Journal of Artificial Intelligence Research 74 (2022) 1037-1057 Submitted 08/2021; published 06/2022

Two-phase Multi-document Event Summarization on Core Event Graphs

Zengjian Chen CHAMKIN1105@GMAIL.COM
WeChat, Tencent Inc.
Huazhong University of Science and Technology
Shenzhen, Guangdong, China

Jin Xu (*corresponding author*) JINXU@SCUT.EDU.CN
School of Future Technology, South China University of Technology
Guangzhou, Guangdong, China

Meng Liao MARICOLIAO@TENCENT.COM
Tong Xue XAVIERXUE@TENCENT.COM
WeChat, Tencent Inc.
Shenzhen, Guangdong, China

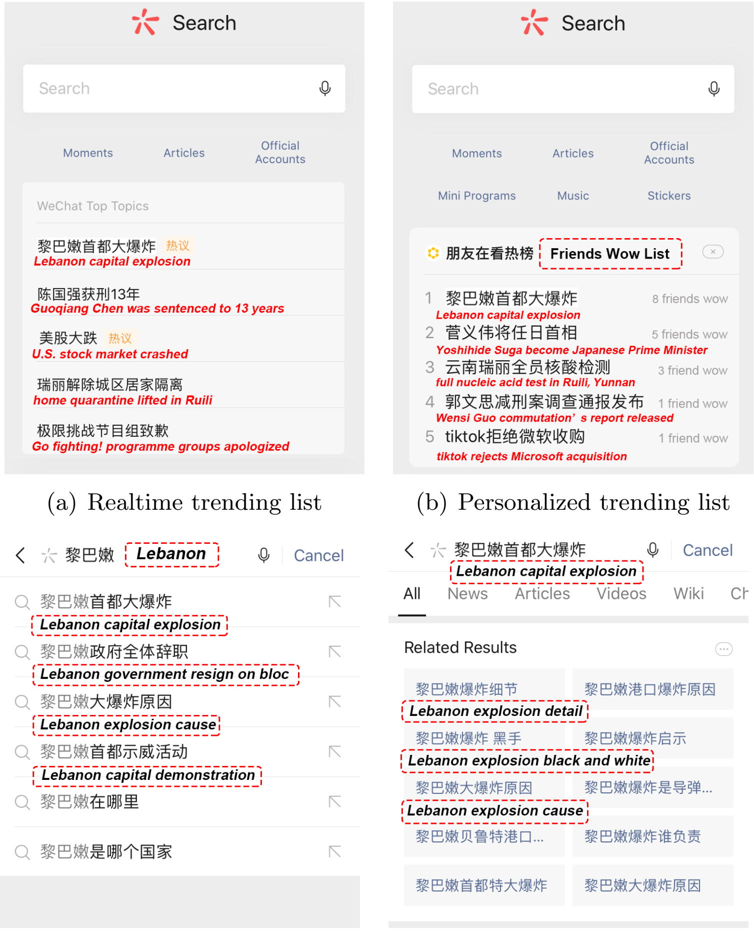
Kun He (*corresponding author*) BROOKLET60@HUST.EDU.CN
Huazhong University of Science and Technology,
Wuhan, Hubei, China

Abstract

Succinct event description based on multiple documents is critical to news systems as well as search engines. Different from existing summarization or event tasks, Multi-document Event Summarization (MES) aims at the query-level event sequence generation, which has extra constraints on event expression and conciseness. Identifying and summarizing the key event from a set of related articles is a challenging task that has not been sufficiently studied, mainly because online articles exhibit characteristics of redundancy and sparsity, and a perfect event summarization needs high level information fusion among diverse sentences and articles. To address these challenges, we propose a two-phase framework for the MES task, that first performs event semantic graph construction and dominant event detection via graph-sequence matching, then summarizes the extracted key event by an event-aware pointer generator. For experiments in the new task, we construct two large-scale real-world datasets for training and assessment. Extensive evaluations show that the proposed framework significantly outperforms the related baseline methods, with the most dominant event of the articles effectively identified and correctly summarized.

1. Introduction

With the information explosion on the web and internet, massive amounts of online articles are constantly being generated by media providers and individuals, drowning the readers in a sea of information. Such trend demands search engines and news systems to grasp the main events from redundant articles and generate more refined event description for guideline and reading. Automatic event summarization from multiple articles is of great value for the reading and search experience, offering readers an overview and quick-perception of trending topics or breaking news. As shown in Figure 1, a news system needs to generate (a) a real-time trending list, or (b) personalized trending recommendation based on article clustering (Topic Detection and Tracking) (Yang et al., 2002) and multi-document event summarization. Search engines can adopt (c) query suggestion and (d) relative search hint for users, which is also an extension of concise event summarization.



黎巴嫩首都大爆炸
Lebanon capital explosion
陈国强获刑13年
Guoqiang Chen was sentenced to 13 years
美股大跌
U.S. stock market crashed
瑞丽解除城区居家隔离
home quarantine lifted in Ruili
极限挑战节目组致歉
Go fighting! programme groups apologized

(a) Realtime trending list

1 黎巴嫩首都大爆炸 8 friends wow
Lebanon capital explosion
2 菅义伟将任日首相 5 friends wow
Yoshihide Suga become Japanese Prime Minister
3 云南瑞丽全员核酸检测 3 friend wow
full nucleic acid test in Ruili, Yunnan
4 郭文思减刑案调查通报发布 1 friend wow
Wensi Guo commutation’ s report released
5 tiktok拒绝微软收购 1 friend wow
tiktok rejects Microsoft acquisition

(b) Personalized trending list

黎巴嫩 Lebanon
黎巴嫩首都大爆炸
Lebanon capital explosion
黎巴嫩政府全体辞职
Lebanon government resign on bloc
黎巴嫩大爆炸原因
Lebanon explosion cause
黎巴嫩首都示威活动
Lebanon capital demonstration
黎巴嫩在哪里
黎巴嫩是哪个国家

(c) Query suggestion

黎巴嫩首都大爆炸
Lebanon capital explosion
黎巴嫩爆炸细节
Lebanon explosion detail
黎巴嫩港口爆炸原因
黎巴嫩爆炸 黑手
Lebanon explosion black and white
黎巴嫩爆炸启示
黎巴嫩大爆炸原因
Lebanon explosion cause
黎巴嫩爆炸是导弹...
黎巴嫩贝鲁特港口...
黎巴嫩爆炸谁负责
黎巴嫩首都特大爆炸
黎巴嫩大爆炸原因

(d) Relative search

Figure 1: Concise event descriptions have wide range of application scenarios and they are crucial to news systems and search engines.

Formally, Multi-document Event Summarization (MES) is the process of distilling the most important information from multiple web-articles to precisely describe the core events. Different from previous summarization (Radev et al., 2004; Yasunaga et al., 2017a) or event tasks (Walker et al., 2006; Mitamura et al., 2017), MES is a new event summarization task with particular constrains in *conciseness* and *event expression*. For instance, conventional multi-document summarization (Barzilay et al., 2002) aims at generating a multi-sentence summary from a collection of documents, while MES is a higher level abstraction that targets at a query-level summary of the core event, which means the topic event should be expressed in a few words (less than 14 Chinese characters in our task). Further, instead of focusing on specified event arguments extraction, MES is essentially an event sequence generation task that summarizes the total event in fluent natural language (see examples in Figure 1).

