Neural Storyline Extraction Model for Storyline Generation from News Articles

Deyu Zhou[†] **Linsen Guo**[†] **Yulan He**[§]
[†] School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
[§] School of Engineering and Applied Science, Aston University, UK
{d.zhou,guolinsen}@seu.edu.cn, y.he@cantab.net

Abstract

Storyline generation aims to extract events described on news articles under a certain topic and reveal how those events evolve over time. Most existing approaches first train supervised models to extract events from news articles published in different time periods and then link relevant events into coherent stories. They are domain dependent and cannot deal with unseen event types. To tackle this problem, approaches based on probabilistic graphic models jointly model the generations of events and storylines without annotated data. However, the parameter inference procedure is too complex and models often require long time to converge. In this paper, we propose a novel neural network based approach to extract structured representations and evolution patterns of storylines without using annotated data. In this model, title and main body of a news article are assumed to share the similar storyline distribution. Moreover, similar documents described in neighboring time periods are assumed to share similar storyline distributions. Based on these assumptions, structured representations and evolution patterns of storylines can be extracted. The proposed model has been evaluated on three news corpora and the experimental results show that it outperforms state-of-the-art approaches accuracy and efficiency.

1 Introduction

With the development of the internet, massive information about current events is generated and propagated continuously on online news media sites. It is difficult for the public to digest such large volumes of information effectively. Storyline generation, aiming at summarizing the development of certain related events, has been intensively studied recently (Diao and Jiang, 2014).

In general, storyline can be considered as an event cluster where event-related news articles are ordered and clustered depending on both content and temporal similarity. Different ways of calculating content and temporal similarity can be used to cluster related events (Yan et al., 2011; Huang and Huang, 2013). Bayesian nonparametric models could also be used to tackle this problem by describing the storyline generating process using probabilistic graphical models (Li and Cardie, 2014; Diao and Jiang, 2014). Nevertheless, most existing approaches extract events independently and link relevant events in a post-processing step. More recently, Zhou et al. (2016) proposed a nonparametric generative model to extract storylines which is combined with Chinese Restaurant Processes (CRPs) to determine the number of storylines automatically. However, the parameter inference procedure is too complex and the model requires long time to converge. This makes it impractical to be deployed in real-world applications.

Recently, deep learning techniques have been successfully applied to various natural language processing tasks. Several approaches (Mikolov et al., 2013; Le and Mikolov, 2014) such as word2vec have been proved efficient in representing rich syntactic and semantic information in text. Therefore, it would be interesting to combine the advantage of both probabilistic graphical model and deep neural networks. There have been some efforts in exploring this in recent years. For example, Yang et al. (2015) proposed a gaussian mixture neural topic model incorporating both the ordering of words and the semantic meaning of sentences into a topic model. Cao et al. (2015) explained topic models from the perspective of neural networks and proposed a neural topic model where the representation of words and documents are combined into a unified framework. However, to the best of our knowledge, there is no attempt in extracting structured representation of storylines from text using neural network based approaches.

In this paper, we propose a novel neural model for storyline generation without the use of any annotated data. In specific, we assume that the storyline distributions of a document's title and its main body are similar. A pairwise ranking approach is used to optimize the model. We also assume that similar documents described in neighboring time periods should share similar storyline distributions. Hence, the model learned in the previous time period can be used for guiding the learning of the model in the current period. Based on the two assumptions, relevant events can be extracted and linked. Furthermore, storyline filtering based on confidence scores is performed. This makes it possible to generate new storylines.

The main contributions of this paper are summarized below:

- We propose a novel neural network based model to extract structured representations and evolution patterns of storylines. To the best of our knowledge, it is the first attempt to perform storyline generation based on neural network without any annotated data.
- The proposed approach has been evaluated on three corpora and a significant improvement on F-measure is achieved when compared to the state-of-the-art approaches. Moreover, the proposed approach only requires a faction of the training time in comparison with the second best approach.

