Preserve Integrity in Realtime Event Summarization

CHEN LIN, ZHICHAO OUYANG, XIAOLI WANG, and HUI LI, School of Informatics, Xiamen University, China
ZHENHUA HUANG, School of Computer Science, South China Normal University, China

Online text streams such as Twitter are the major information source for users when they are looking for ongoing events. Realtime event summarization aims to generate and update coherent and concise summaries to describe the state of a given event. Due to the enormous volume of continuously coming texts, realtime event summarization has become the de facto tool to facilitate information acquisition. However, there exists a challenging yet unexplored issue in current text summarization techniques: how to preserve the integrity, i.e. the accuracy and consistency of summaries during the update process. The issue is critical since online text stream is dynamic and conflicting information could spread during the event period. For example, conflicting numbers of death and injuries might be reported after an earthquake. Such misleading information should not appear in the earthquake summary at any timestamp. In this paper, we present a novel realtime event summarization framework called IAEA (i.e., Integrity-Aware Extractive-Abstractive realtime event summarization). Our key idea is to integrate an inconsistency detection module into a unified extractive-abstractive framework. In each update, important new tweets are first extracted in an extractive module, and the extraction is refined by explicitly detecting inconsistency between new tweets and previous summaries. The extractive module is able to capture the sentence-level attention which is later used by an abstractive module to obtain the word-level attention. Finally, the word-level attention is leveraged to rephrase words. We conduct comprehensive experiments on real-world data sets. To reduce efforts required for building sufficient training data, we also provide automatic labeling steps of which the effectiveness has been empirically verified. Through experiments, we demonstrate that IAEA can generate better summaries with consistent information than state-of-the-art approaches.

CCS Concepts: • **Information systems → Summarization**.

Additional Key Words and Phrases: Tweet Summarization, Data Integrity, Hierarchical Deep Neural Network, Real-time Event Summarization

ACM Reference Format:
Chen Lin, Zhichao Ouyang, Xiaoli Wang, Hui Li, and Zhenhua Huang. 2020. Preserve Integrity in Realtime Event Summarization. *ACM Trans. Knowl. Discov. Data.* 1, 1 (December 2020), 30 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Microblogging platforms have revolutionized the way people access information, especially for information about ongoing affairs or events. Microblogging service allows people to post short messages, such as tweets. With hundreds of millions of daily users all around the world continuously

Authors' addresses: Chen Lin, chenlin@xmu.edu.cn; Zhichao Ouyang, ouyangzhichao@stu.xmu.edu.cn; Xiaoli Wang, xlwang@xmu.edu.cn; Hui Li, hui@xmu.edu.cn, School of Informatics, Xiamen University, 422 Siming Nan Road, Xiamen, China, 361000; Zhenhua Huang, huangzhenhua@m.scnu.edu.cn, School of Computer Science, South China Normal University, 55 Zhongshan Avenue West, Guangzhou, China, 510631.

1556-4681/2020/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1. Illustrative example: event summaries for Boston marathon bombing at three timestamps. Underlines indicate information which should be updated to preserve integrity of the summary at each timestamp. Tweets are modified and shortened for visualization purpose.

April 15, 2013	April 16, 2013	April 17, 2013
Attack in Boston marathon kills two people and wounds 28.	More than 100 injured, 2 dead after explosions in Boston Marathon.	Two homemade bombs exploded near the finish line, killing three people, injuring several hundred others.

sharing what they observe and know in their surroundings, microblogging is acting as a set of social sensors [1, 2]. From natural disasters to socio-political movements, microblogging platforms such as Twitter[1] and Weibo[2] have become the dominant places for publishing and discussing breaking news. Today, not only people rely on tweets to seek for first hand reports, but also governments [3, 4] and media agencies [5] have acknowledged the significance of microblogs as the major source of information for them.

It may be easy to generate new information by posting new tweets on Twitter, but it is not convenient for people to collect and digest useful information from thousands and hundreds of tweets. Such difficulty is termed *information overload* in the literature, and it is a result of the overwhelming amount of tweets and a severe problem which hinders the process of information dissemination. For example, during the 2009-2010 Iranian election protests and the 2011 Egypt revolution, Twitter was inundated with similar and redundant "voices" of protest. Users had difficulties grasping the main idea and understand the evolvement of the event. Hence, there is a strong incentive to develop a *realtime event summarization system (RESS)* which is able to assist people in retrieving useful information out of noises on microblogs.

An RESS aims at delivering coherent and concise summaries in natural language about an event at any desired moment. A critical issue in an RESS is to preserve the *integrity*[3] of summaries at each update. Due to the dynamic and evolving nature of events, an ongoing event will generate changing and oftentimes conflicting information. To convey the most recent and accurate information, the RESS must exclude any inconsistent information in previous summaries and update the current summary. An example of the text summaries for Boston marathon bombing is illustrated in Table 1. From Table 1, we can observe that the numbers of deaths/injuries and the nature of attack had changed as time went on. Thus, to preserve the integrity, i.e. accuracy and consistency, the summary at the third timestamp must replace "attack" with "two homemade bombs", "kills two" with "killing three", and "wounds 28"/"100 injured" with "injuring several hundred others".

Contemporary event summarization techniques can be classified as extractive and abstractive approaches. Extractive approaches extract representative text units (e.g., sentences) and organize them into a summary. As tweets are self-contained short sentences, the majority of microblog summarization systems adopt extractive summarization techniques [6–13]. Though extractive approaches are easy to implement, the resulted summary suffers from the low coherence [14]. Abstractive approaches [15], on the other hand, rephrase words and construct sentences. Therefore, they are better at generating unseen phrases compared to extractive methods. However, abstractive approach may reproduce inaccurate factual details [16]. Recently, deep neural networks (DNN) have

[1]https://www.twitter.com

[2]https://weibo.com

[3]Data integrity in database domain refers to the accuracy and consistency of data over its lifecycle. In this paper, we adopt the concept of integrity to describe a summary which accurately updates salient information and provides consistent information.

fostered numerous extractive [17–20] and abstractive approaches [16, 21–23] in multi-document summarization and they show promising results. Nevertheless, applying modern DNN technologies in preserving the integrity of realtime event summarization is not a trivial task due to three challenges:

- It is inefficient to recompute the complete tweet stream in realtime event summarization.
- Current methods lacks the ability to explicitly detect inconsistent tweets at different timestamps, which results in undesired summaries containing conflicting information.
- It is widely acknowledged that insufficient training data has become the bottleneck for developing DNN based systems. Massive training data is required for effective DNN based realtime event summarization.

To address the aforementioned challenges, we propose a unified ***I**ntegrity-**A**ware **E**xtractive-**A**bstractive RESS (IAEA)* in this paper. Our contributions are four-fold:

- To overcome the first challenge, IAEA *incrementally replaces inconsistent sentences* in the previous summary with new tweets extracted based on sentence-level attention scores. Then, the sentence-level attention scores are used to modulate word-level attention scores. Finally, IAEA learns to rephrase the extracted summary by generating words based on word-level attention scores.
- To deal with the second challenge, we exploit the hierarchical nature of the inconsistency problem, i.e., two tweets are inconsistent only if they are relevant, and design a hierarchical inconsistency detection module for IAEA. The module is embed into IAEA so that the sentence-level attention is refined by the inconsistency probability in the hierarchical deep neural network.
- For the third challenge, we utilize simple text analysis techniques to automatically construct weak supervisions instead of manually labelling training instances. Labels (i.e., a pair of inconsistent tweets) are assigned by comparing the longest common subsequence and the value of named entities.
- We provide a comprehensive experimental analysis of real-world data to verify the effectiveness of IAEA. Experimental results show that, compared to state-of-the-art summarization methods, IAEA generates summaries with zero inconsistency rates, and best quality in terms of automatic evaluation metrics and human evaluations.

The remainder of the paper is structured as follows. We start by reviewing the related work in Section 2. We provide an overview of IAEA in Section 3. We describe in detail the inconsistency detection module, extractive module and abstractive module in Sections 4, 5 and 6, respectively. We evaluate IAEA on a real twitter data set and analyze the results in Section 7. We conclude the paper with a discussion on future work in Section 8.

2 RELATED WORK

We discuss two lines of related work which are relevant to IAEA in this section.

2.1 Event Centric Tweet Summarization

The emergence of Twitter motivates recent research works on summarizing microblogging contents. Tweet summarization systems are successfully applied in entity centric opinion summarization [24] and event centric tweet summarization, i.e., summarizing tweets for sport, natural or social events [6–12, 15, 25, 26]. Similar to multi-document summarization, event centric tweet summarization systems can generally be categorized into two types: extractive and abstractive methods.

Extractive event centric tweet summarization methods extract representative textual units (i.e., tweets) from the entire tweet collection and combine the tweets into an event summary without

modifying them. Most existing systems fall into this group and they can be unsupervised or supervised. Most unsupervised extractive methods are graph-based: they construct a graph of tweets based on tweet similarity [27, 28], then rank and select a small set of tweets based on different measures of centrality or event relevance [7, 11, 29, 30]. Clustering methods including k-means (and its variant) [6, 10, 25] and topic model based approaches [24, 29, 31] are also applicable after the graph has been constructed. These methods can integrate rich side-information in the extraction, e.g., lexical and temporal information [6], tweet influence [32], recency with respect to the event [33], and location distribution of tweets [8], etc. Other NLP techniques are adopted to generate coherent and readable summaries [34]

With the development of neural network techniques, supervised extractive methods have recently attracted a great attention. In these methods, neural networks are used to map tweets into vectorized representations and to learn to select tweets from gold-standard tweet summarization [19]. Vectorized representation of tweets can also be learned via other tasks (e.g., joint [19] or independent [35] event detection, or sentiment classification [36]).

Compared with extractive methods, much fewer event centric tweet summarization systems adopt abstractive methods to paraphrase and generate unseen words/phrases in the source tweets. Previous abstractive event centric tweet summarization systems construct a phrase graph consisting of high-frequency phrases [37] and then perform graph algorithms [38, 39] to combine phrases. With high accuracy achieved by neural networks, a few recent work [40] extract named entities to fill in slots in a pre-defined template.

It is worthy to point out that the realtime requirement of event centric tweet summarization, i.e., efficiently update summary and convey up-to-date information, has been identified as an important issue in the literature [9, 10, 12, 15, 26]. As tweets are continuously coming, extractive methods are more efficient to update the summary, e.g., with incremental clustering [10], by shrinking the range of selection [13, 41], or via sub-event detection [11, 42]. To convey the most up-to-date information, previous studies have shown that conflicting information must be detected and excluded [9, 12, 15, 26]. The TAC Guided Summarization Task[4] addresses the problem of using information extraction techniques to generate and update summary in a predefined topic template for newswire article streams. However, to the best of our knowledge, none of the previous work has exploited the competency of modern neural networks to address the realtime issues of event centric tweet summarization.

2.2 Neural Summarization

Based on vectorized representations of sentences, the extractive method can also be adapted in deep neural network framework, i.e., sentence salience is learned with different neural network structures [20, 21, 43–45]. The basic structures for encoding the sentences include LSTM [46], GRU [47], transformer [48] and so on.

Recent neural network based abstractive summarization systems usually adopt an encoder-decoder framework [16, 22, 23, 49, 50] that uses an encoder component to represent the original text and a decoder component to generate words in the supervised summary. Neural networks are also applied recently in unsupervised abstractive summarization [51].

Several recent studies attempt to combine the strength of extractive and abstractive summarization. For example, a two-stage model is presented in [52] which first extracts salient sentences and then uses only a decoder to generate the summary. Similarly, the decoder-only architecture is adapted in a hierarchical manner in [48]. A unified model is proposed in [14] proposes which

[4]https://tac.nist.gov/2011/Summarization/Guided-Summ.2011.guidelines.html

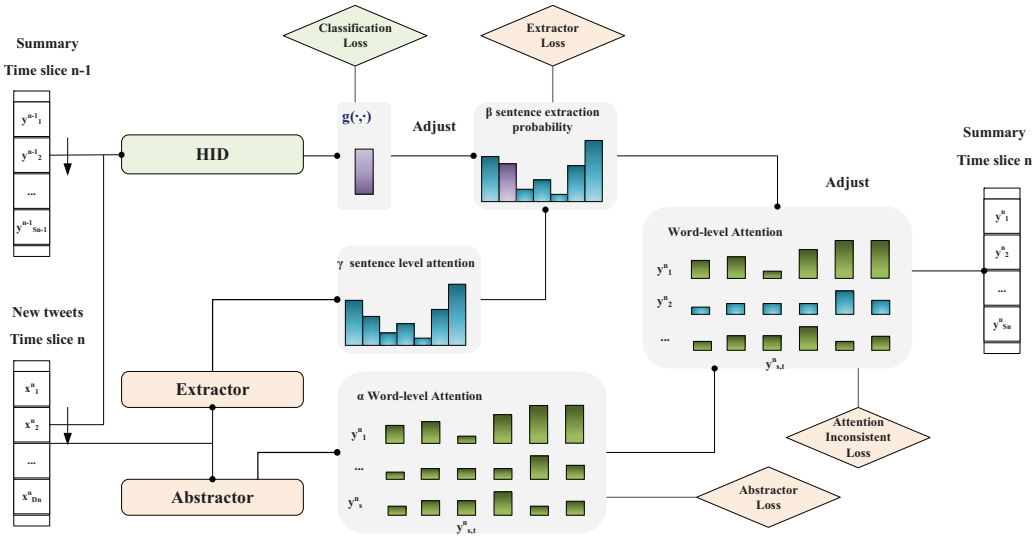

Fig. 1. Overview of IAEA

first obtains sentence-level attention and then uses the sentence-level attention to fine tune the word-level attention in the abstractive module.