However, summarizing the query-level event from multiple redundant articles is not straightforward. Articles of a cluster often differ in focus or point of view for a topic, and sometimes contain several sub-events related to the topic event, making it difficult to extract the most dominant event from an article cluster. Meanwhile, the event elements that constitute a complete event expression may be located in different sentences or articles, requiring proper coordination and reorganization of the event elements. Hence, the following questions lie to be addressed: *How to identify the core event from multiple related articles with complete event elements? How to summarize the extracted event in a fairly concise and fluent manner?*

The encoder-decoder neural network models exhibit strong representation capability in the generation (Radford et al., 2019) and summarization tasks (Bahdanau et al., 2015; Vougiouklis et al., 2020). However, they still meet issues when dealing with multi-document setting that requires additional redundancy elimination and cross-document relation capturing. Graph-based methods (Glavaš et al., 2014; Li et al., 2016) are feasible approaches to extract relationships among various sentences or documents and thus could identify the core event structure, but graphs are built at the cost of syntactic information loss.

In this work, we propose a two-phase framework combined with both graph extraction and neural summarization to address the particular challenges of the MES task, in which we first extract the core event and then summarize the event summary. At the first phase, we adopt a graph-based method to extract the topic event in sentence manner and graph manner, where the core event semantic graph is first constructed with key sentence selection and dependency parsing, and then a representative sentence is chosen with a graph-sentence matching procedure. At the second phase, an event-aware pointer generator (Event-Pg) is introduced to summarize the extracted event sequence and event semantic graph into succinct event sequence, which possesses the ability to integrate different event elements of separate articles and ensure fluency. Specifically, a context-aware event pointer is included compared with the original pointer generator (See et al., 2017) and the event distribution can be iteratively updated to better explain the target word given the context of the source material and inherent semantics in texts, making the learned event pointer points to the most suitable and expressive event elements.

Although there exist many datasets for text summarization (Over, 2003; Hermann et al., 2015; Grusky et al., 2018; Fabbri et al., 2019), the query-level summarization based on multiple articles is a largely unexplored area without any public labeled dataset. To facilitate evaluation and further research on MES, we have created two large-scale datasets, one annotated by professional editors, while the other be collected from crawling and search results.

In a nutshell, our contributions are summarized as follows:

- We propose to address a new and challenging task: multi-document event summarization (MES), which aims at a query-level event summarization from multiple articles. We have also constructed two large scale real-world datasets for the training and evaluation.

- We propose a two-phase framework to address this challenging MES task, in which we first adopt graph-based event identification and then integrate the event sentence and event graph with an event-aware pointer generator for sequence generation.

- Experimental results show that the proposed model significantly outperforms the baselines designed for related tasks, demonstrating the validity and superiority of the proposed model.

2. Related Work

Our work touches several strands of research, including multi-document summarization, event extraction and headline generation.

2.1 Multi-document Summarization

Existing multi-document summarization methods mainly focus on sentence-level summarization and can be categorized into extractive and abstractive methods. Most extractive methods are operated over graph-based representations of sentences or passages with edge weights computed by tf-idf (Erkan et al., 2004), discourse relations (Christensen et al., 2013) or sentence embeddings (Yasunaga et al., 2017b) and then a specific algorithm is further adopted for ranking text units for inclusion in the final summary. More recently, some extractive summarization works also utilize graph convolutional networks for salient sentences estimation (Kipf et al., 2016) and sentence ordering (Yin et al., 2019). Abstractive models, especially neural abstractive ones, have achieved promising results on single-document summarization (See et al., 2017; Paulus et al., 2018; Lewis et al., 2019). However, the extension of sequence-to-sequence architectures to multi-document summarization is less straightforward due to the lack of sufficient training data and the computational challenge of processing multiple documents. Intuitively, graph-based extractive methods are suitable to identify relationship of different sentences or documents and extract salient information, while neural sequence-to-sequence architectures are effective in abstraction and content rewriting. Hence, abstractive models based on graphs gain much attention (Yasunaga et al., 2017b; Li et al., 2020). Our model, which also combines a graph-based extractive module and neural abstractive sequence-to-sequence architecture, is a higher level summarization and focuses on the core event summarization from multiple documents, which is more challenging.

2.2 Event Extraction

With similar target that focuses on event, the event extraction tasks (e.g., ACE2005 (Walker et al., 2006) and TAC-KBP2017 (Mitamura et al., 2017)) typically assume that all event schemata are given, and event components are recognized into the knowledge based structure, such as event triggers labeling (Bronstein et al., 2015), event nuggets detection (Reimers et al., 2015) and event arguments extraction (Li et al., 2013). With the introduction of deep neural networks like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Graph Convolutional Networks (GCNs), event extraction tasks have made considerable progress in recent years (Nguyen et al., 2016; Zeng et al., 2016; Mehta et al., 2019). However, to further generate a fluent and complete event sequence, it usually requires additional coordination and reorganization among various event components beyond the above work, which is also a challenging task due to the topic diversity and unstructured event description. Differently, the MES task, which is essentially an event sequence generation task, aims at generating concise event description sequence instead of focusing on some specified arguments of the event, and summarize the event in an unstructured schema from a cluster of articles.

2.3 Headline Generation

Similar to MES, headline generation is also a high level summarization task, which aims to construct a headline-style abstract describing the salient theme in a single document. As a specific type of single-document summarization, headline generation has made great progress by taking advantage of the sequence-to-sequence architecture (Sutskever et al., 2014; Bahdanau et al., 2015) along with the large dataset English Gigaword (Napoles et al., 2012). Following the task setting, encoder-decoder models equipped with syntactic information (Takase et al., 2016), selective gate (Zhou et al., 2017), template (Wang et al., 2019) and pre-training (Dong et al., 2019; Song et al., 2019) are successively proposed for better representation and more precise generation. However, existing methods mainly focus on summarization from a single document (without title) instead of considering various event elements contained in multiple documents of the same topic, which means they may meet challenges in the recognition of core event information and further coordination. Besides, a perfect short event description usually has more constraint in conciseness and event-centric expression than headlines, and has wider potential applications (Niu et al., 2014; Yang et al., 2021). For example, "Beirut explosion victim Isaac Oehlers was fatally struck by glass in his highchair." is a good headline for a single article, but may be too fragmentary and redundant for searching or recommendation of a total event topic.

To conclude, MES is essentially different to the above tasks in conciseness, event expression and article redundancy, in which previous related methods cannot properly address.

3. Methodology

3.1 Task Definition and Model Overview

We treat the multi-document event summarization as a natural language generation task (Gatt et al., 2018) that automatically abstracts short event description from the input documents. Given N documents $\mathcal{D} = \{d_1, d_2, ..., d_N\}$, our goal is to generate the core event sequence $Y = (y_1, y_2, ..., y_M)$ with M words.