2 Related Work

Considering storyline as hidden topic, storyline extraction can be casted into the topic detection and tracking (TDT) problem. One popular way to deal with TDT is through topic models. However, traditional topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) do not detect the dynamics of topic over time. Griffiths and Steyvers (2004) clustered texts using LDA and then mapped the topics into corresponding time periods. Blei and Lafferty (2006) developed a dynamic topic model which captures the evolution of topics in a sequentially organized corpus of documents by using Gaussian time series on the natural parameter of the multinomial topics and logistic normal topic proportion models. Unlike early work that relied on Markov assumptions or discretization of time, Wang and McCallum (2006) proposed a topic-over-time (TOT) model where each topic is associated with a continuous distribution over timestamps. For each document, the mixture distribution over topics is influenced by both word co-occurrences and the document's timestamp. As a storyline might include more than one topic, Kawamae (2011) made an improvement over TOT and proposed a trend analysis model which generates storylines based on the model trained in the previous time period. Ahmed and Xing (2008) employed Recurrent Chinese Restaurant Processes (RCRPs) to cluster texts from discrete time slice while the number of clusters can grows automatically with the data at each epoch. Following this, many approaches were proposed for storyline extraction by combining RCRP with LDA (Ahmed et al., 2011a,b; Ahmed and Xing, 2013). Considering dependencies among clusters in different time periods, a distance-dependent CRP model was proposed by (Blei and Frazier, 2011) which defines a weight function to quantify the dependency in different clusters. Huang et al. (2015) proposed a Dynamic Chinese Restaurant Process (DCRP) model which considers the birth, survival and death of a storyline.

Recently, there have been increasing interests in exploring neural network based approaches for topic detection from text. These approaches can be divided into two categories, solely based on neural networks and a combination of topic models and neural networks. For the first category, topic distributions of documents are modeled by a hidden layer in neural networks. For example, Hinton and Salakhutdinov (2009) proposed a two layer probabilistic graphical model which is a generalization of the restricted Boltzmann machine, called a "Replicate Softmax". It can be used to automatically extract low-dimensional latent semantic representations from a large unstructured collection of documents. Larochelle and Lauly (2012) proposed a neural autoregressive topic model to compute the hidden units of the network efficiently. There are also many approaches trying to combine neural networks with topic models. For example, Yang et al. (2015) presented a Gaussian mixture neural topic model which incorporates both the ordering of words and the semantic meaning of sentences into topic modeling. To make the neural network based model more interpretable, Cao et al. (2015) explained topic models from the perspective of neural networks and proposed a neural topic model where the representation of words

and documents are combined into a unified framework. Tian et al. (2016) proposed a sentence level recurrent topic model assuming the generation of each word within a sentence is dependent on both the topic of the sentence and the the historical context of its preceding words in the sentence. Wan et al. (2012) introduced a hybrid model which combines a neural networks with a latent topic models. The neural network provides a low dimensional embedding for the input data while the subsequent distribution is captured by the topic model. However, most of the aforementioned models are solely for topic detection. They do not consider evolutionary topic clustering for storyline generation.

3 Methodology

To model the generation of a storyline in consecutive time periods from a stream of documents, we propose a neural network based approach, called Neural Storyline Extraction Model (NSEM), as shown in Figure 1. In this model, we have the following assumptions:

Assumption 1: *for a document, the storyline distribution of its title and main body should be similar.*

In general, for any given document, its title and main body should discuss the same storyline. Although title may exist metaphor and metonymy to catch the reader's eye ball, the key entities and words will not change such as name, location and so on. Therefore, it is reasonable to assume that the title h and its main body d of a document share a similar storyline distribution. The storyline distributions of title and main body are denoted as $\boldsymbol{p}(\boldsymbol{s}_h)$ and $\boldsymbol{p}(\boldsymbol{s}_d)$. Hence, $\boldsymbol{p}(\boldsymbol{s}_h)$ and $\boldsymbol{p}(\boldsymbol{s}_d)$ should be similar. Based on this assumption, documents at time period t can be clustered into several storylines in such a way. Let h_{pos} denotes the correct title to the main body d (positive example), and h_{neg} denotes an irrelevant title (negative example), the similarity of the storyline distribution derived from the main body d and that obtained from the correct title h_{pos} should be far more greater than that obtained from irrelevant titles h_{neg}, i.e. $sim(\boldsymbol{p}(\boldsymbol{s}_d), \boldsymbol{p}(\boldsymbol{s}_{h_{pos}})) \gg sim(\boldsymbol{p}(\boldsymbol{s}_d), \boldsymbol{p}(\boldsymbol{s}_{h_{neg}}))$. Different similarity metrics can be used to measure the similarity between two distributions.

Assumption 2: *for similar documents in neighboring time periods, they should share similar storyline distribution.*

It is assumed that similar documents in the neighboring time periods tend to share the same storyline. For example, a document with the title "Indian Election 2014: What are minorities to do?" and another document in the next time period with the title "The efficiency of Indian elections is time tested" should belong to the same storyline "*India election*". Based on this assumption, events extracted in different time period can be linked into storylines. As main body contains more information than title, we only use the storyline distribution of the main body, $\boldsymbol{p}(\boldsymbol{s}_d)$, in order to simplify the model structure. The learned information in the previous time period is used to supervise the learning in the current time period.

