0.85	0.90
Adaboost	0.80	0.83
BiLSTM	0.91	0.93

The experimental results show that the BiLSTM network is the best choice for classification. It is better at learning long-sequence dependence than a CNN, and it can combine packing techniques to deal with different sentence lengths in batch training.

3.3 Analyzing topic models

As discussed in Section 1, directly interpreting and visualizing deep learning models is challenging. To meet the requirement for text sentiment visualization and semantic interpretation, topic modeling is also integrated in our framework. We use LDA [4] to extract topics.

As LDA is a bag-of-words model, we have to subdivide the text into words first. We apply the Jieba module for Chinese corpora and NLTK for English corpora. We filter out stop words and also add terminology from specific domains to the dictionary, e.g., Bitcoin, Dapp, for the blockchain domain. For English corpora, we also convert all words to lowercase.

We utilize the online LDA training method based on an online variational Bayes algorithm [30], as it is more efficient than the original one. It utilizes a stochastic adaptive strategy to update the posterior parameters of the topic:

$$\lambda \leftarrow (1-\rho_t)\lambda + \rho_t \tilde{\lambda}$$

where $\tilde{\lambda}$ is calculated by mini-batch to reduce noise and ρ_t is similar to the weight used in the simulated annealing algorithm [31]:

$$\rho_t \triangleq (\tau + t)^{-\kappa}$$

In online LDA model training, the best metric for evaluating goodness of model fit might be thought to be log perplexity:

$$P(n_i, \lambda, \alpha) \triangleq \exp\left(\sum_i \log p(n_i \mid (\alpha, \beta)) / \sum_{i,w} n_{iw}\right)$$

where n_i denotes the vector of word counts for the ith document. α and β are the posterior parameters of the LDA model. However, log perplexity mainly measures the model's likelihood function. Since our data contains 70,000 corpora, there should be no overfitting; models with more topics fit better. However, such results are unhelpful as too many topics can lead to memory overload for users of our system. Instead, other metrics, such as AIC and BIC, can better help balance the simplicity and performance of the model.

Therefore, we applied LDAVIS [32], a visualization method, to investigate the feasibility of our LDA model, since our initial goal is to allow end-users to decide which model to choose intuitively. Also, topic model selection is not a clear-cut issue. A higher loss does not necessarily mean the model is suitable for the event analysis process. A visualization of the model itself, including the most relevant terms for each topic and the relative relationships between each topic, would be more acceptable for non-technical users; it can also be used to customize the process of topic model selection. This visualization technique will be further discussed in Section 4.1; a graphical interface is provided to help users managing which model they want.

3.4 News event detection

The last part of our analysis is event detection, which is also the main goal of our system. Since the definition of a news event is often vague, and there is no open dataset of labeled Internet events for us to conduct supervised learning, we provide our event detection for exploratory purposes. As for the main focus of our analysis is to use the sentiment score to extract and detect abnormal moments, we do not elaborate on our method by adding additional neural networks or sophisticated algorithms. In our blockchain tests, we use the Bollinger band [33], as it is frequently used in quantitative finance to seek trading opportunities. Later in our work, it will be shown that it is excellent for detecting events when dealing with smooth data.

Besides sentiment score, inspired by social event processing which often uses tweet volume as an essential indicator, we deploy the number of reads R of a given article to provide additional control. We provide a parameter β for users to adjust; it is used to generate a synthetic feature called *power*, P, to detect events:

$$P = \beta \mid S \mid + (1-\beta)R$$

where S and R denote the sentiment score and number of reads respectively, both scaled using z-score normalization.

In order to filter important news events, we generate the Bollinger band based on the power feature P:

$$U = A(P, n) + b\sigma(P, n)$$

$$D = A(P, n) - b\sigma(P, n)$$

where U, D denote the upper and lower boundary of the Bollinger band, $A(P, n)$ and $\sigma(P, n)$ denote

the moving average and standard deviation of n power values within the chosen time window, respectively. Coefficient b is the bandwidth, which can be customized by the user.

We provide a parameter of window size $n/2$ for users to adjust time granularity. Since number of reads and the absolute value of our sentiment score both have positive correlation with event occurrence, we only use the upper bound of the Bollinger band when choosing an event. Further discussions is presented in Section 4.2.1.

4 Visualization

This section introduces the structure and function of our visualization system and its user interface. Also, we discuss several tasks and visual analysis using the system.

4.1 Design consideration

To design the visualization framework, we held interviews and discussion sessions with several domain experts and students: one blockchain trader, one blockchain market researcher, one expert in visualization, and ten social psychology students. They were chosen for having some aspects of domain knowledge, expertise, or experience in exploring social events or blockchain market trading. We obtained feedback from our users and integrated their ideas into the visual design goals below.

- **T1. Multi-phase event exploration.** Since news events are often complicated and ever-changing, the process of event exploration should contain multiple phases.
  - **T1.1** News events often incorporate multiple sub-events. Therefore our system should allow users to track events at different time granularities and switch between different granularities easily.
  - **T1.2** Our system should highlight events it detects as suggestions and provide a timeline structure to unveil the inner order of sub-events inside an event.
- **T2. Event trend analysis.** To reflect change and explore reasons for change, our system should present the development of events comprehensively.
  - **T2.1** During different stages of events, our system should enable end-users to understand the development of events and reflect the entry point, directionality, and endpoint of an event, using the suggested points discussed above.
  - **T2.2** Moreover, during each stage of an event, change of topic composition within it should be presented, to enable users to better understand changes in sentiment during visualization.
- **T3. Event pattern analysis.** Event patterns are of interest. Our system should include a topic document map to visualize underlying patterns of sub-events within an an event. It should project suggested points in corresponding positions in a map to reveal potential event clustering patterns and relationships between sub-events within an event.