Our method summarizes the event sequence in two phases: dominant event identification and neural event sequence summarization. Figure 2 illustrates the overall architecture of our two-phase framework. At the first phase, we first build an event semantic graph from the input articles and then adopt graph-level extraction for core event semantic graph G_e and sentence-level extraction for representative event sentence S_{event}. At the second phase, we further adopt an event-aware pointer generator for event sequence generation, which is a neural summarization model that incorporates the semantic information of event graph and the copy mechanism of pointer network.

3.2 Dominant Event Detection

The goal of *phase 1* is to extract the most dominant event information from redundant articles in sequence manner and global manner, respectively. To this end, we propose two kinds of event detection: sentence-level event detection and graph-level event detection.

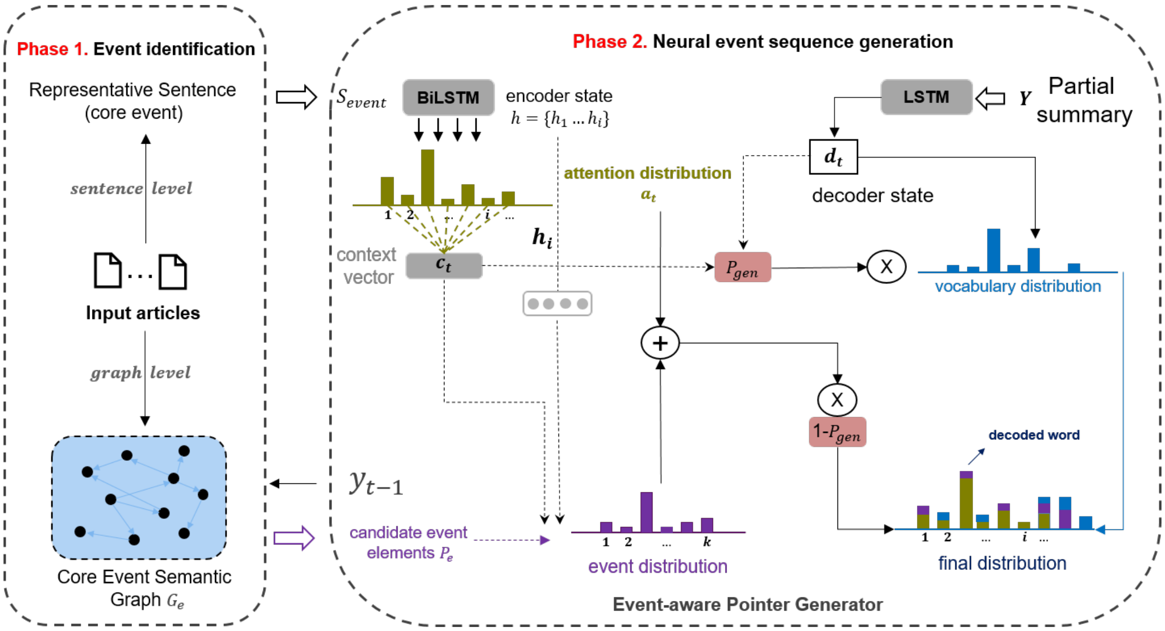

Phase 1. Event identification
Representative Sentence (core event)
sentence level
Input articles
graph level
Core Event Semantic Graph G_e
Phase 2. Neural event sequence generation
S_{event}
BiLSTM
encoder state $h = \{h_1 \dots h_i\}$
LSTM
Y
Partial summary
d_t
decoder state
attention distribution a_t
h_i
context vector
c_t
P_{gen}
X
vocabulary distribution
1-P_{gen}
decoded word
y_{t-1}
candidate event elements P_e
event distribution
final distribution
Event-aware Pointer Generator
1 2 ... i ...
1 2 ... k

Figure 2: The overall architecture of our two-phase MES framework. At the event identification phase, input articles are fed into a graph-based extraction network for sentence-level and graph-level event detection. Then, at the generation phase, the extracted representative sentence S_{event} and event semantic graph G_e are further summarized in succinct event sequence through the proposed event-aware pointer generator.

The former targets at the representative sentence of core event, while the latter aims at extracting the complete event as a core event semantic graph.

3.2.1 Event Semantic Graph Construction

Given a document cluster $\mathcal{D}$, we first do word segmentation and named entity recognition with off-the-shelf tools such as Stanford Core [1]. Then, we further extract topic keywords kw_D of document cluster and article keywords kw_{d_i} for each document based on TF-IDF and NER results. Although there are more sophisticated algorithms that may achieve better performance for the keyword extraction, we found that TF-IDF with NER basically covers most of the event elements and name entities, and is more efficient.

Aside from keywords, locating the key event sentences of cluster and reducing redundancy of original long articles is also important for better event detection (Yang et al., 2018). Note that the titles of articles usually play an important role in expressing the event but sometimes are misleading like the title party, we select the title along with a core content sentence of each article as the key event sentence candidates. In particular, the core content sentence is chosen according to similarity between article keywords kw_{d_i} and each

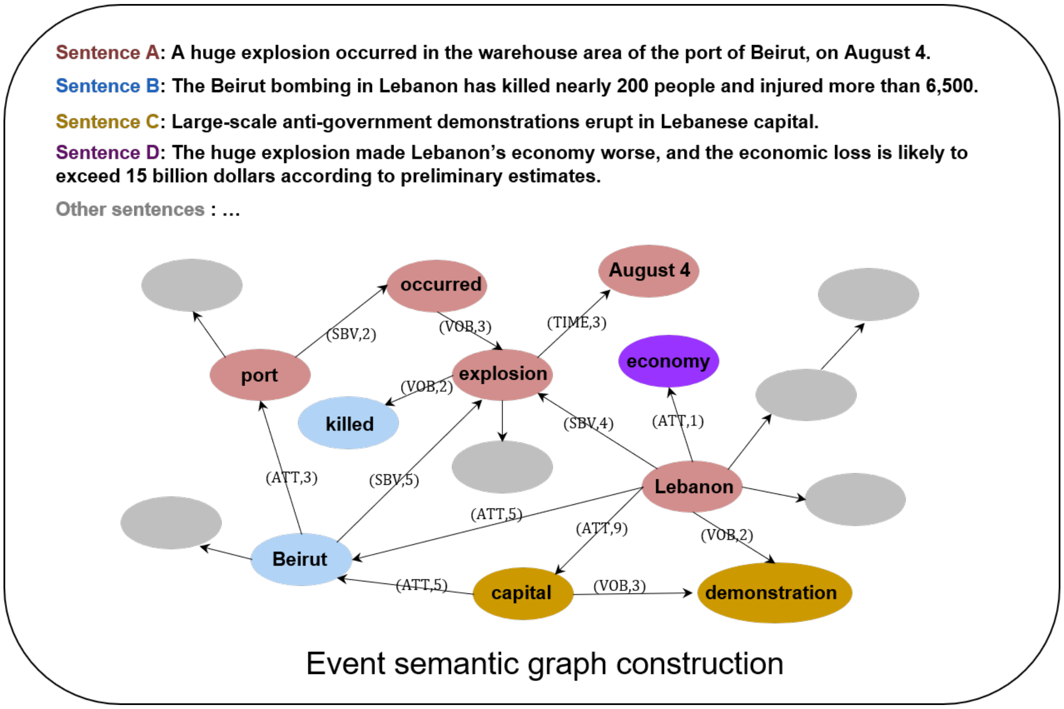
[1] https://stanfordnlp.github.io/CoreNLP

sentence $s^j_{d_i}$:

$$sim(kw_{d_i}, s^j_{d_i}) = \frac{|kw_{d_i}| \cap |s^j_{d_i}|}{|kw_{d_i}| \cup |s^j_{d_i}|}. \tag{1}$$