Based on the above two assumptions, the proposed NSEM as shown in Figure 1 contains the following four layers: (1) *Input layer* shown at the left bottom part of Figure 1, takes d, h_{pos} and h_{neg} as the input and transforms these texts into vectors; (2) *Main body-Storyline layer* and *Title-Storyline layer*, both are designed to generate storyline distributions; (3) *Similarity layer* aims to calculate the similarity between the storyline distribution of the main body and that of the title. In the top part of Figure 1, the model learned in previous time period is used to guide the storyline distribution learning in current time period. We explain the structure and function of each layer of NSEM in more details below:

Input Layer (d, h): the input layer aims to represent the main body d and title h with distributed embedding $\vec{d}$ and $\vec{h}$. Let the subscript pos denotes the relevant title h_{pos} (positive example) and subscript neg denotes an irrelevant title h_{neg} (negative example). For news articles, we pay more attention to the key elements of events such as location l, person p , organization o and keywords w. Thus an event is described by a quadruple $\langle l, p, o, w\rangle$. We extract these elements from the main body and concatenate their word embeddings as the feature vector $\vec{d} = [\vec{l}, \vec{p}, \vec{o}, \vec{w}]$. We obtain the title feature $\vec{h}$ in the same way.

We first identify named entities and treat those named entities with multi-word expressions (e.g., "Donald Trump") as single tokens. Then we train word2vec (Mikolov et al., 2013) to represent each entity with a 100-dimensional embedding vector. We also filter out less important keywords and entities based on some criteria such as TFIDF. For a document containing more than one entity for the same event element type, for example, a document

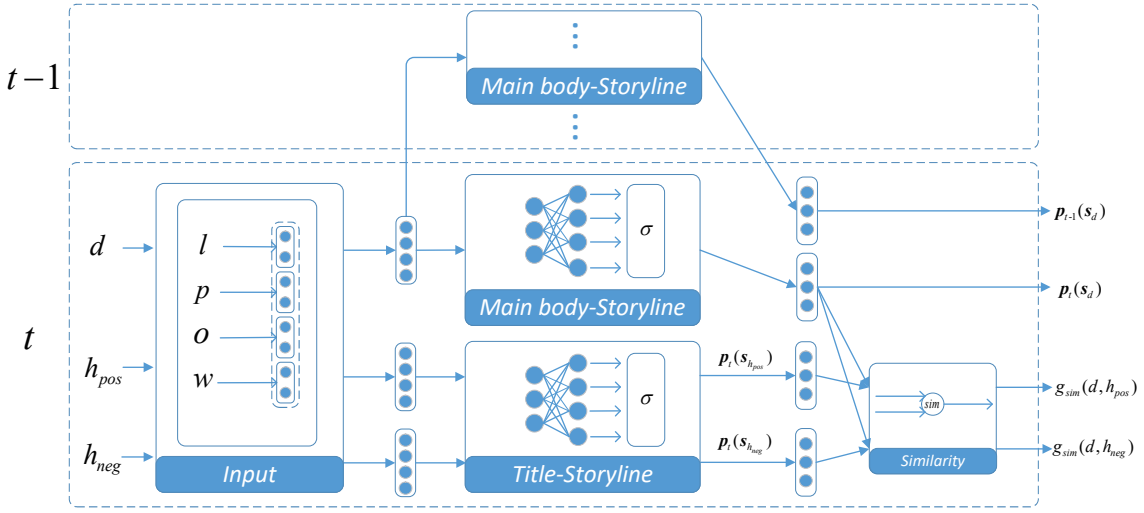


Figure 1: Overall architecture of the Neural Storyline Extraction Model (NSEM).

might contain mentions of different locations, we calculate the weighted sum of all location embeddings according to their occurrence number. If a certain event element is missing from a document, we set it to "null". After concatenating the four key event elements, each document or title is represented by a 400-dimensional embedding vector.

Main body-Storyline Layer ($\boldsymbol{p}(\boldsymbol{s}_d) \in \mathbb{R}^{1\times S}$): this layer aims to represent the storyline distribution $\boldsymbol{p}(\boldsymbol{s}_d)$ of main body d. Suppose there are a total of S storylines, the storyline distribution $\boldsymbol{p}(\boldsymbol{s}_d)$ is a S-dimensional vector, denoted as $\boldsymbol{p}(\boldsymbol{s}_d) = \{p(s_d = 1), \cdots, p(s_d = S)\}$. It can be formulated as below:

$$\boldsymbol{p}(\boldsymbol{s}_d) = f(\vec{d} \cdot \boldsymbol{W}_1 + \boldsymbol{b}_1) \quad (1)$$

where $\boldsymbol{W_1} \in \mathbb{R}^{K\times S}$ denotes the weight matrix, $\boldsymbol{b}$ denote the bias, $K = 400$ is the dimension of the document representation, and f denotes the activation function. Here we use the Softmax function. The probability of the main body d belonging to the storyline i can be written below:

$$p(s_d = i) = \frac{\exp(\vec{d} \cdot \boldsymbol{W}_{1i} + \boldsymbol{b}_{1i})}{\sum_{i=1}^{S} \exp(\vec{d} \cdot \boldsymbol{W}_{1i} + \boldsymbol{b}_{1i})} \quad (2)$$