4.2 System design

As shown in Fig. 2, our visualization system consists of five parts: a sentiment line view, a tree visualization view, a wordcloud view, a topic document map, and an LDAVIS view on the LDA channel. Our visualization system is built using D3.js, which is interactive, fast, and functional. All sub-systems provide users with multi-stage exploration. Since BERT encoding is computationally expensive, we store the precomputed sentiment scores in a database and implement detached timeline visualization.

4.2.1 Timeline visualization of sentiment score

As suggested by experts and our end-users, a sentiment line should be displayed to show the sentiment score trend—see Fig. 3.

Aggregation and smoothing. The scaled sentiment score from text semantic analysis is used. Since our data contains about 70,000 texts, aggregation and smoothing methods are applied to visualize sentiment curves.

First, we aggregate our data by time granularity. We use average values as the results of aggregation; the time granularity can be chosen by the user. In the multi-phase event exploration process, our system initializes a suitable granularity for visualization; granularity is reduced by one level automatically when users drill into sub-events of an event.

Next, we smooth the aggregated data to remove noise and simplify visualization. Many smoothing methods are widely used, such as linear smoothing

[34], local polynomial smoothing, and spline methods [35]. To balance computational effort and quality, as a special case of binomial smoothing [36], we employ the simple scheme of linear combination weights:

$$w_k = \frac{C_h^k}{\sum_{i=1}^{h} C_h^i}, \quad k = 1, \ldots, h$$

where h is the window size, taken to be an odd number to make our smoothing weights symmetric. Smoothed sentiment scores are calculated using:

$$\hat{S}_k = \sum_{i=k-(h-1)/2}^{k+(h-1)/2} w_i S_i$$

where the window size is chosen by the user.

Interactive event detection. The red dots in Fig. 3(a) denote events automatically detected by our system in the chosen time range. These are highlighted on the timeline visualization to draw the user's attention. The user can click these red dots to explore the text further, in terms of sentiment as well as topic semantics around the chosen event point. The next phase is based on the point in the window of the current event point (**T1.1, T1.2**).

For instance, when users click the red dots in Fig. 3(a), the initial time window is 5, and the time granularity is one day. When users click one of the red dots in the sentiment curve view, next-stage exploration is triggered. All points within the time window of the center point are selected. The current time granularity of Fig. 3(a) is one day. So five days within the window will be selected and all the data is aggregated by hour (granularity becomes finer by one level) as shown in Fig. 3(b). New events and red dots are now determined and presented in Fig. 3(b). The user can click these event dots in turn for further exploration until no event is detected or the time granularity is at the smallest value (one minute).

4.2.2 Event tree visualization and wordcloud

We next introduce two auxiliary tools, namely an event tree and a wordcloud view, which help users to explore their interests more efficiently.

Event tree. A tree view helps users to keep track of the current timeline and their drilling into subevents of the current chosen event. Figure 2 shows our tree visualization for the event node. It is displayed when end-users click the red dots suggested by our system in the sentiment curve; the time span within the time window is used to select child nodes. Users can also click to further explore interior features of a parent node, which is divided at the next level of granularity. When this reaches one minute, an accurate time will be displayed since one minute is the finest granularity (**T2.1**). Each node holds a donut chart characterized by three features: sentiment,

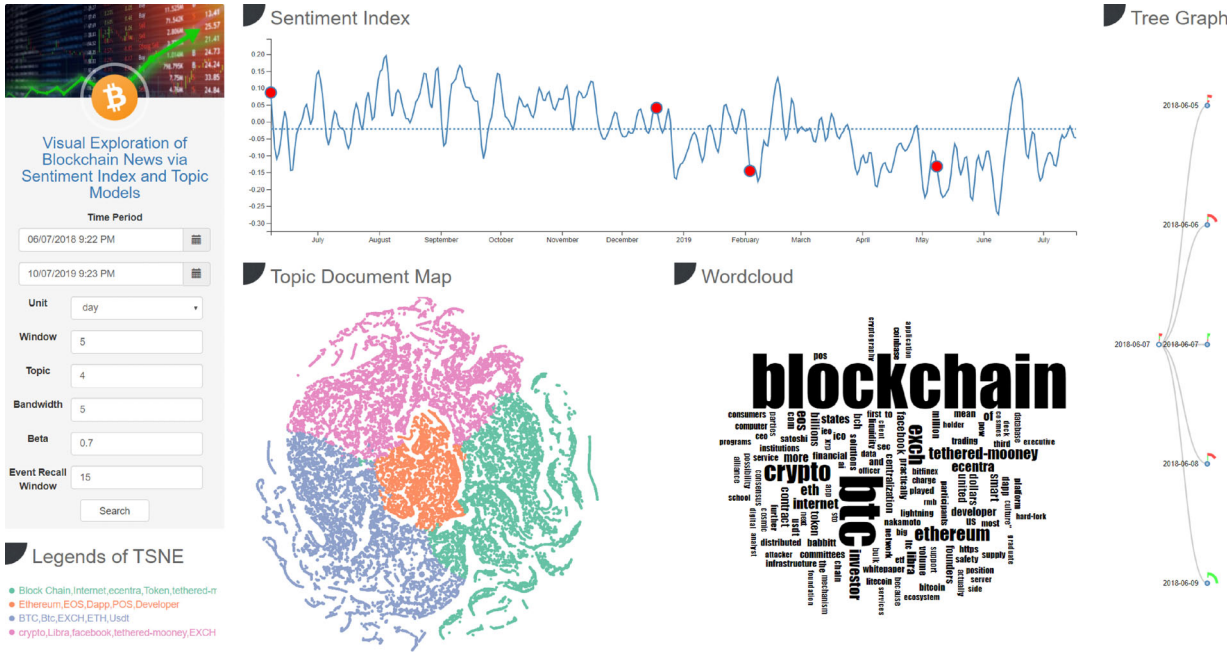


Fig. 2 Visualization system overview. Our system has five main parts: a sentiment curve view, a tree visualization view, a wordcloud view, a topic document map, and an LDAVIS view on the LDA channel.