Thus, $2N$ sentences $s_{cand} = \{s^1_{cand}, s^2_{cand}, ..., s^{2N}_{cand}\}$ are selected as candidates for core event detection. With the candidate sentences, we utilize dependency parsing method to extract semantic structure of different event elements. Noticing that not all words in the candidate sentences are needed event elements, we prune each dependency tree. Particularly, we reserve three important relation edges, including SBV (subject-verb), VOB (verb-object), IOB (indirect-object) and edges that at least one end node word is in the topic keywords kw_D for graph construction. The weight $\hat{w}_{ij}$ of edge $edge_{ij}$ between vertices v_i and v_j is the number of occurrences in dependency trees of candidate sentences. Therefore, an edge between two nodes is defined as $edge_{ij} = (t_{ij}, \hat{w}_{ij})$, where t_{ij} is the type of dependency between nodes.

Finally, the core event semantic graph G_e is constructed from the original articles, as illustrated in Figure 3.



Sentence A: A huge explosion occurred in the warehouse area of the port of Beirut, on August 4.

Sentence B: The Beirut bombing in Lebanon has killed nearly 200 people and injured more than 6,500.

Sentence C: Large-scale anti-government demonstrations erupt in Lebanese capital.

Sentence D: The huge explosion made Lebanon's economy worse, and the economic loss is likely to exceed 15 billion dollars according to preliminary estimates.

Other sentences : ...

Event semantic graph construction

Figure 3: Illustration on the event semantic graph construction.

3.2.2 Representative Event Sentence Selection

Generally, the core event semantic graph can represent the total event in a global manner, which contains core event elements and the inner semantic relation between them. However, summarizing the event sequence directly from the semantic graph is not straightforward. For the event graph, though it can better capture the event relation of different sentences and documents, the graph structure usually loses the original grammatical information in the meantime, which is important to ensure more fluent generation.

Therefore, we also extract the most representative sentence for a neural sequence-to-sequence model to ensure more fluent generation. We utilize a graph-based extraction method to select the most representative sentence from candidate sentences according to the similarity between candidate sentence and core event semantic graph, which considers both keyword similarity and semantic link similarity. Given the core event semantic graph G_e and a candidate sentence $s^i_{cand} = \{w_1, w_2, ...w_L\}$, the similarity score is calculated as:

$$\begin{aligned} score(s^i_{cand}) &= sim_{kw}(s^i_{cand}, kw_D) + \lambda sim_{link}(s^i_{cand}, G_e) \\ &= \frac{N_t}{L} \sum_{j:w_j \in kw_D}^{L} tfidf_{w_j} + \frac{\lambda}{N} \sum_{j=1}^{L} \sum_{t=j}^{L} \hat{w}_{jt}, \end{aligned} \tag{2}$$

where $sim_{kw}(s^i_{cand}, kw_D)$ indicates the similarity between candidate sentence and topic keywords, $sim_{link}(s^i_{cand}, G_e)$ indicates the semantic link similarity between candidate sentence and event graph, N_t is the number of words contained in both candidate sentence and topic keywords and λ is a tunable hyper-parameter to leverage keyword similarity and link similarity. Particularly, $\frac{N_t}{L}$ is designed to ensure choosing the most expressive sentence instead of the longest sentence.

A sentence S_{event} with the highest score is chosen as the representative event sentence, which is regarded as the most similar one with the core event graph.

3.3 Neural Event Sequence Generation

At *phase 1*, we obtain a representative event sentence S_{event} and core event semantic network G_e. At *phase 2*, our goal is then to further summarize the event description Y from the extracted sequence event information and graph event information with a neural summarization system.

3.3.1 Basic Seq2Seq Architecture

For neural summarization, we begin with a basic seq2seq framework, which consists of an encoder and an attention-equipped decoder. We use a two-layer bi-directional LSTM-RNN encoder and a one-layer uni-directional LSTM-RNN decoder along with the attention mechanism (Bahdanau et al., 2015).

Formally, the encoder produces sequential hidden states as $(\overset{\rightharpoonup}{h_1}, ..., \overrightarrow{h_N})$ and $(\overleftarrow{h_1}, ..., \overleftarrow{h_N})$ in the corresponding positions, and the bi-directional $h_i = f_{LSTM}(h_{i-1}, w_i)$. Each word w_i in the sequence can be represented as a concatenation of the bi-directional hidden states, i.e., $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$. The decoder generates a target summary from a vocabulary distribution $P_{vocab}(w)$, which is based on the context vector c_t through the following process:

$$\begin{aligned} P_{vocab}(w) &= P(y_t | y_{<t}, S_{event}; \theta) \\ &= softmax(W_2(W_1[d_t, c_t] + b_1) + b_2), \end{aligned} \tag{3}$$

where d_t is the hidden state of the decoder and c_t is the context vector at time step t. W_1, W_2, b_1, b_2 are trainable parameters.

The context vector c_t is computed by a weighted sum of the hidden representations of the source text, and the weight is denoted as attention $a_{t,i}$:

$$\begin{aligned} c_t &= \sum\nolimits_{i=1}^{N} a_{t,i} h_i, \\ a_{t,i} &= softmax(v^T tanh(W_h h_i + W_d d_t + b)). \end{aligned} \tag{4}$$

The softmax function normalizes the vector of a distribution over the input position, and v, W_h, W_d, b are trainable parameters.

3.3.2 Event-aware Pointer Generator

Pointer networks use attention as a pointer to select segments of the input as outputs (Vinyals et al., 2015). As such, a pointer network is a suitable mechanism for extracting salient event information, while remaining enough flexibility to interface with a seq2seq model for generating an abstractive summarization (See et al., 2017). Our proposed model is essentially an upgrade to this configuration that integrates event semantic information within a unified framework.

Context-aware event attention. The event semantic graph specifies the weight and direction that each event element associated with query word, and we further calculate the probability that query word points to. At time step t, when given the last decoded word y_{t-1}, the probability is defined as $p(e|y_{t-1}) = \hat{w}(y_{t-1}, e)/N$ and then we have a distribution over a set of related event elements. Yet, this raises the question of how to identify a context-appropriate event element for a word from the distributional set of event element candidates. In other words, linked elements with the highest probability may not be most suitable for the context. Formally, given a decoded word y_{t-1}, a set of k event element candidates, $E_t = e_t^1, e_t^2, ...e_t^k$, is pointed to by the word y_{t-1} according to G_e, with distributional probabilities over the event elements, i.e., $P(E|y_{t-1}) = p(e_t^1), p(e_t^2), ...p(e_t^k)$. The task is to find the most suitable event word e_t^j to fit the updated context, represented by the vector c_t in Eq. 2, at time step t.