Title-Storyline Layer ($\boldsymbol{p}(\boldsymbol{s}_h) \in \mathbb{R}^{1\times S}$): this layer aims to represent the storyline distribution $\boldsymbol{p}(\boldsymbol{s}_h)$ of title h. Similar to the Main body-Storyline layer, we can obtain $\boldsymbol{p}(\boldsymbol{s}_h)$ and $p(s_h = i)$ of title h in the following way:

$$\boldsymbol{p}(\boldsymbol{s}_h) = f(\vec{h} \cdot \boldsymbol{W}_2 + \boldsymbol{b}_2) \quad (3)$$

$$p(s_h = i) = \frac{\exp(\vec{h} \cdot \boldsymbol{W}_{2i} + \boldsymbol{b}_{2i})}{\sum_{i=1}^{S} \exp(\vec{h} \cdot \boldsymbol{W}_{2i} + \boldsymbol{b}_{2i})} \quad (4)$$

Similarity Layer ($g_{sim} \in \mathbb{R}$): this layer aims to calculate the similarity of the distributions between $\boldsymbol{p}(\boldsymbol{s}_d)$ and $\boldsymbol{p}(\boldsymbol{s}_h)$. The similarity score g_{sim} is calculated by the Kullback-Leibler (KL) divergence:

$$g_{sim}(d, h) = -\sum \boldsymbol{p}(\boldsymbol{s}_d) \log \frac{\boldsymbol{p}(\boldsymbol{s}_h)}{\boldsymbol{p}(\boldsymbol{s}_d)} \quad (5)$$

The similarity can be also calculated by other metric methods.

3.1 Storyline Construction

Different from the common way which link relevant events into storyline, we extract it in a unified framework. According to our second assumption, for the current time period t, we employ the storyline generation results in the previous time period $t-1$ as constraints to guide the storyline generation process in t. For a document d_t (we only use the main body here) in the time period t, we first use the model trained in $t-1$ to predict its storyline distribution $\boldsymbol{p}_{t-1}(\boldsymbol{s}_{d_t})$. Hence when we learn $\boldsymbol{p}_t(\boldsymbol{s}_{d_t})$, we would expect it to be similar to $\boldsymbol{p}_{t-1}(\boldsymbol{s}_{d_t})$. By doing so, we can link relevant events in different time periods together. For cases where intermittent storylines are observed, i.e., the related events occur initially, but disappear in certain time periods and re-occur later, we select documents randomly from all previous time periods and make them participate in the learning of current model.

3.2 Training

Our first assumption assumes that *for a document, its title and main body should share similar storyline distributions*. Hence, we use a pairwise ranking approach (Collobert et al., 2011) to optimize $\boldsymbol{p}(\boldsymbol{s}_d)$ and $\boldsymbol{p}(\boldsymbol{s}_h)$. The basic idea is that the storyline distribution of the main body d should be more similar to that of the relevant title than irrelevant ones. We first define the loss function as below:

$$\mathcal{L}_1(d, h_{pos}, h_{neg}) = \max(0, \Omega - g_{sim}(d, h_{pos}) + g_{sim}(d, h_{neg})) \quad (6)$$

where Ω denotes the margin parameter, h_{pos} denotes the relevant title and h_{neg} denotes an irrelevant title. We choose titles whose elements $\langle l, p, o, k\rangle$ have no intersection with those positive titles from the current time period as negative examples.

Our second assumption assume that *for similar documents in neighboring time periods, they should share similar storyline distribution*. Hence, the model learned in the previous time period can be used for guiding the learning of the model in the current period. Hence, when constructing storyline for the main body d in current time period t, we use the model in previous time period $t-1$ and predict the storyline distribution $\boldsymbol{p}_{t-1}(\boldsymbol{s}_d)$. Then we measure current storyline distribution $\boldsymbol{p}_t(\boldsymbol{s}_d)$ and predicted distribution $\boldsymbol{p}_{t-1}(\boldsymbol{s}_d)$ by KL divergence which can be defined as below:

$$\mathcal{L}_2(d) = \sum \boldsymbol{p}_{t-1}(\boldsymbol{s}_d) \log \frac{\boldsymbol{p}_t(\boldsymbol{s}_d)}{\boldsymbol{p}_{t-1}(\boldsymbol{s}_d)} \quad (7)$$

Therefore, the final objective function is to minimize:

$$\mathcal{L} = \sum_d (\alpha \mathcal{L}_1(d, h_{pos}, h_{neg}) + \beta \mathcal{L}_2(d)) \quad (8)$$

where α and β are the weights controlling the contributions of the two loss terms.

