News Stream Summarization using Burst Information Networks

Tao Ge[1,2]*, **Lei Cui**[3], **Baobao Chang**[1,2], **Sujian Li**[1,2], **Ming Zhou**[3], **Zhifang Sui**[1,2]
[1]Key Laboratory of Computational Linguistics, Ministry of Education,
School of EECS, Peking University, Beijing, 100871, China
[2]Collaborative Innovation Center for Language Ability, Xuzhou, Jiangsu, 221009, China
[3]Microsoft Research
getao@pku.edu.cn, lecu@microsoft.com, chbb@pku.edu.cn
lisujian@pku.edu.cn, mingzhou@microsoft.com, szf@pku.edu.cn

Abstract

This paper studies summarizing key information from news streams. We propose simple yet effective models to solve the problem based on a novel and promising representation of text streams – *Burst Information Networks (BINets)*. A BINet can be aware of redundant information, allows global analysis of a text stream, and can be efficiently built and dynamically updated, which perfectly fits the demands of text stream summarization. Extensive experiments show that the BINet-based approaches are not only efficient and can be used in a real-time online summarization setting, but also can generate high-quality summaries, outperforming the state-of-the-art approach.

1 Introduction

Text stream summarization aims to summarize key information from a text stream containing huge numbers of documents, which is an important and useful task that can be used for many real-world applications. For example, a news portal website editor needs to summarize news streams in the past day for generating a list of headline news; an editor of Sports Weekly may want a summary of the past week news stream for editing the magazine; and geologists and meteorologists will benefit from a summary of disaster events from the past year news stream (as shown in Table 1) for their study.

In contrast to traditional text summarization tasks (e.g., single and multi-document summarization) that have been extensively studied for decades, the task of stream summarization is a younger research problem which attempts to solve a summarization problem in the big-data setting. For a text stream with millions of documents involving various topics and events, traditional single- and multi-document summarization approaches cannot address the information overload challenge. For example, a single-document summarization model will generate 1 million document summaries for a text stream with 1 million documents, which are still overwhelming for a person to learn the key information in the stream. In such cases, one needs to a summary of the whole stream instead of summaries of each document.

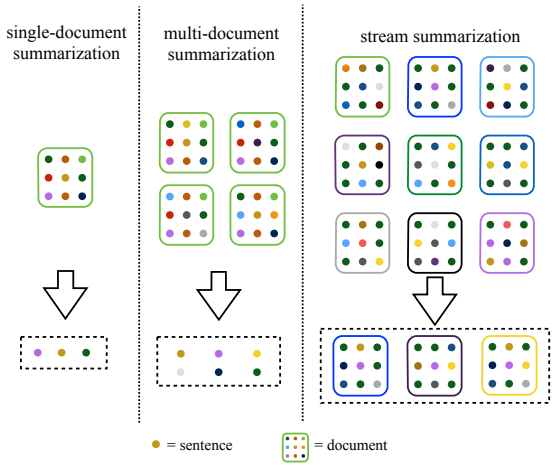

Figure 1: Stream summarization paradigm.

Figure 1 shows the paradigm of stream summarization. Compared with single- and multi-document summarization, stream summarization has three differences: (1) it summarizes a text stream containing millions of documents involving a variety of topics and events while single- and

* This work was done when the first author was visiting Microsoft Research Asia

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 784–794,
Austin, Texas, November 1-5, 2016.

2009 disaster summary	2010 disaster summary
• ... • Sep 2, 2009: About 60 people die when a 7.1-magnitude earthquake hit the island of Java. • Sep 9, 2009: More than 30 people are killed when fast moving floods caused by heavy rain sweep through Istanbul. • Sep 30, 2009: A 7.6-magnitude earthquake hit the island of Sumatra, leaving more than 1,000 people dead and thousands injured. • ...	• ... • Jan 12, 2010: A 7.0-magnitude earthquake hit Haiti, killing about 200,000 people. • Feb 27, 2010: An 8.8-magnitude earthquake rocked Chile, killing at least 700 people dead and affecting more than 1.5 million people. • Apr 5, 2010: An explosion in a West Virginia coal mine kills at least 25 people and leaves 4 unaccounted for. • ...

Table 1: Stream summary about disasters in 2009 and 2010. The disaster summary of 2009 can be used a reference summary to supervise generating a disaster summary for the 2010 news stream.

multi-document summarization summarizes one or a handful of documents about the same news event; (2) instead of selecting sentences to generate a summary, stream summarization selects representative documents to summarize a text stream; (3) summaries for a text stream may vary significantly for users who have different interests and preferences (e.g., summaries for an environmental expert and a sports fan should not be the same). Therefore, in order to generate targeted summaries for specific users, a stream summary needs to be generated based on a reference summary. For instance, one can use the 2009 disaster summary (the left part in Table 1) as a reference to learn how to write the 2010 disaster summary (the right part in Table 1).