In the case of generating summaries given updated contexts, a weighted update of the distributional event element candidates needs to be performed. In the model, the updated weight, denoted as Φ_i^j, is estimated by a softmax classifier that is jointly conditioned on the hidden representation of the word h_i, the context vector c_t, and each of the event element vectors:

$$\Phi_j^i = softmax(W_h[h_i; c_t; e_t^j]), \tag{5}$$

where $j \in [1, k]$, W_h is a trainable parameter, and e_t^j is the vector of the j^th event element candidate, which is a representation of the input embedding. Together with the event association probability from the event semantic graph $p(e_t^j)$ and the updated weights based on the context e_t^j, an event-aware semantic probability of the j^{th} event element for time step t, $P_{t;j}^e$ is finally estimated as:

$$P_{t,j}^e = p(e_i^t) + \gamma \Phi_t^j, \tag{6}$$

where γ is a tunable hyper-parameter. Theoretically, we will end up with a number of k relevant event elements for each word $E_t = \{e_t^1, ..., e_t^k\}$ with a probability distribution over the set, which is learned as an event semantic distribution $P_t^e = \{P_{t,1}^e, ..., P_{t,j}^e, ..., P_{t,k}^e\}$.

Following the basic pointer generator network (See et al., 2017), we combine the baseline generation distribution and both copy distributions (attention distribution, event distribution) with a generation probability p_{gen}:

$$\begin{aligned} P_{final}(w) &= p_{gen} P_{vocab}(w) + (1 - p_{gen})(\sum_{i:w_i=w} a_{ti} + \sum_{j:w_j=w} P^e_{t,j}), \\ p_{gen} &= \sigma(W^*_c c_t + W^*_d d_t + W^*_y y_{t-1} + b_{gen}), \end{aligned} \tag{7}$$

where σ is a sigmoid function, $P_{final}(w)$ is the final output distribution of the model and W^*_c, W^*_d, W^*_y, b_{gen} are parameters to be learned.

To this end, both the extracted representative sentence and event semantic graph information are utilized for neural summarization in a context-aware manner.

4. Experiments

4.1 Datasets

For evaluation purpose, since there is no public benchmark dataset for the MES task yet, we construct a "Topic Multi-document Event Summarization" (TMES) dataset by manual edition and a large "Search Multi-document Event Summarization" (SMES) dataset from scratch. All articles of the datasets are collected from WeChat, a widely used mobile social application in China, where both media organizations and personal users can set up their official accounts for publishing news and articles.

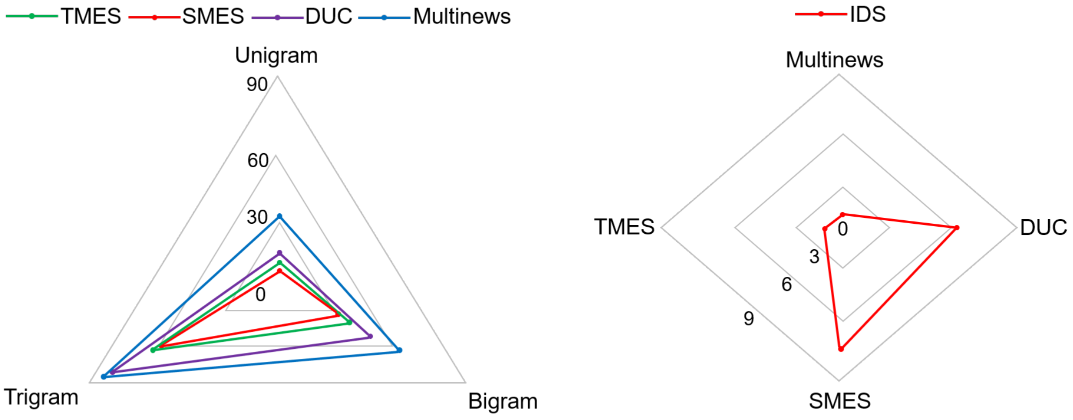
TMES. Articles in a topic set are selected by two steps. First, the total articles are aggregated by a clustering procedure (Topic Detection and Tracking(Allan, 2012)), then less than 10 articles are further selected by professional editors according to the representative degree and diversity. For the event summary annotation of each article set, two editors are asked to edit and review the summaries based on the total articles. Specifically, summaries are first edited by the two editors following some detailed instructions like "core event definition" and "length limit". Then editors act as the quality inspector of each other and conduct revision to avoid low-level errors like typos or grammatical errors. In the end, the annotation consistency of two annotators is up to 87% (without revision) and the rewritten summaries will be further discussed for agreement. To ensure the diversity of event topic, the publication timestamps of the collected articles range from August 2018 to August 2020, and the event categories include disaster, technology, finance, daily life, etc. Finally, we get 8,289 event clusters with a total of 42,341 articles.

SMES. Noticed that large-scale labeled data are essential for the training of neural summarization systems, especially for the event sequence generation task in open domain, we construct another large multi-document event summarization dataset based on search engine. Due to the laborious workload in writing summaries, we crawl event summaries from Chinese social platforms and news apps including Weibo, Zhihu, Tencent News, Baidu and Toutiao. Then, we take the crawled summaries as the queries and get the related event news articles from a news search engine, WeChat Search. In this way, we generate 53,787 pairs of summary-articles automatically, containing the major events from August 2019 to August 2021.

Comparison. Statistics of our datasets [2] are shown in Table 1. To illustrate the difference with previous sumarization datasets, we have also included two well-known multi-document summarization datasets: DUC (Over, 2003; Chali et al., 2004) and Multi-News (Fabbri et al., 2019). As discussed before, the essential difference lies in the summary length, where our event summary is much more concise and the average length is about four words compared to over one hundred words of traditional summaries. None of existing summarization datasets can address our MES problem due to such gap. On the other hand, it has been a long time that datasets are the bottleneck of multi-document summarization with only one dataset of DUC until the large-scale Multi-News dataset be released, revealing that large-scale labeled dataset is critical for deep neural model training. For this consideration, both TMES and SMES contain over 8,000 cluster pairs and the total number of articles of SMES is even more than 661,477, which is nearly five times of the articles in Multi-News. With the comparable size to Multi-News, our two large datasets will be beneficial for deep neural summarization system training and future research.

Dataset	# pairs	# total size (articles)	# average size (articles cluster)	# words (average article)	# words (summary)	# characters (average article)	# characters (summary)
TMES	8,289	42,341	5.11	928.52	4.72	1,573.14	9.05
SMES	53,787	661,477	12.29	802.32	5.20	1,348.29	9.97
DUC03+04	320	1,984	6.20	747.74	109.58	-	-
Multi-News	44,972	125,417	2.78	756.47	263.66	-	-

Table 1: Comparison of our collected event summarization datasets to other multi-document summarization datasets. For the two new Chinese datasets (TMES, SMES), we do word segmentation with the Jieba (Sun, 2012) tool for word counting.



(a) Abstractness.

(b) Inter Document Similarity.

Figure 4: Corpus metrics across datasets.

Analysis. Following (Dey et al., 2020), we further adopt two corpus metrics for analysis, Abstractness and Inter Document Similarity (IDS). Abstractness is defined as the percentage of non-overlapping higher order n-grams between the reference summary and candidate

[2] https://drive.google.com/drive/folders/1QX28zDhkhoHziVyOvtYm0GvmcPTAdI99

documents while IDS is an indicator of the degree of overlap between candidate documents (higher score indicates more similar distributions). As shown in Figure 4(a), TMES and SMES are more extractive compared with previous summarization datasets because the core event elements are mainly extracted from articles for truthfulness and too much novel words may lead to factual error in event expression. For the Inter Document Similarity, TMES has a more similar document distribution compared with SMES as depicted in Figure 4(b) and such distribution difference has an apparent impact on the performance of multi-document summarization systems that will be discussed in Section 5.