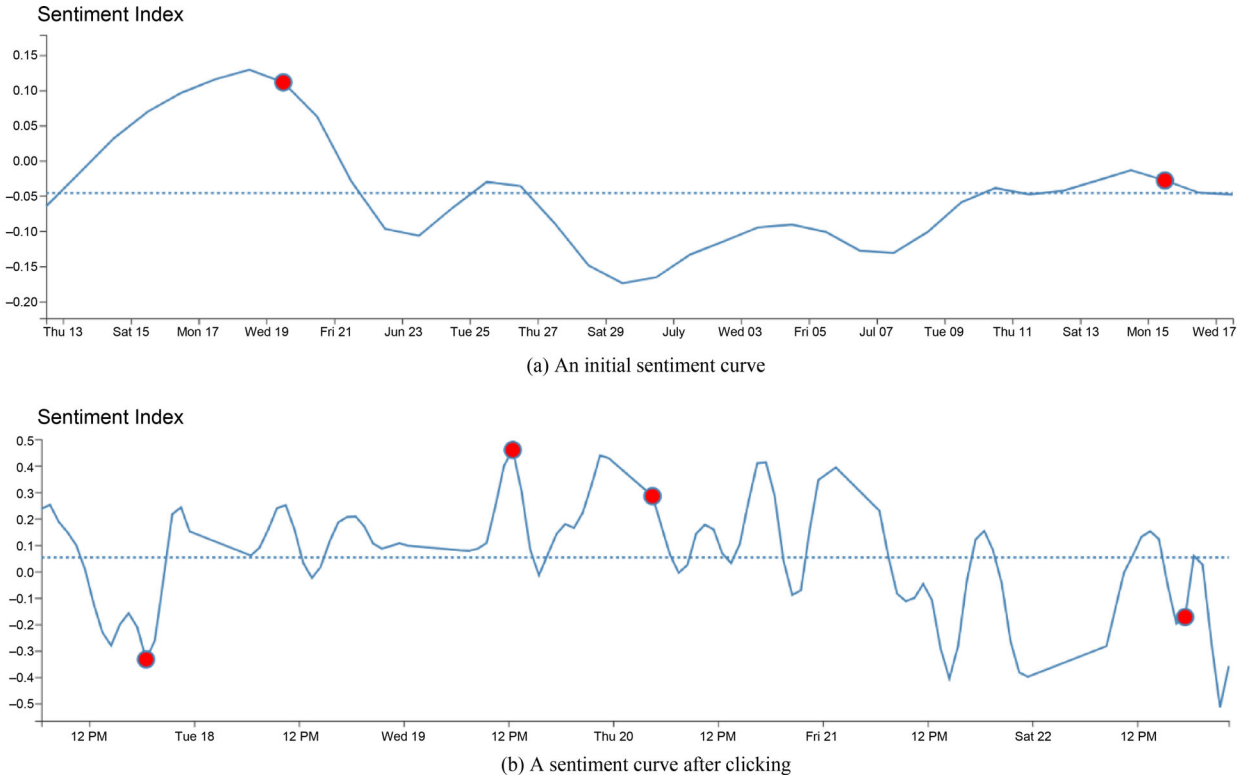


Fig. 3 Sentiment curve interaction.

number of reads, and number of author's followers. All are scaled from −1 to 1. Red and green represent positive and negative value respectively; a parent node value is computed as the average value of its children.

Wordcloud. Our system also provides users a wordcloud view to visualize word frequencies in texts within the chosen time range. It is a useful tool for text semantic analysis and extraction, as demonstrated in Fig. 2.

4.2.3 Topic model visualization

We visualize the chosen LDA model using LDAVIS [32], a web-based visualization tool. In LDAVIS, the relationships between topics and terms are evaluated by relevance:

$$r(w,k \mid \lambda) = \lambda \log(\phi_{kw}) + (1-\lambda)\log(\phi_{kw}/p_w) \quad (9)$$

where ϕ_{kw} denotes the probability of term $w \in \{1,\ldots,V\}$ for topic $k \in \{1,\ldots,K\}$, V is the total number of terms, K is the number of topics, p_w denotes the marginal probability of term w in the corpus, and λ is a free weight for users to adjust. The user can choose a topic in the left panel. In Fig. 4, the top 30 relevant terms for the chosen topic are shown in the right panel. The width of the red bar for a term denotes the term's frequency within the chosen topic, while grey gives the overall frequency of the term. For simplicity of topic interpretation, we initially set $\lambda = 1$, which makes term frequency within the topic the sole decider of relevance.

In the left panel, two main features of topics are quantified. Firstly, the size of a topic is proportional to its prevalence in all texts. Its relative position to other topics is computed by Jensen–Shannon divergence; principal component analysis is used to scale inter-topic distances.

Thus, the web-based LDAVIS is integrated into our system to provide LDA model interpretation. Users can judge the interpretability of their chosen model by examining most relevant terms for each topic, as well as their relative positions and distances. In our analysis, the feasibility of a topic model is evaluated primarily by examining the overlap among all topics and their most relevant terms.