Note that applying the aforementioned neural network techniques for multi-document summarization tasks in the tweet domain is not straightforward. Though a few recent work on the news domain [53] has addressed the relevant problems (i.e., input documents may differ in focus and point of view for an event), the multiple documents in these problems generally do not have conflicting information. Furthermore, none of existing works has dealt with dynamic changing information like IAEA.

3 FRAMEWORK OVERVIEW

The problem of realtime event summarization assumes a tweet stream is segmented into several time slices. In each time slice, given a set of tweets up to the current time slice, generate a sequence of sentences, i.e., summary.

The proposed ***I**ntegrity-**A**ware **E**xtractive-**A**bstractive RESS (IAEA)* processes the time slices sequentially and incrementally. At the end of each time slice, the tweet set in the current time slice are fed into the pipeline, then the previously generated summary will be updated. As shown in Figure 1, IAEA consists of three major modules: the hierarchical inconsistency detection module (HID), the extractor, and the abstractor. In a nutshell, IAEA works as follows:

(1) HID traverses the tweet set in the current time slice and compares each new tweet with each previous summary sentence. HID outputs the probability of inconsistency between them.
(2) Extractor takes two input, the output of HID and the tweet set in current time slice. The output is the probability of each new tweet being picked and each sentence in the previously generated summary being kept. This process is achieved by learning the sentence-level attention score, and adjusting it by the inconsistency probability (i.e. output of HID).
(3) Abstractor takes the concatenated representation of the updated summary as input, and outputs a rephrased summary. This process is achieved by learning the word-level attention

Table 2. Major notations for the IAEA framework

Notation	Definition
$\mathbf{x}_d^n = \langle \mathbf{x}_{d,1}^n, \cdots, \mathbf{x}_{d,T_d}^n \rangle, 1 \le d \le D_n$	A tweet d is a sequence of tokens in time slice n.
$\mathbf{y}_s^n = \langle \mathbf{y}_{s,1}^n, \cdots, \mathbf{y}_{s,T_s}^n \rangle, 1 \le s \le S_n$	A summary sentence s is a sequence of tokens in time slice n.
$g(d,s) \in (0,1), 1 \le d \le D_n, 1 \le s \le S_{n-1}$	The output of HID is the probability of new tweet d and old summary sentence s being inconsistent.
$\beta_d \in (0,1), 1 \le d \le D_n, \beta_s \in (0,1), 1 \le s \le S_{n-1}$	The output of Extractor is the probability of each new tweet being picked in the current summary.

weights for each sentence, and adjusting it by the sentence probability of being picked to replace the previous summary (i.e. output of Extractor).

3.1 Notations

Hereafter, unless stated otherwise, we use lower-case letters for variables, upper-case letters for constants, lower-case bold letters for vectors, upper-case bold-face letters for matrices, upper-case calligraphic letters for sets, superscripts for time index, and subscripts for vector index. We omit time index whenever there is no ambiguity. As the Tweet universe is segmented into time slices, we assume within each time slice n ($n \in \{1, \cdots, N\}$), there is a sequence of tweets $\mathbf{x}^n = \langle \mathbf{x}_1^n, \cdots, \mathbf{x}_{D_n}^n \rangle$. Each tweet in $\mathbf{x}^n$ is denoted as $\mathbf{x}_d^n$ ($1 \le d \le D_n$), where we use D_n to denote the largest tweet index in time slice n. Each tweet $\mathbf{x}_d^n = \langle \mathbf{x}_{d,1}^n, \cdots, \mathbf{x}_{d,T_d}^n \rangle$ is a sequence of tokens, where T_d is the length of tweet d. Each time slice can also be represented as a long sequence of tokens by concatenating tweets.

At the end of each time slice n, IAEA will deliver a summary $\mathbf{y}^n = \langle \mathbf{y}_1^n, \cdots, \mathbf{y}_{S_n}^n \rangle$ which is a sequence of sentences and S_n is the number of sentences. Each sentence in $\mathbf{y}^n$ can be denoted as a sequence of tokens $\mathbf{y}_s^n = \langle \mathbf{y}_{s,1}^n, \cdots, \mathbf{y}_{s,T_s}^n \rangle$ with $1 \le s \le S_n$ and T_s being the number of tokens in s-th sentence. Each summary can also be represented as a long sequence by concatenating sentences. Note that all the tokens in tweets and summaries constitute the vocabulary $\mathcal{V}$, i.e., $\mathbf{x}_{d,t}^n, \mathbf{y}_{s,t}^n \in \mathcal{V}$.

4 HIERARCHICAL INCONSISTENCY DETECTION (HID)

The aim of HID is to determine the probability of the two given text sequences x_d^n and y_s^{n-1} being inconsistent. We define that a pair of text sequences are inconsistent when (1) they are relevant, and (2) they contain conflicting information. Obviously, a desirable definition for the inconsistency is hierarchical, i.e., inconsistent pairs must be relevant pairs. Otherwise, as shown in Figure 2, if we do not explicitly define inconsistent tweets to be relevant, then in updating the summary, we might mistakenly replace a sentence with an inconsistent but irrelevant tweet, the information which should be conveyed by the replaced sentence will be missing, leading to a poor topic coverage.

4.1 Model Architecture

Due to the hierarchical nature of the definition, we design a Hierarchical Inconsistency Detection network for HID, which comprises the *relevance classification* part and the *inconsistency classification* part. As shown in Figure 3, this network first outputs relevance identification for $\mathbf{x}_d^n$ and $\mathbf{y}_s^{n-1}$ (we will use d for tweet $\mathbf{x}_d^n$ and s for summary sentence $\mathbf{y}_s^{n-1}$ later to simplify notations). The relevance is denoted as $f(d,s) \in (0,1)$. Then, if the two sequences are relevant (i.e., $f(d,s) > 0.5$), HID will proceed to output inconsistency identification $g(d,s) \in (0,1)$. Larger $g(d,s)$ indicates d, s are more likely to be inconsistent.

The input of HID are the word embeddings of $\mathbf{x}_d^n$ and $\mathbf{y}_s^{n-1}$. We use the word embedding set Glove [54] as the pre-trained embedding weights for each word. Then, word embeddings flow

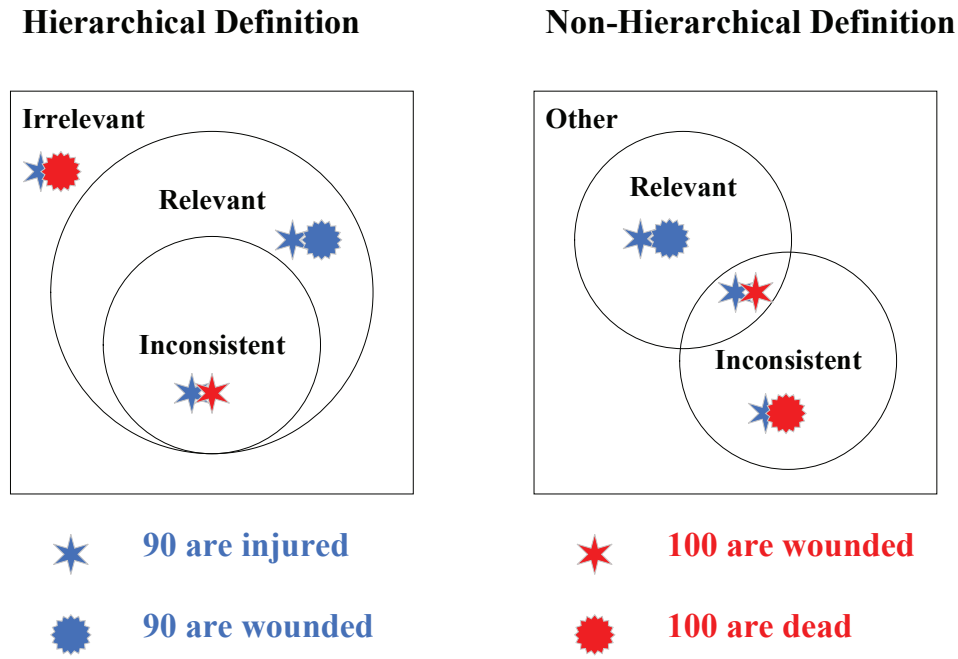

Fig. 2. Hierarchical definition of inconsistency is necessary to replace inconsistent and relevant sentence. Given the tweet "90 are injured" (blue star) , there are different combinations of class labels. In the above figure, "90 are wounded" (blue dots) is "relevant and not inconsistent". "100 are wounded" (red star) is "relevant and inconsistent". "100 are dead" (red dot) is "not relevant and inconsistent". In a hierarchical definition, "90 are injured" (blue star) can be replaced by "100 are wounded" (red star) as summarization goes on. Otherwise, an inconsistent but irrelevant sentence "100 are dead" (red dot) might be used as a replacement and the information about injuries will be missing in the updated summary.

through two Bi-directional Gated Recurrent Units (Bi-GRU) [47] layers. As the tweet d and summary sentence s are processed in the same manner, we will use d as an example in the following to explain the operation of HID.

A Bi-GRU consists of a forward-GRU and a backward-GRU. The forward-GRU processes the sequence from the first token to the last and the backward-GRU processes the sequence in reverse order. We denote the hidden state from forward-GRU at the t-th token as $\mathbf{h}^f_{d,t}$. Each of the GRU units in the forward-GRU updates the next hidden state $\mathbf{h}^f_{d,t}$ based on its previous hidden state $\mathbf{h}^f_{d,t-1}$ through reset and update gates:

$$\begin{aligned}
\mathbf{r}^f_{d,t} &= \sigma_r(\mathbf{W}_r\mathbf{x}_{d,t} + \mathbf{U}_r\mathbf{h}^f_{d,t-1} + \mathbf{b}_r) \\
\mathbf{z}^f_{d,t} &= \sigma_z(\mathbf{W}_z\mathbf{x}_{d,t} + \mathbf{U}_z\mathbf{h}^f_{d,t-1} + \mathbf{b}_z) \\
\tilde{\mathbf{h}}^f_{d,t} &= \sigma_h\big(\mathbf{W}_h\mathbf{x}_{d,t} + \mathbf{U}_h(\mathbf{r}_{d,t} \odot \mathbf{h}^f_{d,t-1}) + \mathbf{b}_h\big) \\
\mathbf{h}_{d,t} &= (1 - \mathbf{z}^f_{d,t}) \odot \mathbf{h}^f_{d,t-1} + \mathbf{z}^f_{d,t} \odot \tilde{\mathbf{h}}^f_{d,t}
\end{aligned} \tag{1}$$

where $\mathbf{r}^f_{d,t}$ and $\mathbf{z}^f_{d,t}$ are the *reset* and *update gates*, respectively. $\tilde{\mathbf{h}}^f_{d,t}$ is the candidate output state, $\mathbf{b}_r$, $\mathbf{b}_z$, $\mathbf{b}_h$ are learnable bias vectors, $\mathbf{W}_r$, $\mathbf{W}_z$, $\mathbf{U}_r$, $\mathbf{U}_z$, $\mathbf{W}_h$ and $\mathbf{U}_h$ are learnable weight matrices, σ is an activation function, and $\mathbf{x}_{d,t}$ is the input word embedding of the t-th token. We use

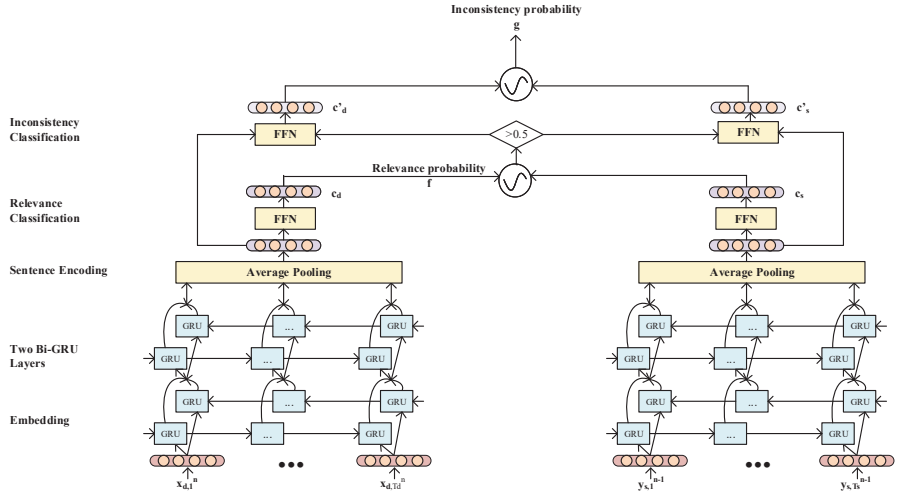

Fig. 3. Architecture of HID.

sigmoid activation function for reset and update gates (σ_r and σ_z), and tanh activation function for the candidate output generation (σ_h). Similarly, the hidden state $\mathbf{h}^b_{d,t}$ in backward-GRU is obtained from its previous hidden state $\mathbf{h}^b_{d,t+1}$. In the first Bi-GRU layer, the token encoding is represented as the concatenation of the hidden states from forward-GRU and backward-GRU: $\mathbf{h}(d,t) = \mathbf{h}(d,t)^f \oplus \mathbf{h}(d,t)^b$.

