PDSum: Prototype-driven Continuous Summarization of Evolving Multi-document Sets Stream

Susik Yoon
UIUC
susik@illinois.edu

Hou Pong Chan
University of Macau
hpchan@um.edu.mo

Jiawei Han
UIUC
hanj@illinois.edu

ABSTRACT

Summarizing text-rich documents has been long studied in the literature, but most of the existing efforts have been made to summarize a static and predefined multi-document set. With the rapid development of online platforms for generating and distributing text-rich documents, there arises an urgent need for continuously summarizing dynamically evolving multi-document sets where the composition of documents and sets is changing over time. This is especially challenging as the summarization should be not only effective in incorporating relevant, novel, and distinctive information from each concurrent multi-document set, but also efficient in serving online applications. In this work, we propose a new summarization problem, Evolving Multi-Document sets stream Summarization (EMDS), and introduce a novel unsupervised algorithm PDSum with the idea of prototype-driven continuous summarization. PDSum builds a lightweight prototype of each multi-document set and exploits it to adapt to new documents while preserving accumulated knowledge from previous documents. To update new summaries, the most representative sentences for each multi-document set are extracted by measuring their similarities to the prototypes. A thorough evaluation with real multi-document sets streams demonstrates that PDSum outperforms state-of-the-art unsupervised multi-document summarization algorithms in EMDS in terms of relevance, novelty, and distinctiveness and is also robust to various evaluation settings.

CCS CONCEPTS

• **Information systems** → **Summarization**; **Web searching and information discovery**; **Data stream mining**.

KEYWORDS

Continuous summarization, Evolving multi-document sets, Unsupervised text summarization

ACM Reference Format:
Susik Yoon, Hou Pong Chan, and Jiawei Han. 2023. PDSum: Prototype-driven Continuous Summarization of Evolving Multi-document Sets Stream. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543507.3583371

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583371

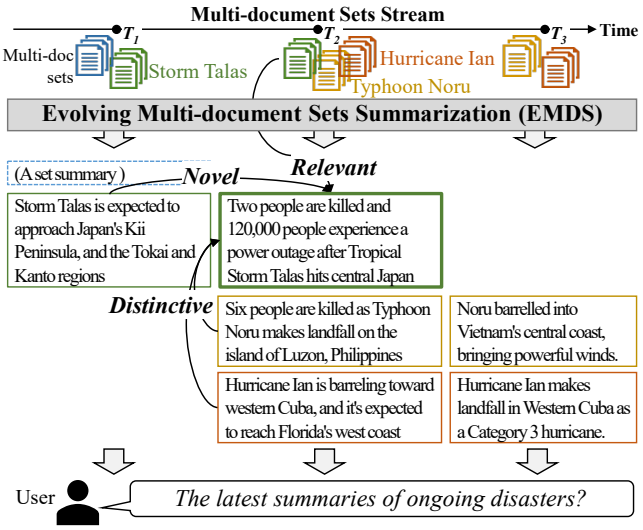


Figure 1: An example of continuous summarization of evolving multi-document sets stream (best viewed in color).

1 INTRODUCTION

The rapid development of web-based platforms and digital journalism leads to abundant text-rich content generated in real-time such as news articles, blog posts, online reviews, and scientific articles [8, 14, 18]. As the scale and speed of the generated document streams are overwhelming, it is not feasible for people to digest all the documents of interest by themselves. There have been long-standing efforts devoted to transforming such text-rich documents into a brief and concise summary. Existing studies assume documents of interest (e.g., articles of a news story) are given and return summaries with an abstractive or extractive approach [2, 8, 28]. The recent large-scale language models have further improved the quality of automatic summarization [28, 47, 55].

The existing studies, however, are not sufficient to meet emerging practical needs for summarization, as they typically target a *fixed document set of a single interest*. In practice, it is common for people to have multiple interests together and stay up-to-date on each interest concurrently [7, 39, 53]. Likewise, a user may track *multiple document sets of different interests* and expect their up-to-date summaries, while the user's interests and the corresponding documents change over time. We refer to the summarization task in this dynamic scenario as *Evolving Multi-Document Sets Stream Summarization (EMDS)*.

Figure 1 illustrates an example scenario of EMDS. A user would like to get the latest summaries of news stories about disasters around the world at a certain time interval (e.g., every day). The queries specifying particular news stories can be manually issued

by the user or automatically provided by an application. For instance, at T_2, news articles about `Storm Talas`, `Typhoon Noru`, and `Hurricane Ian` are published. The related articles arrive at varying rates as news stories emerge, continue, and expire. By continuously providing the latest summaries for each news story, the user can easily stay up-to-date with recent news stories of interest. These summaries can be utilized in various downstream tasks such as news curation, event detection, and topic mining [16, 20, 21, 52].

Despite the practicality and benefits of EMDS, there are unique challenges in EMDS that limit the adoption of existing studies.
(1) The summarization should consider *documents-, sets- and time-aware themes comprehensively*. For instance, in Figure 1, the summary for the story (i.e., set) `Storm Talas` at T_2 should be *relevant* to the current articles (i.e., documents) in the story at T_2, *novel* to the previous articles in the story at T_1, and *distinctive* from the articles in the other stories. Existing methods typically focus on the relevance of a summary to a target document set but do not consider its novelty compared with the previous documents and/or its distinctiveness to the other sets of documents.
(2) The summarization should be conducted *with single-pass processing without access to previous documents*. As it is not feasible for online applications to store continuous and unbounded data streams, streaming algorithms typically adopt single-pass processing of data streams for efficiency [7, 50, 51]. Similarly in EMDS, once news summaries are derived from the latest document sets, it is more practical to discard them immediately. This makes it more difficult for existing methods to keep track of relevant, novel, and distinctive themes of evolving multi-document sets.

To fill this gap, we propose a novel *unsupervised* summarization method, **PDSum** (Prototype-Driven continuous Summarization for evolving multi-document sets stream), targetting the newly introduced EMDS task. PDSum builds *lightweight prototypes* for multi-document sets to embed and summarize new documents over evolving multi-document sets stream. The set prototypes incorporate the unique *symbolic and semantic themes* of each set and are continuously updated to be distinctive from one another through a contrastive learning objective. Then, in every temporal context, new summaries for each set are derived by extracting the representative sentences in the set prioritized by their symbolic and semantic similarities to the corresponding set prototype. In the meantime, *accumulated knowledge distillation* regularizes the set prototypes to control the balance between the consistent relevance and the novelty of summaries over time.

In summary, the main contributions of this work are as follows:

- We introduce **a new summarization problem EMDS** (evolving multi-document sets stream summarization), which is more suitable for dynamic online scenarios than existing summarization tasks and thus is expected to bring huge benefits to users and relevant online applications.
- We propose **a novel method PDSum** designed for EMDS, exploiting lightweight set prototypes learned through contrastive learning with accumulated knowledge distillation. The source code is available at https://github.com/cliveyn/PDSum.
- In experiments with real benchmark datasets, PDSum shows **the state-of-the-art performance** compared with existing methods in the comprehensive evaluation with *relevance*, *novelty*, and *distinctiveness* measures specifically designed for EMDS.

Table 1: A comparison of the proposed summarization task (EMDS) with existing summarization tasks

	Multi-doc	Multi-set	Streaming docs	Evolving sets
SDS	×	×	×	×
MDS	○	×	×	×
QFS	○	×	×	×
TLS	○	×	×	×
RTS	○	×	○	×
EMDS	○	○	○	○

2 RELATED WORK

Text summarization has been actively studied in recent decades [2, 8, 28]. We briefly introduce existing relevant summarization tasks, compared with EMDS in Table 1, and the representative approaches.

2.1 Sigle/multi-document summarization

Single document summarization (SDS) and multi-document summarization (MDS) are the most popular summarization tasks, where the former assumes a single document as input while the latter assumes a set of documents of a certain interest (e.g., topic, query, or theme). Typical SDS and MDS methods can be classified as abstractive or extractive approaches, depending on how the summaries are derived, or as supervised or unsupervised approaches, depending on the use of reference (gold) summaries for model training.

Various approaches for SDS and MDS have been studied in the literature. Centroid-based methods [13, 35, 38] are one of the widely used approaches that cluster input documents and pick the most central sentences as a summary. Graph-based methods embed documents in a graph structure [9, 32, 57, 59]. A popular method LexRank [9] constructs a graph by connecting sentences based on their similarities and applies PageRank [34] to extract the most salience sentences. An unsupervised method SummPip [57] with graph clustering and compression techniques shows comparable performances with supervised methods. Recently, deep neural network (DNN)-based methods have been actively proposed [28], where deep reinforcement learning [29], semantic text matching [58], hierarchical transformer [26], or graph neural network [23, 45, 46] are used, to name a few. While most existing DNN-based methods adopt supervised training with reference summaries, Zhang et al. [55] proposed a self-supervised approach with the Gap Sentence Generation objective. PRIMERA [47] further improves the self-supervision by using the Entity Pyramid for masking sentences and provides state-of-the-art pretrained MDS models.

Nevertheless, the existing work for SDS and MDS inherently considers a *static* and *single* set of documents for summarization, which fall too short for continuous summarization of *streaming* documents from *evolving* sets in EMDS. Moreover, some supervised methods require reference summaries which are not readily available in an online scenario.

Another related line of work centers on query-focused summarization (QFS) [5, 42–44, 48]. QFS aims to summarize a fixed set of documents with respect to a user-specified query (e.g., questions, entities, or keywords). On the other hand, our EMDS task focuses on summarizing *multiple* document sets of different interests that are *evolving* over time. The summaries in our task should also take into account its distinctiveness to other sets of documents and its novelty to past documents.

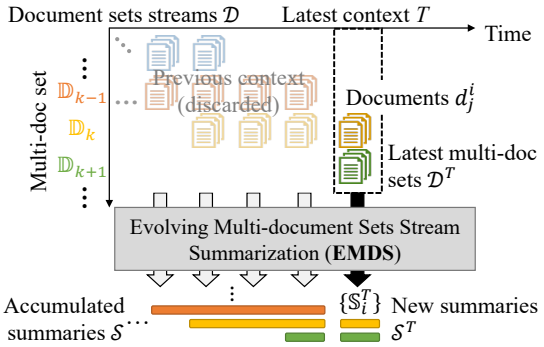


Figure 2: An illustration of the EMDS task.

2.2 Timeline/real-time summarization

Other relevant tasks are timeline summarization (TLS) and real-time summarization (RTS). TLS generates a timeline of events from a given set of documents on a particular topic, by typically conducting two individual subtasks: date selection and date summarization [15]. Graph-based [4, 22], time-event memory-based [6], or affinity propagation-based approaches [54] have been introduced for TLS. While most of them derive a single timeline from a single document set, a recent work [54] generates multiple timelines with different topics. While they are *retrospective* methods for deriving the complete timeline(s) from given documents, EMDS requires continuous updates of summaries from evolving document sets.

RTS [25] (and similar tasks such as temporal summarization [3], update summarization [31], stream summarization [12]) aims to summarize new documents from document streams considering their relevance, redundancy, and timeliness [28]. Some of the existing studies model the task as a sequential decision-making problem and apply deep reinforcement learning [40, 49]. Others represent documents in an information network and apply variants of PageRank [12]. However, their summarization goals are much simpler, predicting a *predefined* relevance labels (e.g., 'not relevant', 'relevant', or 'highly relevant') for documents or sentences stream [3, 12, 40, 49] retrieved with a *single* and *static* query [31, 40, 49]. Thus, they are not directly applicable to EMDS.

3 PROBLEM SETTING

Let a document $d = [s_1, s_2, \ldots, s_{|d|}]$ be a set of sentences s and a multi-document set (or simply *set*) $\mathbb{D} = [d_1, d_2, \ldots, d_{|\mathbb{D}|}]$ be a set of documents under a certain interest (e.g., topic, story, or theme). Then, Definition 3.1 formally introduces an *evolving multi-document sets stream* considered in this work.

Definition 3.1. (**Evolving multi-document sets stream**) An evolving multi-document sets stream $\mathcal{D} = \{\mathbb{D}_i\}$ is composed of unbounded multi-document sets $\mathbb{D}_i$ of documents d_j^i. The composition of sets and corresponding documents in $\mathcal{D}$ is continuously changing over time as new documents arrive in new (or existing) sets. A temporal context (or simply *context*) T indicates a certain temporal scope of interest in $\mathcal{D}$. Then, $\mathcal{D}^T = \{\mathbb{D}_i^T\}$ represents the sets and corresponding documents arrived at T.

A typical example of evolving multi-document sets stream is a stream of news stories curated by news applications. If a user subscribes to certain categories or topics, news articles (documents) in the relevant news stories (sets) are continuously delivered. Then, the user wants to continuously get new summaries for each ongoing story in every latest context (e.g., every day) of the evolving news stories stream. Definition 3.2 formalizes the summarization task proposed in this work and Figure 2 illustrates it.

Table 2: Notations frequently used in this paper.

Notation	Description
$\mathcal{D}$	an evolving multi-document sets stream
$s, d, \mathbb{D}$	a sentence, a document, a multi-document set,
$\mathbb{S}, \mathcal{S}$	a set summary, a collection of set summaries
$p, \mathbb{P}, \mathcal{P}$	a phrase, set phrases, a collection of set phrases
$R, \mathcal{R}$	a set prototype, a collection of set prototypes
T	a context for summarization
γ	a distillation ratio

Definition 3.2. (**Evolving Multi-Document sets stream Summarization (EMDS)**) From evolving multi-document sets stream $\mathcal{D}$, for $\mathcal{D}^T$ in every latest context T, a goal of EMDS is to derive new set summaries $\mathcal{S}^T = \{\mathbb{S}_i^T\}$ for each set $\mathbb{D}_i^T \in \mathcal{D}^T$, where $\mathcal{D}^T$ is discarded after being summarized.

Please note that EMDS naturally follows an *unsupervised* approach since obtaining reference summaries for multiple sets in different contexts is prohibitively expensive in a streaming setting. In this work, we adopt an *extractive* summarization approach and select representative sentences in each set as a summary. An abstractive approach may cause the hallucination problem [30], which could be more critical when summarizing dynamic and diverse themes of evolving multi-document sets stream. Table 2 summarizes the notations frequently used in this paper.

4 METHODOLOGY

4.1 Overview

4.1.1 ***Main Idea.*** A common goal of summarization is to identify a shared theme of documents and derive a concise and informative summary that best describes the theme. Additionally in EMDS, the theme and summary identification should keep up with the evolving contexts of multi-document sets streams; the theme incorporated in a new summary should be not only *relevant* to the target set but also be *novel* compared with previous summaries and *distinctive* from other sets. These goals also need to be achieved efficiently without accessing previous documents.

To this end, we build a lightweight data structure, called *set prototype*, to manage the lifelong theme of a set accumulated over time and use it for efficient and effective continuous summarization. Following the example scenario in Figure 1, we illustrate the idea of **prototype-driven continuous summarization** in Figure 3.
(1) First, we identify *the symbolic theme and the semantic theme of a set given concurrent sets*; the former is obtained by identifying top phrases included in documents in each set (i.e., set phrases), while the latter is obtained by representing documents in an embedding space. The two themes complement each other in clarifying unique themes of sets in the current context. For instance, at T_2 in Figure 3, the phrases '*Talas*', '*Japan*', and '*Power outage*' can collectively represent the symbolic theme of `Storm Talas`. Note that other phrases such as '*Die*', '*Kill*', or '*Approach*' may also be frequently

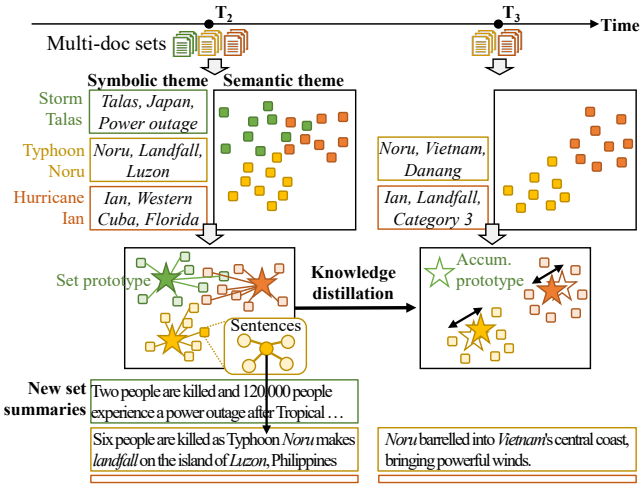


Figure 3: Prototype-driven continuous summarization.

found in the concurrent sets and thus do not represent the unique symbolic themes. These set phrases, however, are sparse and explicit features that can not fully reflect the implicit semantics of documents (e.g., describing victims of disasters). On the other hand, documents of a set represented in an embedding space (marked in rectangles) reflect the semantic themes of the set with dense and contextualized features. The document embeddings of different sets, however, might be overlapped if the documents are written with similar perspectives (e.g., describing evacuation processes for different disasters).

(2) Thus, we build the prototype of a set by *leveraging the two types of themes together*. Specifically, we consolidate them into set representations (marked in stars in Figure 3) by averaging document embeddings weighted by set phrases; the documents including more highly-ranked set phrases contribute more to representing the set prototype. Meanwhile, the embedding space is updated to promote documents being closer to their set prototype while being further from the other set prototypes. The embedding learning is regularized through knowledge distillation by balancing the accumulated set prototype (from previous documents) and the new set prototype (from new documents), as shown at T_3.

(3) Then, we identify set summaries as top sentences in each set *prioritized by their similarities to the set prototype*. The prioritization comprehensively considers the document-, sentence-, and phrase-level similarities. As shown in Figure 3, the chosen summary of `Typhoon Noru` at T_2 is relevant to the set but distinctive from the concurrent sets, and the new summary at T_3 conveys novel information while keeping its relevance and distinctiveness.

4.1.2 ***Overall Procedure of PDSum.*** We systematically implement the prototype-driven continuous summarization through PDSum, of which overall procedure is outlined in Algorithm 1 and illustrated in Figure 4. In every latest context T, the current sets of documents are fed into three sequential components to get new set summaries: (1) *Phrase Ranker* (Line 3) to identify set phrases, (2) *Prototype Encoder* (Lines 4–7 and the upper right part of Figure 4) to embed the documents and sets, and (3) *Summary Identifier* (Line 8 and the lower right part of Figure 4) to summarize the sets. Each component is explained in detail in the following subsections.

Algorithm 1: Overall Procedure of PDSum

Input: an evolving multi-document sets stream $\mathcal{D}$, a distillation ratio γ, the number e of epochs, the batch size b
Output: New summaries $\mathcal{S}^T$ in every context T

1 $\mathcal{R}, \mathcal{P} \leftarrow \emptyset$ `// Initialize set prototypes and set phrases.`
2 **for** *every multi-document sets* $\mathcal{D}^T$ *at the latest context* T *in* $\mathcal{D}$ **do**
`/* Identifying set phrases (Section 4.2) */`
3 $\mathcal{P}, \mathcal{P}^T \leftarrow \mathrm{PR}(\mathbb{D}_i^T \in \mathcal{D}^T)$
`/* Encoding documents and sets (Section 4.3) */`
4 **for** *each epoch in* e **do**
5 $\mathcal{R}, \mathcal{R}^T \leftarrow \mathrm{PE}(\mathbb{D}_i^T \in \mathcal{D}^T)$
6 $\mathcal{R}' \leftarrow \{\gamma R_i + (1-\gamma) R_i^T \,|\, R_i \in \mathcal{R}, R_i^T \in \mathcal{R}^T\}$
7 Update PE with $\sum_n \mathcal{L}_{\mathrm{RegCon}}(d \in \mathcal{D}^T, \mathcal{R}')$ for $|\mathcal{D}^T|/b$ itrs
`/* Summarizing sets (Section 4.4) */`
8 $\mathcal{S}^T \leftarrow \mathrm{SI}(\mathbb{D}_i^T \in \mathcal{D}^T, \gamma)$
9 Report $\mathcal{S}^T$;

4.2 Phrase Ranker (PR)

For each input set, a phrase ranker finds the set phrases representing its symbolic theme in the current context. The set phrases must be found more frequently and uniquely in the set than in any other concurrent set. While any existing phrase mining techniques [1, 17, 27] can be adopted, TFIDF [1] (default in PDSum) is a simple but effective choice as it considers term frequencies as well as inverse document frequencies (i.e., inverse set frequencies). By ranking the salience of phrases in a set, the top-N phrases are selected to form set phrases as follows.

Definition 4.1. **(Set Phrases)** Given sets $\mathcal{D}^T$ in T, set phrases $\mathbb{P}_i^T$ of a set $\mathbb{D}_i^T \in \mathcal{D}^T$ is obtained by a phrase ranker PR(·):

$$\mathrm{PR}(\mathbb{D}_i^T) = \mathbb{P}_i^T = \{(p_1^i, r_1^i), (p_2^i, r_2^i), \ldots, (p_N^i, r_N^i)\}, \text{where} \quad (1)$$

phrases p_k^i in $\mathbb{D}_i^T$ are ordered by their score r_k^i (e.g., TFIDF scores). Note that a collection $\mathcal{P}^T = \{\mathbb{P}_i^T\}$ of set phrases in T is continuously added to the accumulated set phrases $\mathcal{P}$ for further use.

4.3 Prototype Encoder (PE)

The second component, prototype encoder, conducts two sub-steps. First, it encodes the semantic theme of documents with a pretrained language model and combines them with the identified set phrases to derive set prototypes. Then, it updates the embedding space further to make each set prototype more distinctive from one another while being regularized by accumulated set prototypes.

4.3.1 ***Encoding Documents and Sets.*** PDSum first obtains initial sentence representations in each document by using a pretrained sentence encoder (e.g., sentence-BERT [36]). This sentence-level initialization with a pretrained model is more effective and efficient than learning from scratch since the pretrained model has a more generalized embedding capability learned from a much larger and diverse corpus. Furthermore, most sentences meet the maximum input length of typical language models (e.g., 512 tokens).

However, the initialized sentence representations do not reflect the mutual relationships between sentences inside a document since they are embedded independently. Thus, PDSum further enhances their *inter-sentence contexts* by fine-tuning them with a multi-head self-attention mechanism [41] as follows.

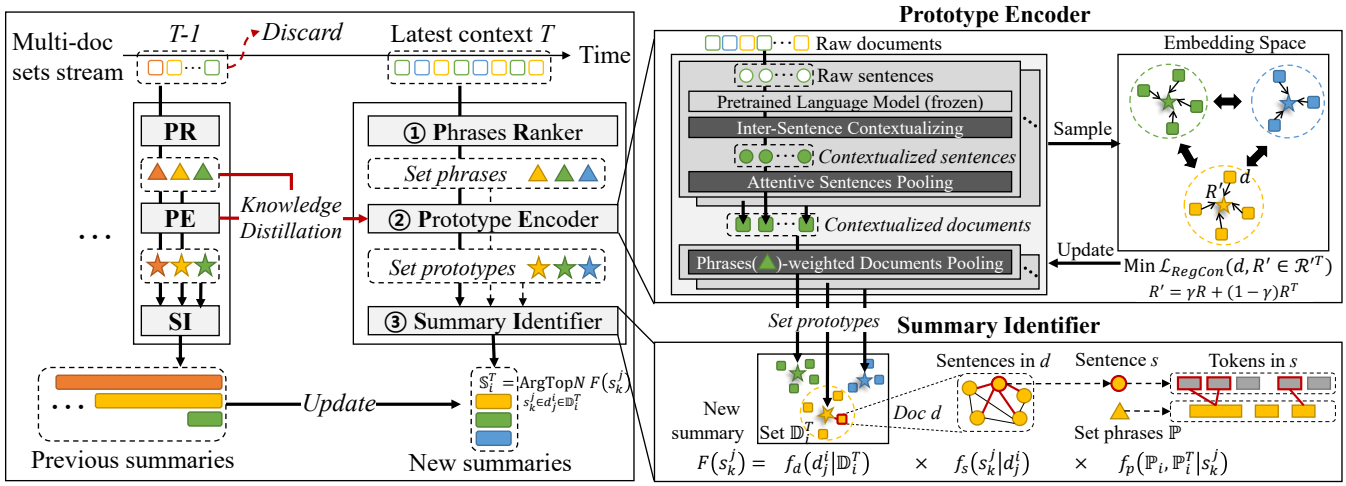


Figure 4: The overall procedure of PDSum.

Definition 4.2. (**Contextualized sentence**) Given a document d_j, the contextualized sentence representations in d_j are:

$$\begin{aligned} CS(d_j) &= [cs_1^j, \ldots, cs_{|d_j|}^j] \in \mathbb{R}^{|d_j| \times h_{cs}} \\ &= l_{\ln}(l_{\text{mhs}}([E(s_k)|s_k \in d_j]) + [E(s_k)|s_k \in d_j])), \end{aligned} \tag{2}$$

where $[E(s_k)|s_k \in d_j]$ is initial sentence representations by a pre-trained language model $\mathrm{E}(\cdot)$, $l_{\text{mhs}}(\cdot)$ is a multi-head self-attention layer, and $l_{\ln}(\cdot)$ is a feed forward layer with layer normalization .

Then, the contextualized sentence representations are pooled into a contextualized document representation through an *attentive sentences pooling*. This is to verify the relative contribution of each sentence in representing a document. Since the semantics of individual sentences are more diverse than the shared semantics of the document (and that of the set), this helps to filter out noisy sentences (e.g., too specific or too general descriptions) and naturally makes representative sentences stand out to represent the document. Definition 4.3 formalizes the pooling procedure.