4.2.4 Topic document map

For further data exploration and topic visualization, we provide a document visualization map. We first utilize the document-topic weights in our topic model to reduce each document to a K-dimensional vector. Then, a dimension reduction method is used to

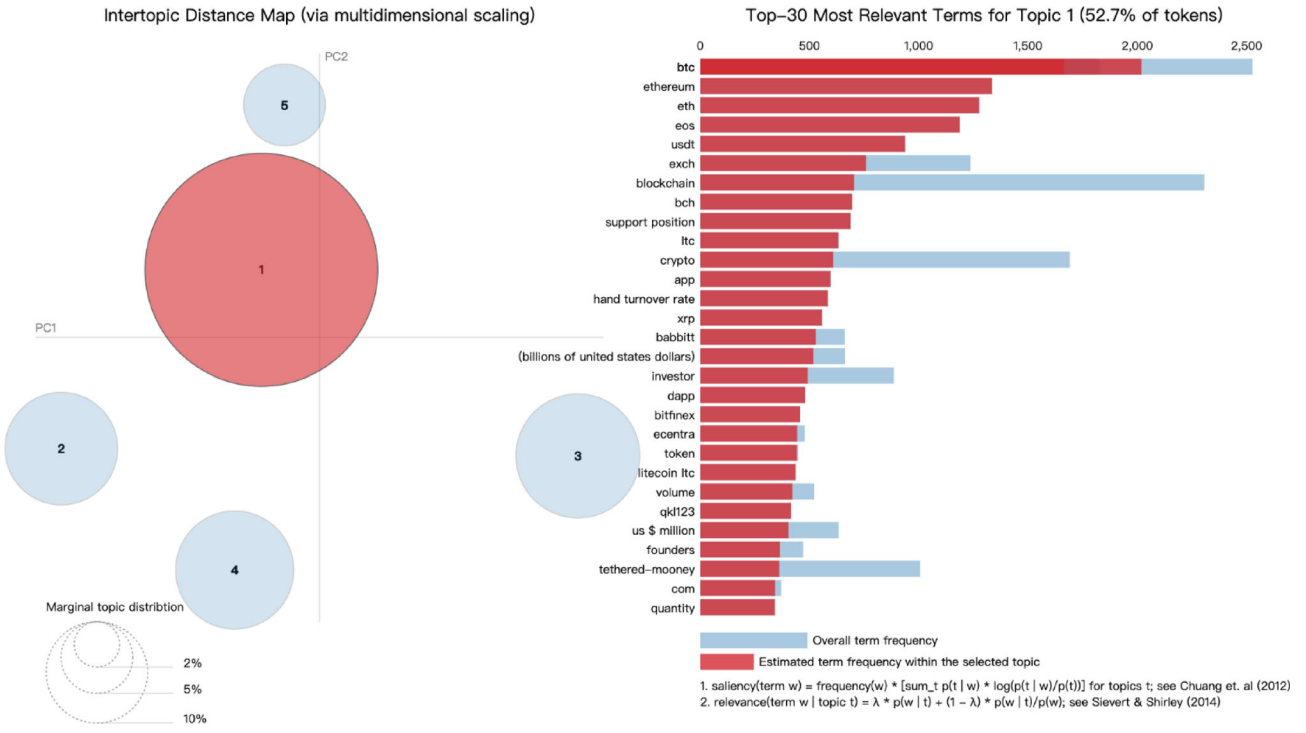


Fig. 4 Topic model visualization: selecting topics via LDAVIS.

visualize each document. We use the t-SNE algorithm [37]. It uses t-distribution to model a probability matrix in low dimensional space. It is suitable for nonlinear data or patterns, but it is computationally expensive, so it is challenging to use for interactive visualization. We thus use an offline method. We first store the LDA model (for fewer than 10 topics), and train the t-SNE model to get and store coordinates for different numbers of topics by using text-topic weight.

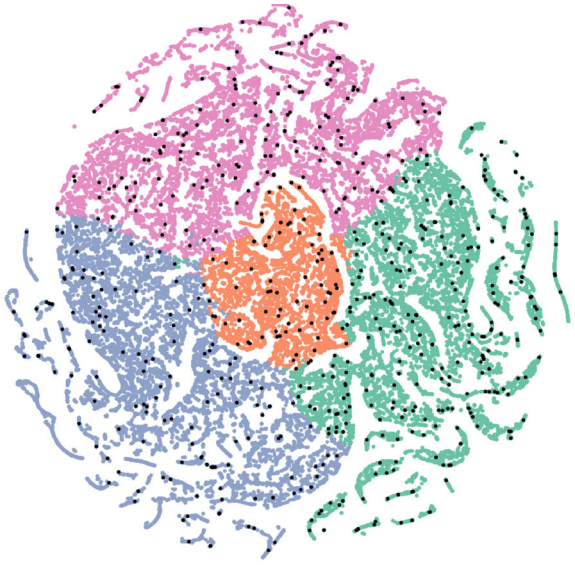
Our topic document map includes the text's data source, as shown in Fig. 5(b), which helps users to consider text directly, facilitating their research. When the mouse is hovered on a red dot in the sentiment curve view, the corresponding event points are displayed as black dots in the topic document map, as shown in Fig. 5(a). Therefore, users can track events by data source and explore topic distribution as well as event patterns by looking into the suggested event points in the topic document map (**T3**).

4.3 User interface

The five parts of our system work together interactively. First, users give a time range of interest to the system, which automatically chooses the granularity depending on the number of points within the time range given. Then, events are detected automatically and presented as red dots. Meanwhile, the topic document map visualizes all texts in 2D space, and the wordcloud visualizes the frequency of hot words in those same texts. Users can hover the mouse over the red dots, and corresponding documents within the window of the hovered red dots are labeled as black points in the topic document map.

It is worth noting that our system also provides users several control parameters. Among them, β is a weight reflecting preference between number of reads and sentiment score. The width of the Bollinger band is also a key control factor; the event recall window is the traceback window used in the Bollinger band. Users can compare the composition and interpretation of different models by LDAVIS and select a number of topics. After choosing time-related content in this way, the user should try to determine their optimal event-related content.