Previous study on deep neural networks has shown that performance can be boosted by stacking multiple neuron layers [48]. Thus, we place another Bi-GRU layer over the first Bi-GRU layer in IAEA. Each token encoding $\mathbf{h}(d,t)$ from first Bi-GRU layer is fed to the second Bi-GRU layer. Then, IAEA performs average pooling of hidden states for each token in the sequence to capture the useful features produced by the second Bi-GRU layer. The output of the average pooling is $\mathbf{h}_d = \sum_{t<T_d} \mathbf{h}(d,t)/T_d$.

The tweet encoding $\mathbf{h}_d$ is fed to a fully connected feed-forward network (FFN) to get the representation of the tweet $\mathbf{c}_d$:

$$\mathbf{c}_d = \sigma_c(\mathbf{W}_F \mathbf{h}_d + \mathbf{b}_F), \tag{2}$$

where the activation function σ_c is tanh activation function. $\mathbf{W}_F$ and $\mathbf{b}_F$ are learnable weight matrix and bias vector, respectively. The representation $\mathbf{c}_s$ of the summary sentence s is obtained similarly.

Relevance. Then, IAEA merges the two encodings $\mathbf{c}_d, \mathbf{c}_s$ and uses a FFN layer to estimate the probability of d and s being relevant. We have experimented with different merge functions, i.e., addition, subtraction, and concatenation. As shown in Section 7, subtraction produces the best results. Thus, subtraction is adopted in the following equation to compute $f(d,s)$:

$$f(d,s) = \sigma_f\big(\mathbf{W}_f(\mathbf{c}_d - \mathbf{c}_s) + \mathbf{b}_f\big), \tag{3}$$

where σ_f is the sigmoid activation function. $\mathbf{W}_f$ and $\mathbf{b}_s$ are learnable weight matrix and bias vector, respectively. The relevance classification in IAEA is optimized with the cross entropy loss.

Inconsistency. After that, IAEA leverages the relevant tweets (i.e., tweets with $f(\cdot) > 0.5$) for the inconsistency detection. The threshold 0.5 is selected manually because it is the boundary between

relevant and irrelevant tweets. The architecture for inconsistency detection part is similar to that of relevance classification part. In inconsistency detection, the hidden states of the second Bi-GRU layer from the bottom module first flow through a FFN to obtain the tweet encoding $\mathbf{c}'_d$ and the summary sentence encoding $\mathbf{c}'_s$. Then a subtracting layer with the sigmoid activation function σ_g is adopted to output the result of inconsistency detection:

$$g(d,s) = \sigma_g\big(\mathbf{W}_g(\mathbf{c}'_d - \mathbf{c}'_s) + \mathbf{b}_g\big), \tag{4}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable weight matrix and bias vector, respectively. Note that the inconsistency detection is not computed for all tweet-sentence pairs as we adopt a hierarchical definition.

4.2 Learning

The HID can be trained independently or jointly with other summarization components. To learn the parameters independently, HID optimizes the cross-entropy loss for relevance and inconsistency classification on labelled training sets.

$$\begin{aligned} L_{classification} = &\sum_{d,s}[f(d,s)\log f(\hat{d},s) + \big(1-f(d,s)\big)\log\big(1-f(\hat{d},s)\big)] \\ &+ \sum_{f(\hat{d},s)>0.5}[g(d,s)\log g(\hat{d},s) + \big(1-g(d,s)\big)\log\big(1-g(\hat{d},s)\big)], \end{aligned} \tag{5}$$

where $f(d,s)$, $g(d,s)$ are the gold standard labels for all tweet sentence pairs in the pipeline (annotation details are explained in Section 7), and $f(\hat{d},s)$, $g(\hat{d},s)$ are the output.

Note that in independent training, $L_{classification}$ is the loss associated solely with HID as shown in Figure 1. Once the training is finished, HID will be used as a static component in the summarization framework and the parameters of HID will not be updated.

To train HID jointly with the other summarization components, we first pre-train HID to optimize $L_{classification}$. Then we fine-tune the parameters in training the extractor (Section 5) and the end-to-end training of the unified model (Section 6.2). However, joint training does not show empirical improvement over independent training. Thus, in the experiments in Section 7, we only report the results obtained by independently training HID.

5 EXTRACTOR

In this section, we present the details of the extractor module, which outputs a probability of a tweet being picked or a previous summary sentence being kept. Intuitively, a tweet should be given more attention in generating a summary if it has a larger probability to be extracted based on the tweet contents or it is inconsistent to the previous summary. Extractor operates on the set of tweets at the current timestamp. Suppose the input is the set of tweets in timestamp n: $x^n_1, \cdots, x^n_{D_n}$, where each tweet is a sequence of tokens, i.e., $x^n_d = \langle x^n_{d,1}, \cdots, x^n_{d,T_d}\rangle$; and the set of summary sentences in timestamp $n-1$: $y^n_1, \cdots, y^{n-1}_{S_{n-1}}$, where each sentence is a sequence of tokens, i.e., $y^{n-1}_s = \langle y^{n-1}_{s,1}, \cdots, y^{n-1}_{s,T_s}\rangle$. Extractor first predicts the probability of each candidate sentence (i.e. including each new tweet d and old summary sentence s) being chosen in the extractive summary, i.e., γ_d, γ_s. And then, it traverses all tweets $d = 1, \cdots, D_n$. Based on the inconsistency prediction $g(d,s)$ between d and each s of the previous summary sentences, extractor adjusts the extraction probability γ_d, γ_s to obtain sentence-level attention score β_d, β_s.

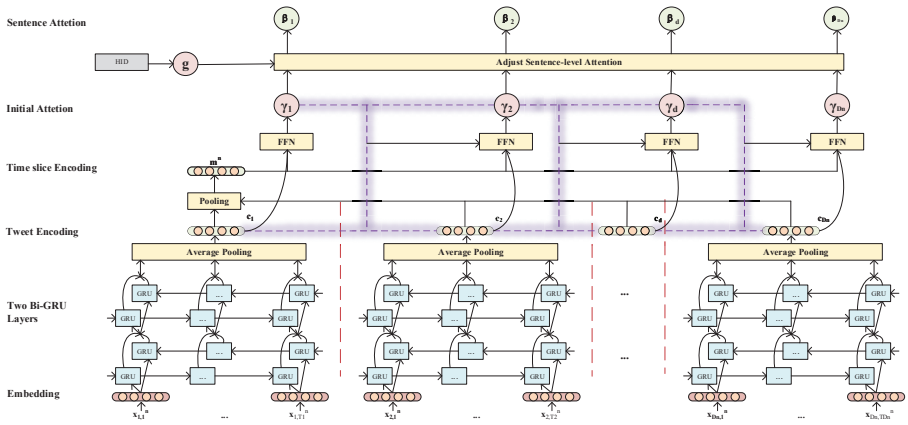

Fig. 4. The architecture of extractor. For simplicity, we omit the input of previous summary sentences, which are treated the same as tweets in the current timestamp.

5.1 Model Architecture

The overall architecture of extractor is demonstrated in Figure 4. Specifically, extractor adopts a similar component as in HID to obtain tweet encoding. The word embeddings of the input tweet d are first passed through two Bi-GRU layers to obtain the hidden state $\mathbf{h}^f_{d,t}$, $\mathbf{h}^b_{d,t}$ from forward-GRU and backward-GRU respectively by Equation 1. The hidden states from backward-GRU and forward-GRU are concatenated as the representation of each token in d, i.e., $\mathbf{h}_{d,t} = \mathbf{h}^b_{d,t} \oplus \mathbf{h}^f_{d,t}$. An average pooling is performed over all tokens to form the sentence encoding $\mathbf{h}_d$, which is then fed to a FFN to obtain the representation of the input tweet d, i.e., $\mathbf{c}_d = \sigma_c(\mathbf{W}_F\mathbf{h}_d + \mathbf{b}_F)$ with σ_c being the tanh activation function. The encoding for old summary sentence s is obtained in the same manner.

We perform average pooling to obtain the *time slice encoding*, i.e., the representation of all tweets and old summary sentences in the current time slice:

$$\mathbf{m}^n = \sigma_m\Big(\frac{1}{D_n + S_{n-1}}\mathbf{W}_P\Big(\sum_{d=1}^{D_n}\mathbf{c}_d + \sum_{s=1}^{S_{n-1}}\mathbf{c}_s\Big) + \mathbf{b}_P\Big), \tag{6}$$

where σ_m is the tanh activation function. $\mathbf{W}_P$ and $\mathbf{b}_p$ are learnable weight matrix and bias vector, respectively.

Inspired by Nallapati et al. [20], the task of predicting γ_d, γ_s is treated as a binary classification problem in IAEA. Extractor predicts the probability of tweet d or sentence s being extracted (i.e., $l_d = 1, l_s = 1$), given the representation of the current time slice $\mathbf{m}^n$ and the tweets/sentences prior to d, s , we have:

$$\gamma_s = p(l_s = 1|\mathbf{c}_s, \mathbf{m}^n, \mathbf{c}_{1:s-1}) = \sigma\big(\mathbf{W}_c\mathbf{c}_s + \mathbf{c}_s\mathbf{W}_s\mathbf{m} - \mathbf{c}_s^T\mathbf{W}_r tanh(\mathbf{c}_{1:s-1})\big) \tag{7}$$

where $\mathbf{c}_{1:s-1} = \sum_{s'=1}^{s-1}\gamma_{s'}\mathbf{c}_{s'}$, and σ is the sigmoid function. Equation 7 consists of three terms,: the first term measures the salience of sentence s based on its encoding $\mathbf{c}_s$, the second term measures the representativeness of sentence s with respect to the current time slice $\mathbf{m}^n$, and the third term measures the redundancy of sentence s with respect to temporal summaries based on previous sentences $\mathbf{c}_1, \cdots, \mathbf{c}_{s-1}$.

Algorithm 1: Adjust sentence-level attention

Input: $\gamma = [\gamma_1, \cdots, \gamma_{D_n}]$ for all tweets d in the current time slice,
$G = \{g(d,s), 1 \le d \le D_n, 1 \le s \le S_{n-1}\}$ for all tweets d in the current time slice and all sentences s in the previous summary, current tweets x^n, previous summary y^{n-1}

Output: $\beta_d \in (0,1), 1 \le d \le D_n, \beta_s \in (0,1), 1 \le s \le S_{n-1}$

1 **for** $1 \le d \le D_n$ **do**
2 $\beta_d = \gamma_d$;
3 **for** $s = 1; s \le S_{n-1}; s++$ **do**
4 **if** $g(d,s) \ge 0.5$ **then**
5 $\beta_s = \gamma_s \times \big(1 - g(d,s)\big)$;
6 Remove s from the summary y^{n-1};
7 $\beta_d = g(d,s) \times \gamma_d + \big(1 - g(d,s)\big) \times p(l_d = 1 | \mathbf{c}_d, \mathbf{m}^n, \mathbf{y}^{n-1})$;
8 Put d in the summary y^{n-1};
9 **end**
10 **end**
11 **end**

The probability γ_d for a new tweet d is computed in a similar manner (i.e. Equation 8).

$$\gamma_d = p(l_d = 1 | \mathbf{c}_d, \mathbf{m}^n, \mathbf{c}_{1:d-1}) = \sigma\big(\mathbf{W}_c \mathbf{c}_d + \mathbf{c}_d \mathbf{W}_s \mathbf{m} - \mathbf{c}_d^T \mathbf{W}_r tanh(\mathbf{c}_{1:S_{n-1},1:d-1})\big) \tag{8}$$

where the third term has been changed accordingly to $\mathbf{c}_{1:S_{n-1},1:d-1} = \sum_{s'=1}^{S_{n-1}} \gamma_{s'} \mathbf{c}_{s'} + \sum_{d'=1}^{d} \gamma_{d'} \mathbf{c}_{d'}$, i.e., tweets indexed from 1 to $d-1$ and all old summary sentences.