For the start time period, we only use $\mathcal{L}_1$ to optimize our model. Let Φ_t denote the model parameter in the time period t. Based on the model structure and the loss function described above, the training procedure for NSEM is given in Algorithm 1.

Algorithm 1 Training procedure for NSEM at the time period t

Require: main bodies $\boldsymbol{d}$; titles $\boldsymbol{h}$; model parameter Φ_{t-1} at the time period $t-1$

```
Initialize Φ_t
for d ∈ d do
    Calculate its storyline distribution based on Φ_{t-1}
end for
repeat
    for every minibatch M in (d, h) do
        for every pair (d_i, h_{i,pos}) in minibatch M do
            Calculate the storyline distribution p(s_{d_i})
            Calculate the storyline distribution p(s_{h_{i,pos}})
            Sample an irrelevant title h_{i,neg} where h_{i,neg} ∩ h_{i,pos} = ∅
            Calculate the storyline distribution p(s_{h_{i,neg}})
            Calculate L_1(d_i, h_{i,pos}, h_{i,neg})
            Calculate L_2(d_i)
        end for
        Calculate minibatch loss L_M = Σ_{d_i}(αL_1 + βL_2) and gradients ∇_{Φ_t} L_M
        Update model parameter Φ_t
    end for
until Convergence
```

3.3 Post-processing

As the number of storylines at each time period is assumed to be the same, some newly emerging storylines might be incorrectly linked with previous storylines. Therefore, post-processing is needed to filter out such erroneous linkings. We assume that if a current storyline does not have any key element in common with previously extracted storyline, it should be flagged as a new storyline. We define the **Coverage** of the storyline s as below:

$$Coverage(s, t, M) = (element)_s^t \cap (element)_s^{t-M} \quad (9)$$

where $(element)_s^t$ denotes the set of event elements in the time period t for storyline s and $(element)_s^{t-M}$ denote the set of event elements in the last M time periods for storyline s. If the coverage $Coverage(s, t, M)$ is less than a threshold N, the current storyline s is considered as a new one. For example, if the current storyline' $Coverage$ with index 5 is less than N, then previous storyline with index 5 stops at current period and the current storyline with index 5 is a new one.

4 Experiments

4.1 Setup

To evaluate the proposed approach, we use the three datasets as in (Zhou et al., 2016). The statistics of the three datasets are presented in Table 4.1. Among which the Dataset III includes 30 different types of manually annotated storylines which are categorized into four types: (1) long-term storylines which last for more than 2 weeks; (2) short-term storylines which last for less than 1 week; (3) intermittent storylines which last for more than 2 weeks in total, but stop for a time and then appear again; (4) new storylines which emerge in the middle of the period, not at the beginning.

Datasets	Documents	Storylines	Dates
I	526,587	N/A	1-30 May 2014
II	101,654	77	1-7 May 2014
III	23,376	30	1-30 May 2014

Table 1: Statistics of the three datasets.

In our experiments, we used the Stanford named entity recognizer [1] for identifying the named entities. In addition, we removed common stopwords and only kept tokens which are verbs, nouns, or adjectives from these news articles.

We chose the following four methods as the baseline approaches.

1. DLDA (Blei and Lafferty, 2006): the dynamic LDA is based on the Markovian assumption that the topic-word distribution at the current time period is only influenced by the topic-word distribution in the previous time period. Moreover, topic-word distributions are linked across time periods by a Markovian chain.

2. RCRP (Ahmed et al., 2011a): it is a non-parametric model for evolutionary clustering based on RCRP, which assumes that the past story popularity is a good prior for current popularity.

3. SDM (Zhou et al., 2015): it assumes that the number of storylines is fixed and the storyline is modeled as a joint distribution over entities and keywords. The dependency of different stories of the same storyline at different time periods is captured by modifying Dirichlet priors.

4. DSEM (Zhou et al., 2016): this model is integrated with CRPs so that the number of storylines can be determined automatically without human intervention. Moreover, per-token Metropolis-Hastings sampler based on light LDA (Yuan et al., 2015) is used to reduce sampling complexity.