For further detail, users can select the red dots to enter the next phase of analysis. Our system can automatically choose the time range and granularity, and although users can override this choice after the first stage, we do not suggest doing so. After selecting a red dot, the wordcloud view synchronously updates



(a) Topic document map with news texts

July 16, Shanxi Evening News reporter from Shanxi Province Public Security Bureau was informed that: recently, Datong police cracked a large fund-raising fraud case, fraudsters under the banner of investment "blockchain" "LCC light cone" in a short period of time to defraud more than 1 million yuan. It is reported that the so-called "LCC light cone coin" is called light cone Coin. According to its promotional information, "LCC, developed by South Africa's top blockchain technology team, is a P2P electronic encrypted digital economy derived from the BTC underlying program created by Nakamoto." If you don't understand the meaning of the term too well, there is a more direct promotional language in the propaganda: "Only up and down." In view of such hype, the "LCC Light Cone" scam quickly accumulated a large amount of money.

(b) Topic document map with event indication

Fig. 5 Topic document map. (a) shows map label events as black dots in this view when users hover on red dots on the sentiment graph. (b) When users click on the dots in the scatter plot, the corresponding text is shown.

to visualize new texts, and the tree view initializes and shows the time and feature of the clicked points as a parent node and the corresponding time divided by granularity within the window as children. The topic document map also changes its visualization with the newly chosen texts (**T2.2**). Events are detected as red dots in the sentiment curve view again on entering the next-stage analysis. When at finest granularity (one minute), or no events are detected, the analysis reaches its endpoint (**T1.2**).

5 Case study

This section presents two case studies according to the goals and tasks of analysis proposed in Section 4.1. Our selected cases concern discoveries made by our end-users' using our visualization system. These typical and persuasive cases demonstrate the effectiveness of our system.

5.1 Hacking attack on Ethereum

We first present a case study of a hacker attack. One of our test users was interested in investigating the blockchain market in 2019. Therefore he choose a time range from January 2019 to the present. Since this represented hundreds of days, our system automatically set the time granularity to one day. The initial sentiment curve is shown in Fig. 6(a). The bandwidth was set large to prevent too many events being detected and causing confusion. We may notice that Fig. 6(b) has a biased topic distribution toward the pink-colored topic, which mainly represents different types of cryptocurrencies and exchanges according to the LDAVIS analysis (Fig. 6(c)). Comparing to another suggested event point on the left, the right dot is preferred because of its lower entropy.

The user stepped into the next stage of exploration by clicking red dots on the sentiment curve and entering the chosen events. Figure 7 shows sentiment development within the chosen event. We can see that newly detected subevents also label the turning point in sentiment for the whole event (**T1, T2**). The market sentiment first increases to the highest point labeled as a sub-event, and then dramatically goes down, finally fluctuating afterwards.

Hovering on the red dots in turn, we can see from Fig. 8 that the distribution is initially concentrated on the pink topic, and scatters to other topics afterwards. The pink topic is highly correlated with different types of cryptocurrencies, including Ethereum. We click each red dot to assess what happened over time. First, the sub-event located at the highest point in Fig. 7, was at around 8:00am on May 6, 2019, and the topic is mainly pink. The associated texts were excited about a prediction of a coming bull market. Therefore the sentiment at the time was abnormally high. Entering the next red dot, after the drastic decline, we can see the associated texts. Although the topics were scattered, a large portion of the news reported that the Ethereum network had been attacked by hackers in the early morning of May 8, 2019. Different reports provide different views about the Ethereum being stolen, which may explain the diversity of themes: technologies such as hard fork and POS mechanism, the effect on the Bitcoin market, and the measures taken to stop the significant losses. Moving on further, we can see the

Fig. 6 Hacker attack case: illustrating topic distribution differences between the two detected events.

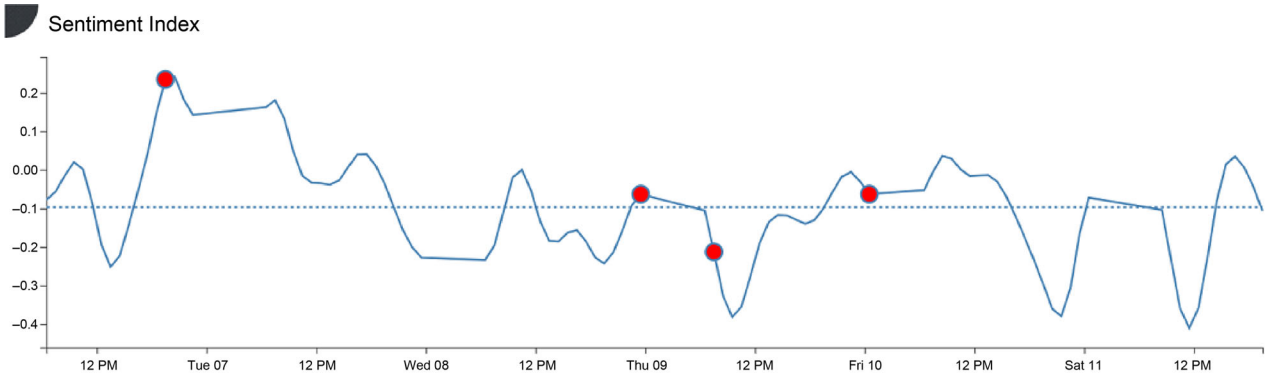


Fig. 7 Hacker attack case: phase2 sentiment curve.