4.2 Baselines

To evaluate the effectiveness of the proposed graph dominant event detection and event-aware pointer generator, we implement three types of baselines: concat-based methods with all articles concatenated as input, traditional and neural extract-based methods with extracted sentence as the input, the ruled-based summarization model based on the extracted event graph.

- **S2S-att-concat / S2S-att-extract:** Attention-equipped sequence-to-sequence is the basic neural network for abstractive summarization that contains a two-layer BiLSTM encoder and a one-layer LSTM decoder equipped with attention (Nallapati et al., 2016).

- **Ptr-Net-concat/ Ptr-extract:** Pointer network directly uses the attention mechanism as a pointer to select tokens from the input as the output (Vinyals et al., 2015).

- **Ptr-gen-concat/ Pg-extract:** Pointer generator is a hybrid model combing Seq2seq-att with pointer network (See et al., 2017).

- **mBART-concat/ mBART-extract:** mBART (Liu et al., 2020) is a sequence-to-sequence denoising auto-encoder pretrained on large-scale monolingual corpora in many languages using the BART objective (Lewis et al., 2019) and utilized for various generation tasks including summarization.

- **Trunc.:** Truncation-based method is a simple traditional baseline where words are kept in the original order until the length limit is reached.

- **ILP:** ILP-based method is an unsupervised method that relies on the preprocessing (i.e., NER, term weighting) results of input sequences (Clarke et al., 2008), which is a strong baseline for traditional sentence compression.

- **Graph-gen** is the graph-based summarization baseline utilizing ILP maximization and links of core event semantic graph to select salient information and generate event sequence following some specific templates.

- **Event-Pg** is our event-aware pointer network that integrates event semantic graph information for summarization.

4.3 Parameter Settings

We implement all the mentioned models in Tensorflow except Trunc., ILP and Graph-gen. For the implementation of our Seq2Seq models (i.e., S2S-att, Ptr-Net, Ptr-gen, Event-Pg), we adopt two 128-dimensional LSTMs for the bidirectional encoder and one 256-dimensional LSTM for the decoder. The vocabulary size is set to 50k for both the source text and the target text. We initialize a 128-dimensional word embedding and other learnable parameters following a normal distribution. All models are trained on a single Tesla M40 GPU, and optimized with AdaGrad (batch size = 128). The initial learning rate and the accumulator value were set to 0.15 and 0.1, respectively. We use gradient clipping with a maximum gradient norm of 2, but with no regularization. For hyper-parameter settings, we tune $\gamma = 0.2$ and $\lambda = 0.3$ for our model. At the test time, our short event summaries are produced with a decoder whose beam search size is set to 8 and the maximum decoding step size is set to 15. We randomly select 80% of the data as the training data, and use the remaining data for development and test (10% for each).

4.4 Automatic Evaluation

Summarization systems are usually evaluated using several variants of the recall-oriented ROUGE metric (Lin, 2004). ROUGE measures the summary quality by counting the overlapping units such as n-grams between the generated summary and reference summaries. Following the common practice, we consider ROUGE-1 (uni-grams), ROUGE-2 (bi-grams) and ROUGE-L (longest common subsequence) as our automatic evaluation metrics.

4.5 Human Evaluation

Like related summarization work (Tan et al., 2017; Wang et al., 2018, 2018), we conduct manual evaluation on the generated short event summary to improve the correctness of quality evaluation. Owing to the laborious evaluation process (reading the long articles), we randomly sampled 300 articles-summary pairs from the test set and asked two professional editors to respectively annotate the quality of the generated short summary. Three perspectives are considered during the manual evaluation process: 1) *Accuracy*: Is the core event of articles correctly extracted? 2) *Informativeness*: How informative is the event summary to express the total event? 3) *Readability*: How fluent, grammatically correct the event summary is?

In particular, we use a strict criteria for core event detection accuracy, the generated short event summaries will be assessed as 1 only if it correctly retains the representative event from the article cluster, otherwise 0. The other two properties are assessed with a score from 1 (worst) to 5 (best). Particularly, the final scores of generated event summaries are the average scores of two annotators, with the annotation consistency of 83.8% (label the same score).

4.6 Evaluation Results

The overall evaluation results are shown in Table 2, where methods are divided into five groups: 1) end to end methods with concatenation of articles as input; 2) two-phase methods with the extracted representative event sentence; 3) rule-based method according to our core event semantic graph; 4) neural summarization methods with the extracted representative event sentence as the input; 5) our proposed two-phase event-aware pointer generator.

Method	TMES						SMES					
	RG-1	RG-2	RG-L	Accu.	Info.	Read.	RG-1	RG-2	RG-L	Accu.	Info.	Read.
S2S-att-concat	37.43	27.15	36.83	56.34%	3.28	3.91	41.66	36.14	40.18	65.34%	3.51	4.39
Ptr-Net-concat	35.91	26.56	33.87	52.32%	3.03	3.79	39.07	34.33	38.11	65.29%	3.46	3.98
Ptr-gen-concat	38.62	29.93	38.21	58.75%	3.41	3.79	42.49	37.41	41.74	70.29%	3.86	4.21
mBART-concat	40.27	31.16	39.69	56.93%	3.40	3.82	44.02	38.66	43.24	71.02%	3.81	4.30
Trunc.	42.66	33.44	41.09	79.12%	3.66	2.97	49.97	41.01	48.45	83.31%	4.03	3.08
ILP	46.23	35.17	44.98	82.78%	3.99	3.86	51.01	43.03	50.95	86.52%	4.23	3.95
Graph-gen	45.24	33.15	43.61	75.76%	3.86	2.83	47.21	36.31	44.54	88.33%	4.03	2.65
S2S-att-extract	49.37	36.15	47.78	89.34%	4.48	4.53	56.66	42.54	54.31	94.34%	4.64	4.79
Ptr-Net-extract	52.68	39.54	51.55	91.32%	4.51	4.45	58.07	44.42	57.16	94.29%	4.70	4.68
Ptr-gen-extract	53.91	40.44	52.71	92.26%	4.53	4.51	59.28	49.47	58.57	95.71%	4.71	**4.83**
mBART-extract	54.12	40.55	52.90	92.29%	4.51	4.53	59.79	49.60	58.98	95.69%	4.69	4.82
Event-Pg	**54.86**	**41.91**	**53.53**	**92.96%**	**4.63**	**4.54**	**60.04**	**49.89**	**59.21**	**96.15%**	**4.74**	4.80

Table 2: Overall performance evaluation, including ROUGE, average core event identification accuracy (Accu.), average informativeness score (Info.) and average readability score (Read.). The best results are in **bold** and the second bests are underlined.