For DLDA, SDM and our model NSEM, the storyline number is set to 100 on both Dataset II and III. In consideration of the dependency to the historical storyline distributions, the number of past epochs M is set to 7 for both SDM and DSEM. For RCRP, the hyperparameter α is set to 1. For our model NSEM, the threshold Ω is set to 0.5 and the loss weight α and β are set to 1 and 0.5 respectively. In postprocess step, we empirically set the N to 7.

To evaluate the performance of the proposed approach, we use precision, recall and F-measure which are commonly used in evaluating information extraction systems. The precision is calculated based on the following criteria: 1) The entities and keywords extracted refer to the same storyline; 2) The duration of the storyline is correct. We assume that the start date (or end date) of a storyline is the publication date of the first (or last) related news article.

As there is no gold standard available for Dataset I, we do manual examination with the experimental result. We search for the same period of news and compare it with our results in the criteria.

4.2 Experimental Results

The experimental results of the proposed approach in comparison to the baselines on Dataset I, II and III are presented in Table 2. For Dataset I, as it is hard to know the ground-truth of storylines, we only report the precision value by manually examining the extracted storylines.

It can be observed from Table 2 that the proposed approach achieves the best performance on the three datasets. In specific, for Dataset I, NSEM extracts more storylines and with a higher precision value. For Dataset II containing 77 storylines, NSEM extracts 81 storylines among which 61 are correct and outperforms DSEM with 2% in F-measure. For dataset III consisting of 30 storylines, NSEM extracted 27 storylines among which 21 are correct. Although its recall value is the same as DSEM, its precision value is nearly 3%

[1] https://nlp.stanford.edu/software/CRF-NER.html

Dataset I		
Method	Precision(%)	# of extracted storylines
SDM	70.20	104
DSEM	75.43	114
NSEM	**76.58**	**121**

Dataset II			
Method	Precision(%)	Recall(%)	F-measure(%)
DLDA	62.67	61.03	61.84
RCRP	67.11	66.23	66.67
SDM	70.67	68.80	69.27
DSEM	73.17	77.92	75.47
NSEM	**75.31**	**79.22**	**77.22**

Dataset III			
Method	Precision(%)	Recall(%)	F-measure(%)
DLDA	46.16	43.33	42.86
RCRP	61.54	53.33	57.14
SDM	54.17	43.33	48.15
DSEM	75.00	70.00	72.41
NSEM	**77.78**	70.00	**73.69**

Table 2: Performance comparison of the storyline extraction results on Dataset I, II and III.

Dataset III			
S	Precision(%)	Recall(%)	F-measure(%)
25	66.67	33.33	44.44
50	73.08	46.67	56.96
75	76.92	53.33	62.99
100	77.78	70.00	73.69
125	78.13	73.33	75.65
150	78.79	70.00	74.13

Table 3: The performances of NSEM with different S.

higher which results in better F-measure.

4.3 Impact of the Number of Storylines S

The proposed approach needs to preset the number of storylines. To study the impact of the number of storylines on the performance of the proposed model, we conducted experiments Dataset III with different numbers of storylines S varying between 25 and 150. Table 3 shows the performance of storyline extraction with different value of S. It can be observed that both precision and recall of NSEM increase with the increasing number of storylines until it reaches 100. If further increasing S, the precision/recall have slight change and the F-measure become relatively stable.

4.4 Structured Browsing

We illustrate the evolution of storylines using structured browsing. The structured information of the storylines such as locations, persons, entities, keywords are presented, together with titles of some related documents. The number of related documents for each storyline is also depicted to allow an easy visualization of storyline popularity over time. Figure 2 illustrates three different types of storylines including "Apple vs Samsung", "Pistorious shoot Steenkamp" and "Egypt election".

For the first storyline "Apple vs Samsung", it starts at the beginning of the month and only lasts for 9 days. Three representative epochs are highlighted. From the extracted organizations, "Apple, Samsung", and keywords, "patent, infringe", it can be easily deduced that this is about "Apple and Samsung infringed patents".

For the storyline "Pistorious shoot Steenkamp", it is an intermittent storyline which lasts for more than 2 weeks but with no related news articles in some of the days in between. From Figure 2, it can be observed that the storyline ceases for 2 days in Day 10 and 11. From the structured representation of the early storylines, it can be observed that there is a shooting event about Pistorious and Steenkamp in South African. After 2 day's silence, in Day 13, public attention was raised once again since Pistorius applied for mental tests.

For the last storyline "Egypt election", it starts in Day 20 and continues beyond the end of May. From the key event elements, location "Egypt" and keywords "presidential, election", it can be easily inferred that there was a presidential election in Egypt. It can also be observed that Sisi and Morsi were both candidates for the Egypt's presidential election from persons extracted, "Sisi, Morsi" in Day 26. In Day 29, the storyline reached to the climax since Sisi won the election, which can be discovered from the title "Sisi elected #Egypt president by landslide".