Finally, IAEA adjusts the sentence-level attention score β based on the output g of the HID module (Section 4) according to Algorithm 1. In Algorithm 1, the Extractor traverses the tweet set in the current time slice (line 1). Note that Extractor does not order tweets within one time slice. Thus, in line 1, the traverse can be implemented in the order of tweets'published timestamps or in a random order. For a tweet d, β_d will be first aligned with the value of γ_d (line 2). If an old summary sentence is inconsistent with a new tweet, i.e., $g(d,s) \ge 0.5$ (line 4), it will no longer be compared with other new tweets (line 6) so that the generated summary will favor more recent information; and the attention for both d, s will be updated (line 5, line 7). The inconsistent previous summary sentence will also be degraded (line 5) by assigning a smaller sentence-level attention score to it.

$$p(l_d = 1 | \mathbf{c}_d, \mathbf{m}^n, \mathbf{y}^{n-1}) = \sigma\big(\mathbf{W}_c \mathbf{c}_d + \mathbf{c}_d \mathbf{W}_s \mathbf{m} - \mathbf{c}_d^T \mathbf{W}_r tanh(\mathbf{y}^{n-1})\big) \tag{9}$$

For the new tweet d, the update will take into account the probability $p(l_d = 1 | \mathbf{c}_d, \mathbf{m}^n, \mathbf{y}^{n-1})$, which is computed by Equation 9 based on the tweet encoding $\mathbf{c}_d$, the representativeness in the time slice $\mathbf{m}^n$, and the previous summary $\mathbf{y}^{n-1}$. We can see that Equation 9 resembles Equation 8. The previous summary is vectorized by $\mathbf{y}^{n-1} = \sum_{s=1}^{S_{n-1}} \gamma_s \mathbf{c}_s$, i.e., the weighted summarization of old summary sentence representations. Note that the previous summary has been updated in line 6. Intuitively, the more the new tweet contains conflicting information (i.e., a larger $g(d,s)$), the more IAEA will depend on the extraction probability γ_d when computing β_d as γ_d is computed on new tweets; The less the new tweet contains conflicting information (i.e., a larger $1 - g(d,s)$), the more IAEA will determinate the representativeness based on previous summary $p(l_d = 1 | \mathbf{c}_d, \mathbf{m}^n, \mathbf{y}^{n-1})$ (line 7).

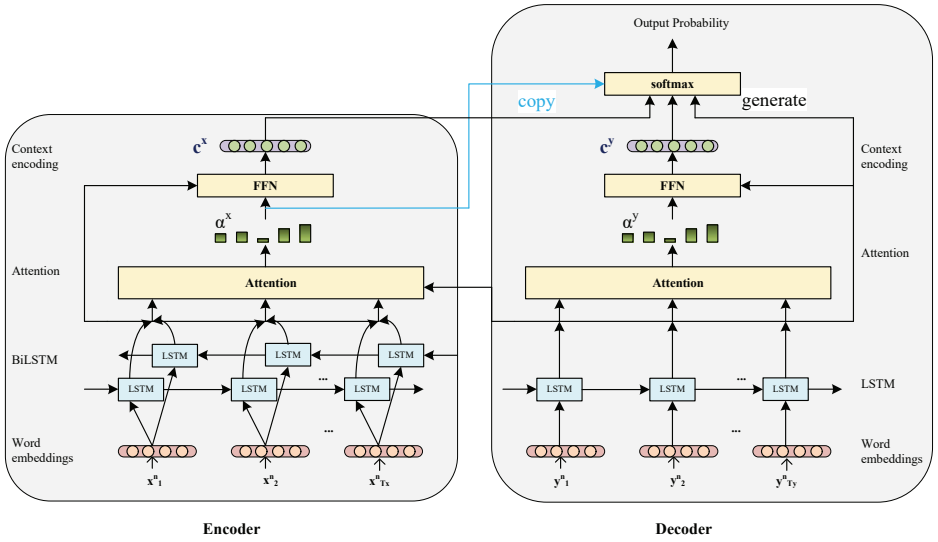

Fig. 5. Architecture of abstractor

5.2 Learning

The extractor is optimized with the following loss:

$$L_{extractor} = -\frac{1}{D_n + S_{n-1}} \sum_{d,s} [l_s \cdot \log\beta_s + (1-l_s) \cdot \log(1-\beta_s) + l_d \cdot \log\beta_d + (1-l_d) \cdot \log(1-\beta_d)], \quad (10)$$

where l_s and l_d represent the labeling of whether a previous summary sentence s or a current tweet d is included in the extractive summary ground-truth data, respectively. D_n is the number of tweets in timestamp n, and S_{n-1} is the number of sentences of previous summary at timestamp $n-1$. We will describe the method of generating extractive summary ground-truth in Section 7.

6 ABSTRACTOR

The last module in IAEA is an abstractor to refine the text in order to generate coherent and concise summaries. Motivated by DeepRL [50], we first train the abstractor independently based on manually generated ground truth (details in Section 7); then we train the extractor-abstractor in an end to end manner. Figure 5 illustrates the architecture of the abstractor.

6.1 Architecture

Each training sample of the abstractor is an input sequence of the tweet set x^n in the time slice n and the output sequence of the summary y^n in the time slice n. The input sequence is organized by tokens rather than tweets, i.e., $\mathbf{x}^n = \langle x_1^n, \cdots, x_{T_x}^n \rangle$, where T_x is the accumulated length in the tweet set: $T_x = \sum_d T_d$. We use mapping function $d(t)$ to denote the tweet index of the t-th token. [5] The output sequence is also organized by tokens, i.e., $\mathbf{y}^n = \langle y_1^n, \cdots, y_{T_y}^n \rangle$, where T_y is the accumulated length of summary y. For simplicity we will drop the superscript n when there is no ambiguity caused.

[5]Note that we do not require the tweets to be concatenated in chronological order. More experimental results regarding the sequential order of input tweets can be found at Section 7.3.

As shown in Figure 5, the abstractor implements an encoder-decoder architecture. The encoder reads the word embeddings of the tokens in $\mathbf{x}^n$ one at a time through a Bi-LSTM layer following by an attention layer. And encoder encapsulates the information for all tokens to the global context encoding vector. The decoder generates the tokens of $\mathbf{y}^n$ one at a time, based on the global context encoding vector and hidden states of previous token in $\mathbf{y}^n$ through a LSTM layer following by an attention layer.

We use the pre-trained word embeddings Glove. In the Bi-LSTM layer of encoder, suppose the output of the forward LSTM for the t-th token is $\mathbf{h}_t^f$, the output of the backward LSTM is $\mathbf{h}_t^b$. We concatenate $\mathbf{h}_t^f$ and $\mathbf{h}_t^b$ to obtain the encoding hidden state $\mathbf{h}_t^x = \mathbf{h}_t^f \oplus \mathbf{h}_t^b$.

Since each token in the input sequence should receive different attentions, IAEA computes attention scores $\alpha_{t,i}^x$ for decoder token i on input token t by looking at all input tokens:

$$\alpha_{t,i}^x = \frac{e_{t,i}^x}{\sum_{t' \le T_x} e_{t',i}^x} \tag{11}$$

$$e_{t,i}^x = \begin{cases} e_{t,i}^{'} & \text{if } i = 1, \\ \frac{e_{t,i}^{'}}{\sum_{j \le i-1} e_{t,j}^{'}} & \text{otherwise} \end{cases} \tag{12}$$

$$e_{t,i}^{'} = \mathbf{w}_e^T \sigma_e(\mathbf{W}_h \mathbf{h}_t^x \mathbf{W}_d \mathbf{h}_i^y + \mathbf{W}_c \mathbf{o}_i^t + \mathbf{b}_e) \tag{13}$$

$$\mathbf{o}_i^t = \sum_{j=1}^{i-1} \alpha_{t,j}^e \tag{14}$$

where σ_e is the tanh activation function, $\mathbf{w}_e$, $\mathbf{W}_h$, $\mathbf{W}_d$ and $\mathbf{W}_c$ are learnable weights, and $\mathbf{b}_e$ is the bias vector. $\mathbf{o}_i^t$ is the coverage vector, which is a key mechanism in [16] to prevent the abstractor from repeatedly attending to the same place. It penalizes tokens that appear in previous positions when computing the attention score, to avoid the decoder generates duplicate tokens.

Then, IAEA encode the information in the input sequence for the i-token in the output as $\mathbf{c}_i^x$, which is the weighted combination of hidden states from all tokens in the input sequence.

$$\mathbf{c}_i^x = \sum_{t \le T_x} \alpha_{t,i}^x \mathbf{h}_t^x \tag{15}$$

In the decoder, word embeddings from summary y^n flow through an LSTM layer. We use $\mathbf{h}_i^y$ to denote the encoding hidden state for the i-th token in y^n, which is also the concatenation of the output of a forward LSTM unit and a backward LSTM unit. We adopt an attention layer in IAEA to compute $\alpha_{i,j}^y$, i.e., the weight for i-th token to attend to other tokens j in y^n:

$$\begin{aligned} e_{i,j}^y &= \mathbf{h}_i^{yT} \mathbf{W}_{attn} \mathbf{h}_j^y \\ \alpha_{i,j}^y &= \frac{\exp(e_{i,j}^y)}{\sum_{j' \le T_y} exp(e_{i,j'}^y)} \end{aligned} \tag{16}$$

where $\mathbf{W}_{attn}$ is the learnable weight matrix.

IAEA obtains the representation of context for decoder at the i-token $\mathbf{c}_i^y$, which is the weighted combination of hidden states from previous tokens in the output sequence. For the first token $i = 1$, the context representation is empty. For $i > 1$, we have:

$$\mathbf{c}_i^y = \sum_{j \le i-1} \alpha_{i,j}^y \mathbf{h}_j^y \tag{17}$$

To reduce the number of OOV (out of vocabulary) tokens, we employ the copy-generate mechanism in pointer-generator network [16]. Specifically, a binary variable u_i for the $i-$th token in decoder is defined to indicate whether the token is copied from the input sequence, i.e. $u_i = 0$, or generated from the vocabulary, i.e. $u_i = 1$:

$$p(u_i = 1|y_1, \cdots, y_{i-1}) = \sigma(\mathbf{W}_{PG}(\mathbf{h}_i^y, \mathbf{c}_i^x, \mathbf{c}_i^y) + \mathbf{b}_{PG}), \tag{18}$$

where $\mathbf{W}_{PG}$ is learnable weights, $\mathbf{b}_{PG}$ is a bias vector and σ is the sigmoid activation function.

Finally, a token is generated by either copying from the input or choosing from the vocabulary.

$$\begin{aligned} p(y_i = t|y_1, \cdots, y_{i-1}) &= p(y_i = v_t|u_i = 1) \cdot p(u_i = 1) + p(y_i = t|u_i = 0) \cdot p(u_i = 0) \\ p(y_i|u_i = 1) &= softmax(\mathbf{W}_{gen}[\mathbf{h}_i^y, \mathbf{c}_i^x, \mathbf{c}_i^y] + \mathbf{b}_{gen}) \\ p(y_i|u_i = 0) &= \alpha_i^x \end{aligned} \tag{19}$$

6.2 Learning

We conduct two-phase training for the abstractor of IAEA. In the first phase, we train the abstractor independently. We adopt the self-critical policy gradient in deep reinforcement learning [55]:

$$L_{rl} = \Big(r\big(\hat{y}\big) - r(y^s)\big)\Big) \sum_{t=1}^{n'} \log P(y_t^s|y_1^s, \ldots, y_{t-1}^s), \tag{20}$$

where $\hat{y}$ is the baseline output by performing greedy search at every decoder step, y^s is the sampled distribution, $r(y)$ is the reward function which is the ROUGE-L F-score of the sequence y, and $P(y_t^s|y_1^s, \ldots, y_{t-1}^s)$ is computed by Equation 19

To generate more natural summaries, we construct the reinforcement loss with two loss terms:

$$L_{abstractor} = \gamma \cdot L_{rl} + (1 - \gamma) \cdot L_{ml} + L_{cov} \tag{21}$$

where γ is the coefficient, $L_{ml} = -\sum_{t=1}^{n'} \log P(y_t^*|y_1^*, \ldots, y_{t-1}^*, \mathbf{x})$ is the negative likelihood loss for the language model, and $L_{cov} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n} \min(\hat{\alpha}_i^t, \mathbf{o}_i^t)$ is the coverage loss for token repetitions.

In the second phase, we fine tune the parameters learned in the first phase, i.e., independently training the abstractor following by training the extractor and abstractor together in an end-to-end manner. We update word attention $\alpha^{\mathbf{x}}$ by combining the sentence-level β as in [14].

$$\hat{\alpha}_{i,t}^x = \frac{tanh(\alpha_{i,t}^x \times \beta_{d(i)})}{\sum_i tanh(\alpha_{i,t}^x \times \beta_{d(i)})}, \tag{22}$$

where $d(i)$ is the sentence index of the i-th token. This ensures that the updated word attention will be high only when the word-level and corresponding sentence-level attentions are both high.

To encourage the attentions by extractor and abstractor consistent with each other, similar as [14], we define

$$L_{attInc} = -\frac{1}{T_x}\sum_t \log(\frac{1}{T_y}\sum_i \alpha_{i,t}^x \times \beta_{d(i)}) \tag{23}$$

Finally, we adopt the loss in the second stage [14]:

$$L_{total} = \lambda_1 \cdot L_{extractor} + \lambda_2 \cdot L_{abstractor} + \lambda_3 \cdot L_{attInc}, \tag{24}$$

where $L_{extractor}$ is the extractor loss defined in Equation 10, $L_{abstractor}$ is the abstractor loss defined in Equation 21, λ_1, λ_2 and λ_3 are coefficients.