Definition 4.3. (**Contextualized document**) Given a document d_j, a contextualized document representation cd_j is obtained through an attentive pooling of contextualized sentences:

$$\begin{aligned} CD(d_j) = cd_j &= \sum_{k=1\ldots|d_j|} \alpha_k cs_k^j \in \mathbb{R}^{h_{cd}} \\ &= \sum_{k=1\ldots|d_j|} \frac{\exp([l_\alpha(CS(d_j))]_k)}{\sum_{n=1\ldots|d_j|}\exp([l_\alpha(CS(d_j))]_n)} cs_k^j, \end{aligned} \tag{3}$$

where an attention weight α_k indicates the relative importance of s_k for representing cd_i, derived by an attention layer $l_\alpha(CS(d_j)) = \tanh(CS(d_j)W + b_W)V$ with learnable weights W, b_W, V.

Finally, PDSum derives a set prototype that represents all documents in the set both in the symbolic and semantic themes, through a *phrase-weighted documents pooling* to combine the contextualized document representations and the accumulated set phrases. Definition 4.4 formalizes the set prototype.

Definition 4.4. (**Set Prototype**) Given a set $\mathbb{D}_i^T$ in T, accumulated set phrases $\mathbb{P}_i$, and contextualized document representations $\{cd_j^i\}$, a set prototype R_i derived by a prototype encoder $PE(\cdot)$ is:

$$PE(\mathbb{D}_i^T) = R_i = \sum_{j=1}^{|\mathbb{D}_i^T|} \left(\frac{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i} |p_k^i \in d_j^i| r_k^i}{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i} |p_k^i \in \mathbb{D}_i^T| r_k^i} \cdot cd_j^i \right) \in \mathbb{R}^{h_{pe}}. \tag{4}$$

4.3.2 ***Optimizing Prototype Encoder.*** To optimize the prototype encoder, PDSum performs *accumulated knowledge distillation* to balance the previously accumulated knowledge and the currently identified new knowledge. Specifically, PDSum employs two types of set prototype for each set $\mathbb{D}_i^T \in \mathcal{D}^T$: an accumulated set prototype R_i and a new set prototype R_i^T, where R_i is obtained by Definition 4.4 while R_i^T is obtained similarly but with the *new set phrases* $\mathbb{P}_i^T$ in T and the *initial document representations* as the mean of initial sentence representations:

$$R_i^T = \sum_{j=1}^{|\mathbb{D}_i^T|} \left(\frac{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i^T} |p_k^i \in d_j^i| r_k^i}{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i^T} |p_k^i \in \mathbb{D}_i^T| r_k^i} \cdot \frac{\sum_{s_k^j \in d_j} E(s_k^j)}{|d_j^i|} \right) \in \mathbb{R}^{h_{pe}}, \tag{5}$$

In other words, the accumulated set prototype R_i reflects the lifelong theme of a set learned through the previous contexts when the set has existed, whereas the new set prototype R_i^T reflects only the new theme identified in the current context T.

Then, a knowledge-distilled set prototype is formulated by combining the two types of set prototypes as follows.

Definition 4.5. (**Knowledge-distilled Set Prototype**) Given an accumulated set prototype R_i, a new set prototype R_i^T, and a distillation ratio γ, a knowledge-distilled set prototype R_i' is:

$$R_i' = \gamma R_i + (1-\gamma) R_i^T, \tag{6}$$

where γ controls the distillation of the accumulated theme to the new theme; the higher γ weighs more on the previously learned theme while the lower γ weighs more on the newly identified theme.

Finally, PDSum promotes documents in each set to highlight their own shared theme; the knowledge-distilled set prototype becomes a positive target for the contextualized document representations in the set to be closer to, while those of the other concurrent sets

become negative targets to be further from. Definition 4.6 formalizes the regularized contrastive loss designed to achieve this goal.

Definition 4.6. (**Regularized Contrastive Loss**) Given sets $\mathcal{D}^T$, knowledge-distilled set prototypes $\mathcal{R}'^T$ in T, and a temperature scaling value τ, a regularized contrastive loss is calculated as:

$$\mathcal{L}_{\text{RegCon}}(d_j^i \in \mathcal{D}^T, \mathcal{R}'^T) = -\log \frac{\exp(\cos(cd_j^i, R_i'))/\tau}{\sum_{R_k' \in \mathcal{R}'^T} \exp(\cos(cd_j^i, R_k')/\tau)}. \quad (7)$$

Note that the regularized contrastive loss is *uniquely designed for EMDS*, different from typical contrastive losses [11, 19, 37]. Rather than optimizing pairwise distances between samples, PDSum makes the best of set prototypes to achieve a similar goal but in a more efficient and effective way; it significantly reduces the similarity computations (i.e., from $O(N_d^2)$ to $O(N_d N_D)$ for N_d articles and N_D sets where $N_d \gg N_D$) and also directly pursues the unique themes identification of sets (i.e., making articles distinctively similar to the set prototype) which is well aligned with the goal of summarization.

With the accordingly optimized prototype encoder, documents in a set are represented more *relevantly* within the set while retaining more *novelty* from earlier documents in the set and more *distinctiveness* from documents in the other sets.

4.4 Summary Identifier (SI)

Finally, a summary identifier picks the most representative sentences in each set as a new summary. For each sentence of a document in a set, three levels of the score are conjunctively estimated: (1) the semantic similarity between the set and the document (i.e., doc-level score), (2) the contribution of a sentence in representing the document (i.e., sentence-level score), and (3) the symbolic similarity between the set phrases and the sentence (i.e., phrase-level score). In the meanwhile, the distillation ratio γ controls the balance between the accumulated knowledge and the new knowledge for scoring. Definition 4.7 formalizes the summary identifier.

Definition 4.7. (**New Summary**) Given a set $\mathbb{D}_i^T$ in T, a summary identifier derives its new summary $\text{SI}(\mathbb{D}_i^T) = \mathbb{S}_i^T$ as the top sentences ranked by a sentence score function $F(s)$:

$$F(s_k^j \in d_j^i \in \mathbb{D}_i^T) = f_d(d_j^i|\mathbb{D}_i^T) \times f_s(s_k^j|d_j^i) \times f_p(\mathbb{P}_i, \mathbb{P}_i^T|s_k^j), \text{ where} \quad (8)$$

$$f_d(d_j^i|\mathbb{D}_i^T) = \gamma \exp(\cos(cd_j^i, R_i)) + (1-\gamma)\exp(\cos(cd_j^i, R_i^T))$$

$$f_s(s_k^j|d_j^i) = \alpha_k \text{ (i.e., an attention weight in Equation 3)}$$

$$f_p(\mathbb{P}_i, \mathbb{P}_i^T|s_k^j) = \gamma \frac{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i} |p_k^i \in s_k^j|r_k^i}{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i} |p_k^i \in d_j^i|r_k^i} + (1-\gamma) \frac{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i^T} |p_k^i \in s_k^j|r_k^i}{\sum_{(p_k^i, r_k^i) \in \mathbb{P}_i^T} |p_k^i \in d_j^i|r_k^i}.$$

The distillation ratio γ, utilized for both the embedding learning and the set summarizing, allows a user to have more freedom in choosing a specific degree of knowledge preservation over contexts (e.g., some users may prefer to get fresh information in summaries, while others may want to get more consistent information in summaries across the entire contexts.). In Section 5.3, we study the effects of the distillation ratio and demonstrate that the value of 0.5 balances the trade-off well and results in quality summaries more conforming to the reference summaries provided by humans.

4.5 Time Complexity of PDSum

Given N_d (# of documents), N_D (# of sets), N_P (# of set phrases), N_{PE} (parameter size in a prototype encoder), N_E (epoch size), and N_B (batch size), (1) the time complexity for a phrase ranker is $O(N_P N_d)$, (2) that for a prototype encoder is $O(N_d N_{PE} + N_P N_D + N_E N_B N_{PE})$ where $O(N_d N_{PE})$ for embedding documents, $O(N_P N_D)$ for encoding set prototypes, and $O(N_E N_B N_D N_{PE})$ for training, and (3) that for a summary identifier is $O(N_d N_D)$. Since $N_d, N_{PE} \gg N_D, N_P, N_E, N_B$, the total time complexity becomes $O(N_d N_{PE})$.

5 EXPERIMENTS

We conducted extensive experiments to evaluate the performance of PDSum, of which results are summarized as follows.

- PDSum achieved *state-of-the-art performances* in terms of relevance, novelty, and distinctiveness in EMDS on two benchmark datasets compared with existing algorithms (Section 5.2) and returned quality summaries (Appendix A.5).
- PDSum was *robust* to the distillation ratio, the number of set phrases, and training hyperparameters (Section 5.3).
- PDSum was more *efficient* than existing algorithms and *scalable* in various streaming settings (Section 5.4).

5.1 Experiment Setting

5.1.1 ***Multi-document Sets Stream.*** We used two real news datasets: WCEP [14] and W2E [18]. To the best of our knowledge, they are the only benchmark summarization datasets suitable for EMDS; they contain timestamped news articles (i.e., documents) of different stories (i.e., sets) over various temporal contexts (i.e., evolving sets) and provide reference summaries annotated by humans for evaluation. We simulated the datasets as multi-document sets stream $\mathcal{D}$ and set contexts T according to reference summaries. Refer to Appendix A.1 for more details.

5.1.2 ***Compared Algorithms.*** For the new task EMDS, we prepared strong baselines by adopting a centroid-based model [13, 38] with a pretrained language model: *DocCent* and *SentCent* with document- and sentence-based centers, respectively, and their incremental versions *IncDocCent*, and *IncSentCent*. We also compared three popular *unsupervised* algorithms for multi-document summarization: the graph-based model *Lexrank* [9], the state-of-the-art extractive model *Summpip* [57], and the state-of-the-art abstractive model *PRIMERA* [47]. We fed each document set in a context to them so that they can only infer the temporally correlated documents to update their set summaries. See Appendix A.2 for details.

5.1.3 ***Evaluation Metrics and Criteria.*** We used two popular metrics for evaluating summarization tasks: ROUGE scores [24] (denoted as *R1*, *R2*, and *RL*) for lexical matching and BERTScore [56] (denoted as *BS*) for semantic matching between output summaries and reference summaries. For a more comprehensive evaluation of EMDS, we derive three scores respectively on the two metrics to evaluate output summaries with the following three criteria (refer to Appendix A.3 for detailed formulation):

- Relevance between an output summary and a reference summary (i.e., denoted by *RL* and *BS* as default).
- Novelty of an output summary from the previous summary in the same set (i.e., *N-RL* and *N-BS*).

Table 3: Overall performance results. The highest and the second highest results are bolded and underlined, respectively.

	WCEP								W2E							
	Relevance				Novelty		Distinctiveness		Relevance				Novelty		Distinctiveness	
	R1	R2	RL	BS	N-RL	N-BS	D-RL	D-BS	R1	R2	RL	BS	N-RL	N-BS	D-RL	D-BS
DocCent	20.06	5.20	16.02	83.59	10.67	79.77	1.19	1.95	16.06	3.88	13.06	82.96	5.86	78.65	1.13	1.17
IncDocCent	20.52	5.40	16.34	83.73	10.51	79.11	1.19	1.97	14.65	2.97	11.39	82.73	5.16	77.89	1.11	1.14
SentCent	19.17	4.87	15.37	83.38	10.30	79.35	1.18	1.92	14.93	3.41	12.15	82.78	5.91	**79.24**	1.12	1.16
IncSentCent	19.64	4.99	15.64	83.46	10.53	79.24	1.19	1.93	14.94	3.39	12.19	82.81	5.76	79.19	1.12	1.15
Lexrank	13.63	3.33	11.28	82.01	9.11	80.58	1.13	1.79	11.40	2.93	9.55	81.24	5.12	77.27	1.09	1.12
Summpip	21.72	7.04	17.28	83.60	11.74	81.21	1.21	1.95	16.12	4.83	12.76	82.68	5.77	78.39	1.12	1.18
PRIMERA	20.87	6.84	18.45	83.84	11.65	80.70	1.23	1.99	16.28	5.69	14.24	83.00	3.67	75.61	1.11	1.13
PDSum	**26.62**	**9.20**	**21.01**	**84.51**	**12.67**	81.27	**1.27**	**2.08**	**19.61**	**6.48**	**15.75**	**83.26**	6.60	76.35	**1.17**	**1.28**
	±0.05	±0.04	±0.05	±0.01	±0.03	±0.04	±0.00	±0.00	±0.10	±0.06	±0.09	±0.02	±0.03	±0.09	±0.00	±0.00
w/o doc-score	26.44	9.13	20.85	84.47	12.64	**81.40**	1.26	**2.08**	19.07	6.23	15.31	83.20	6.56	76.60	**1.17**	1.27
	±0.07	±0.04	±0.06	±0.01	±0.05	±0.04	±0.00	±0.00	±0.01	±0.05	±0.08	±0.02	±0.03	±0.08	±0.00	±0.00
w/o sent-score	22.75	7.42	17.81	83.60	11.27	80.47	1.22	1.95	14.22	4.02	11.38	81.26	**7.37**	74.27	1.11	1.21
	±0.01	±0.01	±0.01	±0.00	±0.01	±0.03	±0.00	±0.00	±0.02	±0.00	±0.01	±0.00	±0.00	±0.07	±0.00	±0.00
w/o phrase-score	20.99	6.33	17.06	83.59	11.2	80.71	1.21	1.96	13.59	3.94	10.91	81.86	5.32	76.16	1.10	1.13
	±0.09	±0.05	±0.06	±0.02	±0.05	±0.08	±0.00	±0.00	±0.19	±0.11	±0.17	±0.04	±0.04	±0.33	±0.00	±0.00

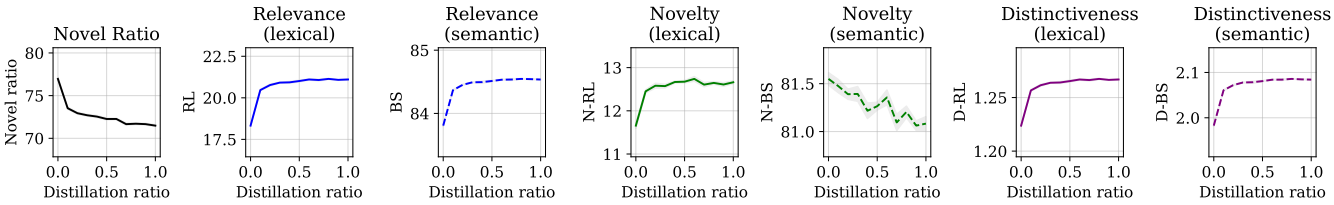


Figure 5: Effects of knowledge distillation ratio in WCEP (the result of W2E is provided in Appendix A.4).

- Distinctiveness between an output summary and the reference summaries of the other concurrent sets (i.e., *D-RL* and *D-BS*).

For each score, the result averaged over stories and contexts is reported to show the overall performance over the entire stream.

5.2 Overall Performance Results

Table 3 shows the overall evaluation results. For brevity, we show the R1 and R2 scores only for relevance, while the others show similar trends. The main observations are summarized as follows:

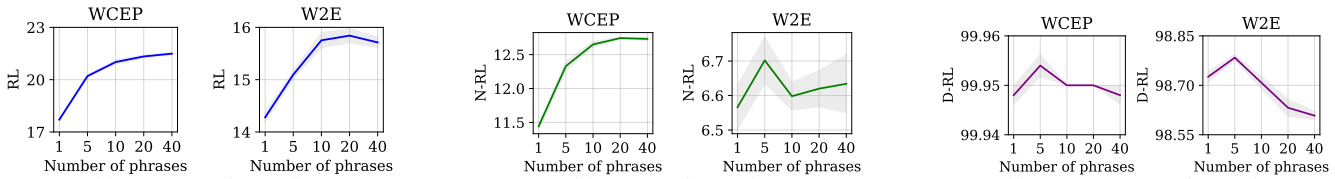
- Comparison of existing algorithms: Among centroid-based variants, IncDocCent achieved the highest scores, indicating the efficacy of document-level and incremental accumulation of knowledge in EMDS. The existing algorithms outperformed them by considering more comprehensive aspects of each set with graph embedding (Summpip) or pretrained Entity Pyramid (PRIMERA). However, since they do not consider previous and concurrent documents conjunctively, their summaries are biased toward the current context and/or the documents in the same set.
- PDSum v.s. existing algorithms: PDSum outperformed existing algorithms in most cases, by achieving significantly higher relevance scores (△30.9%), much higher novelty scores (△22.0%), and moderately higher distinctiveness scores (△7.2%) when averaged over all cases. This indicates the efficacy of set prototypes in identifying distinctive knowledge in concurrent sets and preserving it over continuous sets stream (note that on average 15.49 (std: 7.56) sets existed in each context in the datasets).
- Ablation study for PDSum: Among the three levels of scores to prioritize sentences for summarization, the phrase-level score contributed the most as it directly affects the symbolic theme of a set. On the other hand, the document-level score contributed marginally because the prototype encoder makes the representations of documents in the same set converge around the set prototype. Nevertheless, as not all sentences in the document are equally important to represent the set theme, the sentence-level score contributed more to the overall performance.

Appendix A.5 also discusses human evaluation results and a qualitative case study with sample news stories.

5.3 Sensitivity Analysis

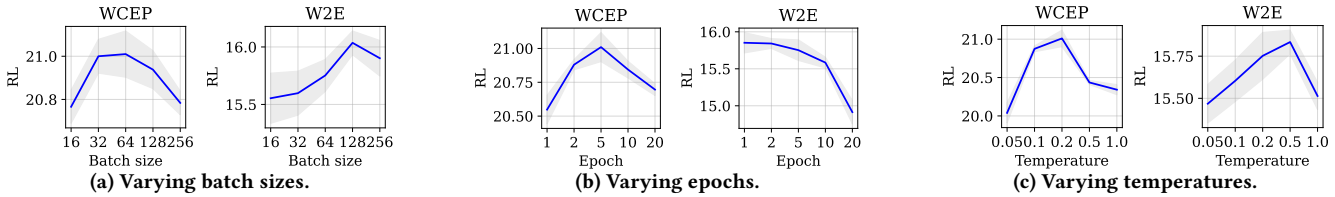
5.3.1 ***Knowledge Distillation Ratio.*** Figure 5 shows the effects of distillation ratio in WCEP (refer to Appendix A.4 for more discussion with the results in W2E). A novel ratio (i.e., the ratio of novel tokens in a new summary) decreases as a distillation ratio increases (i.e., PDSum weighs more on the accumulated knowledge in previous contexts). A high novel ratio also leads to higher relevance scores to a reference summary since it could preserve the lifelong theme of a set. Interestingly, the lexical novelty score is positively correlated to the distillation ratio, while the semantic novelty score is not. This is because the novel tokens in a new summary become more conforming to the tokens in the reference summary with more preserved knowledge, but their collective semantics become less similar to the semantics of the reference summary with fewer tokens. Both of the scores for distinctiveness, however, are positively correlated with the distillation ratio which again shows preserving previous knowledge helps the distinctiveness of new summaries. Regardless of the trade-offs in various aspects of the output summaries discussed, the distillation ratio of 0.5 smoothly balanced the trade-offs and lead to the quality results in both datasets.

5.3.2 ***Number of Set Phrases.*** Figure 6 shows the effects of the number of set phrases. We show the results of RL for brevity, while those of other metrics showed similar trends. Overall, considering more set phrases helps get more relevant and novel summaries but less distinct summaries in both datasets, which is expected as more phrases may help specify the theme of each set but also can overlap over concurrent sets. It is empirically observed that a moderate number of phrases, around five to ten, is the optimal value balancing the trade-off, which is also consistent with real-world



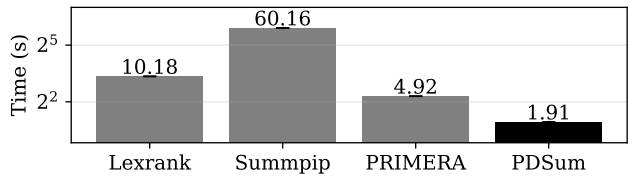
(a) Relevance (lexical). (b) Novelty (lexical). (c) Distinctiveness (lexical).

Figure 6: Effects of the number of set phrases.

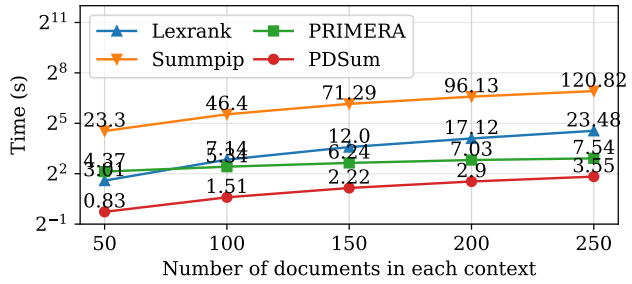


(a) Varying batch sizes. (b) Varying epochs. (c) Varying temperatures.

Figure 7: Sensitivity analysis on training hyperparameters.



(a) Average running time in each context.



(b) Average running time by varying the number of documents.

Figure 8: Scalability analysis results in WCEP.

practices (e.g., it is common to use around five keywords to describe a certain theme in news articles, scientific papers, or products).

5.3.3 ***Training Hyperparameters.*** We studied the effects of main hyperparameters used for training PDSum: the batch size, the number of epochs, and the temperature value. Due to space limitation, in Figure 7, we show the results of relevance (i.e., RL, where BS showed similar trends) which is the most important criterion in the summarization task. For both datasets, each hyperparameter has an optimal point for the highest scores, which conforms to the default value used in PDSum. However, overall, PDSum consistently achieved good performances as even its lowest scores in extreme settings were higher than the best score of existing methods (e.g., the best RL score by PRIMERA was 18.45 in WCEP and 14.24 in W2E as reported in Table 3). This again demonstrates the merits and robustness of PDSum over various settings in EMDS.

5.4 Scalability Analysis

Besides the summarization quality, the algorithms for EMDS should be efficient and scalable to deal with dynamic streaming environments. We measured the average running time of compared algorithms in summarizing sets in each context. As shown in Figure 8 for WCEP (the results for W2E showed similar trends), PDSum was the fastest among the compared algorithms by taking only a few seconds to summarize hundreds of documents. This is attributed to lightweight prototype-based sets processing while the existing algorithms not only process each set independently but also use expensive graph-based processing (Lexrank and Summpip) or a large language model (PRIMERA). Furthermore, when we varied the input rate of an evolving multi-document sets stream by controlling the number of documents in each context, PDSum consistently achieved the lowest running time with a comparable increase rate. The trend of scalability also conforms to the time complexity of PDSum which is linear to the number of documents.

6 DISCUSSION AND CONCLUSION

Before concluding, we discuss two interesting directions to facilitate future work for EMDS. First, an abstractive summarization approach can be alternatively considered. For instance, in PDSum, the summary identifier can incorporate a decoding module that considers a set prototype as a key signal to decode the output summary from documents. However, to prevent the hallucination problem, the factual consistency over different contexts in the same set needs to be specifically considered in accumulating previous knowledge and generating summaries. Second, while an unsupervised approach is practical in EMDS, some information can be provided as delayed feedback such as reference summaries by annotators or user ratings on new summaries. Then, they can be used to refine the embedding and summarizing processes as previous auxiliary knowledge (e.g., to additionally regularize set prototypes or sentence score functions in PDSum).

In conclusion, we introduced a new summarization task EMDS for continuously summarizing evolving multi-document sets stream. We proposed a novel unsupervised method PDSum for EMDS, that builds *lightweight prototypes of multi-document sets* used for embedding and summarizing. The relevance, novelty, and distinctiveness of summaries are achieved by continuously updating set prototypes over contexts through a contrastive learning objective, while being regularized by accumulated knowledge distillation. We demonstrated the superiority of PDSum over existing state-of-the-art unsupervised summarization algorithms in benchmark datasets. We believe this work opens a promising direction for summarization.

ACKNOWLEDGMENTS

The first author was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A3A14043765). The second author was supported by the Science and Technology Development Fund, Macau SAR (Grant No. 060/2017/AFJ and 070/2022/AMJ). The research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management* 39, 1 (2003), 45–65.

[2] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268* (2017).

[3] James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. 10–18.

[4] Jeffery Ansah, Lin Liu, Wei Kang, Selasie Kwashie, Jixue Li, and Jiuyong Li. 2019. A graph is worth a thousand words: Telling event stories using timeline summarization graphs. In *The World Wide Web Conference*. 2565–2571.

[5] Hou Pong Chan, Lu Wang, and Irwin King. 2021. Controllable summarization with constrained markov decision process. *Transactions of the Association for Computational Linguistics* 9 (2021), 1213–1232.

[6] Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan. 2019. Learning towards Abstractive Timeline Summarization.. In *IJCAI*. 4939–4945.

[7] Gianpaolo Cugola and Alessandro Margara. 2012. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)* 44, 3 (2012), 1–62.

[8] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165 (2021), 113679.

[9] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479.

[10] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021), 391–409.

[11] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6894–6910.

[12] Tao Ge, Lei Cui, Baobao Chang, Sujian Li, Ming Zhou, and Zhifang Sui. 2016. News stream summarization using burst information networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 784–794.

[13] Demian Gholipour Ghalandari. 2017. Revisiting the Centroid-based Method: A Strong Baseline for Multi-Document Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. 85–90.

[14] Demian Gholipour Ghalandari, Chris Hokamp, John Glover, Georgiana Ifrim, et al. 2020. A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1302–1308.

[15] Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the State-of-the-Art in News Timeline Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1322–1334.

[16] Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, You Wu, Cong Yu, Daniel Finnie, Hongkun Yu, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating representative headlines for news stories. In *Proceedings of The Web Conference 2020*. 1773–1784.

[17] Xiaotao Gu, Zihan Wang, Zhenyu Bi, Yu Meng, Liyuan Liu, Jiawei Han, and Jingbo Shang. 2021. UCPhrase: Unsupervised Context-aware Quality Phrase Tagging. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 478–486.

[18] Tuan-Anh Hoang, Khoi Duy Vo, and Wolfgang Nejdl. 2018. W2E: a worldwide-event benchmark dataset for topic detection and tracking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1847–1850.

[19] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access* 8 (2020), 193907–193934.

[20] Dongha Lee, Jiaming Shen, SeongKu Kang, Susik Yoon, Jiawei Han, and Hwanjo Yu. 2022. TaxoCom: Topic Taxonomy Completion with Hierarchical Discovery of Novel Topic Clusters. In *Proceedings of the ACM Web Conference 2022*. 2819–2829.

[21] Dongha Lee, Jiaming Shen, Seonghyeon Lee, Susik Yoon, Hwanjo Yu, and Jiawei Han. 2022. TopicExpan: Topic Taxonomy Expansion via Hierarchy-Aware Topic Phrase Generation. (2022).

[22] Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown. 2021. Timeline summarization based on event graph compression via time-aware optimal transport. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6443–6456.

[23] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6232–6243.

[24] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[25] Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen M Voorhees, and Fernando Diaz. 2016. Overview of the TREC 2016 Real-Time Summarization Track.. In *TREC*.

[26] Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 5070–5081. https://doi.org/10.18653/v1/p19-1500

[27] Yuanhua Lv and ChengXiang Zhai. 2011. When Documents are Very Long, BM25 Fails!. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[28] Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z Sheng. 2020. Multi-document summarization via deep learning techniques: A survey. *ACM Computing Surveys (CSUR)* (2020).

[29] Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document Summarization with Maximal Marginal Relevance-guided Reinforcement Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1737–1751.

[30] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1906–1919.

[31] Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. Incremental update summarization: Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 301–310.

[32] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[34] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[35] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938.

[36] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese Bert-Networks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[37] Nils Rethmeier and Isabelle Augenstein. 2021. A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned & Perspectives. *ACM Computing Surveys (CSUR)* (2021).

[38] Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*. 12–21.

[39] Yooju Shin, Susik Yoon, Patara Trirat, and Jae-Gil Lee. 2019. CEP-Wizard: Automatic Deployment of Distributed Complex Event Processing. In *2019 IEEE International Conference on Data Engineering (ICDE)*. IEEE, 2004–2007.

[40] Haihui Tan, Ziyu Lu, and Wenjie Li. 2017. Neural network based reinforcement learning for real-time pushing on text stream. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 913–916.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[42] Jesse Vig, Alexander Richard Fabbri, Wojciech Kryściński, Chien-Sheng Wu, and Wenhao Liu. 2022. Exploring Neural Models for Query-Focused Summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 1455–1468.

[43] Xiaojun Wan. 2009. Topic analysis for topic-focused multi-document summarization. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 1609–1612.

[44] Xiaojun Wan and Jianmin Zhang. 2014. CTSUM: extracting more certain summaries for news articles. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 787–796.

[45] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing Huang. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6209–6219.

[46] Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting Abstractive Summarization with Unified Semantic Graph. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 6052–6067.

[47] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5245–5263.

[48] Yumo Xu and Mirella Lapata. 2021. Generating Query Focused Summaries from Query-Free Resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 6096–6109.

[49] Min Yang, Chengming Li, Fei Sun, Zhou Zhao, Ying Shen, and Chenglin Wu. 2020. Be relevant, non-redundant, and timely: Deep reinforcement learning for real-time event summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9410–9417.

[50] Susik Yoon, Jae-Gil Lee, and Byung Suk Lee. 2019. NETS: extremely fast outlier detection from a data stream via set-based processing. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1303–1315.

[51] Susik Yoon, Jae-Gil Lee, and Byung Suk Lee. 2020. Ultrafast local outlier detection from a data stream with stationary region skipping. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1181–1191.

[52] Susik Yoon, Youngjun Lee, Jae-Gil Lee, and Byung Suk Lee. 2022. Adaptive Model Pooling for Online Deep Anomaly Detection from a Complex Evolving Data Stream. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2347–2357.

[53] Susik Yoon, Yooju Shin, Jae-Gil Lee, and Byung Suk Lee. 2021. Multiple dynamic outlier-detection from a data stream by exploiting duality of data and queries. In *Proceedings of the 2021 International Conference on Management of Data*. 2063–2075.

[54] Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 2021. Multi-timeline summarization (MLTS): Improving timeline summarization by generating multiple summaries. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 377–387.

[55] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*. PMLR, 11328–11339.

[56] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[57] Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. 2020. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval*. 1949–1952.

[58] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6197–6208.

[59] Qi Zhu, Fang Guo, Jingjing Tian, Yuning Mao, and Jiawei Han. 2021. SUMDocS: Surrounding-aware Unsupervised Multi-Document Summarization. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 477–485.

A SUPPLEMENTAL MATERIAL

A.1 Details of Datasets

Table 4: Statistics of the datasets.

Dataset	Period	#Docs	#Sets	#RefSummaries
WCEP [14]	Jan~Dec, 2019	29,931	519	519
W2E [18]	Jan~Dec, 2016	14,475	47	248

The statistics of two datasets are summarized in Table 4.

- WCEP [14] is a large-scale benchmark data set with reference summaries. We chose news stories with at least 50 news articles over their whole lifespan and used the news articles of the stories published in 2019. We simulated an evolving multi-document set stream by feeding articles ordered by their publication date, thus each day becomes a context for summarization. As each story has a single reference summary, it is used for evaluation in each context of the story.
- W2E [18] is another large-scale benchmark data set. As it provides multiple reference summaries spanned over the lifespan of news stories, for each story, we regarded the temporal gaps between consecutive reference summaries as the contexts (i.e., to compare with a reference summary at t, a new summary is generated from the articles published from $t-1$ to t). We used news stories including at least 10 articles in each context. Similar to WCEP, we simulated an evolving multi-document set stream by feeding articles ordered by their publication date. We conducted summarization in each unique context in the stream, where the overlapped temporal period among the contexts naturally leads to concurrent articles and sets in the context.

A.2 Implementation of Compared Algorithms

- *Variants of centroid-based model* [13, 35, 38]: Instead of using symbolic [1] or shallow [33] approach for embedding sentences and documents, we used a deep contextualized embedding with a pretrained language model (sentence-BERT [36]) which shows the state-of-the-art performances in semantic tasks.
  - *SentCent* and *DocCent*: For each context, the center of each set is calculated by the mean of sentence (or document) representations and the sentences closest to the center are returned.
  - *IncSentCent* and *IncDocCent*: Similar to the above, but the centers are incrementally updated from the previous contexts.
- *Lexrank* [9]: A widely used unsupervised graph-based summarization algorithm that uses centrality scores based on PageRank [34] for ranking sentences. We used the default settings following the original work.
- *Summpip* [57]: A state-of-the-art unsupervised extractive summarization algorithm with graph clustering and compression. We used sentence-BERT [36] for embedding sentences and the default settings following the original work.
- *PRIMERA* [47]: A state-of-the-art abstractive summarization algorithm with self-supervision by the Gap Sentence Generation and the Entity Pyramid. We used a pretrained model provided by the authors trained with a large-scale news data set (Newshead [16]) without reference summaries (i.e., unsupervised).
- ***PDSum*** (proposed): We used TFIDF [1] for a phrase ranker and sentence-BERT [36] for initializing sentence representations. We set the number of heads for MHS to 2, the hidden dimensionalities to 1024, the learning rate to 1e-5, the batch size to 64, the number of epochs to 5, the temperature value to 0.2, and the distillation ratio to 0.5. We reported the mean and standard error of the scores measured from ten independent experiments (the other algorithms, including a pretrained PRIMERA, are deterministic).

For sentence-BERT [36], we used a popular pretrained model *all-roberta-large-v1*. For existing algorithms, the documents in each set in each context are used as the input for summarization. For all algorithms, we set the size of an output summary to one sentence (for extractive approaches) or 40 tokens (for abstractive approaches), following the average size of reference summaries in the datasets used. All experiments are conducted with a 2.5GHz 32-Core CPU with 1TB RAM and an RTX A6000 GPU with 48GB RAM.

A.3 Details of Evaluation Criteria

Given an output summary $\mathbb{S}$ and a reference summary $\mathbb{G}$, let $\mathcal{F}(\mathbb{S}, \mathbb{G})$ be an evaluation metric (e.g., ROUGE [24] or BERTScore [56] in our evaluation). Then, we derive the three measures with $\mathcal{F}(\cdot,\cdot)$ for evaluating EMDS by different criteria as follows.

Given an output summary $\mathbb{S}_i^T$ for a set i in a context T and the corresponding reference summary $\mathbb{G}_i^T$,

- ***Relevance*** measures the default score by $\mathcal{F}$:
$$\text{Score}_{relevance} = \mathcal{F}(\mathbb{S}_i^T, \mathbb{G}_i^T) \tag{9}$$
- ***Novelty*** measures how relevant the novel part of a new summary from the previous summary is to the reference summary:
$$\text{Score}_{novelty} = \mathcal{F}(\mathbb{S}_i^T \setminus \mathbb{S}_i^{T-1}, \mathbb{G}_i^T), \tag{10}$$
where $\mathbb{S} \setminus \mathbb{S}'$ denotes the token-level differences.
- ***Distinctiveness*** measures how distinctive a new summary is from other reference summaries. Specifically, the distinctiveness of a summary from other reference summaries is calculated by $1 - \mathcal{F}(\mathbb{S}, \mathbb{G})$ and normalized by that from its reference summary (i.e., the higher the distinctiveness score, the less similar an output summary is to other reference summaries):
$$\text{Score}_{distinctiveness} = \frac{\frac{1}{|\mathcal{G}^T|-1} \sum_{\mathbb{G}_j^T \in \mathcal{G}^T, j \neq i} (1 - \mathcal{F}(\mathbb{S}_i^T, \mathbb{G}_j^T))}{1 - \mathcal{F}(\mathbb{S}_i^T, \mathbb{G}_i^T)}, \tag{11}$$
where $\mathcal{G}^T$ is a collection of reference summaries in all sets in T.

A.4 Knowledge Distillation in W2E

Following the effects of knowledge distillation ratio in WCEP discussed in Section 5.3, Figure 9 shows the results in W2E. Unlike the trends observed in WCEP, as the distillation ratio increases, the lexical score in each criterion is consistent or marginally decreases while the semantic score is also consistent or marginally increases (except for novelty). This is because in W2E output summaries are evaluated with individual reference summaries in each context, which may have different thematic keywords (i.e., thus not necessarily "lexically" relevant) but should have some degree of similar semantics (i.e., as they are about the same set). In the case of the semantic novelty score, however, considering the previous knowledge more should make the overall semantics of a new summary less similar to that of a reference summary, as conforming to the observation in WCEP. Overall, the default distillation ratio of 0.5 balances the trade-off well, showing competitive scores in most cases in W2E, as well as in WCEP.

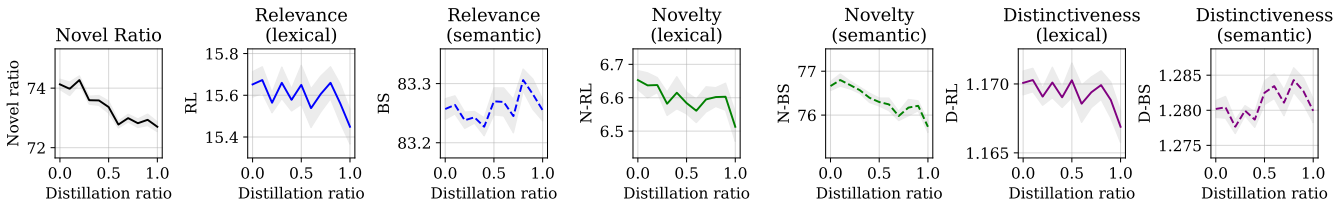


Figure 9: Effects of knowledge distillation ratio in W2E.

	Set A (US-Venezuela Talk)	Set B (US National Emergency)	Set C (US Sanction on Iran)
Reference	A senior United States administration official says the U.S. is willing to meet with Venezuelan President Nicolás Maduro to negotiate an exit to the crisis, as the U.S. military helps deliver aid to Venezuela.	U.S. President Donald Trump declares a National Emergency Concerning the Southern Border of the United States.	U.S. Vice President Mike Pence accuses European Union members of trying to break U.S. sanctions against Iran, and calls on the EU to withdraw from the Joint Comprehensive Plan of Action.
Summpip	U.S. president Nicolas Maduro said on Friday.	The deal to avert a government shutdown.	European allies of trying to undermine U.S. sanctions against Iran.
PRIMERA	A month into Venezuela's high-stakes political crisis, President Nicolas Maduro revealed in an AP interview that his government was in secret talks with the Trump administration.	Congress lopsidedly approved a border security compromise Thursday that would avert a second painful government shutdown, but a new confrontation was ignited — this time over President Donald Trump's plan.	U.S. Vice President Mike Pence visited the memorial site of Auschwitz on Friday along with the Polish president. It was the first visit for Pence, a conservative Christian.
PDSum	**As the US continues to work to bolster the administration of self-declared interim Venezuelan President Juan Guaido, a senior administration official says the US would be willing to meet with Nicolas Maduro to negotiate his exit.**	**President Donald Trump will sign a border security bill to avert another government shutdown, but also declare a national emergency to obtain funds for his promised U.S.-Mexico border wall, the top Senate Republican said on Thursday.**	**US Vice President Mike Pence accused Washington's European allies on Thursday of trying to break US sanctions against Tehran and called on them to withdraw from the Iran nuclear deal.**

(a) Summaries of *three different sets* of articles about `the United States` published on the same date in WCEP.

	4/7/2016	4/23/2016	4/25/2016
Reference	Attackers in Bangladesh wielding machetes kill Nazimuddin Samad, a liberal blogger.	Professor Rezaul Karim Siddique is killed in the Bangladeshi city of Rajshahi in an ISIL attack.	Xulhaz Mannan, a Bangladeshi employee of a U.S. charity (USAID), who is also an editor for the country's only LGBT magazine, and a friend are hacked to death in Dhaka, Bangladesh, by suspected Islamist militants. Guards at the building were also injured.
Summpip	The killing on wednesday, police said thursday.	Bangladesh professor was hacked to death a university .	Advertisement U.S. agency for international development, police said.
PRIMERA	Attackers in Bangladesh wielding machetes killed a liberal blogger, police said on Thursday, the latest in series of murders of secular activists by suspected Islamist militants.	In Bangladesh, men attacked and hacked to death liberal blogger Nazimuddin Samad, 26, with machetes on Monday night.Samad left his law classes at Jagganath University	Unidentified assailants fatally stabbed two men in Bangladesh's capital Monday night, including a gay rights activist who also worked for the U.S. Agency for International Development, police said.
PDSum	**Men armed with machetes killed a secular activist at a crowded intersection in Dhaka, the Bangladeshi capital, a police official said on Thursday, the latest in a series of grisly attacks on intellectuals and bloggers who have written critically about militant Islam on social media.**	**A professor of English was hacked to death and nearly beheaded near his home in northwestern Bangladesh on Saturday, in what the police said they suspected was the latest in a series of targeted killings by Islamist militants.**	**Top News Gay rights activist killed in Bangladesh Posted Islamist militants are suspected of hacking to death a leading gay rights activist and a friend at an apartment in the Bangladeshi capital of Dhaka.**

(b) Summaries *over three different days* on a set about `Attacks on secularists in Bangladesh` in W2E.

Figure 10: Example summaries of compared algorithms, where abstract, irrelevant, or redundant ones are highlighted in colors.

Table 5: Human evaluation results.

	Relevance		Novelty		Distinctiveness	
	WCEP	W2E	WCEP	W2E	WCEP	W2E
Summpip	1.56	1.08	1.80	1.80	1.80	1.73
PRIMERA	2.44	2.16	1.88	2.76	3.40	3.07
PDSum	**4.08**	**3.68**	**3.40**	**3.64**	**3.60**	**3.93**

A.5 Qualitative Analysis

We conducted a qualitative analysis of PDSum in comparison with the two existing algorithms based on extractive (Summpip) and abstractive (PRIMERA) approaches, respectively. For human evaluation, we recruited five graduates to evaluate the compared algorithms' summaries for 25 news stories of different categories (e.g., Politics, Sports, etc.) in the two datasets. The raw articles and reference summaries were provided together. Following the widely used protocol [10], we asked them to rate summaries on a Likert scale from 1 to 5 (i.e., from very poor to excellent) for relevance, novelty, and distinctiveness. As shown in Table 5, PDSum achieved higher scores than the existing algorithms by a large margin.

We further analyzed the quality of summaries by each algorithm with two case studies. Figure 10 shows the case studies on example summaries generated by PDSum and the two existing algorithms. Figure 10a shows the case study in WCEP, where the summaries of the three *different sets in the same context* with similar themes about `the United States` are provided. The existing algorithms were not effective in returning relevant and distinctive summaries for different sets, by including too abstract or irrelevant summaries. On the other hand, the summaries returned by PDSum were distinctive from each other as well as relevant to the reference summaries. Figure 10b shows the case study in W2E where summaries of the *same set in different contexts* are provided. Similarly, on each day, PDSum could get the consistently relevant summaries to the set about `Attacks on secularists in Bangladesh`, while being specific to each accident that happened each day. The two existing algorithms, however, returned too abstract or irrelevant summaries (Summpip), without capturing the relevance and distinctiveness effectively, or a redundant summary (PRIMERA) without capturing the novelty effectively.