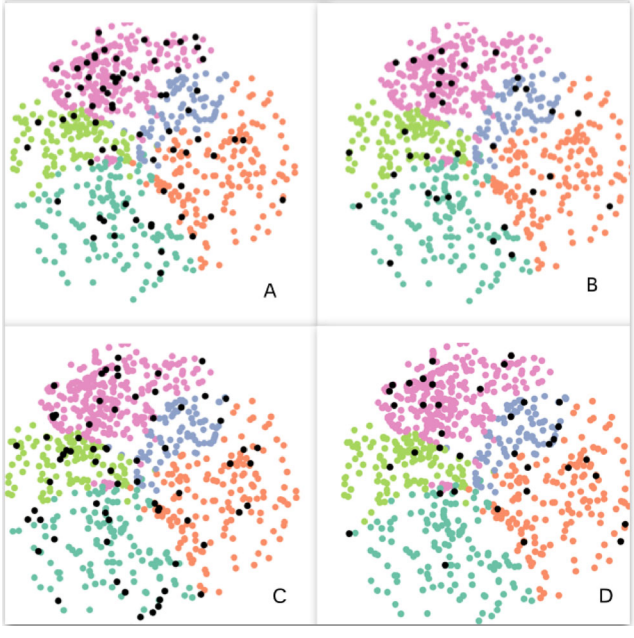


Fig. 8 Hacker attack case: topic distribution changes inside the event, and distribution shifts in alphabetical order.

focus shifting to other issues like the Bitfinex and Tether dispute, the strength of the Bitcoin market, and potential dangers of the Dex system. After May 10, 2019, few news was still focusing on the theft of Ethereum, with a more disorganized distribution of topics, which symbolizes the endpoint of the event (**T3**).

5.2 Facebook releasing Libra

This case shows the usefulness of our system in detecting new events, multi-phase extraction of an event, and trend analysis of an event. One of our users was interested in the announcement of Libra by Facebook, so a relatively short period was chosen around this event, from June 2019 to August 2019.

Figure 9(a) shows phase 1 during the chosen period, when the cyan-colored topic was more dominant than before. According to LDAVIS, the cyan-labeled topic is highly correlated with Libra-related terms. Also, the two words "Libra" and "Facebook" became more significant in the wordcloud view. The sentiment curve peaked around June 17, which may indicate the entrypoint of the event. Also an event was detected by our system around the peak. Hence the left red dot is more important for exploring the event (**T2.1, T2.2**).

Further investigating the event, phase 2 in Fig. 9(b) indicates that sentiment reached a highest point around June 17. Event-related words such as "Libra", "Facebook", and "Ethereum" dominated the wordcloud view in Fig. 9(b). Exploring the topic document map, most events were still distributed in the cyan-colored topic, which mainly consisted of Libra-related terms according to LDAVIS. Unveiling the texts around the peak, which is phase 3 as shown in Fig. 9(c), most of the news was claiming the release of Libra to be an unprecedented milestone in cryptocurrency technology and about to transform the whole industry. The cyan-related topics become more dominant in the document topic map, and as related terms in the wordcloud. The sentiment curve began fluctuating after the highest point. Clicking the red dot and viewing the news texts, it seems that the market became more uncertain about the release of Libra. Some persons still held the previous positive attitude, but other negative views became more prevalent: the government's antitrust policy against Libra, disputes about the wallet and the white paper from Facebook, and the underlying fraudulent behavior it might bring about (**T1, T3**).

After June 22, there was still quite a lot of derived dispute such as the impact on other cryptocurrencies and Paypal's cooperation with Libra. But the cyan-colored topic and Libra-related terms became less significant, tending to indicate the end of the event.

6 Discussions and future work

The results of these and other case studies show that our Internet news visualization system is of value to users and experts, allowing them to find Internet news events and explore sentiment behind news texts interactively and efficiently. However, feedback from our test users and experts also indicates that our work has some downsides.

First, since labeled news text is available, a labeled LDA model might be able to better interpret underlying topic semantics inside texts. Second, a labeled LDA model can be employed to categorize each news item by the type of currency, for example, facilitating single currency analysis.

Furthermore, our event detection algorithm provides no guarantees. Since few blockchain or other news data are labeled with events, we cannot implement supervised learning to implement model selection and parameter optimization. Also, an event itself cannot be strictly defined, which makes the