4.5 Time Complexity

To explore the efficiency of the proposed approach, we conducted an experiment by comparing the proposed approach NSEM with DSEM. DSEM employs the Metropolis-Hastings sampler to boost the sampling complexity in order to achieve faster convergence. We train both models on training data varying from 1,000 to 10,000 documents. Figure 3 illustrates the logarithm of

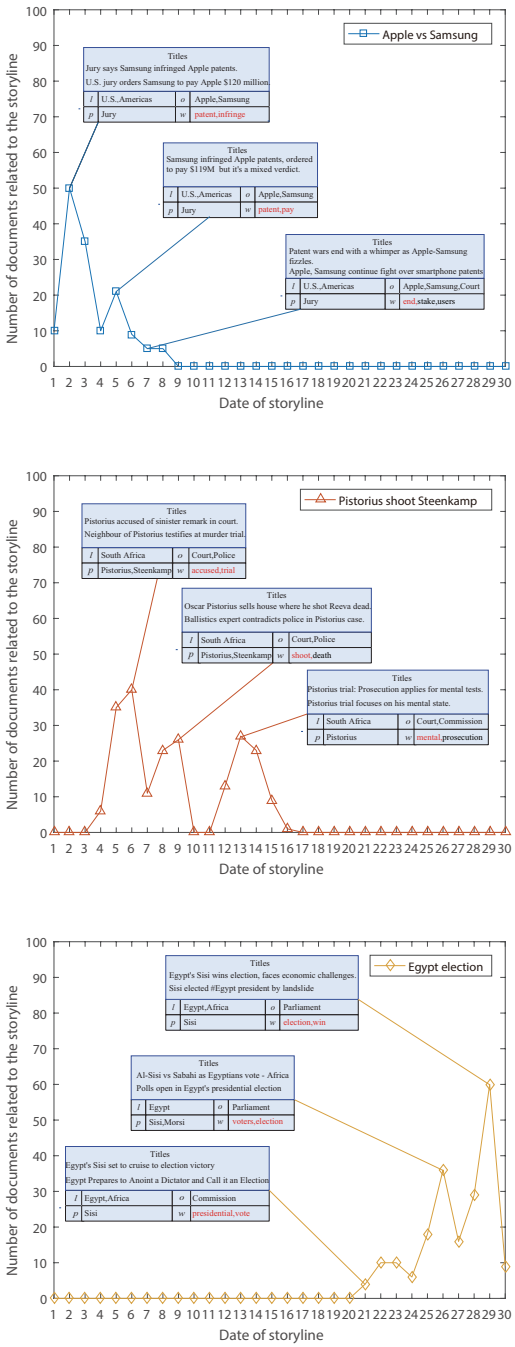


Figure 2: The structured representations of three example storylines.

time consumed for each training set. It can be observed that NSEM trains 30 times faster compared to DSEM, showing the advantage of using a neural network based approach in comparison with a Bayesian model based method.

4.6 Visualization of the Learned Distribution

Our proposed model is based on the two distribution similarity assumptions which we presented in the Methodology section. To investigate the quality of the learned storyline distribution, we conducted an experiment on Dataset III where the storyline number S is set to 100. We randomly choose three documents and calculate the storyline distribution of theirs title and main body based on our learned NSEM. We also randomly select three pairs similar documents in different time periods and draw their main body storyline distributions based on the learned NSEM. It can be observed from Figure 4 that the storyline distributions of the title and the main body of a document are similar. Moreover, the storyline distributions of two similar documents in different time periods are also similar.

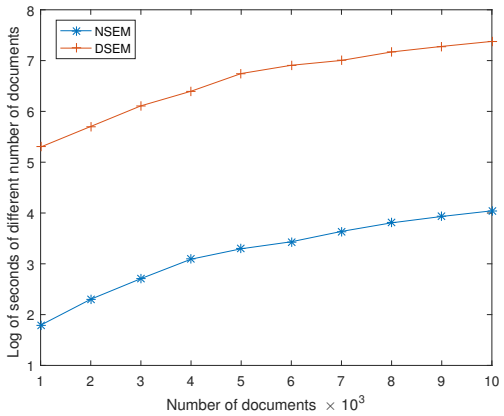
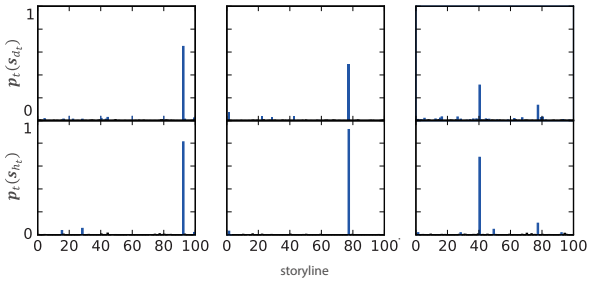