7 EXPERIMENTAL STUDY

In this section, we design comprehensive experiments to answer the following research questions.

(1) Does HID identify inconsistent tweets accurately?
(2) Does IAEA preserve integrity in updating summaries?
(3) Does IAEA produce high quality summary?

7.1 Data Set Preparation

We conduct experiments on a large scale real-world tweet data set. As there is no public data available for integrity-aware realtime event summarization which contains ground-truth summary updates at each time stamp and annotations for inconsistent tweets, part of this paper's contribution is to build a data set in the following efficient and effective steps. The data set and code are published[6]. The data set is built on the collection of 9, 154, 025 tweets, where each tweet is assigned to one of 30 events [56] . The event statistics of the data set is described in Table 3.

Pre-process. To simulate the real-time event-centric summarization scenario, we order the tweets of each event based on their published timestamps and update summaries for each event whenever there are 3, 000 new tweets, i.e. a time segment n contains 3, 000 tweets. As shown in Table 3, most events contain five time segments. To reduce the workload for neural summarization system, at each time segment, we feed the pipeline with 100 relevant and credible tweets [12]. These 100 relevant and credible tweets are selected by the following steps. Firstly, for each input event keyword (i.e. "Keyword" shown in Table 3), we run ten search models of the search engine Terrier[7], including BM25, PL2, TF_IDF, InL2, LGD, DLH13, BB2, IFB2, In_expC2 and DPH. Suppose the rank of each tweet d by the ten search models is denoted as $r(d)_o, o = 1, \cdots, 10$, we obtain the average reciprocal rank $\bar{r}(d) = \sum_o 1/(10 \times r(d)_o)$. In addition to textual content, four attributes of each tweet d in the dataset are available [56], i.e. number of views $e(d)$, number of comments $c(d)$, number of retweets $t(d)$ and number of thumb-ups $u(d)$. We normalize the four attributes and obtain $\bar{e}(d), \bar{c}(d), \bar{t}(d), \bar{u}(d)$. For example, the normalized number of views $\bar{e}(d)$ is obtained by dividing $e(d)$ (i.e., the number of views) by the maximal number of views a tweet receives in the time segment. Then, we delete tweets that are published by any author who has fewer than five tweets [12]. Finally, we score each remaining tweet by $s(d) = (\bar{r}(d) + \bar{e}(d) + \bar{c}(d) + \bar{t}(d) + \bar{u}(d))/5$. The top 100 tweets with highest score $s(d)$ are fed into the pipeline. Since the score $s(d)$ is based on both content relevance (i.e., $\bar{r}(d)$) and popularity (i.e., $\bar{e}(d), \bar{c}(d), \bar{t}(d), \bar{u}(d)$), the pre-processing avoids rumor and inaccurate information as much as possible. The 100 relevant and credible tweets are also used to generate gold-standard extractive and abstractive summaries, as shown in the following.

Annotation for inconsistency detection. For each event, we rank the tweets in chronological order. Note that the summarization is conducted under each event, thus we do not label cross-event tweets as ground truth. We observe that for each event, adjacent tweets tend to be duplicate tweets, thus we form tweet pairs $\langle i, j \rangle, i < j \leq i + 4$ for each tweet ranked at $i, i = 1, \cdots, N$, where N is the total number of tweets under this event. Labeling such massive number of tweet pairs is a huge labor cost, thus we adopt the following strategy to automatically generate *weak supervisions*. To be specific, we label pairs of tweets i, j as irrelevant (denoted as $f(\hat{i}, j) = 0$) and relevant (denoted as $f(\hat{i}, j) = 1$). We also label relevant tweets as relevant and consistent (denoted as $g(\hat{i}, j) = 0$) and relevant and inconsistent (denoted as $g(\hat{i}, j) = 1$), respectively.

To perform the labeling, we first perform case-folding, lemmatization, stop-word removal, marker deletion and Named Entity Recognition (NER) [57]. Then, the Named Entities are replaced by special

[6]https://github.com/XMUDM/IAEA
[7]www.terrier.org

Table 3. Event statistics of the data set. N is the number of time slices.

Keyword	# Tweet	Start Time	End Time	Event description	N
Superbowl	376,707	2012/02/03	2012/02/07	An American football game to decide the National Football League (NFL) champion .	5
SXSW	455,867	2012/03/08	2012/03/22	An annual conglomerate of film, interactive media, and music festivals and conferences .	5
Euro	904,237	2012/06/02	2012/07/04	The 14th European Championship for men's national football teams organised by UEFA.	5
Mexico Election	9,000	2012/07/01	2012/07/03	Mexico Elections 2012.	3
Hurricane sandy	2,230,689	2012/10/25	2012/11/02	The deadliest and most destructive hurricane of the 2012 Atlantic hurricane season.	5
ObamaRomney	1,178,182	2012/11/05	2012/11/08	Two presidential candidates of the 57th quadrennial American presidential election.	5
US Election	313,341	2012/11/05	2012/11/08	The 57th quadrennial American presidential election .	5
BMBombing	302,733	2013/04/15	2013/04/16	Two homemade bombs detonated near the finish line of the annual Boston Marathon.	5
SPatricksDay	392,338	2014/03/15	2014/03/18	A cultural and religious celebration held on 17 March, the traditional death date of Saint Patrick.	5
GUAttack	56,999	2014/06/02	2014/07/17	Military operation launched by Israel in the Hamas-ruled Gaza Strip.	5
EOutbreak	91,128	2014/07/01	2014/07/31	Large-scale virus outbreaks in West Africa.	5
FUnrest	485,180	2014/08/09	2014/08/25	A policeman shoots an Afro-American man, causing citizens in 170 U.S. cities to get involved in the parade.	5
Indyref	175,584	2014/09/17	2014/09/20	A referendum on Scottish independence from the United Kingdom took place on 18 September 2014.	5
HProtests	50,295	2014/09/26	2014/10/17	A series of civil disobedience campaigns in Hong Kong.	4
OShooting	115,527	2014/10/22	2014/10/24	At the Canadian National War Memorial, Michael Zehaf-Bibeau fatally shot Corporal Nathan Cirillo, a Canadian soldier on ceremonial sentry duty.	5
THagupit	16,796	2014/11/05	2014/11/11	A strong cyclone code-named Typhoon Hagupit hit Philippines .	4
SydneySiege	169,684	2014/12/14	2014/12/17	A lone gunman held hostage ten customers and eight employees of a Lindt chocolate cafe located at Martin Place in Sydney, Australia.	5
CHShoot	159,255	2015/01/07	2015/01/07	Two brothers forced their way into the offices of the French satirical weekly newspaper Charlie Hebdo in Paris.	5
GPCrash	70,188	2015/03/24	2015/03/30	On 24 March 2015, the aircraft, an Airbus A320-211, crashed 100 kilometres (62 mi) north-west of Nice in the French Alps.	5
NEarthquake	401,889	2015/04/25	2015/05/18	Earthquake occurred at 11:56 Nepal Standard Time on 25 April, with a magnitude of 7.8Mw or 8.1Ms.	5
RWelcome	69,393	2015/09/02	2015/11/24	Rising numbers of people arrived in the European Union because of European refugee crisis .	5
HPatricia	15,224	2015/10/24	2015/11/08	The second-most intense tropical cyclone on record worldwide.	5
ParisAttack	732,145	2015/11/13	2015/11/24	A series of coordinated terrorist attacks that occurred on Friday, 13 November 2015 in Paris.	5
IElection	33,362	2016/02/03	2016/03/06	Irish Elections 2016.	5
Brexit	67,482	2016/02/24	2016/04/23	Prospective withdrawal of the United Kingdom (UK) from the European Union (EU).	5
BAExplossion	184,783	2016/03/22	2016/03/22	Three coordinated suicide bombings occurred in Belgium.	5
LBlast	23,103	2016/03/27	2016/03/30	A suicide bombing that hit the main entrance of Gulshan-e-Iqbal Park.	5
HPCyprus	21,258	2016/03/29	2016/03/30	A domestic passenger flight was hijacked by an Egyptian man in Cyprus.	5
PanamaPapers	36,656	2016/04/03	2016/05/03	Panama document leaks (the biggest offshore money laundering secret ever leaked).	4
SEcuador	15,000	2016/04/17	2016/04/28	Ecuador 7.8 magnitude earthquake.	5

tokens such as"place, organization, people". The numerals are replaced by a special token "numeral'. Comparing two tweets depends on the skeleton of a pair of tweets, which is defined as the Longest Common Subsequence of tokens (LCS). Intuitively, if two tweets have nothing in common, the LCS will be relatively short. Otherwise, it will be long. If the tweets are inconsistent, the key values (i.e. the exact place, organization, people names and the numbers) will be different. Note that we do not consider the position of special tokens (i.e., "place, organization, people, numeral" we use to replace Named Entities and numerals) in finding the LCS. The special tokens are trimmed before the LCS mining, and are automatically appended to the end of the LCS.

Thus, we determine the label based on the ratio of length of LCS. Suppose $|LCS(i, j)|$ is the length of LCS of tweets i, j, o_i is the number of tokens in tweet i, o_j is the number of tokens

in tweet j. We label a pair of tweets as irrelevant $f(i,j) = 0$ if the LCS ratios are both small, i.e. $|LCS(i,j)|/\min(o_i, o_j) \leq 0.3$. We label a pair of tweets as relevant and inconsistent $f(i,j) = 1, g(i,j) = 1$ if (1) the LCS ratios are large, i.e. $|LCS(i,j)|/\max(o_i, o_j) \geq 0.5$; and (2) the special tokens have different values. Other pairs of tweets are labeled as relevant, i.e. $|LCS(i,j)|/\min(o_i, o_j) > 0.3$ and $|LCS(i,j)|/\max(o_i, o_j) < 0.5, \rightarrow f(i,j) = 1, g(i,j) = 0$.

For example, the LCS of tweets "Death toll from earthquake in Nepal rises to 449" and "Death toll from Nepal earthquake reaches at least 688" is "death toll earthquake [place] [numeral]" (length 5). The smallest LCS ratio is 5/7, the place is both Nepal, but the numeral values are different. Thus we label these two tweets as relevant and inconsistent $f(\hat{i}, j) = 1, g(\hat{i}, j) = 1$.

The above automatic labeling process will generate highly imbalanced training set, i.e. overwhelming irrelevant tweet pairs. To reduce the affect of class imbalance, we under-sample the irrelevant tweet pairs and obtain weak supervisions of 67, 968 pairs of irrelevant tweets, 61, 778 pairs of relevant tweets, within which 27, 174 are inconsistent and 34, 604 are not.

Gold standard abstractive summary. The judge views the 100 relevant and credible tweets and manually generate $\mathbf{y}^n$ the gold standard abstractive summary at time segment n. New information from related tweets must be added and conflicting information must be removed from the previous summary. For example, if we have the previous summary "Boston Marathon, 2 died, 4 injured " and a new relevant tweet "4 died, 100+ injured" in hand. The new summary will be "Boston Marathon, 4 died, 100+ injured". This abstractive gold-standard summaries are used as output for training the abstractor and end-to-end training the proposed framework. The average length of gold standard abstractive summary is 70.63 tokens.

Annotation for extractive summarization. We further annotate the 100 relevant and credible tweets, i.e. whether a tweet will be extracted to form the summary, by the dynamic selection process in [14]. A ROUGE-L score $RL(\mathbf{x}^{\mathbf{n}}_{\mathbf{d}}, \mathbf{y}^n)$ is computed against the gold standard abstractive summary $\mathbf{y}^n$ for every tweet d in the time segment n. Then we construct a list of tweets based on descending ROUGE-L score. Starting from the top tweet in the list with the largest ROUGE-L score, we add the next tweet d' into the extractive ground truth, i.e. $l_{d'} = 1$, if (1) it increases the ROUGE-L score of $RL(\mathbf{x}^n_{d'}, \mathbf{y}^n)$, where $\mathbf{x}^n_{d'}$ is the concatenation of tweets that are currently added, and (2) it is not inconsistent with previous tweets, i.e. $\forall d, l_d = 1, g(d', d) = 0$ determined by the automatic rules described in **Annotation for inconsistency detection**. The extractive gold-standard summaries are used to train extractor independently. They are also used as the input to train the abstractor independently. The average length of annotation for extractive summary is 8.02 sentences and 147.2 tokens.

7.2 Inconsistency Detection

We first show that the HID module in IAEA is able to accurately identify inconsistent tweets. Since HID is a hierarchical method, for fair comparison, the competitors in this experiment are hierarchical classifiers [58]. We used the hierarchical implementation[8] of eight state-of-the-art text encoders, which include:

- **TextCNN** [59]: convolutional neural networks (CNN) trained on top of pre-trained word vectors.
- **TextRNN** [60]: recurrent neural network with shared layer between classification tasks.
- **Transformer** [61]: stacked encoder layers based on self-attention mechanism.
- **TextRCNN** [62]: recurrent convolutional neural network which defines a left context and a right context for each token to allow bi-directional information flow.

[8]https://github.com/Tencent/NeuralNLP-NeuralClassifier

Table 4. Inconsistency detection results of different methods. Best results in bold font. Numbers marked with $^{++}$ are significant higher than competitors, with $p < 0.01$.