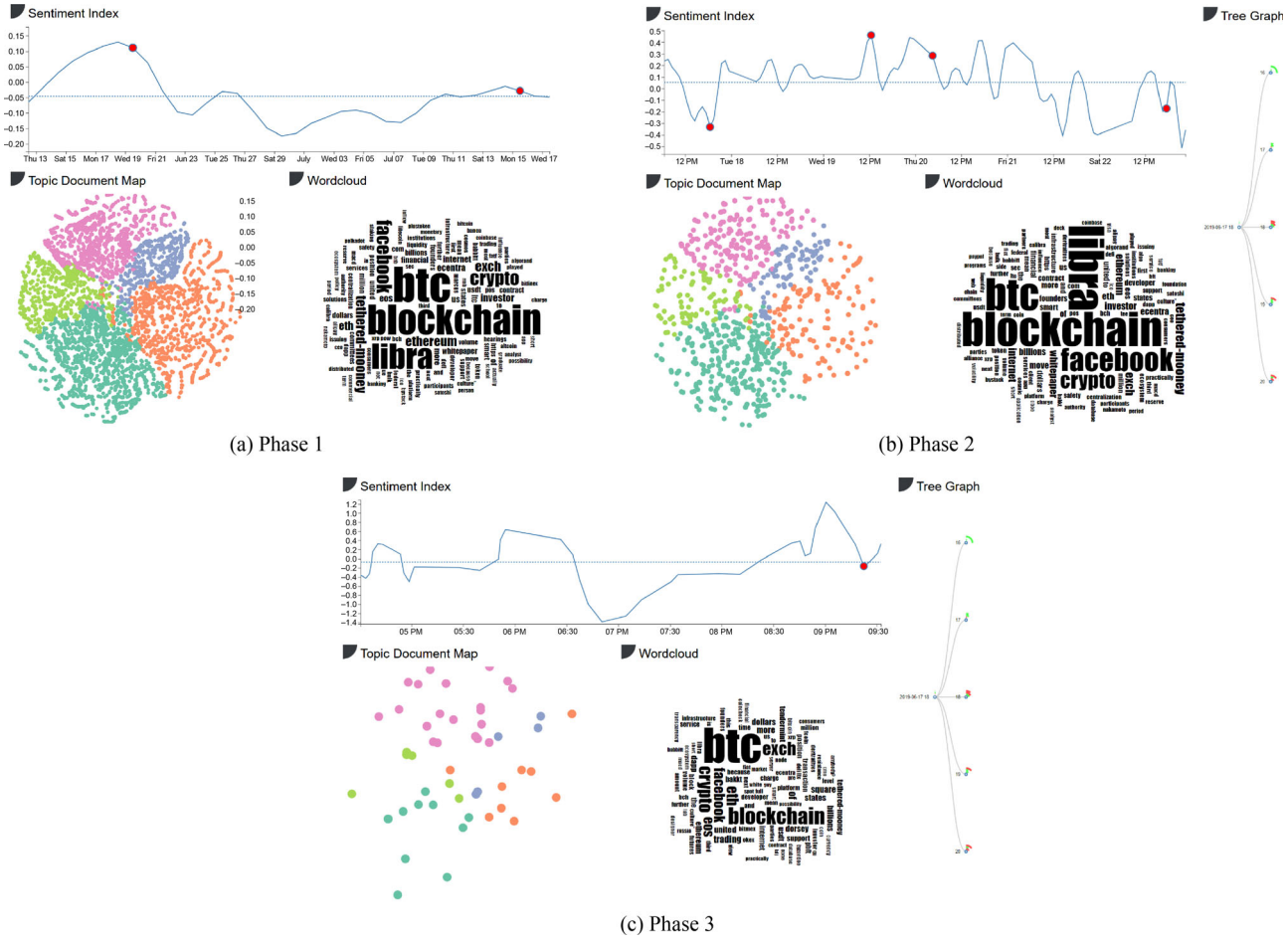


Fig. 9 Libra case: sentiment score, topic document map, and wordcloud in the three phases.

evaluation of our work difficult. Moreover, some events might be simply classified as positive or negative. Our event detection algorithm might be unable to detect subtle events with different aspects since we use a one-dimensional sentiment score. We hope to explore incorporating our topic weights as indicators for event detection in our future work.

7 Conclusions

This paper has proposed a novel framework for news event detection, text sentiment analysis, and topic modeling visualization. A multi-stage interactive visualization system allows users to drill into the pattern of events. In future, we hope to optimize our system for higher performance, as well as improving our topic models and enabling user to specify multiple topics in their domains of interest. While our case studies focused on blockchain news analysis, it could readily be applied to exploration and investigation of other types of news.

Acknowledgements

This work was supported by the National Key Research and Development Project of China (No. 2017YFC0804401) and the National Natural Science Foundation of China (No. U1909204). The work was supported by Prof. Wei Chen, who provided suggestions on how to build an interactive visualization system, and Mrs. Liyan Liu, who provided valuable ideas on how to conduct standardized tests.

Electronic Supplementary Material Electronic supplementary material is available in the online version of this article at `https://doi.org/10.1007/s41095-020-0178-4`.

References

[1] Wu, Y. C.; Liu, S. X.; Yan, K.; Liu, M. C.; Wu, F. Z. OpinionFlow: Visual analysis of opinion diffusion on social media. *IEEE Transactions on Visualization and Computer Graphics* Vol. 20, No. 12, 1763–1772, 2014.

[2] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding.*arXiv preprint* arXiv:1810.04805, 2018.

[3] Gers, F. A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. In: Proceedings of the 9th International Conference on Artificial Neural Networks, 850–855, 1999.

[4] Blei, D. M.; Ng, A. Y.; Jordan, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research* Vol. 3, 993–1022, 2003.

[5] Liu, X.; Tang, K. Z.; Hancock, J., Han, J. W.; Song, M., Xu, R.; Pokorny, B. A text cube approach to human, social and cultural behavior in the twitter stream. In: *Social Computing, Behavioral-Cultural Modeling and Prediction. Lecture Notes in Computer Science, Vol. 7812*. Greenberg, A. M.; Kennedy, W. G.; Bos, N. D. Eds. Springer Berlin Heidelberg, 321–330, 2013.

[6] Zhu, M. F.; Chen, W.; Xia, J. Z.; Ma, Y. X.; Zhang, Y. K.; Luo, Y. T.; Huang, Z.; Liu, L. Location2vec: A situation-aware representation for visual exploration of urban locations. *IEEE Transactions on Intelligent Transportation Systems* Vol. 20, No. 10, 3981–3990, 2019.

[7] Yuan, N. J., Zheng, Y.; Xie, X.; Wang, Y. Z.; Zheng, K.; Xiong, H. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering* Vol. 27, No. 3, 712–725, 2015.

[8] Doumit, S.; Minai, A. Online news media bias analysis using an LDA-NLP approach. In: Proceedings of the International Conference on Complex Systems, 2011.

[9] Liu, B.; Hu, M. Q.; Cheng, J. S. Opinion observer: Analyzing and comparing opinions on the Web. In: Proceedings of the 14th International Conference on World Wide Web, 342–351, 2005.

[10] Oelke, D.; Hao, M.; Rohrdantz, C.; Keim, D. A.; Dayal, U.; Haug, L.-E.; Janetzko, H. Visual opinion analysis of customer feedback data. In: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, 187–194, 2009.

[11] Morinaga, S.; Yamanishi, K.; Tateishi, K.; Fukushima, T. Mining product reputations on the Web. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 341–349, 2002.

[12] Chen, C. M.; Ibekwe-Sanjuan, F.; SanJuan, E.; Weaver, C. Visual analysis of conflicting opinions. In: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, 59–66, 2006.