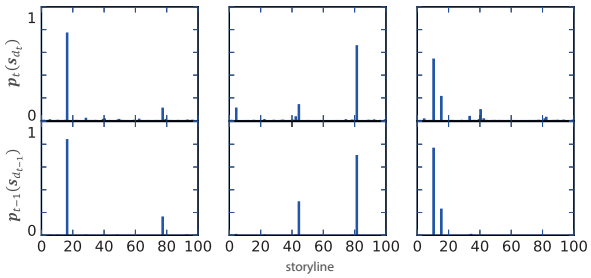
Figure 3: Comparison of training time between NSEM and DSEM.

5 Conclusions and Future Work

In this paper, we have proposed a neural network based storyline extraction model, called NSEM, to extract structured representations of storyline from news articles. NSEM was designed based on the two assumptions about the similarity of storyline distributions of the title and the main body of the same document, and the similarity of storyline distributions of similar documents in different time periods. Experimental results show that our proposed model outperforms the state-of-the-art approaches and only requires a fraction of training time. In future work, we will explore the extension of our proposed model to cater for varying number of storylines automatically and also better deal with intermittent storylines.



(a) Storyline distributions of title and main body.



(b) Storyline distributions of similar documents in different time periods.

Figure 4: Visualization of the learned storyline distributions.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (61772132), the Natural Science Foundation of Jiangsu Province of China (BK20161430) and Innovate UK (103652).

References

Amr Ahmed, Qirong Ho, Jacob Eisenstein, Eric Xing, Alexander J Smola, and Choon Hui Teo. 2011a. Unified analysis of streaming news. In *Proceedings of the 20th international conference on World wide web*. ACM, pages 267–276.

Amr Ahmed, Qirong Ho, Choon Hui Teo, Jacob Eisenstein, Alex Smola, and Eric Xing. 2011b. Online inference for the infinite topic-cluster model: Storylines from streaming text. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. pages 101–109.

Amr Ahmed and Eric Xing. 2008. Dynamic nonparametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, pages 219–230.

Amr Ahmed and Eric P Xing. 2013. Scalable dynamic nonparametric bayesian models of content and users. In *IJCAI*. pages 3111–3115.

David M Blei and Peter I Frazier. 2011. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research* 12(Aug):2461–2488.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. ACM, pages 113–120.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A novel neural topic model and its supervised extension. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. pages 2210–2216.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Qiming Diao and Jing Jiang. 2014. Recurrent chinese restaurant process with a duration-based discount for event identification from twitter. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, pages 388–397.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* 101(suppl 1):5228–5235.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*. pages 1607–1614.

Lifu Huang and Lian'en Huang. 2013. Optimized event storyline generation based on mixture-event-aspect model. In *EMNLP*. pages 726–735.

Rui Huang, Fengyuan Zhu, and Pheng-Ann Heng. 2015. The dynamic chinese restaurant process via birth and death processes. In *AAAI*. pages 2687–2693.

Noriaki Kawamae. 2011. Trend analysis model: trend consists of temporal words, topics, and timestamps. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pages 317–326.

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*. pages 2708–2716.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1188–1196.

Jiwei Li and Claire Cardie. 2014. Timeline generation: Tracking individuals on twitter. In *Proceedings of the 23rd international conference on World wide web*. ACM, pages 643–652.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Computer Science* .

Fei Tian, Bin Gao, Di He, and Tie-Yan Liu. 2016. Sentence level recurrent topic model: Letting topics speak for themselves. *arXiv preprint arXiv:1604.02038* .

Li Wan, Leo Zhu, and Rob Fergus. 2012. A hybrid neural network-latent topic model. In *International Conference on Artificial Intelligence and Statistics*. pages 1287–1294.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 424–433.

Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 433–443.

Min Yang, Tianyi Cui, and Wenting Tu. 2015. Ordering-sensitive and semantic-aware topic modeling. In *AAAI*. pages 2353–2360.

Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. 2015. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pages 1351–1361.

Deyu Zhou, Haiyang Xu, Xin-Yu Dai, and Yulan He. 2016. Unsupervised storyline extraction from news articles. In *IJCAI*. pages 3014–3021.

Deyu Zhou, Haiyang Xu, and Yulan He. 2015. An unsupervised bayesian modelling approach for storyline detection on news articles. In *EMNLP*. pages 1943–1948.