Method	Accuracy	Precision	Recall	F1-Score	AUC
TextCNN	0.654	0.207	0.225	0.216	0.497
TextRNN	0.676	0.213	0.196	0.204	0.501
Transformer	0.670	0.212	0.204	0.208	0.500
TextRCNN	0.670	0.213	0.206	0.209	0.500
DRNN	0.666	0.210	0.208	0.209	0.499
AttentiveConvNet	0.749	0.217	0.071	0.106	0.501
DPCNN	0.670	0.210	0.203	0.206	0.499
TextVDCNN	0.670	0.208	0.197	0.202	0.497
HID-subtract	**0.880**$^{++}$	0.713	**0.710**$^{++}$	**0.705**$^{++}$	**0.814**$^{++}$
HID-concatenate	0.863	**0.723**$^{++}$	0.681	0.702	0.805
HID-add	0.855	0.710	0.638	0.683	0.788

- **DRNN** [63]: disconnected recurrent neural network which incorporates position-invariance and limits the distance of information flow.
- **AttentiveConvNet** [64]: a CNN-style model which extends the context scope of the convolution operation to include nonlocal attention.
- **DPCNN** [65]: word-level CNN which increases the network depth by a pyramid architecture.
- **TextVDCNN** [66]: character-level CNN which uses only small convolutions and pooling operations.
- **HID**: our proposed hierarchical model with two Bi-GRU layers for inconsistency detection in IAEA. We compare different HID variants based on how the two tweet encodings are merged, i.e., subtract, add or concatenate.

For all methods, the size of word embedding is set to 100. For HID, max sequence length, maximal iteration number, batch size and dropout rate 0.2 are set to 500, 100, 64 and 0.2, respectively. In each run, the training-test generated in Section 7.1 set are randomly split to $70-30$. We repeat 10 runs and report average results. Our model is fed with all labels (i.e. irrelevant, relevant, relevant and inconsistent) as our model is a hierarchical classification method. The competitors are fed with binary training data, i.e. the inconsistent pairs of tweets are positive instances, and the rest (including relevant and irrelevant tweets) are negative instances.

We evaluate the inconsistency detection results with standard binary classification metrics. Suppose P denotes the set of positive instances (i.e., inconsistent tweet pairs), N is the negative instances (i.e., other tweet pairs), TP denotes the truly positive instances that the classifier has correctly labeled as positive, TN indicates the truly negative instances that the classifier has predicted as negative, FP is the positive instances that the classifier has wrongly labeled as positive, and FN denotes the false negative instances. We report evaluation results for competitors and the proposed HID model on the following evaluation metrics: (1) $Accuracy = \frac{TP+TN}{P+N}$ (2) $Precision = \frac{TP}{TP+FP}$ (3) $Recall = \frac{TP}{TP+FN}$ (4) $F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$ (5) AUC: the area under ROC curve. We use AUC as our main evaluation metric.

As shown in Table 4, HID obtains the best classification performance in terms of Accuracy, Precision, Recall, F1-score and AUC. HID performances are significant better than all competitors. However, among the HID variants, we do not see significant difference which implies that how to merge the tweet encodings do not affect detection performance.

Recently pre-training [67, 68] has shown promising results in many NLP tasks. Thus, to understand the impact of sentence representation learning, we evaluate HID's performance by replacing the Bi-GRU layers with two sentence-level pre-trained representation, i.e., doc2vec [67] and

Table 5. Inconsistency detection results of different sentence representations. Best results in bold font. Numbers marked with $^{++}$ are significant with $p < 0.01$.

Method	Accuracy	Precision	Recall	F1-Score	AUC
doc2vec	0.732	0.440	**0.958**	0.602	0.802
Sentence-BERT	0.705	0.415	0.952	0.578	0.796
HID	**0.880**$^{++}$	**0.713**$^{++}$	0.710	**0.705**$^{++}$	**0.814**$^{++}$

Sentence-BERT [68]. As shown in Table 5, Bi-GRU is better than pre-trained sentence representation vectors, such as doc2vec and Sentence-Bert, in terms of most metrics (i.e. Accuracy, Precision, F1-Score and AUC). This suggests that a task specific representation learning is more powerful than a pre-trained representation.

7.3 Quantitative Summarization Evaluation

To evaluate the quality of summary produced by IAEA, we first provide quantitative evaluation based on manually generated gold standard summaries. We perform 70 − 30 random split. That is, we randomly select 21 events out of the 30 events for training, repeat for 10 times and report the average results. We perform splitting on events to avoid influence of event specific information, i.e. if the following tweets in an event are too similar to the previous summary, the system will produce biased results. Furthermore, such a training protocol allows us to test the ability of summarization system to generate summaries for unseen events.

We compare our method to several state-of-the-art summarization approaches:

- **MSSF** [69]: It is an abstractive approach of multi-document summarization based on submodularity hidden in textual-unit similarity property.
- **SNMF** [70]: It is a summarization method based on symmetric non-negative matrix decomposition.
- **MWDS** [7]: It is a language model which obtains relevant tweets using dynamic pseudo relevance feedback and then generate storylines via graph optimization.
- **Sumblr** [10]: It is the online tweet summarization approach based on incremental clustering.
- **Simplex** [12]: It models the realtime summarization problem as multiple integer programming problems and solves the relaxed linear programming form by an improved simplex update method. To reduce the storage and computational cost of expensive inconsistency detection, it embeds a novel fast inconsistency detection strategy in the simplex update algorithm.
- **RL Abstractor** [50]: It is a neural network model with intra-attention and a new training method. This method combines standard supervised word prediction and reinforcement learning (RL).
- **Unified Model** [14]: It is a unified model combining the strength of extractive and abstractive summarization. The simple extractive model can obtain the sentence-level attention with high ROUGE scores and a more complicated abstractive model can capture the word-level dynamic attention to generate a more readable paragraph. The loss coefficient for Unified Model and IAEA are set as [14] $\lambda_1 = 5, \lambda_2 = 1, \lambda_3 = 1$.

For the ablation study, we compare IAEA and its two variants IAEA$_{-H}$ and IAEA$_r$. To measure the impact of HID, we remove HID in the summarization framework and provide IAEA$_{-H}$, which does not perform inconsistency detection in learning sentence-level attention. Note that IAEA$_{-H}$ is different compared to Unified Model [14] in the design of extractor and abstractor. To measure the impact of sentence orders, we provide IAEA$_r$, which concatenates sentences in a random order in the abstractor.

We used the standard ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [71] and BLEU [72] metrics for evaluating the quality of the summaries generated. ROUGE automatically determines the quality of a summary by comparing it with the gold-standard summaries through counting the number of their overlapping textual units (e.g., n-gram, word sequences, and etc.). There are different ROUGE measures. We report ROUGE-1, ROUGE-2 and ROUGE-L. For $n = 1, 2$, ROUGE-n is computed depending on the number of matching n-grams.

$$\text{ROUGE} - \text{n} = \frac{\sum_{S \in \{gold-standard\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{gold-standard\}} \sum_{gram_n \in S} Count(gram_n)} \tag{25}$$

where $Count_{match}(gram_n)$ is the number of matching n-grams in the result and $Count(gram_n)$ is the number of n-grams in the gold-standard summary. ROUGE-L is computed based on the longest common sequence.

$$\begin{aligned} R &= \frac{LCS(gold - standard, summary)}{m} \\ P &= \frac{LCS(gold - standard, summary)}{n} \\ ROUGE - L &= \frac{2RP}{R + P} \end{aligned} \tag{26}$$

where $LCS(gold - standard, summary)$ is the longest common sequence between the gold-standard and the result summary, m is the length of gold-standard summary, n is the length of result summary.

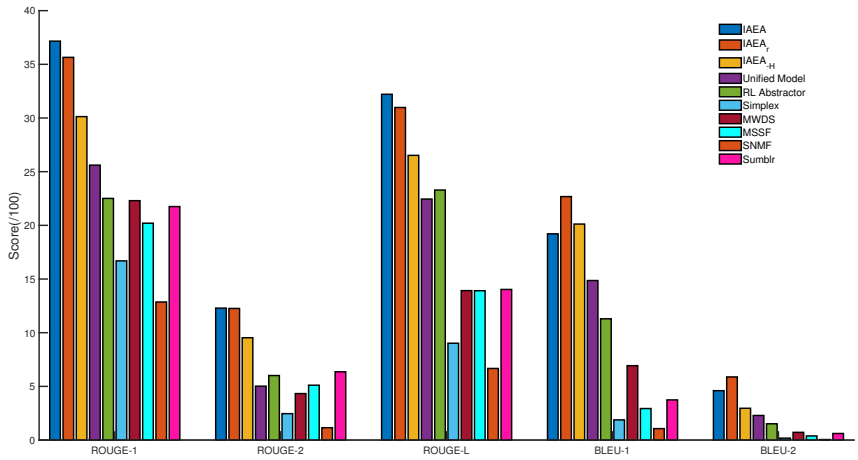

Fig. 6. Average Rouge and BLEU scores

The results are shown in Figure 6. We have several interesting observations: (1) Our proposed system IAEA and its two variants perform significantly better than all the competitors in terms of all evaluation metrics. It is clear that, despite of the nature of events, the proposed system is able to model the summarization task well. (2) Removing HID has a negative impact on summarization performance, as IAEA$_{-H}$ performs worse than IAEA in most metrics, i.e. ROUGE-1, ROUGE-2, ROUGE-L and BLEU-2. This observation suggests that inconsistency detection is a major contributor

Table 6. Automatic readability evaluation results, best results shown in bold font.

Method	Flesch Kincaid	SMOG	Dale Chall Readability	Coleman Liau Index	Gunning Fog
Simplex	10.02	12.75	10.74	11.52	11.4
MWDS	10.43	10.58	10.66	12.02	11.59
MSSF	10.53	12.25	11.59	13.84	12.57
SNMF	9.99	11.47	11.42	12.58	11.42
Sumblr	10.39	10.58	9.88	11.69	11.67
Unified Model	7.94	10.72	9.60	10.75	10.22
RL Abstractor	9.49	9.03	**12.03**	**14.53**	8.23
IAEA	**12.35**	**13.71**	11.57	14.25	**14.95**

that boosts the summary quality. The difference on BLEU-1 is relatively small, i.e., BLEU-1 is 20.12 without HID and 19.21 with HID. One possible explanation for the slightly improved BLEU-1 score is that, by removing an inconsistency detection component HID, the summaries will be less often updated and thus more unigram matches. (3) On ROUGE-2 and ROUGE-3, Unified Model [14], which is also an extractive-abstractive model, is less powerful than the sole abstractive model [50]. However, our method generates better summaries on all metrics, which verifies the improvement by incorporating reinforcement learning in extractive-abstractive model. (4) According to the comparison between IAEA and IAEA$_r$, the sequential orders of sentences has ambivalent effects on summarization performance. IAEA$_r$ generates worse ROUGE-1 and ROUGE-L results than IAEA, but better BLEU-1 and BLEU-2 results, and similar ROUGE-2 results. In general, neither of the chronological order and the random order has a dominant performance than the other. (5) Finally, IAEA is especially more powerful than Sumblr [10], which is also a real-time tweet summarization system and is the best of all non-NN competitors, on Rouge-L.

7.4 Readability

Furthermore, we provide qualitative evaluations on the automatically generated summaries.

The first qualitative evaluation is to automatically calculate the readability of the resulted summaries based on readability formulas[9]. The automatic readability formulas estimate the years of education a reader needs to understand the summary. Thus, they could be good and objective surrogate to measure the writing quality of the summaries. The competitors are the same as in the previous subsection.

As shown in Table 6, the proposed IAEA system performs best in terms of most automatic readability tests, i.e. Flesch Kincaid, SMOG, and Gunning Fog. It is comparable to RL Abstractor on Coleman Liau Index. Both of them are the only two methods that have a Coleman Liau Index higher than 14. The result shows that IAEA generates readable summaries.

Besides automatically calculated readability evaluation, we also provide quality metrics based on human evaluation, i.e. readability, completeness, compactness and correctness [73]. As in [73], we request five evaluators to complete evaluations of 25 summaries on 5 events, for a total of 625 ratings. Each evaluation metric is considered separately. We ask each evaluator to use a Likert scale rating, where rating one is for “Not at all"; rating two for “Not very"; rating three for “Somewhat"; rating four for “Very"; and rating five for “Absolutely". The ratings are assigned to four summary criteria: (1) readability: a summary is easy to read and understand; (2) completeness: a summary captures all relevant topics in the current event; (3) compactness: a summary does not repeat information; (4) correctness: a summary covers the current status of the event.

[9]https://py-readability-metrics.readthedocs.io/en/latest/

Table 7. Qualitative evaluation results based on human evaluation, best results in bold font.

Method	Readability	Completeness	Compactness	Correctness
Simplex	3.28	3.17	3.12	3.11
MWDS	3.10	3.01	2.88	2.92
MSSF	3.13	2.85	2.77	2.95
SNMF	2.89	2.93	2.95	3.01
Sumblr	3.32	3.13	3.14	3.07
Unified Model	3.37	3.22	3.42	3.33
RL Abstractor	2.77	2.89	2.83	2.97
IAEA	**3.52**	**3.78**	**3.65**	**3.61**

As shown in Table 7, the proposed IAEA system performs best in terms of all human evaluation metrics. The result shows that IAEA generates complete, compact and correct summary, which are important for understanding an event. Compared to the results in Table 6, RL Abstractor performs better than Unified Model in human readability evaluation. This is because in human evaluation, we can observe that RL Abstractor tends to generate more repeated terms and thus reduces readability drastically.

7.5 Integrity of Summaries

Next we evaluate the integrity of summaries produced by different systems. Our goal is to show how much inconsistent information is contained in each summary update for each time segment given each event, we perform automatic and manual test. For **automatic test**, we adopt the LCS rules in Section 7.1 to detect inconsistent sentence pairs in the summary generated for the time segment. Then, the number of detected inconsistent sentence pairs will be divided by the number of possible sentence pairs to obtain the percentage in time segment n,, i.e. $InconsistencyRatio = \frac{\#inconsistent}{(S_n(S_n-1)/2}$ where S_n is the number of sentences in time n. For **manual test**, we manually check all results and select inconsistent sentence pairs in each summary update generated for the time segment.

We report the distribution of inconsistency ratio and the mean inconsistency ratio by different methods, over all events and all time segments, in Figure 7 and Figure 8. We have the following observations. (1) The inconsistency ratio of our proposed method is near zero by both automatic test and manual test. This shows that IAEA indeed preserves the integrity of summaries. (2) There exists a behavior gap for the competitors, i.e. most methods perform differently on the automatic check and manual check. For example, Simplex performs well on automatic check and less well on manual check, as it implements a LCS based inconsistency check [12] in updating the summary which is exactly the steps of automatic check. The same behavior gap exists for other methods. For example, MSSF, SNMF and Sumbl have narrower inconsistency ratio distribution on manual check. On the contrary, IAEA does not have a behavior gap. Although the weak supervision for the HID component of IAEA is based on LCS (i.e. the same as the automatic check), IAEA still has the best performance on manual check. (3) IAEA performs robustly well on all events, i.e., the box is narrow. However, the state-of-the-art comparative methods can not generate a consistent summary along the timeline for all events. They do not have consistency check between the tweets, hence it is possible that some inconsistent tweets got selected into the summary at some timepoints and are never replaced later. For example, Unified model performs best among all competitors without explicit inconsistency detection. However, it has many outliers, i.e. the mean inconsistency ratio is larger than the medium on both automatic check and manual check. This shows that unified model does not have a stable performance over all events and all time segments. (4) Removing the inconsistency detection module (i.e., HID) significantly increases the amount of conflicting information. IAEA$_{-H}$ has larger inconsistency ratio than Unified Model on both automatic check

and manual check. Thus, we confirm the importance of including an explicit inconsistency detection module. (5) The order of incoming tweet sequence to feed abstractor has little impact on the integrity of summaries. The performance of IAEA$_r$ is generally indistinguishable with IAEA.

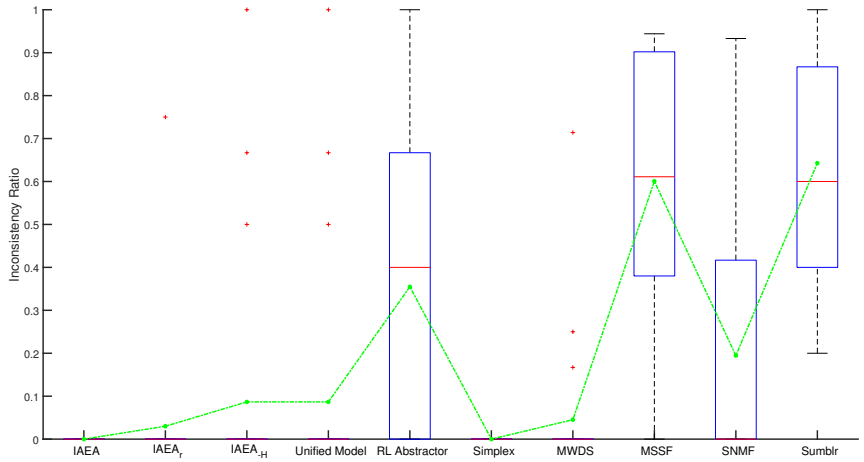

Fig. 7. Inconsistency ratio by automatic test. Best shown in color. Green line for the mean inconsistency ratio.

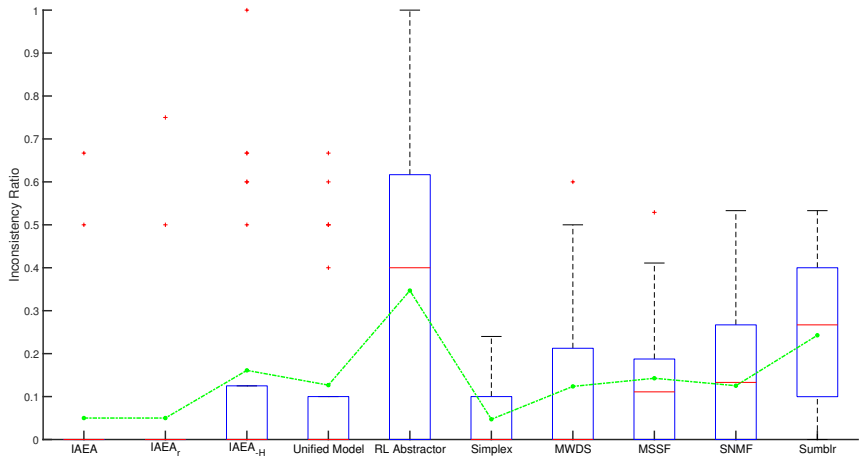

Fig. 8. Inconsistency ratio by manual test. Best shown in color. Green line for the mean inconsistency ratio.

7.6 Efficiency

The speed of IAEA is of equal magnitude to the competitors. As shown in Table 8, the training time of IAEA takes about 30 hours, most of which is spent on inconsistency detection. Removing HID

Table 8. Training time and summary update time for each model

Method	IAEA	IAEA_H	Unified Model	RL Abstractor	Simplex	MWDS	MSSF	SNMF	Sumblr
Training(hour)	30	10	10	9	N/A	N/A	N/A	N/A	N/A
Restore(minute)	8	4	4	2	N/A	N/A	N/A	N/A	N/A
Output(minute)	4	4	4	4	2.6	36	1.1	19.5	1

reduces the training time to roughly 10 hours, which is comparable to the Unified Model (10 hours) and RL Abstractor (9 hours). However, this is a common trade-off in DNN models. The inclusion of HID boosts the performance of summarization, as shown in the experimental section, while it requires more training time. We believe the training speed is not a crucial issue, as training is usually implemented offline. Once training is finished, modern deep neural network models are as fast as conventional unsupervised methods in updating the summary. Generally, the NN models spent equal time to restore model and output summary (i.e., in several minutes) in each time slice. Removing the HID can speed up model restore, but does not affect output speed.

7.7 Case Study

We give a case study of an event where a series of coordinated terrorist attacks Paris. We present the result summaries by different competitors in Table 9 and Table 10. We underline the obvious errors, including duplicate, redundant and inconsistent ones. Note the listed examples are only a small portion of all the errors. We highlight up-to-date information that should be conveyed in each summary update in bold font. We have the following observations:

- Our proposed IAEA method accurately updates the information without providing inconsistent information. For example, IAEA correctly updates the number of death to 12 in Time 1.
- Simplex is the only competitor that explicitly deletes inconsistent information in updating summary. However, it fails to update critical information. For example, it does not mention the number of deaths at all in Time 1.
- The rest of the competitors provide conflicting numbers of death and injuries, which will be very confusing for readers.
- Extractive methods generally generate more readable summaries than abstractive neural summarization systems. However, they do not cover the most important information. For example, the summaries generated by Simplex and Sumblr are very verbose with much longer summaries, and contain redundant information. MSSF provides brief and fluent sentences. But the most crucial information such as the number of deaths is not included.

8 CONCLUSION

Although text summarization has been extensively studied, there is one critical problem which has not been solved in realtime event summarization scenario: how to preserve the integrity of summaries at each update. We believe this problem is very important when dealing with the flooding of information nowadays, especially on online social networking platforms. Given that the problem stays largely unexplored, we believe our work opens a new direction for realtime event monitoring and summarization. In this article, we tackle this problem systematically. We formulate the problem of detecting inconsistent tweet pairs as a hierarchical classification problem, and propose a hierarchical neural network model to deliver accurate predictions. We present a novel summarization system IAEA, which is a unified extractive-abstractive framework with inconsistency detection module embedded. The system has several advantages: (1) It implements incremental update where previous inconsistent sentences are replaced by the extracted new and important

Table 9. Real-time summaries output by different summarization methods: Part I. Errors including duplicate phrases, redundant or inconsistent information are underlined, up-to-date information in bold.

Method	Time 0	Time 1	Time 2
IAEA	tells fired inside paris office of offices shooting charlie hebdo , media reports say france : 10 killed as multiple shooting at hq of satirical weekly newspaper #charliehebdo, according to witnesses fired fired in shooting hebdo attack deadly albaghdadi	respond charlie hebdo, paris france: **12** raises dead, french president says it s director victims injured at confirmed newspaper s office	12 killed in shooting at paris magazine charlie hebdo, **included 2 police, france raises national alert**. #jesuischarlie **gunmen at large , paris on 3 gunmen** storm newspaper charlie hebdo attack in paris offices storm
Unified model	people have dead after shooting six, policeman hit in france shooting in charlie hebdo attack. hq people 11 dead in paris shooting seriously people in involved at french shooting stormed .	10 people people dead after shooting reportedly paris , dead in paris.12 dead media #paris after french paris a attacks hebdo in #paris after gunmen stormed the office dead #charliehebdo shooting of attack hq has anything to in a shooting at , reports .	people dead shooting at have been used the escape used and survivors show solidarity with shooting victims of after shooting . hq of killed after armed muslims at the ' the newspaper france charlie, reports .
RL Abstractor	paris hebdo 11 dead. paris hebdo dead 11. *11 gunmen dead.*	12 dead in charlie 12 in hebdo 12 in shooting charlie in the charlie.	12 in in in in in killed in in shooting in in charlie in in hebdo
Simplex	Deadly attack on office of French magazine Charlie Hebdo. Unconfirmed reports claim 10 people died. BBC News - Charlie Hebdo: Gun attack on French magazine kills 11 shocking news of a dispicable terrorist attack. 11 people killed in carnage at satirical magazine 'Charlie Hebdo'office in Paris, which published blasphemous caricatures a few years ago. Shots fired! Paris police say shots fired at satirical newspaper Charlie Hebdo; witness says multiple gunmen involved.	Live Updates on Deadly Shooting at Paris Newspaper - NYTimes. Sickest thing on Twitter today. I RT an Israeli friend's tweet about the #CharlieHebdo shooting we get a reply that Hitler should return. #Freespeech is a non-negotiable human right. We condemn the appalling attack on Charlie Hebdo, and any attempt to silence the free press. #CharlieHebdo shooting we get a reply that Hitler should return.	Live Updates on Deadly Shooting at Paris Newspaper - NYTimes. Sickest thing on Twitter today. I RT an Israeli friend's tweet about the #CharlieHebdo shooting we get a reply that Hitler should return. Mayor to react to #CharlieHebdo shooting at city hall at 13h; vigil planned there this evening at 17h #cbcmtl

tweets. (2) It preserves the integrity of summaries at each update by explicitly predicting inconsistent information. (3) It inherits the ability of abstractive summarization to boost coherence for better readability, while avoiding factual errors by combining extractive summarization. We further address the common bottleneck of training neural networks by introducing weak supervisions which are empirically verified and easy to obtain. These methodology contributions are beneficial for developing practical solutions in text generation.

For future work, it is worthy to study the efficiency issue in realtime event summarization systems. The efficiency issue will be more appealing when the summarization system is combined

Table 10. Real-time summaries output by different summarization methods: Part II. Errors including duplicate phrases, redundant or inconsistent information are underlined, up-to-date information in bold.