[13] Wu, Y. C.; Wei, F. R.; Liu, S. X.; Au, N., Cui, W. W.; Zhou, H.; Qu, H. OpinionSeer: Interactive visualization of hotel customer feedback. *IEEE Transactions on Visualization and Computer Graphics* Vol. 16, No. 6, 1109–1118, 2010.

[14] Wu, Y. C.; Chen, Z. T.; Sun, G. D.; Xie, X.; Cao, N.; Liu, S. X.; Cui, W. StreamExplorer: A multi-stage system for visually exploring events in social streams. *IEEE Transactions on Visualization and Computer Graphics* Vol. 24, No. 10, 2758–2772, 2018.

[15] Reuter, T.; Papadopoulos, S.; Petkos, G.; Mezaris, V.; Kompatsiaris, Y.; Cimiano, P.; de Vries, C.; Geva, S. Social event detection at mediaeval 2013: Challenges, datasets, and evaluation. In: Proceedings of the MediaEval Multimedia Benchmark Workshop Barcelona, 2013.

[16] Fernando, T.; Denman, S.; Sridharan, S.; Fookes, C. Soft+hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection. *Neural Networks* Vol. 108, 466–478, 2018.

[17] Dörk, M.; Gruen, D.; Williamson, C.; Carpendale, S. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics* Vol. 16, No. 6, 1129–1138, 2010.

[18] Zhao, J.; Cao, N.; Wen, Z.; Song, Y. L.; Lin, Y. R.; Collins, C. FluxFlow: Visual analysis of anomalous information spreading on social media. *IEEE Transactions on Visualization and Computer Graphics* Vol. 20, No. 12, 1773–1782, 2014.

[19] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. 2019. Available at `https://git.dhimmel.com/bitcoin-whitepaper/`.

[20] Yli-Huumo, J.; Ko, D.; Choi, S.; Park, S.; Smolander, K. Where is current research on blockchain technology? A systematic review. *PLoS One* Vol. 11, No. 10, e0163477, 2016.

[21] Yue, X. W.; Shu, X. H.; Zhu, X. Y.; Du, X. N.; Yu, Z. Q.; Papadopoulos, D.; Liu, S. BitExTract: Interactive visualization for extracting bitcoin exchange intelligence. *IEEE Transactions on Visualization and Computer Graphics* Vol. 25, No. 1, 162–171, 2018.

[22] Battista, G. D.; Donato, V. D.; Patrignani, M.; Pizzonia, M.; Roselli, V.; Tamassia, R. Bitconeview: Visualization of flows in the bitcoin transaction graph. In: Proceedings of the IEEE Symposium on Visualization for Cyber Security, 1–8, 2015.

[23] Ranshous, S.; Joslyn, C. A.; Kreyling, S.; Nowak, K.; Samatova, N. F.; West, C. L.; Winters, S. Exchange pattern mining in the bitcoin transaction directed hypergraph. In: *Financial Cryptography and Data Security. Lecture Notes in Computer Science, Vol. 10323*. Brenner, M. et al. Eds. Springer Cham, 248–263, 2017.

[24] McGinn, D.; McIlwraith, D.; Guo, Y. Towards open data blockchain analytics: A Bitcoin perspective. *Royal Society Open Science* Vol. 5, No. 8, 180298, 2018.

[25] Information on `https://www.8btc.com/`.

[26] Information on `http://www.bitcoin86.com/`.

[27] Goldberg, Y.; Levy, O. Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint* arXiv:1402.3722, 2014.

[28] Mousa, A., Schuller, B. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 1, 1023–1032, 2017.

[29] Yang, Z. C.; Yang, D. Y.; Dyer, C., He, X. D.; Smola, A., Hovy, E. Hierarchical attention networks for document classification. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1480–1489, 2016.

[30] Hoffman, M.; Bach, F. R.; Blei, D. M. Online learning for latent dirichlet allocation. In: Proceedings of the Advances in Neural Information Processing Systems 23, 856–864, 2010.

[31] Van Laarhoven, P. J. M.; Aarts, E. H. L. Simulated annealing. In: *Simulated Annealing: Theory and Applications, Vol. 37*. Dordrecht: Springer Netherlands, 7–15, 1987.

[32] Sievert, C.; Shirley, K. LDAvis: A method for visualizing and interpreting topics. In: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, 63–70, 2014.

[33] Bollinger, J. Using bollinger bands. *Stocks & Commodities* Vol. 10, No. 2, 47–51, 1992.

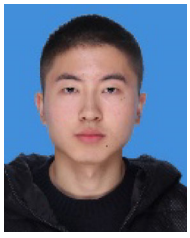
[34] Kailath, T.; Frost, P. An innovations approach to least-squares estimation—Part II: Linear smoothing in additive white noise. *IEEE Transactions on Automatic Control* Vol. 13, No. 6, 655–660, 1968.

[35] Eubank, R. L. *Nonparametric Regression and Spline Smoothing*. CRC Press, 1999.

[36] Marchand, P.; Marmet, L. Binomial smoothing filter: A way to avoid some pitfalls of least-squares polynomial smoothing. *Review of Scientific Instruments* Vol. 54, No. 8, 1034–1041, 1983.

[37] Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research* Vol. 9, 2579–2605, 2008.

Songye Han is currently an undergraduate of Zhejiang University and will receive his B.S. degree in 2020. His research interests lie in visual analytics and natural language processing.



Shaojie Ye is currently an undergraduate of Zhejiang University and will receive his B.S. degree in 2020. His research interests lie in visual analytics and machine learning.

Hongxin Zhang is an associate professor of the State Key Laboratory of CAD & CG, Zhejiang University. He received his Ph.D. degree in applied mathematics from Zhejiang University in 2002. His research interests include geometric modeling, visual analytics, and machine learning.