In general, there are three challenges for summarizing a text stream. First, a stream summarization model should be able to be aware of redundant information in the stream for avoiding generating redundant content in the summary; second, a stream summarization algorithm should be capable of analyzing text content on the stream level for identifying the most important information in the stream; third, a stream summarization model should be efficient, scalable and able to run in an online fashion because data size of a text stream is usually huge, and it is dynamic and updated every second.

The previous approaches (e.g., (Ge et al., 2015b)) tend to cluster similar documents as event detection to avoid redundancy, rank the clusters based on their sizes and topical relevance to the reference summaries, and select one document from each cluster as representative documents. Due to the high time complexity of clustering models, their approaches usually run slowly and are not scalable.

To overcome the limitations, we propose Burst Information Networks (BINet) as a novel representation of a text stream. In a BINet (Figure 2), a node is a burst word (including entities) with the time span of one of its burst periods, and an edge between two nodes indicates how strongly they are related. Based on the BINet representation, we propose two models – NodeRank and AreaRank – for summarizing a news stream. We conduct extensive experiments to evaluate our approaches by comparing several baselines and the state-of-the-art approaches in various settings and show that the BINet-based approaches are efficient, scalable and can work in an online fashion and that they can generate high-quality summaries for a news stream, outperforming the state-of-the-art.

The major contributions of this paper are:

- We propose BINets as a novel representation of text streams. BINets can perfectly address the challenges of text stream summarization, which can be aware of information redundancy (Section 3), enables global analysis of the text stream (Section 4.1 and 4.2), and be efficiently built and updated incrementally (Section 4.3).
- We propose two ranking-based models based on the BINet representation, which can effectively learn to summarize a text stream from a reference summary, and outperform the state-of-the-art model.
- We create and release a new benchmark dataset for evaluating real-time stream summarization.

2 Stream Summarization

The task of text stream summarization is to generate a summary including key information from a

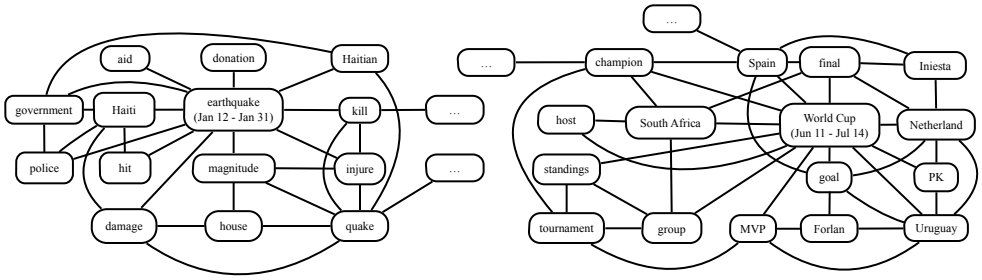

Figure 2: Illustration of a BINet. Due to space limitation, we only show the burst period of some nodes.

given text stream (e.g., 1-year news stream). In contrast to traditional summarization tasks which summarize a single or a handful of documents related to the same event by extracting sentences, the task of stream summarization aims to summarize a text stream which contains huge numbers of documents involving a variety of topics and events by selecting representative documents, as Figure 1 shows. In a stream summary, each selected document is considered as an entry which can be shown using the title or the first paragraph of the document. Since documents in a news stream are always about news events, we also call an entry as *an event entry* and call a stream summary as an *event chronicle* which is a list of event entries, as shown in Table 1. In a stream summary, entries should not be redundant. Formally, we define a stream summary (i.e., event chronicle) $\mathcal{E} = \{\boldsymbol{e_1}, \boldsymbol{e_2}, \cdots, \boldsymbol{e_K}\}$ where $\boldsymbol{e_k} = (t_{\boldsymbol{e_k}}, \boldsymbol{w_{e_k}})$ is an event entry including the event's time information $t_{\boldsymbol{e_k}}$ and text description $\boldsymbol{w_{e_k}}$ which is set of words in text.

Due to the diversity of ways to summarize a text stream as Section 1 discusses, we use a reference summary of a text stream during an early period to supervise summary generation for new text streams. It is a practical setting since many historical manually edited summaries of early streams are available and can be used as an example to demonstrate what kind of information is preferred in a stream summary.

3 Representing a text stream using Burst Information Network

3.1 Burst

A word's burst refers to a remarkable increase in the number of occurrences of the word during a period and might indicate important events or trending topics. For example, as shown in Figure 3, the word *earthquake* has bursts from the Jan 12 to Jan 31, 2010 and from Feb 27 to Mar 8, 2010 because of the strong earthquakes occurring in Haiti and Chile respectively.