Method	Time 0	Time 1	Time 2
Sumblr	Police: 11 people dead and 10 injured in shooting at satirical magazine Charlie Hebdo's Paris office. 11 people killed in an attack at the #Paris office of French satirical magazine Charlie Hebdo. BREAKING; Gunmen Attack Paris Office Of French Satirical Magazine Charlie Hebdo, killing 11 People - BBC. LIVE - 11 people killed after shooting at French satirical newspaper #CharlieHebdo. 11 dead in attack on Paris office of satirical magazine Charlie Hebdo. Charlie Hebdo Shooting: 10 dead in shooting at French satirical weekly Charlie Hebdo in Paris. Gunmen attacked a Paris office of French satirical magazine Charlie Hebdo, killing **10** people.	Shocking news from Paris Charlie Hebdo attack: **12** dead at Paris offices of satirical magazine. 11 people killed in an attack at the #Paris office of French satirical magazine Charlie Hebdo. BREAKING; Gunmen Attack Paris Office Of French Satirical Magazine Charlie Hebdo, killing 11 People - BBC. Charlie Hebdo Shooting: 10 dead in shooting at French satirical weekly Charlie Hebdo in Paris. Video: At least one dead in two explosions at Brussels Airport, reports of blast at metro station. At least one dead in shooting at #Paris offices of satirical magazine #CharlieHebdo. Shocking news from Paris Charlie Hebdo attack: 12 dead at Paris offices of satirical magazine. #CharlieHebdo paris shooting think it's now 12 dead.	My thoughts are with #Paris right now. Charlie Hebdo Shooting: 12 Dead In Attack On Paris Satirical Newspaper. Gunmen attacked a Paris office of French satirical magazine Charlie Hebdo, killing 10 people. BREAKING; Gunmen Attack Paris Office Of French Satirical Magazine Charlie Hebdo, killing 11People - BBC. Police: 11 people dead and 10 injured in shooting at satirical magazine Charlie Hebdo's Paris office. Charlie Hebdo Shooting: 10 dead in shooting at French satirical weekly Charlie Hebdo in Paris. Shocking news from Paris Charlie Hebdo attack: 12 dead at Paris offices of satirical magazine. #CharlieHebdo paris shooting think it's now 12 dead.
MSSF	Charlie Hebdo Headquarter attacked, 10 killed. Ten people killed at Charlie Hebdo. Charlie Hebdo - Wikipedia, the free encyclopedia. So sad for Charlie Hebdo, so sad for Paris.	Terrorist attack at Charlie Hebdo Paris. Charlie Hebdo Shooting: Paris Terror Attack. Attack on Charlie Hebdo Paris Headquarters. Tweets from Charlie Hebdo attack scene. Charlie Hebdo: an irreverent French fixture.	News: Attack on Charlie Hebdo Office. French journalists from Charlie Hebdo. Shocking video of Charlie Hebdo attack.

with event monitoring, i.e., detect the burst of an event and deliver realtime summaries. As an event is usually associated with multiple entities, it is also interesting to incorporate the concepts of entities and specify the inconsistency at entity-level.

9 ACKNOWLEDGEMENTS

Chen Lin is the corresponding author. Chen Lin is supported by the Natural Science Foundation of China (No. 61972328), Joint Innovation Research Program of Fujian Province China (No.2020R0130). Hui Li is supported by the Natural Science Foundation of China (No. 62002303), Natural Science Foundation of Fujian Province China (No. 2020J05001). Xiaoli Wang is supported by the International

Cooperation Projects of Fujian Province China (No. 201810016). Huang Zhenhua is supported by the National Natural Science Foundation of China under Grant 61772366, the National Natural Science Foundation of China (Key Program) under Grant U1811263, and the Shanghai Committee of Science and Technology under Grant 17ZR1445900, 17070502800, 18ZR1428300.

REFERENCES

[1] Murat Demirbas, Murat Ali Bayir, Cuneyt Gurcan Akcora, Yavuz Selim Yilmaz, and Hakan Ferhatosmanoglu. Crowd-sourced sensing and collaboration using twitter. In WOWMOM, pages 1–9. IEEE Computer Society, 2010.

[2] M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. Identifying local events by using microblogs as social sensors. In OAIR, pages 173–180. ACM, 2013.

[3] Merrin Fabre. Use of social media for internal communication: A case study in a government organisation. In Social Media for Government Services, pages 51–74. Springer, 2015.

[4] Gina Ciancio and Amanda Dennett. Social media for government services: A case study of human services. In Social Media for Government Services, pages 25–49. Springer, 2015.

[5] Na Yeon Lee, Yonghwan Kim, and Yoonmo Sang. How do journalists leverage twitter? expressive and consumptive use of twitter. The Social Science Journal, 54(2):139 – 147, 2017.

[6] Mehreen Gillani, Muhammad U. Ilyas, Saad Saleh, Jalal S. Alowibdi, Naif R. Aljohani, and Fahad S. Alotaibi. Post summarization of microblogs of sporting events. In WWW (Companion Volume), pages 59–68. ACM, 2017.

[7] Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. Generating event storylines from microblogs. In CIKM, pages 175–184. ACM, 2012.

[8] Zhi Liu, Yan Huang, and Joshua R. Trampier. LEDS: local event discovery and summarization from tweets. In SIGSPATIAL/GIS, pages 53:1–53:4. ACM, 2016.

[9] Koustav Rudra, Subham Ghosh, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. Extracting situational information from microblogs during disaster events: a classification-summarization approach. In CIKM, pages 583–592. ACM, 2015.

[10] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. Sumblr: continuous summarization of evolving tweet streams. In SIGIR, pages 533–542. ACM, 2013.

[11] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In HT, pages 319–320. ACM, 2012.

[12] Lingting Lin, Chen Lin, and Yongxuan Lai. Realtime event summarization from tweets with inconsistency detection. In ER, volume 11157, pages 555–570. Springer, 2018.

[13] Qunhui Wu, Jianghua Lv, and Shilong Ma. Continuous summarization for microblog streams based on clustering. In ICONIP, volume 9490, pages 371–379. Springer, 2015.

[14] Wan Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. A unified model for extractive and abstractive summarization using inconsistency loss. In ACL, pages 132–141. ACL, 2018.

[15] Koustav Rudra, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, and Prasenjit Mitra. Summarizing situational tweets in crisis scenario. In HT, pages 137–147. ACM, 2016.

[16] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In ACL, pages 1073–1083. Association for Computational Linguistics, 2017.

[17] Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. Multiview convolutional neural networks for multi-document extractive summarization. IEEE Trans. Cybernetics, 47(10):3230–3242, 2017.

[18] Yong Zhang, Meng Joo Er, and Mahardhika Pratama. Extractive document summarization based on convolutional neural networks. In IECON, pages 918–922. IEEE, 2016.

[19] Zhongqing Wang and Yue Zhang. A neural model for joint event detection and summarization. In IJCAI, pages 4158–4164. ijcai.org, 2017.

[20] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In AAAI, pages 3075–3081. AAAI Press, 2017.

[21] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir R. Radev. Graph-based neural multi-document summarization. In CoNLL, pages 452–462. Association for Computational Linguistics, 2017.

[22] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In EMNLP, pages 379–389. The Association for Computational Linguistics, 2015.

[23] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In CoNLL, pages 280–290. ACL, 2016.

[24] Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. Entity-centric topic-oriented opinion summarization in twitter. In KDD, pages 379–387. ACM, 2012.

[25] Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. Summarizing a document stream. In ECIR, volume 6611, pages 177–188. Springer, 2011.

[26] Koustav Rudra, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. Extracting and summarizing situational information from the twitter social media during disasters. TWEB, 12(3):17:1–17:35, 2018.

[27] A. P. Naik and S. Bojewar. Tweet analytics and tweet summarization using graph mining. In ICECA, volume 1, pages 17–21, April 2017.

[28] Ilkin Huseynli and M. Elif Karsligil. Determination and summarization of important tweets after natural disasters. In SIU, pages 1–4. IEEE, 2017.
[29] Muhammad Asif Hossain Khan, Danushka Bollegala, Guangwen Liu, and Kaoru Sezaki. Multi-tweet summarization of real-time events. In SocialCom, pages 128–133. IEEE Computer Society, 2013.
[30] Soumi Dutta, Asit Kumar Das, Abhishek Bhattacharya, Gourav Dutta, Komal K. Parikh, Atin Das, and Dipsa Ganguly. Community detection based tweet summarization. In Emerging Technologies in Data Mining and Information Security, pages 797–808. Springer Singapore, 2019.
[31] Jin Yao Chin, Sourav S. Bhowmick, and Adam Jatowt. TOTEM: personal tweets summarization on mobile devices. In SIGIR, pages 1305–1308. ACM, 2017.
[32] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In RecSys, pages 199–206. ACM, 2010.
[33] Miles Efron and Gene Golovchinsky. Estimation methods for ranking recent information. In SIGIR, pages 495–504. ACM, 2011.
[34] Liuqing Li, Jack Geissinger, William A. Ingram, and Edward A. Fox. Teaching natural language processing through big data text summarization with problem-based learning. Data and Information Management, 4(1):18–43, 2020.
[35] Meng Xu, Xin Zhang, and Lixiang Guo. Jointly detecting and extracting social events from twitter using gated bilstm-crf. IEEE Access, 7:148462–148471, 2019.
[36] Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W. Cohen. Tweet2vec: Character-based distributed representations for social media. In ACL. The Association for Computer Linguistics, 2016.
[37] Sara Melvin, Wenchao Yu, Peng Ju, Sean D. Young, and Wei Wang. Event detection and summarization using phrase network. In ECML/PKDD, volume 10536, pages 89–101. Springer, 2017.
[38] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K. Kalita. Summarizing microblogs automatically. In HLT-NAACL, pages 685–688. The Association for Computational Linguistics, 2010.
[39] Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. A tweet summarization method based on maximal association rules. In ICCCI, volume 11055, pages 373–382. Springer, 2018.
[40] Nuno Dionísio, Fernando Alves, Pedro Miguel Ferreira, and Alysson Bessani. Cyberthreat detection from twitter using deep neural networks. In IJCNN, pages 1–8. IEEE, 2019.
[41] Abdelhamid Chellal and Mohand Boughanem. Optimization framework model for retrospective tweet summarization. In SAC, pages 704–711. ACM, 2018.
[42] Yue Huang, Chao Shen, and Tao Li. Event summarization for sports games using twitter streams. World Wide Web, 21(3):609–627, 2018.
[43] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In ACL. The Association for Computer Linguistics, 2016.
[44] Shashi Narayan, Nikos Papasarantopoulos, Mirella Lapata, and Shay B. Cohen. Neural extractive summarization with side information. arXiv Preprint, 2017. URL https://arxiv.org/abs/1704.04530.
[45] Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Liqiang Nie, Jun Ma, and Maarten de Rijke. Sentence relations for extractive summarization with deep neural networks. ACM Trans. Inf. Syst., 36(4):39:1–39:32, 2018.
[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.
[47] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, pages 1724–1734. ACL, 2014.
[48] Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. In ACL, pages 5070–5081. Association for Computational Linguistics, 2019.
[49] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. Abstractive document summarization with a graph-based attentional neural model. In ACL, pages 1171–1181. ACL, 2017.
[50] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In ICLR. OpenReview.net, 2018.
[51] Eric Chu and Peter J. Liu. Meansum: A neural model for unsupervised multi-document abstractive summarization. In ICML, volume 97 of Proceedings of Machine Learning Research, pages 1223–1232. PMLR, 2019.
[52] Yang Liu and Mirella Lapata. Learning structured text representations. Trans. Assoc. Comput. Linguistics, 6:63–75, 2018.
[53] Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In ACL, pages 1074–1084. Association for Computational Linguistics, 2019.
[54] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In EMNLP, pages 1532–1543. ACL, 2014.

[55] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In CVPR, pages 1179–1195. IEEE Computer Society, 2017.
[56] Arkaitz Zubiaga. A longitudinal assessment of the persistence of twitter datasets. J. Assoc. Inf. Sci. Technol., 69(8): 974–984, 2018.
[57] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In ACL (System Demonstrations), pages 55–60. The Association for Computer Linguistics, 2014.
[58] Liqun Liu, Funan Mu, Pengyu Li, Xin Mu, Jing Tang, Xingsheng Ai, Ran Fu, Lifeng Wang, and Xing Zhou. Neuralclassifier: An open-source neural hierarchical multi-label text classification toolkit. In ACL (3), pages 87–92, 2019.
[59] Yoon Kim. Convolutional neural networks for sentence classification. In EMNLP, pages 1746–1751. ACL, 2014.
[60] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. In IJCAI, pages 2873–2879. IJCAI/AAAI Press, 2016.
[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, pages 5998–6008, 2017.
[62] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In AAAI, pages 2267–2273. AAAI Press, 2015.
[63] Baoxin Wang. Disconnected recurrent neural networks for text categorization. In ACL, volume 1, pages 2311–2320. Association for Computational Linguistics, 2018.
[64] Wenpeng Yin and Hinrich Schütze. Attentive convolution: Equipping cnns with rnn-style attention mechanisms. Trans. Assoc. Comput. Linguistics, 6:687–702, 2018.
[65] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. In ACL, volume 1, pages 562–570. Association for Computational Linguistics, 2017.
[66] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. Very deep convolutional networks for text classification. In EACL, volume 1, pages 1107–1116. Association for Computational Linguistics, 2017.
[67] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In Rep4NLP@ACL, pages 78–86. Association for Computational Linguistics, 2016.
[68] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In EMNLP/IJCNLP, volume 1, pages 3980–3990, 2019.
[69] Jingxuan Li, Lei Li, and Tao Li. MSSF: a multi-document summarization framework based on submodularity. In SIGIR, pages 1247–1248. ACM, 2011.
[70] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris H. Q. Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In SIGIR, pages 307–314. ACM, 2008.
[71] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pages 74–81. Association for Computational Linguistics, 2004.
[72] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In ACL, pages 311–318. ACL, 2002.
[73] Giuseppe Di Fabbrizio, Amanda Stent, and Robert J. Gaizauskas. A hybrid approach to multi-document summarization of opinions in reviews. In INLG, pages 54–63, 2014.