Automatic evaluation. As expected, concatenation-based methods perform worst on all metrics due to the redundancy and sparsity of long concatenated sequence, making it extremely difficult to locate the core event information or summarize in concise event summary. With the extracted representative event sentence in advance, extract-then-summarize frameworks perform significantly better than end-to-end methods by a clear gap on ROUGE scores. Even the simplest baseline Trunc. achieves an obvious performance improvement compared to concat-based ones. ILP, which is acknowledged as a strong traditional sentence compression method, performs much better. Though Graph-gen utilizes the core event semantic graph and total event information for event summary generation, it cannot beat sentence-based methods like ILP or S2S-att-extract because summarizing various event sequences from a graph is not direct and often results in organization error (large gap in RG-2). Therefore, neural summarization models with the pre-extracted event sentence are the strongest baselines for MES, where all achieve fairly high ROUGE scores in our experiments. Particularly, for the pretrained abstractive models like mBART, since event summarization tasks are more extractive, they can not vastly outperform pointer-based networks as in other abstractive datasets, and the event information they summarized is limited in the extracted representative sentence. However, with a graph-Seq2Seq framework, Event-Pg integrates both the representative sentence and semantic graph for summarization, which means Event-Pg can retain the strong sequence generation ability of neural Seq2Seq models and meanwhile utilize the important event elements occurred only in other sentences. In summary, Event-Pg gains the best performance on all ROUGE metrics.

Manual evaluation. The results on "readability" metric show that all models built on Seq2Seq architecture can generate more fluent summaries compared to traditional sentence compression methods and Graph-gen. Particularly, Graph-gen performs worst in "readability", revealing that it is infeasible to generate unstructured event summaries directly from semantic graph though it contains complete event elements and semantic information. Our

Event-Pg, which fuses event semantic information and neural summarization, can inherent semantics in texts and achieve better performance in readability. For the core event identification accuracy (Accu.), the large gap between concatenation-based and extract-based methods demonstrates that it is more feasible to summarize the core event sequence in an extract-then-summarize framework when the sources are multiple articles. With the extracted event sentence, even the traditional sentence compression methods can achieve a considerable accuracy on core event detection, which demonstrates effectiveness of our graph-based event extraction procedure. Lastly, the results on "informative" indicate that our two-phase Event-Pg can retain more key event information and generate more informative summaries compared with the baselines. This is because sometimes a complete event does not lie in a single sentence but lies in several sentences or several articles. With the inclusion of context-aware event attention, Event-Pg utilizes event elements occurred only in other sentences or articles and summarizes the event in a global manner.

Considering all the three metrics, our Event-Pg produces more accurate and more informative event summaries, and achieve comparable performance in readability compared with sequence-to-sequence models, showing the advantage of our two-phase framework and event-aware pointer generator.

5. Further Discussion and Analysis

In this section, we first make comparison across TMES and SMES datasets and discuss the reason behind performance deviation. Then we analyze the performance of the dominant event identification phase and compare Event-Pg with various key sentence extraction and event identification methods. Lastly, we conduct study on the ability of event extraction with a public Chinese event detection benchmark.

5.1 Comparison across Datasets

We notice that for both automatic evaluation and manual evaluation, all models (including ours) perform worse on TMES than on SMES. The deviation of the results is mainly due to the differences on data characteristics. For TMES, the event summary are edited manually and written in a more general thinking with the consideration of total article clusters and summary attraction, which means quite a few event sequences can not be summarized from a single representative sentence and often requires supplement other key event elements that are unique in other sentences or articles. In contrast, for SMES, which is collected by query searching, articles of a cluster are closely related to the given event query and have a more similar distribution as shown in Figure 4(b), while articles of TMES are additionally filtered and chosen taking consideration of the diversity and reading experience.

Despite of the data characteristic difference, with the ability of detecting core event in both sequence and graph manner and utilizing the extracted information with an event-aware summarization network, our model yields the best performance for nearly all automatic and manual evaluation metrics (except "readability" of SMES). For the sentence that can express the total event, neural sequence-to-sequence models are good enough for event summarization and Event-Pg is slightly inferior to them in fluency. However, for the TMES, which requires more semantic event information beyond representative sentence, Event-Pg achieves much more performance improvement compared with the baselines, especially in core event identification accuracy and informativeness, demonstrating the effectiveness of the inclusion of context-aware event attention.

5.2 Analysis on Event Identification

Here we focus on the analysis on dominant event identification phase and conduct comparison with various key sentence extraction and event identification methods for subsequent summarization. Specifically, we implement the following extraction methods: 1) Lead-title: choose the title of first article as selected sentence; 2) Text-rank (Mihalcea et al., 2004); 3) Event-ext: sentence selection based on event extraction (Yang et al., 2018); 4) Event-graph: choose representative sentence according to event semantic graph construction and matching; 5) Event-Pg: utilize both graph information and representative sentence for the summarization. Except for Event-Pg, other methods all adopt pointer generator as the summarization model at *phase 2* and evaluate on TMES, which is more challenging and usually contains various sub-events.

The results are shown in Figure 5. Note that the simplest sentence selection method Lead-title gains decent improvements compared to the well-known Text-rank. The reason behind is that editors usually choose a more attractive article or an overview article describing the dominant event at the top position, and the titles, except title party, usually focus on the topic of articles in a concise manner and are reasonable to be chosen as representative sentence to some extent. However, the methods considering event (Event-ext) can select a more suitable and expressive event sentence for the later summarization compared to traditional sentence extraction. Then, considering the semantic links of event elements, Event-graph gains better performance than Event-ext in ROUGE scores. Lastly, our Event-Pg, which selects event sentence according to semantic graph and further fuses both the extracted sequence information and semantic graph information, gains the best performance among all the event sentence selection baselines. Generally, the results on identification analysis show that our framework is a more sensible approach to extract dominant event in both sentence manner and graph manner, ensuring a more representative and more complete event summary generation at the subsequent phase.

5.3 Study on Event Extraction

To further illustrate the ability of Event-Pg on core event element location, we conduct additional event assessment with a public Chinese event detection benchmark called ACE2005 (Walker et al., 2006), where the golden event mentions of each sentence are marked. Take the following sentence as an example "Earlier documents in the case have included embarrassing details about perks Welch received as part of his retirement package from GE.", "retirement", "Welch" and "GE" are marked as "trigger", "Individual" and "Organization" respectively as the golden event mentions. Intuitively, the performance in core event element detection is closely associated with how many golden event mentions are contained in the generated event summary. Therefore, we quantify the assessment criteria as Event Element Extraction Score $EEES(Y) = 10 * \frac{N_c * N_c}{L_e * N_e}$, where N_c is the number of golden event words contained in the generated summary (counted by string matching), N_e is the total number of golden event mentions, L_e is the length of the generated summary. The results are shown in Table 3, revealing that the event summaries generated by Event-Pg contains