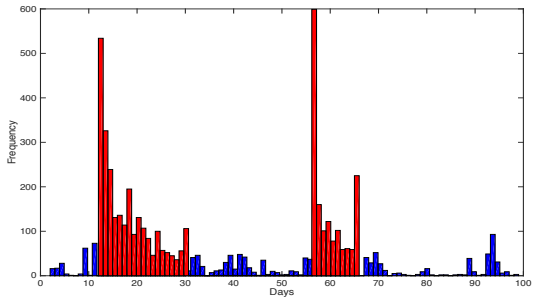

Figure 3: Frequency of *earthquake* during the first 90 days in the 2010 news stream.

Specifically, if a word w is in a burst state at every time t during a period, we call this period as a burst period of w, and w has a burst during this period. In Figure 3, *earthquake* has 2 burst periods (i.e., (Jan 12 - Jan 31) and (Feb 27 - Mar 8))

Formally, we define $\mathcal{P}$ as one burst period of the word w. $\mathcal{P}$ is a consecutive time sequence during which w bursts at every time epoch t:

$$\mathcal{P} = (t_i, t_{i+1}, t_{i+2}, ..., t_{i+n})$$
$$\forall t \in \mathcal{P} \;\; s_t = 1$$

where s_t is a binary indicator of the burst state of w at time t.

3.2 Burst Information Network

To build an information network which can represent associations between key facts in a text stream, we propose a new representation called "*Burst Information Network (BINet)*" by using burst elements as nodes:

A Burst Element is a burst of a word. It can be represented by a tuple: $\langle w, \mathcal{P} \rangle$ where w denotes the

word and $\mathcal{P}$ denotes one burst period of w.

According to the above definition, a burst element is a joint representation of a word type and one of its burst periods. A word may have multiple burst periods while a burst element only has one burst period. A word during its different burst periods will be regarded as different burst elements.

Formally, we define the BINet $G = \langle V, E \rangle$ as follows. Each node $v \in V$ is a burst element and each edge $e \in E$ denotes the association between burst elements. Intuitively, if two burst elements frequently co-occur, the edge between them should be highly weighted. We define $\omega_{i,j}$ as the weight of an edge between v_i and v_j, which is equal to the number of documents where v_i and v_j co-occur.

Besides $w(v)$ and $\mathcal{P}(v)$ that denote a node v's word and burst period respectively, we also record a node's context words[1] and its source documents which the node is from during constructing a BINet. Formally, we use $\mathcal{C}(v)$ and $\mathcal{D}(v)$ to denote the context word set and source document set of v. Also, for a document d in the stream, we use $\mathcal{A}(d)$ to denote the set of nodes whose source documents include d. Since nodes in $\mathcal{A}(d)$ are usually adjacent, we also call $\mathcal{A}(d)$ document d's area on the BINet. The construction of a BINet is efficient: the time complexity of building a BINet is $O(n)$ where n is the number of documents in a stream.

BINets can be properly aware of redundant information: since nodes in a community in a BINet are topically and temporally coherent, information about the same news event tends to be adjacent and redundant information of the same event is naturally removed. For example, assuming that there are hundreds of documents about *Haiti earthquake* in a text stream, by using the BINet representation, the information is concentrated in a few adjacent nodes without redundancy (left part in Figure 2). Moreover, information about different events is not considered as redundant. For example, the information regarding *Haiti earthquake* and *Chile earthquake* is not treated as redundant, which is allocated to different areas in the BINet, as Figure 2 shows. Therefore, as long as we do not select overlapping areas on the BINet, we can avoid selecting redundant content as entries.

[1]Here, the context window size is set to 10. Note that in our experiments, only words frequently (more than 5 times) co-occur in the context will be reserved.

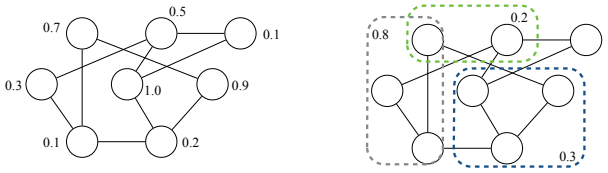

Figure 4: NodeRank (left) and AreaRank (right).

In addition to the awareness of information redundancy, BINets also allow global importance analysis on the stream level and online stream summarization, which will be discussed in Section 4.

4 Summarizing a text stream on the BINet

Based on the BINet representation, we propose two models – NodeRank and AreaRank – to summarize a text stream by generating entries of the summary. As Figure 4 shows, the NodeRank model scores every node on the BINet independently for identifying the most valuable information to be included in the stream summary, while the AreaRank model attempts to score an area that covers a handful of nodes for locating the most informative information blocks.

To train NodeRank and AreaRank models, we use reference summaries and the (reference) BINets built from the text stream during the reference summary's period as supervision.

4.1 NodeRank

Intuitively, if we can find the most valuable information on the BINet that should be included in the summary, then we can generate a high-quality summary of a text stream. For this goal, we label the corresponding nodes of words appearing in the reference summary on the reference BINet as score 1 (positive). Formally, for a reference summary $\mathcal{E}$, we label the following set of nodes in the reference BINet $G_r = \langle V_r, E_r \rangle$ as score 1:

$$V_{pos} = \bigcup_{\boldsymbol{e_k} \in \mathcal{E}} \{v | v \in V_r \wedge w(v) \in \boldsymbol{w}_{\boldsymbol{e_k}} \wedge t_{\boldsymbol{e_k}} \in \mathcal{P}(v)\} \quad (1)$$

where $w(v)$ and $\mathcal{P}(v)$ are word and burst period of node v respectively, $\boldsymbol{e_k}$ is an event entry in the reference summary $\mathcal{E}$, $\boldsymbol{w}_{\boldsymbol{e_k}}$ is the set of words in $\boldsymbol{e_k}$'s text, and $t_{\boldsymbol{e_k}}$ is $\boldsymbol{e_k}$'s time. The nodes that are not in V_{pos} in the reference BINet will be labeled as 0 (negative).

After labeling the reference BINet, we train a learning to rank (L2R) model[2] using the following features for scoring nodes in the target BINet $G_\tau = \{V_\tau, E_\tau\}$ (shown in Figure 4):

- $w(v)$: the word of node v, indicating its semantic information.
- $pr(v)$: node v's PageRank value can reflect the global importance of the node on the stream level, which can be easily obtained by running the PageRank algorithm on the BINet.
- $\mathcal{C}(v)$: the context words of node v defined in Section 3.2, indicating the topic information.

After scoring nodes in the target BINet, we greedily choose a document area $\mathcal{A}(d)$ that covers a set of nodes whose score is the largest:

$$d^* = arg \max_{d \in D_\tau} \sum_{v \in \mathcal{A}(d)} score_{NR}(v) \quad (2)$$

where D_τ is the document sets in the target stream and $score_{NR}(v)$ is the score of node v outputted by NodeRank model. Document d^*'s first paragraph and its document creation time (DCT) will be used to generate an event entry for the summary of the target stream. Note that though we do not normalize the length of a document in Eq (2), we constrain the maximum length of a document's first paragraph is 50 words and will not select the document whose first paragraph is longer than 50 words.

By repeating this step for k times, we can generate a stream summary with k event entries. Note that in order to avoid generating redundant entries in the summary, we will not choose d^* if its document area $\mathcal{A}(d^*)$ overlaps with the areas of the documents that have been already chosen as entries.

4.2 AreaRank

Instead of scoring nodes independently like NodeRank, we propose AreaRank model for scoring an area on the BINet for finding areas that corresponds to the most important news events in the stream.

Different from NodeRank where each instance is one node in the BINet, instances are areas on the BINet in the AreaRank model, as shown in Figure 4. In this paper, we mainly consider document area $\mathcal{A}(d)$ since we select representative documents as entries in the summary.

As NodeRank, we first label reference BINet using the reference summary. In the AreaRank model, we find the areas on the reference BINet corresponding to each event entry in the reference summary and label such areas as score 1 (positive). Formally, for a reference summary $\mathcal{E}$, the positive areas are in the following set:

$$\mathbb{A}_{pos} = \bigcup_{\boldsymbol{e_k} \in \mathcal{E}} \{\mathcal{A} | \mathcal{A} = V_{\boldsymbol{e_k}}\} \quad (3)$$

where $V_{\boldsymbol{e_k}} = \{v | v \in V_r \wedge w(v) \in \boldsymbol{w}_{\boldsymbol{e_k}} \wedge t_{\boldsymbol{e_k}} \in \mathcal{P}(v)\}$ is the set of nodes to which words in $\boldsymbol{e_k}$ correspond in the reference BINet.

We label other document areas that do not overlap any positive area on the reference BINet as score 0. Then, we use the training data to train AreaRank using the following features:

- $\boldsymbol{w}(\mathcal{A})$: words of nodes in area $\mathcal{A}$, indicating the area's semantic and topic information.
- $\boldsymbol{pr}(\mathcal{A})$: this feature includes maximum, sum and average of PageRank value of nodes in the area and sum of top 3 PageRank value of nodes in the area, indicating the area's general importance, which can reflects the impact of the events corresponding to the area in the stream.
- $\mathcal{C}(\mathcal{A})$: context of nodes in area $\mathcal{A}$. This feature is useful for indicating topical information.

In the test phase, we use AreaRank model to score all possible document areas on the target BINet. Then, we greedily choose the document area with the top score to generate an event entry for the summary:

$$d^* = arg \max_{d \in D_\tau} score_{AR}(\mathcal{A}(d)) \quad (4)$$

As NodeRank, d^*'s first paragraph and DCT will be used to generate an event entry for the stream summary if d^*'s area $\mathcal{A}(d)$ does not overlap the areas of the documents that have been already selected for generating event entries. The maximum length of the first paragraph of a document is 50 words. This step will be repeated for multiple times for generating event entries of the summary.

[2]We use SVMRank (Joachims, 2006). During training, we randomly sample 50% of negative examples which are used to generate the training set with positive examples.

4.3 Online stream summarization

An advantage of the BINet is that it can be incrementally updated when new streams arrive, which is useful for online stream summarization. Assuming we have a news stream from time t_0 to t_k at hand, we can detect word bursts and construct a BINet G based on the stream. When the news stream at t_{k+1} comes, we first detect burst words in the newly arriving data, update the BINet and calculate the PageRank value for $G(t_{k+1})$ which denotes the slice of BINet G at time t_{k+1}, which is defined as follows:

$$G(t) = \langle V(t), E(t) \rangle$$

where $V(t) = \{v|t \in \mathcal{P}(v)\}$ and $E(t) = \{e_{i,j}|e_{i,j} \in E \wedge i \in V(t) \wedge j \in V(t)\}$. Then, we can apply NodeRank and AreaRank on $G(t_{k+1})$ to generate a stream summary at t_{k+1}.

5 Experiments and Evaluations

5.1 Experiments on Gigaword corpus

For comparison to the previous work, we use the same data with Ge et al. (2015b) (i.e., 2009 and 2010 APW and XIN news stories in English Gigaword (Graff et al., 2003)) as a news stream. We detect burst words using Kleinberg algorithm (Kleinberg, 2003), which models word burst detection as a burst state decoding problem. In total, there are 140,557 documents in the dataset.

Topic	#Entry	#Entry in corpus
Disaster	35	28
Sports	19	12
Politics	8	5
Military	14	13
Comprehensive	85	64

Table 2: The number of event entries in the reference summaries. The third column is the number of event entries excluding those events that do not appear in the corpus.

We removed stopwords and used Stanford CoreNLP (Manning et al., 2014) to do lemmatization and named tagging, and built BINets on the news stream during 2009 and 2010 separately. On the 2009 news stream, there are 31,888 nodes and 833,313 edges while there are 32,997 nodes and 825,976 edges on the 2010 stream.

Ge et al. (2015b) used manually edited event chronicles of various topics on the web[3] during 2009 as reference summaries for summarizing the news stream during 2010. The information of the reference summaries is summarized in Table 2. In evaluation, they pooled entries in stream sumamries generated by various approaches, annotated each entry based on the reference summary and the manually edited event chronicles on the web, and used *precision@K* to evaluate the quality of top K event entries in a stream summary instead of using *ROUGE* (Lin, 2004) because news stream summaries are event-centric.

In this paper, we adopt the same evaluation setting and use the same reference summaries and the annotations with our previous work (Ge et al., 2015b) to evaluate our summaries' quality. For the event entries that are not in Ge et al. (2015b)'s annotations, we have 3 human judges annotate them according to the previous annotation guideline and consider an entry correct if it is annotated as correct by at least 2 judges.

We evaluate our approaches by comparing to Ge et al. (2015b)'s approach and the baselines in their work:

- RANDOM: this baseline randomly selects documents in the dataset as event entries.
- NB: this baseline uses Naive Bayes to cluster documents for event detection and ranks the clusters based on the combination score of topical relevance and the event impact (i.e., event cluster size). The earliest documents in the top-ranked clusters are selected as entries.
- B-HAC: similar to NB except that BurstVSM representation (Zhao et al., 2012) is used for event detection using Hierarchical Agglomerative Clustering algorithm.
- TAHBM: similar to NB except that the state-of-the-art event detection model (TaHBM) proposed by Ge et al. (2015b) is used for event detection.
- Ge et al. (2015b): the state-of-the-art stream summarization approach which used TaHBM to detect events and L2R model to rank events.

Note that we did not compare with previous multi-document summarization models because the goal and setting of stream summarization are different from multi-document summarization, as Section 1

[3] http://www.mapreport.com; http://www.infoplease.com; https://en.wikipedia.org/wiki/2009

	sports		politics		disaster		military		comprehensive	
	P@50	P@100	P@50	P@100	P@50	P@100	P@50	P@100	P@50	P@100
Random	0.02	0.08	0	0	0.02	0.04	0	0	0.02	0.03
NB	0.08	0.12	0.18	0.19	0.42	0.36	0.18	0.17	0.38	0.31
B-HAC	0.10	0.13	0.30	0.26	0.50	0.47	0.30	0.22	0.36	0.32
TaHBM	0.18	0.15	0.30	0.29	0.50	0.43	0.46	0.36	0.38	**0.33**
Ge et al. (2015b)	0.20	0.15	0.38	**0.36**	0.64	0.53	**0.54**	0.41	**0.40**	**0.33**
BINet-NodeRank	0.24	0.20	0.38	0.30	0.54	0.51	0.48	0.43	0.36	**0.33**
BINet-AreaRank	**0.40**	**0.33**	**0.40**	0.34	**0.80**	**0.62**	0.50	**0.49**	0.32	0.30

Table 3: Performance of various approaches on stream summarization on five topics.

discussed. Moreover, these two tasks differ greatly in the data size and redundancy identification mechanism. Therefore, it is not feasible to directly compare multi-document summarization models to our approaches unless they are adapted for our setting.

The results are shown in Table 3. It can be clearly observed that BINet-based approaches outperform baselines and perform comparably to the state-of-the-art model on generating the summaries on most topics: AreaRank achieves the significant improvement over the state-of-the-art model on sports and disasters, and performs comparably on politics and military and NodeRank's performance achieves the comparable performance to previous state-of-the-art model though it is inferior to AreaRank on most topics. Among these five topics, almost all models perform well on disaster and military topics because disaster and military reference summaries have more entries than the topics such as politics and sports and topics of event entries in the summaries are focused. The high-quality training data benefits models' performance especially for AreaRank which is purely data-driven. In contrast, on sports and politics, the number of entries in the reference summaries is small, which results in weaker supervision and affect the performance of models. It is notable that AreaRank does not perform well on generating the comprehensive summary in which topics of event entries are miscellaneous. The reason for the undesirable performance is that the topics of event entries in the comprehensive reference summary are not focused, which results in very few reference (positive) examples for each topic. As a result, the miscellaneousness of topics of positive examples makes them tend to be overwhelmed by large numbers of negative examples during training the model, leading to very week supervision and making it difficult for AreaRank to learn the patterns of positive examples. Compared to AreaRank, the strategy of selecting documents for generating event entries in other baselines and NodeRank use more or less heuristic knowledge, which makes these models perform stably even if the training examples are not sufficient.

Model	**Features**	**Precision@100**
NodeRank	$w(v)$	0.18
	$w(v)$+$pr(v)$	0.22
	$w(v)$+$\mathcal{C}(v)$	0.46
	$w(v)$+$pr(v)$+$\mathcal{C}(v)$	**0.51**
AreaRank	$\boldsymbol{w}(\mathcal{A})$	0.25
	$\boldsymbol{w}(\mathcal{A})$ + $\boldsymbol{pr}(\mathcal{A})$	0.34
	$\boldsymbol{w}(\mathcal{A})$+$\mathcal{C}(\mathcal{A})$	0.58
	$\boldsymbol{w}(\mathcal{A})$+$\boldsymbol{pr}(\mathcal{A})$+$\mathcal{C}(\mathcal{A})$	**0.62**

Table 4: Ablation test on feature combination for generating disaster summaries.

Model	Topic	Irrelevant	Minor	Redundant
NodeRank	disaster	35.3%	64.7%	0
	sports	21.3%	77.5%	1.3%
	comprehensive	-	100%	0
AreaRank	disaster	34.2%	63.1%	2.6%
	sports	7.5%	91.1%	1.5%
	comprehensive	-	100%	0

Table 5: Error analysis of BINet-based approaches.

We conducted an ablation test to study the effects of features on generating summaries in our model. Table 4 shows the performance of models using various feature combination on generating disaster summaries. In both NodeRank and AreaRank models, PageRank features enhance the models that only use word features of nodes, demonstrating the effects of global importance analysis on the stream level. Context features are also useful for improving the results because words (both burst and non-burst words) in context can help the model learn the preference of topics and styles from the reference summary.

We conducted error analysis for NodeRank and AreaRank, shown in Table 5. Among topically irrelevant, minor and redundant event entries, minor (i.e.,

Model	Module	Run time	Can be run in parallel
BINet	burst detection	14ms per word	Yes
	BINet construction	213.88s on 1-year news	Partially
	PageRank	1.36s per iteration	No
	Ranking	negligible	No
Ge et al. (2015b)	Event detection	1,018s per iteration	No
	Ranking	negligible	No

Table 6: Run time of BINet-based approaches and Ge et al. (2015b)'s approach

trivial) event entries that are not important enough to be included in the stream summary account for the majority of errors for both models. This is because it is difficult to distinguish these trivial events since the corpus we used as a text stream is not as ideal as the assumption that the more important events, the more times they are reported. As shown in Table 2, many entries in the reference summaries even do not appear or burst in our corpus because the Gigaword corpus used is just a small sample of news stream during the period. As a result, the importance features (e.g., PageRank value) in our ranking model do not work very well for distinguishing trivial events.

At last, we tested the run time of our BINet approach and compare to the state-of-the-art model proposed by Ge et al. (2015b) in terms of efficiency. The results are shown in Table 6. The run time is tested on a workstation with Intel Xeon 3.5 GHz CPU and 64GB RAM. The efficiency of our model is much better than Ge et al. (2015b)'s approach whose event detection model takes much time to iterate thousands of times for Gibbs sampling. For memory cost, the peak memory cost of our BINet-based approaches is 5GB while Ge et al. (2015b)'s approach needs more than 10GB memory to run the event detection model and thus cannot work on a large dataset.

5.2 Experiments on a real-time news stream

To evaluate our approaches in a real setting, we create a benchmark dataset[4] containing 7.9 million English news stories (without exact duplication) during Feb 5 to Mar 31, 2015, collecting from Bing news portal[5]. On average, there are approximately 150,000 news documents per day.

We applied our BINet-based approaches (i.e., NodeRank and AreaRank) on the real-time stream. Specifically, we used news stream during Feb 5 to Mar 23 for training to generate news summaries for every day during Mar 24 to Mar 30 in an online fashion. This is a practical setting and can be useful for automatically generating headline news every day.

Models	Disaster	Attack
Random	0.012	0.019
Online-B-HAC	0.096	0.138
NodeRank	0.111	0.153
AreaRank	**0.182**	**0.157**

Table 7: MRR of BINet-based approaches on generating summaries for the real-time news stream.

Daily news summaries in *Current Event Portal*[6] at Wikipedia are used as reference summaries for training and gold standard for evaluating our approaches. In this paper, we tested on generating summaries on *Disaster and accident* (Disaster) and *Armed conflicts and attacks* (Attack) topics. Instead of evaluating *Precision@K* as we did on the Gigaword corpus which is a small dataset, we used *Mean Reciprocal Rank (MRR)* which is defined as follows to see the ranking position of event entries of the gold standard in the summaries generated by our approaches:

$$MRR = \frac{\sum_{t \in T_{test}} (\sum_{\boldsymbol{e_k} \in \mathcal{E}^{(t)}_{gold}} \frac{1}{rank^{(t)}_{\boldsymbol{e_k}}})}{\sum_{t \in T_{test}} |\mathcal{E}^{(t)}_{gold}|} \quad (5)$$

where $\mathcal{E}^{(t)}_{gold}$ is the gold standard summaries at time t, T_{test} is the period of test set (i.e., Mar 24 to Mar 30) and $rank^{(t)}_{\boldsymbol{e_k}}$ is the highest rank of an event entry $\boldsymbol{e_k}$ of the gold standard summary in our summary at t. A high MRR means the event entries of gold standard tend to be ranked at top positions in our generated summaries. The evaluation is conducted manually.

Table 7 shows the performance of BINet-based

[4]The dataset and the gold standard are available at http://getao.github.io

[5]https://www.bing.com/news

[6]https://en.wikipedia.org/wiki/Portal:Current_events/

approaches on the real-time news stream. The BINet-based approaches achieve better results than the online version of B-HAC model on both topics, demonstrating the advantages of the BINet representation. It is also notable that AreaRank performs better than NodeRank because it scores a document area as a whole by taking into account various information of the area. For AreaRank, MRR on the disaster topic is about 0.2, meaning that the average ranking position of gold standard event entries is 5, which is a promising result and shows our approach can be effective to find key information. More importantly, it only takes 500 seconds to build a BINet and 388 seconds to run PageRank for 1,000 iterations for global importance analysis on the 7.9 million documents while other methods in Table 3 even cannot be applied on the stream because they cannot handle so large scale of data or work in an online fashion, which is why we did not compare to them in this setting.

6 Related Work

Stream summarization is not a hot topic in NLP community. Despite the related work that studies corpus summarization of research papers (Sipos et al., 2012), Ge et al. (2015b) is the only work exactly dealing with the news stream summarization challenge. However, they studied the problem on a static timestamped corpus instead of on a dynamic text stream and their proposed pipeline-style approach cannot be applied on a real-time text stream due to high complexity in time and space. Other previous work dealing with stream data is mainly focused on topic and event detection (Yang et al., 1998; Swan and Allan, 2000; Allan, 2002; He et al., 2007; Sayyadi et al., 2009; Sakaki et al., 2010; Zhao et al., 2012; Ge et al., 2015a), dynamic language and topic modelling (Blei and Lafferty, 2006; Iwata et al., 2010; Wang et al., 2012; Yogatama et al., 2014), incremental (temporal) summarization and timeline generation for one major news event (Allan et al., 2001; Hu et al., 2011; Yan et al., 2011; Lin et al., 2012; Li and Li, 2013; Kedzie et al., 2015; Tran et al., 2015; Yao et al., 2016), a sports match (Takamura et al., 2011) or users on the social network (Li and Cardie, 2014).

Different from traditional single and multi-document summarization (Carbonell and Goldstein, 1998; Lin, 2004; Erkan and Radev, 2004; Conroy et al., 2004; Li et al., 2007; Wan and Yang, 2008; Chen and Chen, 2012; Wan and Zhang, 2014) whose focus is to select important sentences, the focus of stream summarization is to select representative documents referring to important news events. The novel paradigm focuses on the summarization problem in the big data age and is useful for many applications.

7 Conclusions and Future work

In this paper, we study the news stream summarization problem by proposing a novel text stream representation – Burst Information Networks and presenting two summarization models based on it. The proposed approaches can efficiently generate high-quality summaries, achieving the state-of-the-art performance. Moreover, the experiments on our created benchmark dataset showed our approach can be effectively applied on the real-time news stream for finding key information, demonstrating its potential values for many real-world applications (e.g., personalized headline news recommendation).

In the future, we plan to generalize the stream summarization problem to various streams such as social (e.g., Twitter), image (e.g., Imgur) and even video streams (e.g., Youtube), which would yield many interesting and practical applications (Lu et al., 2016) to deal with the information overload challenge in the big data era.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. We also want to specially thank Prof. Heng Ji for her valuable suggestions and discussion on the early ideas of this work. This work is supported by the National Key Basic Research Program of China (No.2014CB340504) and the National Natural Science Foundation of China (No.61375074,61273318). The contact author is Zhifang Sui.

References

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *SIGIR*.

James Allan. 2002. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *ICML*.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*.

Chien Chin Chen and Meng Chang Chen. 2012. Tscan: A content anatomy approach to temporal topic summarization. *Knowledge and Data Engineering, IEEE Transactions on*, 24(1):170–183.

John M Conroy, Judith D Schlesinger, Jade Goldstein, and Dianne P Oleary. 2004. Left-brain/right-brain multi-document summarization. In *DUC*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479.

Tao Ge, Wenzhe Pei, Baobao Chang, and Zhifang Sui. 2015a. Distinguishing specific and daily topics. In *APWeb*.

Tao Ge, Wenzhe Pei, Heng Ji, Sujian Li, Baobao Chang, and Zhifang Sui. 2015b. Bring you to the past: Automatic generation of topically relevant event chronicles. In *ACL*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

Qi He, Kuiyu Chang, and Ee-Peng Lim. 2007. Using burstiness to improve clustering of topics in news streams. In *ICDM*.

Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K Usadi, and Xiaoyan Zhu. 2011. Generating breakpoint-based timeline overview for news topic retrospection. In *ICDM*.

Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. 2010. Online multiscale dynamic topic models. In *KDD*.

Thorsten Joachims. 2006. Training linear svms in linear time. In *SIGKDD*.

Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting salient updates for disaster summarization.

Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

Jiwei Li and Claire Cardie. 2014. Timeline generation: Tracking individuals on twitter. In *WWW*.

Jiwei Li and Sujian Li. 2013. Evolutionary hierarchical dirichlet process for timeline summarization. In *ACL*.

Sujian Li, You Ouyang, Wei Wang, and Bin Sun. 2007. Multi-document summarization using support vector regression. In *DUC*. Citeseer.

Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. 2012. Generating event storylines from microblogs. In *CIKM*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8.

Di Lu, Clare Voss, Fangbo Tao, Xiang Ren, Rachel Guan, Rostyslav Korolov, Tongtao Zhang, Dongang Wang, Hongzhi Li, Taylor Cassidy, Heng Ji, Shih-fu Chang, Jiawei Han, William Wallace, James Hendler, Mei Si, and Lance Kaplan. 2016. Cross-media event extraction and recommendation. In *NAACL Demo Session*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*.

Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. 2009. Event detection and tracking in social streams. In *ICWSM*.

Ruben Sipos, Adith Swaminathan, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Temporal corpus summarization using submodular word coverage. In *CIKM*.

Russell Swan and James Allan. 2000. Automatic generation of overview timelines. In *SIGIR*.

Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. 2011. Summarizing a document stream. In *Advances in Information Retrieval*, pages 177–188. Springer.

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *Advances in Information Retrieval*.

Xiaojun Wan and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. In *SIGIR*.

Xiaojun Wan and Jianmin Zhang. 2014. Ctsum: extracting more certain summaries for news articles. In *SIGIR*.

Chong Wang, David Blei, and David Heckerman. 2012. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *SIGIR*.

Yiming Yang, Tom Pierce, and Jaime Carbonell. 1998. A study of retrospective and on-line event detection. In *SIGIR*.

Jin-ge Yao, Feifan Fan, Wayne Xin Zhao, Xiaojun Wan, Edward Chang, and Jianguo Xiao. 2016. Tweet timeline generation with determinantal point processes. In *AAAI*.

Dani Yogatama, Chong Wang, Bryan R Routledge, Noah A Smith, and Eric P Xing. 2014. Dynamic language models for streaming text. *Transactions of the Association for Computational Linguistics*, 2:181–192.

Wayne Xin Zhao, Rishan Chen, Kai Fan, Hongfei Yan, and Xiaoming Li. 2012. A novel burst-based text representation model for scalable event detection. In *ACL*.