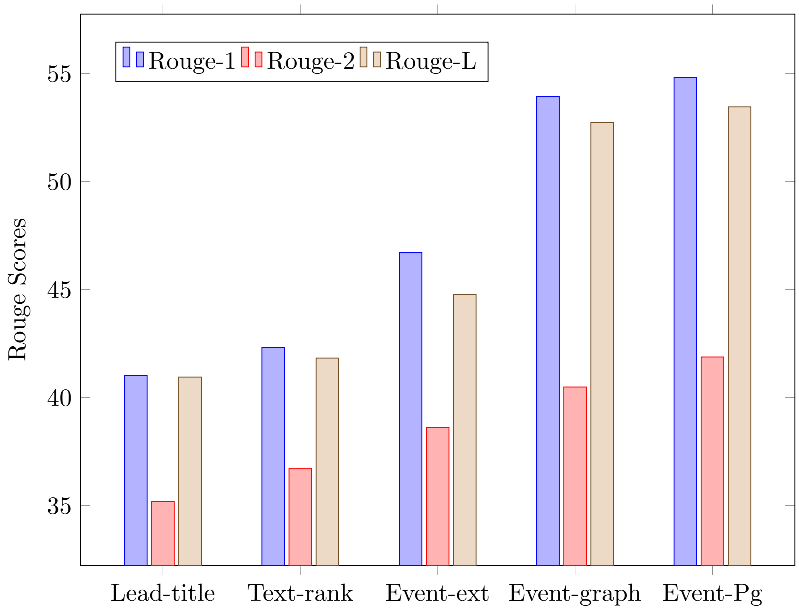


Figure 5: Rouge scores of event identification methods.

more event mentions and acquires the best ability in core event mentions retention. The difference between Event-Pg and Ptr-gen/mBART shows that the inclusion of event semantic information helps the models focus more on the event mentions and thereby generates more informative description sequence expressing the complete event.

Model	**average** L_e	**average** N_c	**EEES**
Trunc.	4.00	1.39	1.07
ILP	4.18	1.93	1.97
S2S-att	4.89	2.32	2.44
mBART	4.52	2.59	3.07
Ptr-gen	4.63	2.65	3.36
Event-Pg	4.59	2.86	**3.95**

Table 3: Comparison on the event assessment.

6. Conclusion

In this work, we propose to address a new and challenging task called multi-document event summarization (MES), which aims at the query-level event sequence generation from multiple related articles. To deal with this task, we propose a two-phase framework, in which we first construct event semantic graph and adopt graph-sequence matching for the graph-level and sentence-level core event identification. Then, we adopt an event-aware pointer generator to summarize the extracted representative event sentence and core event semantic graph, which both utilize the sequence syntactic information and event graph semantic link information. For the purpose of training and assessment, we construct two datasets suitable for the MES task. Extensive experimental results demonstrate the effectiveness of the proposed method compared with the competitive baselines designed for related tasks.

Though pointer-based models are feasible in integrating semantic event graph information and gain better performance over pretrained models like BART, pretrained abstractive methods have been proved to outperform pointer-based models in the generation phase. In future work, we will try to explore the pretrained summarization models based on event graph to better capture the unique event elements of different articles.

References

Allan, J. (2012). *Topic detection and tracking: event-based information organization*, Vol. 12. Springer Science & Business Media.

Bahdanau, D., et al. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Barzilay, R., et al. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, *17*, 35–55.

Bronstein, O., et al. (2015). Seed-based event trigger labeling: How far can event descriptions get us?. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 372–376.

Chali, Y., et al. (2004). Summarization techniques at duc 2004. In *Proceedings of the document understanding conference*, pp. 105–111. National Institute of Standards in Technology (NIST).

Christensen, J., et al. (2013). Towards coherent multi-document summarization. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1163–1173.

Clarke, J., et al. (2008). Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, *31*, 399–429.

Dey, A., et al. (2020). Corpora evaluation and system bias detection in multi-document summarization. *arXiv preprint arXiv:2010.01786*.

Dong, L., et al. (2019). Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pp. 13063–13075.

Erkan, G., et al. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457–479.

Fabbri, A. R., et al. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Gatt, A., et al. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170.

Glavaš, G., et al. (2014). Event graphs for information retrieval and multi-document summarization. *Expert systems with applications*, *41*(15), 6904–6916.

Grusky, M., et al. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 708–719.

Hermann, K. M., et al. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pp. 1693–1701.

Kipf, T. N., et al. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lewis, M., et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, P., et al. (2013). Argument inference from relevant event mentions in chinese argument extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1477–1487.

Li, W., et al. (2016). Abstractive news summarization based on event semantic link network.. Association for Computational Linguistics.

Li, W., et al. (2020). Leveraging graph to improve abstractive multi-document summarization..

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81.

Liu, Y., et al. (2020). Multilingual denoising pre-training for neural machine translation.. Vol. 8, pp. 726–742. MIT Press.

Mehta, S., et al. (2019). Event detection using hierarchical multi-aspect attention. In *The World Wide Web Conference*, pp. 3079–3085.

Mihalcea, R., et al. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411.

Mitamura, T., et al. (2017). Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track.. In *TAC*.

Nallapati, R., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Napoles, C., et al. (2012). Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pp. 95–100.

Nguyen, T. H., et al. (2016). Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 300–309.

Niu, X., et al. (2014). The use of query suggestions during information search. *Information Processing & Management*, *50*(1), 218–234.

Over, P. (2003). An introduction to duc 2003: Intrinsic evaluation of generic news text summarization systems. In *Proceedings of Document Understanding Conference 2003*.

Paulus, R., et al. (2018). A deep reinforced model for abstractive summarization. In *Proceedings of the 6th International Conference on Learning Representations*.

Radev, D. R., et al. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, *40*(6), 919–938.

Radford, A., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Reimers, N., et al. (2015). Event nugget detection, classification and coreference resolution using deep neural networks and gradient boosted decision trees. *Transfer*, *551*, 554.

See, A., et al. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083. Association for Computational Linguistics.

Song, K., et al. (2019). Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Sun, J. (2012). Jieba chinese word segmentation tool..

Sutskever, I., et al. (2014). Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Takase, S., et al. (2016). Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1054–1059.

Tan, J., et al. (2017). Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1171–1181.

Vinyals, O., et al. (2015). Pointer networks. In *Advances in Neural Information Processing Systems*, pp. 2692–2700.

Vougiouklis, P., et al. (2020). Point at the triple: Generation of text summaries from knowledge base triples. *Journal of Artificial Intelligence Research*, *69*, 1–31.

Walker, C., et al. (2006). Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, *57*, 45.

Wang, J., et al. (2018). A multi-task learning approach for improving product title compression with user search log data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, K., et al. (2019). BiSET: Bi-directional selective encoding with template for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Wang, L., et al. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4453–4460.

Yang, H., et al. (2018). Dcfee: A document-level chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pp. 50–55.

Yang, X., et al. (2021). Limited-energy output formation for multiagent systems with intermittent interactions. *Journal of the Franklin Institute*, *358*(13), 6462–6489.

Yang, Y., et al. (2002). Multi-strategy learning for topic detection and tracking. In *Topic detection and tracking*, pp. 85–114. Springer.

Yasunaga, M., et al. (2017a). Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Vancouver, Canada. Association for Computational Linguistics.

Yasunaga, M., et al. (2017b). Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*.

Yin, Y., et al. (2019). Graph-based neural sentence ordering. *arXiv preprint arXiv:1912.07225*.

Zeng, Y., et al. (2016). A convolution bilstm neural network model for chinese event extraction. In *Natural Language Understanding and Intelligent Applications*, pp. 275–287. Springer.

Zhou, Q., et al. (2017). Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics.