Natural Language Engineering (2024), **30**, pp. 525–553
doi:10.1017/S1351324923000177

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Joint learning of text alignment and abstractive summarization for long documents via unbalanced optimal transport

Xin Shen[1], Wai Lam[1], Shumin Ma[2] and Huadong Wang[3]

[1]Department of System Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China, [2]Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, BNU-HKBU United International College, Zhuhai, China, and [3]Department of Computer Science and Technology, Tsinghua University, Beijing, China
Corresponding author: X. Shen; Email: xshen@se.cuhk.edu.hk

(Received 3 January 2022; revised 15 March 2023; accepted 17 April 2023; first published online 15 May 2023)

Abstract

Recently, neural abstractive text summarization (NATS) models based on sequence-to-sequence architecture have drawn a lot of attention. Real-world texts that need to be summarized range from short news with dozens of words to long reports with thousands of words. However, most existing NATS models are not good at summarizing long documents, due to the inherent limitations of their underlying neural architectures. In this paper, we focus on the task of long document summarization (LDS). Based on the inherent section structures of source documents, we divide an abstractive LDS problem into several smaller-sized problems. In this circumstance, how to provide a less-biased target summary as the supervision for each section is vital for the model's performance. As a preliminary, we formally describe the section-to-summary-sentence (S2SS) alignment for LDS. Based on this, we propose a novel NATS framework for the LDS task. Our framework is built based on the theory of unbalanced optimal transport (UOT), and it is named as UOTSumm. It jointly learns three targets in a unified training objective, including the optimal S2SS alignment, a section-level NATS summarizer, and the number of aligned summary sentences for each section. In this way, UOTSumm directly learns the text alignment from summarization data, without resorting to any biased tool such as ROUGE. UOTSumm can be easily adapted to most existing NATS models. And we implement two versions of UOTSumm, with and without the pretrain-finetune technique. We evaluate UOTSumm on three publicly available LDS benchmarks: *PubMed*, *arXiv*, and *GovReport*. UOTSumm obviously outperforms its counterparts that use ROUGE for the text alignment. When combined with UOTSumm, the performance of two vanilla NATS models improves by a large margin. Besides, UOTSumm achieves better or comparable performance when compared with some recent strong baselines.

Keywords: Abstractive text summarization; Text alignment; Optimal transport; Long document summarization

1. Introduction

Text summarization is the procedure of identifying the most important information from source text and producing a concise and readable summary (Mani, 1999). Generally speaking, the summarization models can be divided into two types: extraction or abstraction (Gambhir and Gupta, 2017). The extractive approach directly extracts snippets, such as sentences or phrases from the original documents as the summary. In contrast, the abstractive approach uses natural language generation techniques to produce fluent summaries and it may generate expressions not directly

Table 1. The statistics of average document and summary lengths on several popular summarization datasets

Dataset	Type	Source document			Target summary	
		♯token	♯sent	♯section	♯token	♯sent
Gigaword (Rush et al., 2015)	news	31	1	–	8	1.0
X-Sum (Narayan, Cohen, and Lapata, 2018)	news	431	20	–	23	1.0
Newsroom (Grusky, Naaman, and Artzi, 2018)	news	751	24	–	30	1.4
CNN/DM (Hermann et al., 2015)	news	790	29	–	56	3.8
BillSum (Kornilova and Eidelman, 2019)	legislation	1686	49	4.4	177	7.5
PubMed (Cohan et al., 2018)	research paper	3224	106	5.7	203	7.7
arXiv (Cohan et al., 2018)	research paper	6446	222	5.3	195	7.1
GovReport (Huang et al., 2021)	government report	9409	296	19.8	553	20.0

For *GovReport*, its documents are organized in a form of multi-level sections. The method of dividing its documents to a form of one-level sections is discussed in Section 5.1.

existing in the source document. Nowadays, neural abstractive text summarization (NATS) models (Shi *et al.*, 2021) based on sequence-to-sequence (Seq2Seq) architecture (Sutskever, Vinyals, and Le, 2014) are prevailing. In practice, different types of documents vary greatly in length. In Table 1, we present the length statistics of several popular summarization datasets. It can be observed that news stories are shorter than 800 words on average. In contrast, the average length of research papers exceeds 3000 words, and the average length of government reports even exceeds 9000 words. Most existing NATS models treat source document and summary as two single sequences, which works well in summarizing documents of short and medium lengths. However, limited by the underlying neural architectures of NATS models, this practice leads to some de facto difficulties when applied to long documents. Previous studies show that vanilla LSTM (Hochreiter and Schmidhuber, 1997) and vanilla Transformer (Vaswani *et al.*, 2017) can effectively handle sequences of several hundred words at most (Khandelwal *et al.*, 2018; Dai *et al.*, 2019). Besides, the memory and time complexities of computing Transformer grow quadratically with the sequence length. This constraint also limits the application of Transformer in long documents, since long sequences easily run out of GPU memory.

In this paper, we study the challenging setting of long document summarization (LDS), where one source document includes thousands of words and one summary includes hundreds of words. Under the length constraint of neural architectures, one common practice adopted by NATS models is to set a length limit and truncate the exceeded part. However, this simple practice discards useful information beyond the prescribed length limit. To handle the longer sequence, one approach is to design sophisticated network structures to capture the long-range dependency, such as hierarchically encoding the discourse structure (Webber and Joshi, 2012) of documents (Cohan *et al.*, 2018), or introducing the long-span attention mechanisms (Zaheer *et al.*, 2020). As another approach, *extractive-and-abstractive* methods extract some snippets first and then paraphrase them (Pilault *et al.*, 2020). Recently, Gidiotis and Tsoumakas (2020) propose a simple and effective method for LDS, which is named as divide-and-conquer (DANCER). DANCER decomposes a LDS problem into multiple smaller problems, reduces computational complexity, and achieves good performance. Concretely, it breaks a long document and its summary into several pairs of document section and corresponding summary. A NATS model is trained to summarize the sections of a document separately, and these partial summaries are then combined as a complete summary. *Text alignment* refers to the correspondence between two pieces of text. For DANCER, the alignment between section and summary sentences is necessary to decide which

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

sentences in summary should be treated as the target of one section. To achieve text alignment, DANCER utilizes ROUGE (Lin, 2004). However, ROUGE only matches tokens in a superficial and exact way, which does not support synonyms or paraphrasing. And as an approach to text comparison, ROUGE deviates from human judgment (Kryscinski *et al*., 2019; Fabbri *et al*., 2021). For these reasons, ROUGE-based text alignment also deviates from human judgment. This gap leaves some room for improving the NATS models that require ROUGE at the training stage. It is natural to ask the following questions: is ROUGE inevitable to achieve text alignment? is it possible to directly learn the text alignment from summarization data without utilizing ROUGE?

In this paper, we propose a novel framework for LDS. Our method treats summarizing a long document as an ensemble of summarizing its contained sections. As a preliminary step, we formally describe the section-to-summary-sentence (S2SS) alignment for LDS. Based on this, we propose a joint training objective to formulate LDS as an unbalanced optimal transport (UOT) (Chizat *et al*., 2015) problem. Accordingly, our method is named as UOT-based summarizer (UOTSumm). UOTSumm achieves multiple goals simultaneously: it jointly learns the optimal S2SS alignment and a section-level NATS summarizer, it also learns the number of aligned summary sentences for each section. At training stage, UOTSumm directly learns S2SS alignment from summarization data, without utilizing any external tool such as ROUGE. In terms of concrete implementation, UOTSumm comprises two modules: a Section-to-Summary (Sec2Summ) module and an aligned summary sentence counter (ASSC) module. The Sec2Summ module takes document sections as the input and outputs the corresponding abstractive summaries. ASSC module records the number of generated sentences for each section. We adopt an alternating optimization technique (Bezdek and Hathaway, 2002) to train UOTSumm, such that ASSC module and Sec2Summ module are alternately updated. UOTSumm includes a universal training objective for LDS, and its Sec2Summ module can be any existing NATS model. In this paper, we implement UOTSumm with two popular NATS models: Pointer-generator networks (PG-Net) (See, Liu, and Manning, 2017) and BART (Lewis *et al*., 2020). They represent two paradigms of NATS models: learning from scratch and fine-tuning from a pre-trained model. We evaluate these two UOTSumm variants on three public LDS benchmarks: *PubMed*, *arXiv* (Cohan *et al*., 2018), and *GovReport* (Huang *et al*., 2021). With a purely data-driven approach to text alignment, UOTSumm obviously outperforms its counterparts that are based on ROUGE. And when combined with UOTSumm, the improved PG-Net and BART also outperform their respective vanilla models by a large margin. On *PubMed* and *arXiv*, UOTSumm fine-tuned from BART outperforms some recent strong baseline models that are specifically designed for the LDS task. On *GovReport*, UOTSumm fine-tuned from BART achieves comparable performance with the state-of-the-art model. Besides, to thoroughly investigate the functions of each component, we introduce and study three ablation models for UOTSumm. At last, we also study some practical cases and conduct a human evaluation to show the advantages of UOTSumm.

The contributions of this paper are as follows:

- We propose a novel framework based on UOT theory for LDS task, which is named as UOTSumm. Under a unified training objective, UOTSumm jointly learns the optimal text alignment, a section-level NATS summarizer, and the number of aligned summary sentences for each section.
- We formalize the concept of S2SS alignment for LDS task. The existing models usually utilize ROUGE to achieve text alignment, while UOTSumm directly learns S2SS alignment from summarization data. The benefits of our practice are twofold. It provides less-biased supervision for training. In inference, it could predict the number of generated sentences for each section without relying on any headline information.
- We evaluate UOTSumm on three public LDS benchmarks: *PubMed*, *arXiv*, and *GovReport*. With a purely data-driven approach to text alignment, UOTSumm outperforms its

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

counterparts that are based on ROUGE. Besides, UOTSumm outperforms several recent competitive baseline models that are particularly designed for LDS.

- UOTSumm includes a universal training objective for LDS, which is applicable to any existing NATS model. When equipped with UOTSumm, two popular NATS models, that is PG-Net and BART, markedly outperform their vanilla implementations.

2. Related work

2.1 Abstractive summarization of long document

Modern NATS models are built based on Seq2Seq architecture (Shi *et al.*, 2021). Seq2Seq architecture first aggregates information from input text sequence with an encoder, and then generate output text sequence with a decoder. Common neural networks that serve as encoder or decoder can be LSTM (Hochreiter and Schmidhuber, 1997) or Transformer (Vaswani *et al.*, 2017). If one source document and its summary are treated as two single sequences, then Seq2Seq architecture can be directly applied to abstractive summarization task. This practice works when the documents to be summarized are not too long, but it is not suitable for long documents with thousands of words. On some standard language modeling benchmarks, it is observed that LSTM language model is capable of using about 200 tokens of context on average (Khandelwal *et al.*, 2018), and the effective context is shorter for vanilla Transformer (Dai *et al.*, 2019). One approach to solving this issue is hierarchical encoding (Nallapati *et al.* 2016; Cohan *et al.*, 2018). This approach decomposes a long document into chunks, where one chunk can be one sentence or one section. Each chunk is encoded by a lower-level encoder first, and then the chunk sequence is encoded by an upper-level encoder. With a hierarchical attention mechanism, the decoder attends to a chunk first and then attends to a concrete word. As another approach to LDS, *extractive-and-abstractive* methods explicitly conduct content selection from source text first, and then rewrite a summary based on the selected content (Jing and McKeown, 1999; Gehrmann, Deng, and Rush, 2018; Liu *et al.*, 2018; Chen and Bansal, 2018; Pilault *et al.*, 2020; Zhao, Saleh, and Liu, 2020).

Compared to training from scratch (Rush, Chopra, and Weston, 2015; See *et al.*, 2017), pretrain-finetune paradigm (Devlin *et al.*, 2019; Lewis *et al.*, 2020) boosts performance of NATS models (Liu and Lapata, 2019; Raffel *et al.*, 2020; Zhang *et al.*, 2020a), since it utilizes the transferred knowledge from large-scale external corpora. Pretrain-finetune paradigm is usually implemented based on Transformer. Self-attention mechanism is a cornerstone component of Transformer. However, the memory and computational requirements of self-attention grow quadratically with sequence length, which limits its application in LDS task. To tackle the quadratic characteristic, one approach is to modify self-attention mechanism such that the quadratic complexity is reduced (Tay *et al.*, 2022). To this end, sparse attention represents a class of methods, that forces each token to attend to only part of the context. For example, BigBird (Zaheer *et al.*, 2020), Longformer (Beltagy, Peters, and Cohan, 2020), LoBART (Manakul and Gales, 2021), and Poolingformer (Zhang *et al.*, 2021) adopt fixed attention patterns on some local contexts. Reformer (Kitaev, Kaiser, and Levskaya, 2019) and Sinkhorn Attention (Tay *et al.*, 2020) try to learn attention patterns. Different from the above methods concentrating on self-attention mechanism, Hepos (Huang *et al.*, 2021) modifies the encoder–decoder attention with head-wise positional strides to pinpoint salient information from source documents. Recently, Koh *et al.* (2022) give an empirical survey on datasets, models, and metrics for LDS task.

2.2 Text alignment in summarization

The concept of text alignment widely exists in both extractive and abstractive summarization tasks. For example, in supervised extractive text summarization, due to abstractive rephrasing of summary sentences, there is no explicit signal about which sentences should be extracted.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

To generate supervision signals, one common approach (Nallapati, Zhai, and Zhou, 2017) is to heuristically label a subset of sentences from source document, which has the maximum ROUGE score with the ground-truth summary. This process finds an *alignment* between summary and some snippets from document. Besides, to achieve training-stage content selection, text alignment also plays an important role in abstractive methods (Manakul and Gales, 2021) and *extractive-and-abstractive* methods (Liu *et al.*, 2018; Pilault *et al.*, 2020). To sum up, for most summarization models, ROUGE is a long-standing and common workhorse for training-stage text alignment.

2.3 Optimal transport

As the foundation of UOTSumm, related works of optimal transport (OT) (Villani, 2008; Peyré and Cuturi, 2019) are discussed in this section. The theory of OT originates from Monge's problem (Monge, 1781) of moving sand with the least effort. Kantorovich (1942, 2006) relaxed Monge's problem to a formulation of moving *mass* between two probability distributions. OT seeks the most efficient way of transforming one *histogram* to another when a cost function is given. It provides a tool to compare empirical probability distributions. In particular, UOT (Chizat *et al.*, 2015; Liero, Mielke, and Savaré, 2018) tackles the case when two histograms have different total *mass*. Recently, OT and UOT have been extensively applied to various machine learning (Frogner *et al.*, 2015; Kolouri *et al.*, 2017) and natural language processing (NLP) (Kusner *et al.*, 2015; Zhang *et al.*, 2017; Clark, Celikyilmaz, and Smith, 2019; Zhao *et al.*, 2019) problems. As the applications in NLP, OT is usually used to compare two sets of embeddings and serves as a distance measure. More specifically, OT distance and its variants can be applied to measure the document distance (Kusner *et al.*, 2015; Yokoi *et al.*, 2020), or to measure the similarity between words across multiple languages (Alvarez-Melis and Jaakkola, 2018; Xu *et al.*, 2021). Moreover, one benefit of applying OT to NLP task is the interpretability, which provides an explicit alignment between tokens. In this line of research, OT is applied to improve text generation (Chen *et al.*, 2019, 2020a), to achieve sparse and explainable text alignment (Swanson, Yu, and Lei, 2020), or to automatically evaluate the machine-generated texts (Clark *et al.*, 2019; Zhao *et al.*, 2019; Zhang *et al.*, 2020b; Chen *et al.*, 2020b).

2.4 A theoretical model of text summarization

In this section, we review some concepts from a theoretic text summarization model proposed by Peyrard (2019), which is helpful for understanding UOTSumm. The theoretic model is established based on information theory (Shannon, 1948). Its basic viewpoint is as follow: texts are represented by probability distributions over *semantic unit*s (Bao *et al.*, 2011). Take summarization data as an example, characters, words, n-grams, phrases, sentences, and sections in documents or summaries can be treated as semantic units. Based on this viewpoint, some intuitively used concepts in summarization, such as *importance*, *redundancy*, *relevance*, and *informativeness* are rigorously defined. This abstract model remains in theory. How to utilize it to guide various real-world summarization tasks is an underexplored but meaningful topic.

3. Preliminaries

3.1 Sequence-to-sequence learning

We first briefly review the training objective of Seq2Seq (Sutskever *et al.*, 2014) architecture, which is a cornerstone of NATS models. Denote input word sequence as $\mathbf{w}^{\text{in}} = \left(\mathbf{w}_1^{\text{in}}, \mathbf{w}_2^{\text{in}}, \cdots, \mathbf{w}_I^{\text{in}}\right)$, and output word sequence as $\mathbf{w}^{\text{out}} = \left(\mathbf{w}_1^{\text{out}}, \mathbf{w}_2^{\text{out}}, \cdots, \mathbf{w}_J^{\text{out}}\right)$. The objective of training a Seq2Seq architecture is to maximize the probability of observing $\mathbf{w}^{\text{out}}$ on condition that $\mathbf{w}^{\text{in}}$ is observed:

$$\max_{\theta} \mathbb{P}_{\theta}\left(\mathbf{w}^{\text{out}} | \mathbf{w}^{\text{in}}\right), \tag{1}$$

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

where θ denotes the trainable parameters of NATS. The typical auto-regressive training objective decomposes $\mathbb{P}_\theta(\mathbf{w}^{\text{out}}|\mathbf{w}^{\text{in}})$ into the product of a series of conditional probability, which predicts next word conditioned on the current context. It is equivalent to minimizing negative log likelihood (NLL) loss $\mathscr{L}^\theta\left(\mathbf{w}^{\text{in}}, \mathbf{w}^{\text{out}}\right)$ as follows:

$$\mathscr{L}^\theta\left(\mathbf{w}^{\text{in}}, \mathbf{w}^{\text{out}}\right) = -\log \mathbb{P}_\theta\left(\mathbf{w}^{\text{out}}|\mathbf{w}^{\text{in}}\right) = -\sum_{j=1}^{J} \log \mathbb{P}_\theta\left(\mathbf{w}_j^{\text{out}}|\mathbf{w}_{<j}^{\text{out}}, \mathbf{w}^{\text{in}}\right). \tag{2}$$

3.2 Unbalanced optimal transport

In this section, we provide some background knowledge on OT and UOT, which would help understand some technical aspects of our proposed framework. Let $\langle\cdot,\cdot\rangle$ stand for the Frobenius dot-product between two matrices of the same size. Given a cost matrix $\mathbf{C} \in \mathbb{R}_+^{m\times n}$ and two positive *histograms* $\mathbf{a} \in \mathbb{R}_+^m$ and $\mathbf{b} \in \mathbb{R}_+^n$, Kantarovich's formulation (Kantorovich, 1942, 2006) of OT problem is as follows:

$$\min_{\mathbf{P}\in\prod(\mathbf{a},\mathbf{b})} \langle \mathbf{P}, \mathbf{C}\rangle, \tag{3}$$

where $\prod(\mathbf{a},\mathbf{b}) = \{\mathbf{P} \in \mathbb{R}_+^{m\times n} | \mathbf{P}\mathbf{1}_n = \mathbf{a}, \mathbf{P}^T\mathbf{1}_m = \mathbf{b}\}$ is the set of all feasible transport plans. A basic requirement of the OT problem in Formula (3) is that two histograms should have the same total mass:

$$\mathbf{a}^T\mathbf{1}_m = \mathbf{b}^T\mathbf{1}_n. \tag{4}$$

However, many practical problems do not satisfy this constraint in nature. To tackle this issue, UOT (Chizat *et al.*, 2015; Liero *et al.*, 2018) relaxes the hard equality constraint $\mathbf{P} \in \prod(\mathbf{a},\mathbf{b})$ in Formula (3) to allow *mass variation*:

$$\min_{\mathbf{P}\in\mathbb{R}_+^{m\times n}} \langle \mathbf{P}, \mathbf{C}\rangle + \tau_1 \mathrm{KL}(\mathbf{P}\mathbf{1}_n||\mathbf{a}) + \tau_2 \mathrm{KL}\left(\mathbf{P}^T\mathbf{1}_m||\mathbf{b}\right). \tag{5}$$

Here, following terminologies in this area, $\mathbf{P}\mathbf{1}_n \in \mathbb{R}^m$ and $\mathbf{P}^T\mathbf{1}_m \in \mathbb{R}^n$ are named as *marginal* vectors. *Mass variation* refers to the discrepancy between marginal of the transport plan $\mathbf{P}$ and the mass of one side. It is measured with Kullback–Leibler (KL) divergence $\mathrm{KL}(\cdot||\cdot)$ defined as $\mathrm{KL}(\boldsymbol{\alpha}||\boldsymbol{\beta}) = \sum_i \left(\boldsymbol{\alpha}_i \log \frac{\boldsymbol{\alpha}_i}{\boldsymbol{\beta}_i} - \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i\right)$. In Formula (5), τ_1 and τ_2 are hyper-parameters for controlling how much mass variation is penalized as opposed to the transportation cost. When $\tau_1 \to +\infty$ and $\tau_2 \to +\infty$, UOT problem in Formula (5) is equivalent to the standard OT problem in Formula (3).

4. Method description

4.1 Section to summary sentence alignment

Section structures are widely existing in long documents of various genres, since it is natural to split a long document into subdivisions to relieve the burden of readers. One section usually consists of a series of sentences. It can be a paragraph for fictions or a section for research papers, as long as one section is coherently organized and related to a single topic. Formally, the training set for text summarization is usually organized as a set of document-summary article pairs. For each document-summary pair, the source long document contains m sections: $\{\mathbf{s}_i\}_{i=1}^m$, where each section $\mathbf{s}_i$ contains ℓ_i sentences $\{\mathbf{x}_k\}_{k=1}^{\ell_i}$; and the summary contains n sentences: $\{\mathbf{y}_j\}_{j=1}^n$. Text summarization is a lossy procedure, and the information in summary is only part of its source document. Correspondingly, as indicated in Table 1, the average summary length is much shorter than the average document length.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

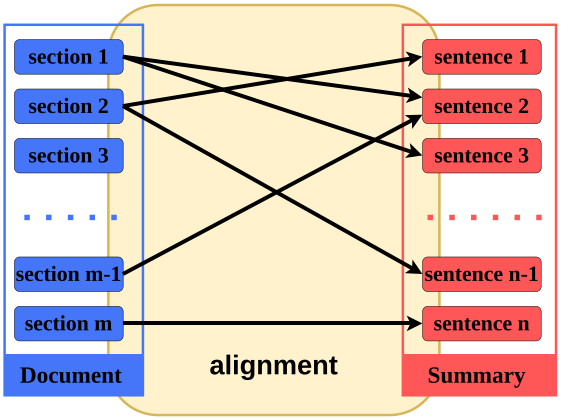

Figure 1. An illustration of S2SS alignment. The arrow from $\mathbf{s}_i$ to $\mathbf{y}_j$, symbolizes $\mathbf{P}_{i,j} > 0$, that is, there exists some degree of alignment between $\mathbf{s}_i$ and $\mathbf{y}_j$. Some sections may be aligned to more than one summary sentence, and some sections may not be aligned to any summary sentence. One summary sentence must be aligned to at least one section.

Now, we introduce the notion of S2SS alignment for the LDS task. We assign one unit score for each summary sentence $\mathbf{y}_j$, which represents the total amount of information contained in $\mathbf{y}_j$. This setting ensures all the summary sentences $\{\mathbf{y}_j\}_{j=1}^n$ are equally treated. We define a score $\mathbf{P}_{i,j} \in [0, 1]$ to measure the amount of information that summary sentence $\mathbf{y}_j$ gets from section $\mathbf{s}_i$. When $\mathbf{P}_{i,j}$ is larger, sentence $\mathbf{y}_j$ gets more information from section $\mathbf{s}_i$. As two extreme cases, $\mathbf{P}_{i,j} = 0$ indicates that $\mathbf{y}_j$ is irrelevant to $\mathbf{s}_i$, and $\mathbf{P}_{i,j} = 1$ indicates that $\mathbf{y}_j$ is generated exclusively from $\mathbf{s}_i$. For each $\mathbf{y}_j$, its information must come from the source document, hence we have: $\forall j, \sum_{i=1}^m \mathbf{P}_{i,j} = 1$. In contrast, each section $\mathbf{s}_i$ may provide information for at most n summary sentences. Besides the above explanation, $\mathbf{P}_{i,j}$ can also be understood from the following perspectives:

1. $\mathbf{P}_{i,j}$ measures the possibility that $\mathbf{y}_j$ is one summary sentence of section $\mathbf{s}_i$.
2. $\mathbf{P}_{i,j}$ measures the degree of S2SS alignment between sentence $\mathbf{y}_j$ and section $\mathbf{s}_i$.

We name the matrix $\mathbf{P}$ as S2SS *alignment plan* between $\{\mathbf{s}_i\}_{i=1}^m$ and $\{\mathbf{y}_j\}_{j=1}^n$. We give an illustration of S2SS alignment in Figure 1. In practice, it is very common that one summary sentence is totally based on one particular section. A natural question is: is it suitable to formulate $\mathbf{P}_{i,j}$ as a continuous variable in [0, 1], instead of a discrete 0–1 variable? In the next, we discuss two situations that justify the rationality of our formulation. First, one sentence summarizes content from several different sections. In this situation, a discrete 0–1 variable cannot measure the amount of information from different source sections. Second, different sections have overlapping information, and content of one summary sentence is based on the overlapping part. In this situation, the summary sentence should be equally possible to be aligned to any section. In Table 2, we present two cases. Case 1 satisfies the first situation, and Case 2 satisfies the second situation.

It should be pointed out that *oracle* S2SS alignment plan is existing, which can be decided by human judgment. But usually, no explicit *oracle alignment* is annotated for real-world documents.

4.2 Divide-and-conquer approach to LDS

To summarize a long document, one direct approach is to treat source document and summary as two single sequences and adopt the following objective for training:

$$\max_{\theta} \mathbb{P}_\theta \left(\overline{\mathbf{y}_1 \mathbf{y}_2 \cdots \mathbf{y}_n} \mid \overline{\mathbf{s}_1 \mathbf{s}_2 \cdots \mathbf{s}_m} \right). \tag{6}$$

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 2. Two cases to support formulating $\mathbf{P}_{i,j}$ as a continuous variable

Case 1	Source dataset: *arXiv*.
characteristic: One summary sentence is based on the content of several sections.	
summary sentence: We provide a broad overview of the theoretical status and phenomenological applications of the color glass condensate effective field theory describing universal properties of saturated gluons in hadron wavefunctions that are extracted from deeply inelastic scattering and hadron-hadron collision experiments at high energies.	
Section 1	*section heading*: Color glass condensate: theoretical status
Section 2	*section heading*: Collisions in the cgc framework
related sentences in this section: . . . The cgc is an effective theory for the wavefunction of a high-energy hadron or nucleus. . . . In this section, we apply it, with particular emphasis on factorization, to deeply inelastic scattering and hadronic collisions. . . .	
Section 3	*section heading*: Phenomenological applications of the cgc
Case 2	Source dataset: *PubMed*.
characteristic: Different sections have overlapping information, which is also the content of one summary sentence.	
summary sentence: We report the unique growth of nanofibers in silica and borosilicate glass using femtosecond laser radiation at 8 mhz repetition rate and a pulse width of 214 fs in air at atmospheric pressure.	
Section 1	*section heading*: Introduction
related sentence in this section: . . . In the present work, we aim to study the unique growth of nanofibers of silica and borosilicate glass using femtosecond laser radiation at mhz repetition rate under ambient condition, which is defined by rather a different mechanism. . . .	
Section 2	*section heading*: Experimental methods
related sentence in this section: . . . In the present case, arrays of microvias were drilled on silica and borosilicate glass specimens using laser radiation with a repetition of 8 mhz and pulse width 214 fs. . . .	
Section 3	*section heading*: Conclusions
related sentence in this section: . . . In summary, we report a characteristic growth of nanofibers of silica and borosilicate glass using femtosecond laser radiation at 8 mhz repetition rate and a pulse width of 214 fs under atmospheric pressure. . . .	

Here, the overline symbol denotes sequential concatenation of texts. However, existing neural architectures are not good at handling very long sequences.

To handle LDS problem, DANCER (Gidiotis and Tsoumakas, 2020) breaks it into several smaller-sized problems. The authors assume that one summary sentence can be aligned to exactly one section. Since oracle alignment is unavailable, they use the ROUGE tool to obtain a *surrogate* S2SS alignment. Concretely, ROUGE-L precision is computed between each summary sentence and each document sentence, and the summary sentence is aligned to section containing document sentence of the highest precision score. Then, a set of source–target pairs is constructed as $\{(\mathbf{s}_i, \{\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \cdots\})\}, (i \in I)$. Here, $\mathbf{s}_i$ is a section with at least one aligned summary sentence, I denotes the set of indices, and $\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \cdots$ follows order in the original summary: $i_1 < i_2 < \cdots$. Based on this surrogate S2SS alignment plan, DANCER adopts the following objective to train a NATS model:

$$\max_{\theta} \frac{1}{|I|} \sum_{i \in I} \mathbb{P}_{\theta} \left(\overline{\mathbf{y}_{i_1} \mathbf{y}_{i_2} \cdots} \,|\mathbf{s}_i \right). \tag{7}$$

Compared with the objective in Formula (6), the sequence length involved in Formula (7) is much shorter, which is computationally easier.

We point out that DANCER's approach to obtain the *surrogate* S2SS alignment has some room for improvement:

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

1. Some studies show that ROUGE is a biased approach to text comparison (Kryscinski *et al.*, 2019; Fabbri *et al.*, 2021), since it only relies on superficial and exact token matching. Hence the alignment constructed by DANCER differs from human judgment. NATS models trained on inexactly aligned source–target pairs are also biased. Besides, other approaches to text evaluation also have respective shortcomings (Kryscinski *et al.*, 2019; Fabbri *et al.*, 2021). Therefore, it is interesting to consider learning text alignment directly from data, without relying on any external tool.
2. At training stage, both source sections $\{\mathbf{s}_i\}_{i=1}^m$ and summary sentences $\{\mathbf{y}_j\}_{j=1}^n$ are available for constructing the *surrogate* S2SS alignment. This process recognizes sections that are useful for training a NATS model. However, at inference stage, no summary sentence is available to decide which sections should be adopted for generation. For DANCER, a heuristic is adopted to match section headings with a prepared keywords list including "*introduction*," "*methods*," "*conclusion*," etc.[a] They recognize the matched sections as important ones and adopt them for the generation at inference stage. This heuristic is less rigorous. It is hard to be transferred to the other domain, or long documents without section headings.
3. For DANCER, there is no way to decide the number of generated sentences for each section at inference stage. One common stopping criterion is to set a threshold length for the generation, which neglects the differences among sections.

4.3 Joint learning of text alignment and abstractive summarization

In this section, we first briefly describe the architecture of UOTSumm, and then introduce its training objective. UOTSumm is made up of two modules: a Section-to-Summary (Sec2Summ) module with trainable parameters Θ_1, and an ASSC module with trainable parameters Θ_2. Sec2Summ module learns to summarize each section from $\{\mathbf{s}_i\}_{i=1}^m$. Any existing NATS model based on encoder–decoder architecture can serve as the Sec2Summ module of UOTSumm. ASSC module adopts the representations of sections $\{\mathbf{s}_i\}_{i=1}^m$, that is, section embeddings from the encoder of Sec2Summ module, as its input. ASSC module is made up of a sequence encoder, for example LSTM, to model the context of document, and predicts the number of aligned summary sentence $\boldsymbol{\varphi}_i^{\Theta_2}$ for each section $\mathbf{s}_i$. And the vector $\boldsymbol{\varphi}^{\Theta_2}$ is defined as: $\boldsymbol{\varphi}^{\Theta_2} = \left[\boldsymbol{\varphi}_1^{\Theta_2}, \boldsymbol{\varphi}_2^{\Theta_2}, \cdots, \boldsymbol{\varphi}_m^{\Theta_2}\right]^T$. The architecture of UOTSumm is presented in Figure 2.

We propose the following joint optimization problem w.r.t. $\mathbf{P}$, Θ_1, and Θ_2, as the training objective for UOTSumm:

$$\begin{aligned} \min_{\mathbf{P},\Theta_1,\Theta_2} \quad & \langle \mathbf{P}, \mathbf{C}^{\Theta_1} \rangle + \tau \mathrm{KL}\left(\mathbf{P}\mathbf{1}_n || \boldsymbol{\varphi}^{\Theta_2}\right) \\ s.t. \quad & \mathbf{P} \in \mathbb{R}_+^{m \times n}, \mathbf{P}^T \mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{8}$$

In Problem (8), $\mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{1}_m \in \mathbb{R}^m$ denote all-one vectors, the cost matrix $\mathbf{C}^{\Theta_1}$ is defined as

$$\mathbf{C}^{\Theta_1}\left(\{\mathbf{s}_i\}_{i=1}^m, \{\mathbf{y}_j\}_{j=1}^n\right) = \begin{bmatrix} \mathscr{L}^{\Theta_1}(\mathbf{s}_1, \mathbf{y}_1) & \mathscr{L}^{\Theta_1}(\mathbf{s}_1, \mathbf{y}_2) & \cdots & \mathscr{L}^{\Theta_1}(\mathbf{s}_1, \mathbf{y}_n) \\ \mathscr{L}^{\Theta_1}(\mathbf{s}_2, \mathbf{y}_1) & \mathscr{L}^{\Theta_1}(\mathbf{s}_2, \mathbf{y}_2) & \cdots & \mathscr{L}^{\Theta_1}(\mathbf{s}_2, \mathbf{y}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \mathscr{L}^{\Theta_1}(\mathbf{s}_m, \mathbf{y}_1) & \mathscr{L}^{\Theta_1}(\mathbf{s}_m, \mathbf{y}_2) & \cdots & \mathscr{L}^{\Theta_1}(\mathbf{s}_m, \mathbf{y}_n) \end{bmatrix}, \tag{9}$$

[a]For *arXiv* and *PubMed*, a list of section types and corresponding keywords adopted by DANCER is presented Table 7.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

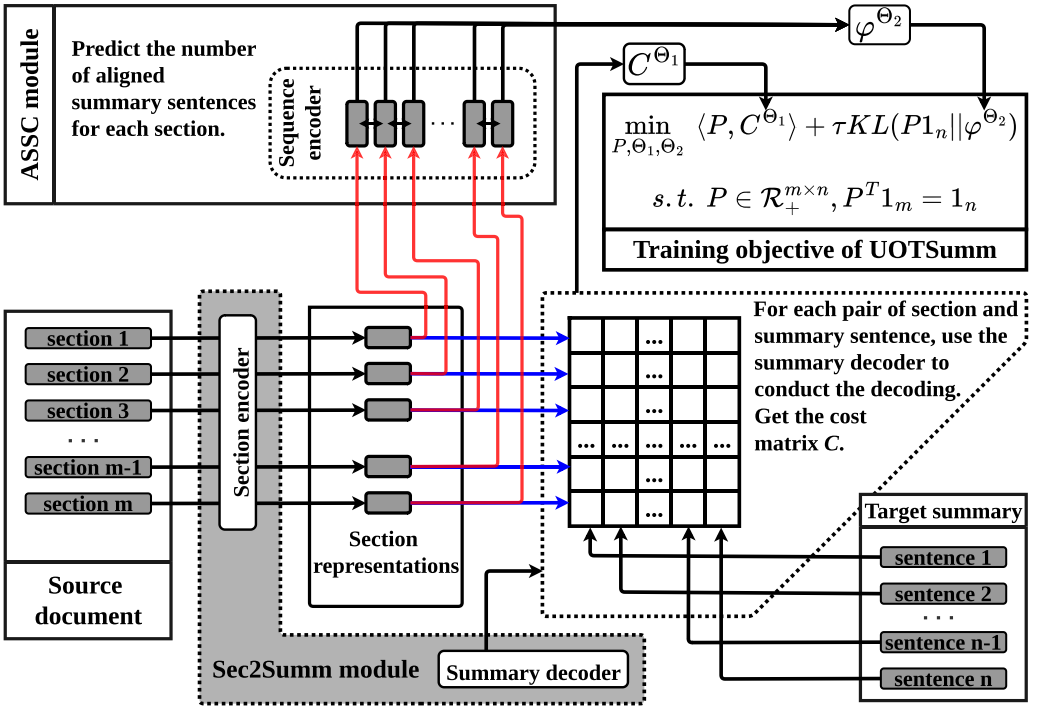

Figure 2. The architecture of UOTSumm.

where $\mathscr{L}^{\Theta_1}(*,*)$ is the loss function of Sec2Summ module. Usually, NLL loss $\mathscr{L}^{\theta}(*,*)$ defined in Equation (2) is adopted.

Roughly speaking, the first term $\langle \mathbf{P}, \mathbf{C}^{\Theta_1}\rangle$ in the objective of Problem (8) conducts abstractive summarization and S2SS alignment jointly, and the second term $\mathrm{KL}(\mathbf{P1}_n||\boldsymbol{\varphi}^{\Theta_2})$ automatically learns the number of aligned summary sentences for each section. In the next, we explain Problem (8) in more detail.

1. When $\mathbf{C}^{\Theta_1}$ and $\boldsymbol{\varphi}^{\Theta_2}$ are fixed as constants, the UOT problem in Formula (8) is a special form of Problem (5), where only the source-document side is relaxed with KL divergence.
2. From the constraint of Problem (8), it can be observed that variable $\mathbf{P}$ satisfies exactly the same requirements as the S2SS *alignment plan* defined in Section 4.1.
3. The term $\langle \mathbf{P}, \mathbf{C}^{\Theta_1}\rangle$ can be written as a summation form:

$$\langle \mathbf{P}, \mathbf{C}^{\Theta_1}\rangle = \sum_{j=1}^{n}\sum_{i=1}^{m} \mathbf{P}_{i,j}\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j), \tag{10}$$

which has the following properties:

(a) For any summary sentence $\mathbf{y}_j$, the constraint $\sum_{i=1}^{m}\mathbf{P}_{i,j} = 1$ ensures that we must use one unit amount of aligned sections $\{\mathbf{s}_i\}$ in total to minimize the term $\sum_{i=1}^{m}\mathbf{P}_{i,j}\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$.

(b) To minimize $\sum_{i=1}^{m}\mathbf{P}_{i,j}\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$ for each j, the i-th term $\mathbf{P}_{i,j}\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$ with a smaller loss value of $\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$ leads to a larger value of $\mathbf{P}_{i,j}$. In contrast, an unaligned pair $(\mathbf{s}_i, \mathbf{y}_j)$, that is, when $\mathbf{P}_{i,j} = 0$, does not serve as training data for Sec2Summ module, since $\mathbf{P}_{i,j}\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j) = 0$.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

(c) Minimizing the term $\langle \mathbf{P}, \mathbf{C}^{\Theta_1} \rangle$ accomplishes two purposes: finding the aligned sections $\{\mathbf{s}_i\}$ for each summary sentence $\mathbf{y}_j$ and using the set of aligned pairs $\{(\mathbf{s}_i, \mathbf{y}_j)\}$ to train the Sec2Summ module.

4. The second term $\mathrm{KL}(\mathbf{P}\mathbf{1}_n || \boldsymbol{\varphi}^{\Theta_2})$ in the training objective is explained as follows:

(a) $\forall i,\ \sum_{j=1}^{n} \mathbf{P}_{i,j}$ is the number of summary sentences that are aligned to the i-th section $\mathbf{s}_i$.
(b) Minimizing the term $\mathrm{KL}(\mathbf{P}\mathbf{1}_n || \boldsymbol{\varphi}^{\Theta_2})$ helps to train the parameters Θ_2 of ASSC module, so that $\boldsymbol{\varphi}_i^{\Theta_2} \geq 0$ is a good estimation of the number of aligned summary sentences for section $\mathbf{s}_i$.
(c) Computing $\boldsymbol{\varphi}^{\Theta_2}$ in forward propagation only requires sections $\{\mathbf{s}_i\}_{i=1}^{m}$ from the source side document. At inference stage, ASSC module can decide the number of generated sentences for each section when ground-truth summary is unavailable.
(d) Learning $\boldsymbol{\varphi}^{\Theta_2}$ does not utilize any section heading, such as "*introduction*," "*methods*," "*conclusion*" in scientific papers. Therefore, different from DANCER, UOTSumm can be directly applied to any type of long articles as long as they are organized in sections, even when section headings are unavailable.

5. As one perspective, the joint training objective in Problem (8) can be understood as: the Sec2Summ module with parameters Θ_1, the ASSC module with parameters Θ_2, and the S2SS alignment plan $\mathbf{P}$ are optimal at the same time.

6. Problem (8) can be understood from the viewpoint of OT (Peyré and Cuturi, 2019) as follows:

(a) Using the terminologies discussed in Section 2.4, we stipulate that section is the *semantic unit* of source documents, and sentence is the *semantic unit* of summaries.
(b) One source document is represented as a probability distribution over sections, where its mass is unknown in advance. Hence, we parameterize its mass as a learnable vector $\boldsymbol{\varphi}^{\Theta_2}$. One target summary is represented as a probability distribution over sentences, where each summary sentence has the equal amount of mass.
(c) UOTSumm tries to *move* information from a distribution of source sections to a distribution of summary sentences with the least-effort way.
(d) The moving *cost* is measured by loss values of a NATS model. This setting is reasonable, since the loss value is smaller when one sentence is more prone to be the summary of one section.
(e) The mass vector $\boldsymbol{\varphi}^{\Theta_2}$ in the source side is learnable. We cannot guarantee that source and target sides always have the same amount of total mass, which is prescribed by the balanced OT problem in Formula (3). In other words, $\sum_{i=1}^{m} \boldsymbol{\varphi}_i^{\Theta_2} = \sum_{j=1}^{n} \sum_{i=1}^{m} \mathbf{P}_{i,j} = n$ is not always guaranteed. Therefore, we adopt the unbalanced formulation in Problem (8).

4.4 Training and inference strategies

We propose to apply an alternating optimization method (Bezdek and Hathaway, 2002) to train the joint objective of UOTSumm in Formula (8). The basic idea is: we fix two variables from $\{\mathbf{P}, \Theta_1, \Theta_2\}$ and optimize the remaining variable and repeat this procedure in a rotating way for each training iteration. The detailed training procedure of UOTSumm is presented in Algorithm 1, and one loop of Algorithm 1 is visualized in Figure 3. Although our methods have two network modules with separate optimizers, their training is alternate and the whole system is in an end-to-end fashion.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Algorithm 1 Training framework of UOTSumm

Require: The whole dataset of paired documents and summaries: $\left\{\left(\{\mathbf{s}_i\}_{i=1}^{m}, \{\mathbf{y}_j\}_{j=1}^{n}\right)\right\}$.

1: **repeat**
2: Get one batch of document-summary pairs.
3: For each document, encode sections $\{\mathbf{s}_i\}_{i=1}^{m}$ with the *section encoder*
4: For each pair of section and summary sentence $(\mathbf{s}_i, \mathbf{y}_j)$, conduct *tentative decoding* with the *summary decoder* and get the loss value $\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$. Compute $\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$ for all the possible combinations $\left\{(\mathbf{s}_i, \mathbf{y}_j)\right\}_{1\le i\le m, 1\le j\le n}$, and construct the cost matrix $\mathbf{C}^{\Theta_1}$ in Formula (9).
5: Use the section embeddings from Step (3) as the input of ASSC module, and compute the number of aligned summary sentences $\boldsymbol{\varphi}_i^{\Theta_2}$ for each section $\mathbf{s}_i$.
6: Solve UOT Problem (11) with Algorithm 2, and get the solution $\mathbf{P}^*$.
7: Assign each summary sentence $\mathbf{y}_j$ to section $\mathbf{s}_i$ with the largest alignment score $\mathbf{P}_{ij}^*$, and get a S2SS alignment set $\left\{(\mathbf{s}_{\tilde{i}}, \{\mathbf{y}_{\tilde{j}1}, \mathbf{y}_{\tilde{j}2}, \cdots\})\right\}$. If one section is not assigned with any summary sentence, it is excluded from the alignment set.
8: For the S2SS alignment set $\left\{(\mathbf{s}_{\tilde{i}}, \{\mathbf{y}_{\tilde{j}1}, \mathbf{y}_{\tilde{j}2}, \cdots\})\right\}$, concatenate the aligned summary sentences for each section and get the set $\left\{(\mathbf{s}_{\tilde{i}}, \overline{\mathbf{y}_{\tilde{j}1}\mathbf{y}_{\tilde{j}2}\cdots})\right\}$. The concatenation operation follows the order in the original summary.
9: Fix the parameters Θ_2 of ASSC module. Conduct decoding with the *summary decoder* for the set $\left\{(\mathbf{s}_{\tilde{i}}, \overline{\mathbf{y}_{\tilde{j}1}\mathbf{y}_{\tilde{j}2}\cdots})\right\}$, compute the average of $\mathscr{L}^{\Theta_1}(\mathbf{s}_{\tilde{i}}, \overline{\mathbf{y}_{\tilde{j}1}\mathbf{y}_{\tilde{j}2}\cdots})$ as the training objective, and update the parameters Θ_1 of Sec2Summ module by back propagation.
10: Fix the parameters Θ_1 of Sec2Summ module. Compute the average of KL $\left(\mathbf{P}^*\mathbf{1}_n||\boldsymbol{\varphi}^{\Theta_2}\right)$ over the whole batch as the training objective, and update the parameters Θ_2 of ASSC module by back propagation.
11: **until** The termination criterion of Sec2Summ module is satisfied on the validation set.

Ensure: The UOTSumm model with trained parameters Θ_1 and Θ_2.

Algorithm 2 Log-domain Sinkhorn algorithm for Problem (11)

Require: $k=0$, $\boldsymbol{u}^0=\mathbf{0}_m$, $\boldsymbol{v}^0=\mathbf{0}_n$, K is the maximum number of iterations allowed.

1: **while** $k<K$ **do**
2: $\boldsymbol{u}^{k+1}=\frac{\tau}{\tau+\varepsilon}\left\{\boldsymbol{u}^k+\varepsilon\log\left(\boldsymbol{\varphi}^{\Theta_2}\right)-\log\left(\mathbf{M}(\boldsymbol{u}^k,\boldsymbol{v}^k)\mathbf{1}_n\right)\right\}$
3: $\boldsymbol{v}^{k+1}=\boldsymbol{v}^k+\varepsilon\log\left(\mathbf{1}_n\right)-\log\left(\mathbf{M}(\boldsymbol{u}^{k+1},\boldsymbol{v}^k)^T\mathbf{1}_m\right)$
4: $k=k+1$
5: **end while**

Ensure: $\mathbf{P}_{\Theta}^*=\mathbf{M}(\boldsymbol{u}^k,\boldsymbol{v}^k)$, where $\Theta=\{\Theta_1,\Theta_2\}$.

When Θ_1 and Θ_2 are fixed, we need to solve the UOT problem in Formula (8), where $\mathbf{C}^{\Theta_1}$ and $\boldsymbol{\varphi}^{\Theta_2}$ are constants in this case. Considering the computational efficiency, we adopt the practice in Cuturi (2013); Frogner *et al.* (2015) and solve the following entropy-regularized UOT problem:

$$\begin{aligned}&\min_{\mathbf{P}}\left\langle\mathbf{P},\mathbf{C}^{\Theta_1}\right\rangle-\varepsilon H(\mathbf{P})+\tau\mathrm{KL}\left(\mathbf{P}\mathbf{1}_n||\boldsymbol{\varphi}^{\Theta_2}\right)\\&s.t.\ \mathbf{P}\in\mathbb{R}_+^{m\times n},\mathbf{P}^T\mathbf{1}_m=\mathbf{1}_n\end{aligned}\tag{11}$$

Here, $H(\mathbf{P})$ is an entropy regularization term defined as: $H(\mathbf{P})=-\sum_{i,j}\mathbf{P}_{ij}(\log(\mathbf{P}_{ij})-1)$. We choose the hyper-parameter ε as a small positive value, so that Problem (11) is a good approximation of the original UOT problem in Formula (8). We utilize Sinkhorn algorithm in log domain (Chizat *et al.*, 2018; Schmitzer, 2019) to solve Problem (11), and its details

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

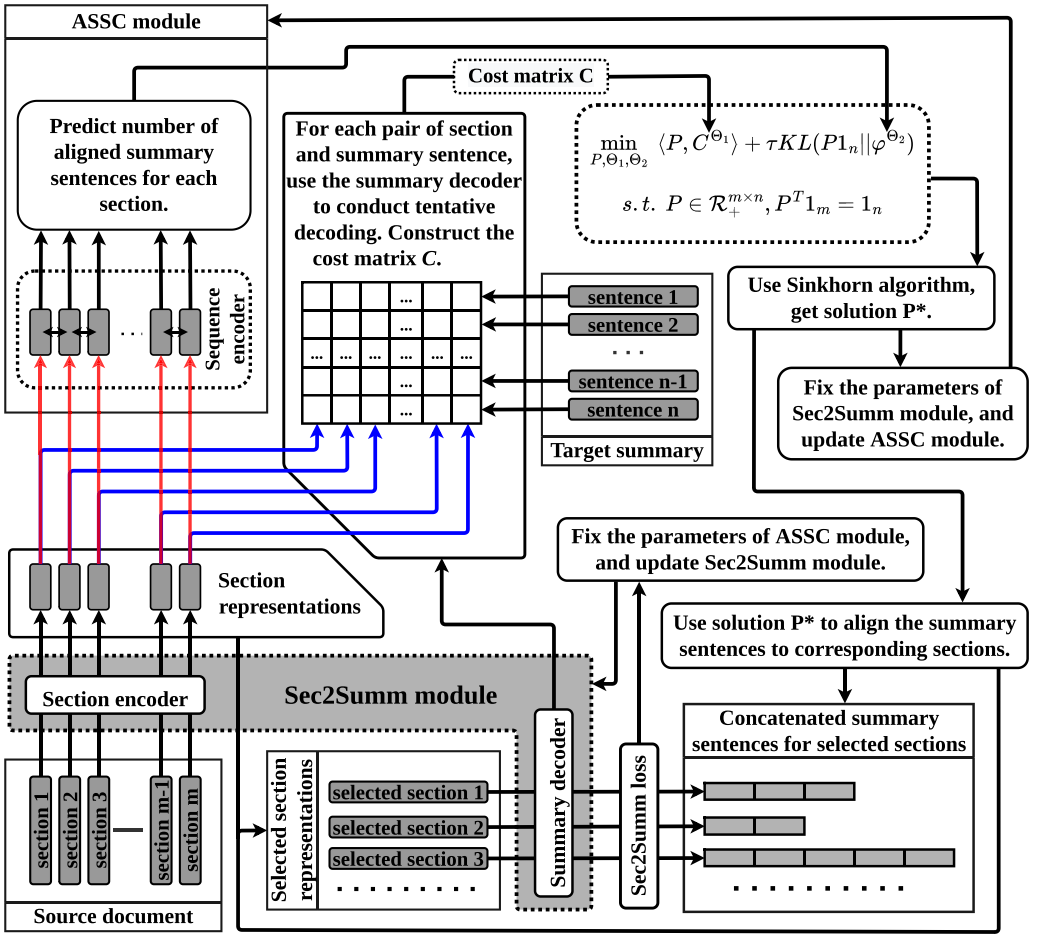

Figure 3. The procedure of training UOTSumm. This figure visualizes one loop of Algorithm 1.

are presented in Algorithm 2. In this algorithm, the function $\mathbf{M}(\boldsymbol{u}, \boldsymbol{v})$ is defined as $\mathbf{M}(\boldsymbol{u}, \boldsymbol{v}) = \text{diag}(e^{\frac{\boldsymbol{u}}{\varepsilon}})\, e^{-\frac{1}{\varepsilon}\mathbf{C}^{\Theta_1}}\, \text{diag}(e^{\frac{\boldsymbol{v}}{\varepsilon}})$.

In Step (4) of Algorithm 1, we compute $\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$ for each pair of section and summary sentence $(\mathbf{s}_i, \mathbf{y}_j)$ given the fixed Θ_1. We name this procedure as *tentative decoding*, since these loss values are not used for back propagation. For document sections $\{\mathbf{s}_i\}_{i=1}^m$ and summary sentences $\{\mathbf{y}_j\}_{j=1}^n$, there are $m \times n$ possible combinations of pairs $(\mathbf{s}_i, \mathbf{y}_j)$ in total. Then a natural question is: whether is *tentative decoding* very slow? Fortunately, we observe that it is not an issue in practice. For UOTSumm implemented with Pytorch (Paszke *et al.*, 2019), in each iteration, we observe that the total time of *tentative decoding* is similar to the total time of decoding step for back propagation. Since it is not an emphasis of our paper, we only infer the reason for this phenomenon. We infer that when the decoded results are not adopted for back propagation, Pytorch stores fewer intermediate variables and requires less computation.

We formulate the alignment score between summary sentence $\mathbf{y}_j$ and section $\mathbf{s}_i$ as a continuous variable $\mathbf{P}_{i,j} \in [0, 1]$. The advantages of this formulation have been discussed in Sections 4.1 and 4.3. However, it brings difficulty for Sec2Summ module in the procedure of alternating optimization. Consider the following situation. After Step (6) of Algorithm 1, for a summary sentence $\mathbf{y}_j$, more than one alignment scores $\mathbf{P}_{i,j}^*$ are positive. To update Sec2Summ module, a unique

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Algorithm 3 Inference procedure of UOTSumm for summary generation

Require: One source document with sections $\{\mathbf{s}_i\}_{i=1}^m$; an empty list to contain the generated summary sentences: `summ_list`=[].

Encode sections $\{\mathbf{s}_i\}_{i=1}^m$ with the *section encoder*.
Compute the number of aligned summary sentences $\boldsymbol{\varphi}_i^{\Theta_2}$ for each section $\mathbf{s}_i$ with ASSC module.
$i = 1$.
while $i \leq m$ **do**
  For section $\mathbf{s}_i$, use the trained Sec2Summ module and *beam search* algorithm to generate a summary. Stop until $\lfloor \boldsymbol{\varphi}_i^{\Theta_2} + 0.5 \rfloor$ sentences are generated for this section, where $\lfloor \cdot \rfloor$ is the floor rounding function.
  Append the generated summary sentences of section $\mathbf{s}_i$ to `summ_list`.
  $i = i + 1$.
end while
`// Trigram blocking is an optional module.`
if *trigram blocking* is required **then**
  `blocked_summ_list=[]`
  for each $\widetilde{\mathbf{y}}$ in `summ_list` **do**
    if there is not a trigram overlapping between $\widetilde{\mathbf{y}}$ and any sentence in `blocked_summ_list` **then**
      Append $\widetilde{\mathbf{y}}$ to `blocked_summ_list`.
    end if
  end for
  `summ_list=blocked_summ_list`
end if

Ensure: Concatenate the sentences in `summ_list` as the summary of this document.

target summary sequence is required for Seq2Seq learning. Then, the difficulty is: for $\mathbf{y}_j$ with positive scores on multiple sections, which section should it be aligned to? As indicated in Step (7) of Algorithm 1, we make a compromise and adopt an approximate strategy. The experimental results in Section 5.3 show that this approximation is suitable for practical usage, and we leave a more accurate algorithm of optimizing the objective in Problem (8) as the future work.

The inference procedure of UOTSumm for summary generation is presented in Algorithm 3. It is made up of two steps: section selection and abstractive summarization of the selected sections. It should be highlighted that computing $\boldsymbol{\varphi}^{\Theta_2}$ only requires sections $\{\mathbf{s}_i\}_{i=1}^m$ from the source side document. Hence at inference stage, ASSC module can decide the number of generated sentences for each section when the ground-truth summary is unavailable.

4.5 Ablation models

In this section, we introduce three ablation models of UOTSumm. As discussed in Section 4.3, for section and summary pair $(\mathbf{s}_i, \mathbf{y}_j)$, a smaller loss value of $\mathscr{L}^{\Theta_1}(\mathbf{s}_i, \mathbf{y}_j)$ leads to a larger alignment score of $\mathbf{P}_{i,j}$. Then, one question is: based on this relationship, can we simplify the alignment procedure in Algorithm 1? To this end, we remove Step (6) and modify the alignment strategy in Step (7) as: assign each summary sentence $\mathbf{y}_j$ to section $\mathbf{s}_i$ with the smallest $\mathbf{C}_{ij}^{\Theta_1}$. Besides, we replace the KL divergence term in Step (10) with a squared loss $\sum_{i=1}^m (\boldsymbol{\varphi}_i^* - \boldsymbol{\varphi}_i^{\Theta_2})^2$, where $\boldsymbol{\varphi}_i^*$ is the number of aligned summary sentences for section $\mathbf{s}_i$. We treat this modified procedure as one ablation model of UOTSumm, and name it as "*simple alignment*."

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

ASSC module learns to record the number of generated sentences for each section. To investigate the effectiveness of this module, we further simplify *simple alignment* as the second ablation model. We replace the regression objective of ASSC module with a classification objective. Concretely, we use 1 as the label if $\boldsymbol{\varphi}_i^* > 0$, use 0 as the label if $\boldsymbol{\varphi}_i^* = 0$, and adopt a binary cross entropy loss for training. Since this ablation model does not learn the number of generated sentences, we set a threshold to restrict the length of generated summary. In practice, the threshold value depends on dataset, we try different values and report the best results in the experiment.

For summarization task that requires to generate multiple sentences, one common problem is: the model may generate repetitive or similar sentences. For DANCER-based methods, this problem may be more serious because summary sentences are independently produced from different sections. To handle this issue, we adopt *trigram blocking* (Paulus, Xiong, and Socher, 2018; Liu and Lapata, 2019) as a post-processing step in the inference procedure. The details are included in Algorithm 3. To investigate its effect, UOTSumm without *trigram blocking* is treated as one ablation model.

5. Experiments

5.1 Datasets and evaluation metrics

We adopt three popular LDS datasets for evaluation: *arXiv*, *PubMed,* and *GovReport*. The statistics of these three datasets are included in Table 1. Their descriptions and pre-processing steps are as follows.

- *arXiv* and *PubMed* (Cohan *et al*., 2018) are two datasets collected from research papers. One research paper is treated as source, and its abstract is treated as summary. We adopt the same way of splitting as in Cohan *et al*. (2018). For *arXiv*, the sizes of `training`/`validation`/`testing` sets are 203, 037/ 6436/ 6440. For *PubMed*, the sizes of `training`/`validation`/`testing` sets are 119, 924/ 6633/ 6658. The documents in these two datasets are pre-processed as one-level sections by the original authors, and we follow the same way of dividing sections.
- *GovReport* (Huang *et al*., 2021) contains long reports published by U.S. Government Accountability Office to fulfill requests by congressional members, and Congressional Research Service covering researches on a broad range of national policy issues. We adopt the same way of splitting as in Huang *et al*. (2021), and the sizes of `training`/`validation`/`testing` sets are 17, 519/ 974/ 973. For *GovReport*, its documents are organized in a form of multi-level sections. Concretely, each document is made up of several *section*s, each *section* contains several *subsection*s and/or several *paragraph*s, and each *subsection* contains several *paragraph*s.[b] To accommodate UOTSumm, we need to transform documents to a form of one-level sections. Our primary consideration is: the length of each section should be moderate. To this end, we use "`paragraphs`" as the key[c] to recursively iterate the multi-level structures, and concatenate all the paragraphs of each "`paragraphs`" as one section.

As shown in Table 1, the documents of *BillSum* dataset are relatively longer than those from news datasets. We manually check *BillSum*, and get the following findings. Although the average section number of documents is 4.4, the sentences are usually concentrated in one or two sections. The documents contain many very short sections with only one useless sentence, whose headings are "short title," "effective date," "funding," etc. Therefore, *BillSum* is not suitable for

[b]To avoid confusion, the italic "*section*," "*subsection*," and "*paragraph*" refer to the structures defined by the original dataset authors (Huang *et al*., 2021).

[c]The data of *GovReport* are stored with a multi-level dictionary data structure, and "`paragraphs`" is one dictionary key.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

DANCER-based summarization models, since the motivation of these models is to reduce input length by splitting long document into several moderate-length sections. We do not consider *BillSum* as an evaluation dataset.

Currently, ROUGE is the default and most popular metric for evaluating summarization models. However, as discussed in Fabbri *et al.* (2021), ROUGE has some shortcomings, and some other evaluation metrics make up for these disadvantages. To make the comparison more complete and convincing, we adopt three automatic evaluation metrics in this paper: ROUGE, BERTScore, and MoverScore. They are described as follows.

- ROUGE (recall-oriented understudy for gisting evaluation) (Lin, 2004) measures the number of overlapping textual units, that is n-grams and word sequences, between a generated summary and its ground-truth reference. F-measures of ROUGE-1, ROUGE-2, and ROUGE-L are reported.
- BERTScore (Zhang *et al.*, 2020b) computes token-level similarity scores by aligning generated summary and ground-truth reference. Instead of exact matches, it computes token similarity using contextualized token embeddings from BERT (Devlin *et al.*, 2019). F1-measure of BERTScore is reported.
- MoverScore (Zhao *et al.*, 2019) utilizes the Word Mover's Distance (Kusner *et al.*, 2015) to compare a generated summary and its ground-truth reference. It operates over n-gram embeddings pooled from BERT representations.

SummEval[d] (Fabbri *et al.*, 2021) is a unified and easy-to-use toolkit, which contains common evaluation metrics for text summarization. We utilize SummEval with its default settings to compute the above three metrics.

5.2 Implementations

UOTSumm includes a general-purpose training objective for LDS task. Its implementation is made up of two modules: a Sec2Summ module and an ASSC module. The Sec2Summ module can be any existing NATS model. NATS models can be grouped into two categories: learning-from-scratch (Rush *et al.*, 2015; See *et al.*, 2017), and adopting the pretrain-finetune paradigm (Liu and Lapata, 2019; Lewis *et al.*, 2020; Zhang *et al.*, 2020a). To demonstrate the universality and effectiveness of UOTSumm, we choose one typical NATS model from each category and adapt it to UOTSumm. Their details are described as follows.

- PG-Net (See *et al.*, 2017) is a representative NATS model of learning-from-scratch. We follow most settings of vanilla PG-Net.[e] The vocabulary size is set to 50, 000. For *arXiv* and *PubMed*, we adopt the vocabulary provided by Cohan *et al.* (2018). For *GovReport*, since Huang *et al.* (2021) did not provide a vocabulary, we consider the most frequent 50, 000 words in the training set as the vocabulary.
- BART (Lewis *et al.*, 2020) is a representative of pretrain-finetune paradigm. We use the publicly released BART model fine-tuned on CNN/DM[f] (Hermann *et al.*, 2015) to initialize model parameters. We implement based on the AllenNLP[g] wrapper of BART.[h] We follow most settings of its vanilla implementation[i] except the learning rate, which is tuned from: $\{1e^{-5}, 1.5e^{-5}, 3e^{-5}, 5e^{-5}\}$.

[d]https://github.com/Yale-LILY/SummEval.
[e]We implement based on the repository in: https://github.com/lipiji/neural-summ-cnndm-pytorch.
[f]The model is available in https://huggingface.co/facebook/bart-large-cnn.
[g]https://allennlp.org.
[h]https://github.com/allenai/allennlp-models/blob/main/allennlp_models/generation/models/bart.py.
[i]https://github.com/allenai/allennlp-models/blob/main/training_config/generation/bart_cnn_dm.jsonnet.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

For ASSC module, Adam (Kingma and Ba, 2014) optimizer is adopted for training with a learning rate of $1e^{-5}$. All the experiments are conducted on one NVIDIA TITAN RTX GPU with 24 GB memory, or one NVIDIA RTX A6000 GPU with 48 GB memory, depending on dataset and model sizes. At the testing stage, we adopt a beam size of 4 for all the variants of UOTSumm. We only implement ablation models for BART-based UOTSumm, which adopt the same experimental settings as the full implementation.

5.3 Baselines and results

In this section, we compare UOTSumm with some competitive NATS baselines. We implement two variants for UOTSumm. Generally speaking, pretrain-finetune-based NATS models are more powerful than learning-from-scratch, since the former benefit from transferred knowledge of external corpus. To ensure fairness of comparison, baselines are accordingly classified into two groups. To compare with PG-Net-based UOTSumm, we adopt learning-from-scratch NATS models as follows.

- Seq2Seq (Chopra, Auli, and Rush, 2016; Nallapati *et al.*, 2016), a Seq2Seq NATS model equipped with attention.
- PG-Net (See *et al.*, 2017), a NATS model featured with the copying (Gu *et al.*, 2016) and the coverage (Tu *et al.*, 2016) mechanisms.
- Discourse-Aware (Cohan *et al.*, 2018), a NATS model equipped with a hierarchical encoder to capture the discourse structure of the document and a discourse-aware decoder.
- Ext + TLM (Pilault *et al.*, 2020), an *extractive-and-abstractive* summarization model based on Transformer. Its extractive stage relies on ROUGE to produce the ground-truth extraction targets.
- reinforce-selected sentence rewriting (RSSR) (Chen and Bansal, 2018), an *extractive-and-abstractive* NATS model.[j] RSSR is made up of an extractor which extracts sentences, and an abstractor which rewrites the extracted sentences as a summary. The extractor and the abstractor are bridged together with policy-based reinforcement learning.
- DANCER + PG-Net (Gidiotis and Tsoumakas, 2020), the DANCER framework combined with PG-Net. The authors did not provide code for this version of DANCER.

To compare with BART based UOTSumm, baselines are chosen from the following pretrain-finetune based NATS models.

- PEGASUS (Zhang *et al.*, 2020a), a self-supervised pre-training objective specifically designed for text summarization. Some important sentences are masked and generated as one output sequence conditioned on the remaining sentences. Pre-trained PEGASUS is often adopted by the other NATS models for fine-tuning.
- BigBird + PEGASUS (Zaheer *et al.*, 2020), fine-tuning PEGASUS for BigBird. BigBird combines sliding window, global, and random token attentions in its encoder.
- DANCER + PEGASUS (Gidiotis and Tsoumakas, 2020), fine-tuning PEGASUS for DANCER.
- BART (Lewis *et al.*, 2020), a denoising auto-encoder for pre-training Seq2Seq models.
- MCS + BART (Manakul and Gales, 2021), a multitask content selection model with sentence-level extractive labeling. Its training-stage content selection relies on ROUGE.

[j] We implement the baseline based on the repository in: https://github.com/ChenRocks/fast_abs_rl.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

- DYLE + RoBERTa + BART (Mao *et al.* 2022), a dynamic latent extraction approach for abstractive LDS. DYLE is made up of an extractor which is initialized with RoBERTa (Liu *et al.*, 2019), and a generator which is initialized with BART.
- LED + BART (Beltagy *et al.*, 2020), fine-tuning BART for a Longformer variant. Longformer's attention mechanism combines a local windowed attention with a task motivated global attention.
- Stride Patterns (Child *et al.* 2019), a sparse factorization of the self-attention matrix which reduces the quadratic computational complexity.
- LSH (Kitaev *et al.*, 2019), which replaces dot-product attention with the locality-sensitive hashing to reduce complexity.
- Sinkhorn Attention (Tay *et al.*, 2020), which segments a sequence into blocks and adopt a learnable Sinkhorn sorting network to reduce complexity.
- Hepos (Huang *et al.*, 2021), an efficient encoder–decoder attention mechanism with head-wise positional strides to pinpoint salient information from source document.

Some of the above baseline models do not have reported results on *arXiv*, *PubMed*, or *GovReport*. Besides, for all the existing results on these three datasets, only ROUGE scores are reported. Based on these two facts, we reproduce the following baseline models: RSSR, PEGASUS, BigBird, PEGASUS-based DANCER, BART, and LED. In this way, the results of BERTScore and MoverScore for these models can be obtained, and baselines for three datasets are more consistent. All the pre-trained Transformer models are downloaded from Huggingface models.[k] We only reproduce DANCER on *arXiv* and *PubMed*,[l] because DANCER requires some human-designed rules for selecting sections, which are unavailable for *GovReport*. For PEGASUS, BigBird, and DANCER, our reproduced ROUGE scores have some minute differences with the scores reported in original papers. We follow the practice of Zaheer *et al.* (2020) and report all the versions of ROUGE scores. The results of UOTSumm and the baseline models on *arXiv*, *PubMed*, and *GovReport* are reported in Tables 3–5, respectively. For UOTSumm, we report results of four variants: the full implementation, and three ablation models introduced in Section 4.5. Baseline models specially designed for LDS task are marked with the symbol ♣. The symbol ‡ denotes that the results are produced by us, while results without ‡ are taken from the original papers.

We can draw the following conclusions from the results.

1. On *arXiv* and *PubMed*, PG-Net-based UOTSumm outperforms PG-Net by a large margin. On all the three datasets, UOTSumm finetuned from BART outperforms BART by a large margin. The performance gain partly comes from the DANCER approach, which is also adopted by DANCER.
2. On *arXiv* and *PubMed*, PG-Net-based UOTSumm outperforms PG-Net based DANCER, and finetuned UOTSumm outperforms finetuned DANCER, in terms of all the evaluation metrics. UOTSumm directly learns S2SS alignment from data, while DANCER achieves S2SS alignment via ROUGE. The improvements demonstrate that our purely data-driven approach captures better text alignment than ROUGE.
3. The listed pretrain-finetune-based baselines are all recent competitive NATS models. We compare BART-based UOTSumm with them and analyze the results as follows.

[k]https://huggingface.co/models.

[l]We use the repository provided by original paper authors: https://github.com/AlexGidiotis/DANCER-summ. In this page, the authors mentioned that this implementation is different from that used in their paper. Hence, the reproduced scores are slightly different from scores reported in Gidiotis and Tsoumakas (2020).

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 3. Results on the test set of *arXiv*

Model	ROUGE			BERT-S	Mover-S
	R-1	R-2	R-L		
Seq2Seq (Chopra et al., 2016)	29.30	6.00	25.56	–	–
PG-Net (See et al., 2017)	32.06	9.04	25.16	–	–
Discourse-Aware (Cohan et al., 2018) ♣	35.80	11.05	31.80	–	–
Ext + TLM (Pilault et al., 2020) ♣	41.62	14.69	38.03	–	–
RSSR (Chen and Bansal, 2018) ♣‡	45.23	16.94	41.04	39.37	17.83
DANCER + PG-Net (Gidiotis and Tsoumakas, 2020) ♣	41.87	15.92	37.61	–	–
UOTSumm + PG-Net (Our method) ♣‡	45.87	17.62	41.57	39.52	17.85
Pretrain-finetune based Models					
PEGASUS (Zhang et al., 2020a)	44.21	16.95	38.83	–	–
reproduced by Zaheer et al. (2020)	43.85	16.83	39.17	–	–
reproduced by us ‡	43.78	16.76	39.14	38.93	16.06
BigBird + PEGASUS (Zaheer et al., 2020) ♣	46.63	19.02	41.77	–	–
reproduced by us ‡	46.98	19.32	42.10	42.38	19.15
DANCER + PEGASUS (Gidiotis and Tsoumakas, 2020) ♣	45.01	17.60	40.56	–	–
reproduced by us ‡	44.84	18.29	40.42	40.24	17.17
BART (Lewis et al., 2020) ‡	44.17	17.43	39.68	40.37	16.91
MCS + BART (Manakul and Gales, 2021) ♣	47.68	19.77	42.25	–	–
DYLE + RoBERTa + BART (Mao et al., 2022) ♣	46.41	17.95	41.54	–	–
LED + BART (Beltagy et al., 2020) ♣‡	46.94	19.71	42.33	41.64	19.41
UOTSumm + BART (Our method) ♣‡					
full implementation	**48.87**	**20.34**	**44.61**	**43.26**	**20.99**
simple alignment	48.67	20.19	44.42	43.06	20.43
simple alignment & w/o ASSC	46.24	19.09	42.26	41.73	19.82
w/o trigram blocking	48.74	20.28	44.49	43.17	20.87

In this table, Tables 4 and 5, we use R-1, R-2, R-L, BERT-S, and Mover-S as the abbreviations for ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, and MoverScore, respectively. The boldface type indicates that the model achieves the best performance in terms of the corresponding evaluation metric. To facilitate a fair comparison, the group of pretrain-finetune-based models is separately presented in the bottom part of each table.

(a) On *arXiv* and *PubMed*, UOTSumm finetuned from BART outperforms all these baselines in terms of all the evaluation metrics. To investigate the statistical significance of the comparison, we further conduct the following experiment. We use the stratified random sampling (Noreen, 1989) to sample document-summary pairs from the testing set of *arXiv* and *PubMed*. We create the subgroups (a.k.a. strata) based on the sentence number of ground-truth summary. Concretely, three subgroups are specified: a subgroup with short summaries, a subgroup with medium-length summaries, and a subgroup with long summaries. We use proportionate sampling to get 1000 document-summary pairs from each subgroup, and get 3000 document-summary pairs in total from each testing dataset. We calculate statistical significance level based on the bootstrap test. On both *arXiv* and *PubMed*, BART-based UOTSumm is statistically significantly better than any pretrain-finetune-based baseline model for all the evaluation metrics, with the statistical significance level $p < 0.05$.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 4. Results on the test set of *PubMed*

Model	ROUGE			BERT-S	Mover-S
	R-1	R-2	R-L		
Seq2Seq (Chopra et al., 2016)	31.55	8.52	27.38	–	–
PG-Net (See et al., 2017)	35.86	10.22	29.69	–	–
Discourse-Aware (Cohan et al., 2018) ♣	38.93	15.37	35.21	–	–
Ext + TLM (Pilault et al., 2020) ♣	42.13	16.27	39.21	–	–
RSSR (Chen and Bansal, 2018) ♣‡	44.68	17.98	41.22	42.06	19.42
DANCER + PG-Net (Gidiotis and Tsoumakas, 2020) ♣	44.09	17.69	40.27	–	–
UOTSumm + PG-Net (Our method) ♣‡	45.06	18.03	41.44	42.23	20.26
Pretrain-finetune based Models					
PEGASUS (Zhang et al., 2020a)	45.97	20.15	41.34	–	–
reproduced by Zaheer et al. (2020)	44.53	19.30	40.70	–	–
reproduced by us ‡	44.51	19.07	40.79	43.08	19.78
BigBird + PEGASUS (Zaheer et al., 2020) ♣	46.32	20.65	42.33	–	–
reproduced by us ‡	46.34	20.36	42.50	45.09	21.74
DANCER + PEGASUS (Gidiotis and Tsoumakas, 2020) ♣	46.34	19.97	42.42	–	–
reproduced by us ‡	45.92	19.89	42.08	44.13	21.55
BART (Lewis et al., 2020) ‡	44.61	19.37	41.01	43.45	20.08
MCS + BART (Manakul and Gales, 2021) ♣	46.49	19.45	42.04	–	–
LED + BART (Beltagy et al., 2020) ♣‡	47.04	20.03	42.93	44.34	22.40
UOTSumm + BART (Our method) ♣‡					
full implementation	**48.43**	**20.99**	**44.95**	46.55	**23.90**
simple alignment	47.55	20.64	44.22	45.83	23.49
simple alignment & w/o ASSC	47.16	20.45	43.80	45.52	23.09
w/o trigram blocking	48.41	20.93	44.90	**46.56**	23.84

The boldface type indicates that the model achieves the best performance in terms of the corresponding evaluation metric.

(b) On *GovReport*, UOTSumm finetuned from BART is comparable with the baseline model DYLE and outperforms all the other baseline models. It should be highlighted that DYLE is the state-of-the-art model on *GovReport*. It inherits knowledge from the powerful pre-trained model RoBERTa besides BART, while our method only utilizes knowledge from BART. Hence the comparison between UOTSumm and DYLE is not fair, and it favors DYLE. As shown in Table 1, the average number of sections and summary sentences is large for *GovReport*. Hence, the performance of UOTSumm demonstrates that it is a suitable choice in this setting.

4. Consider the ablation model *simple alignment*, its results are close to the full implementation of UOTSumm. Roughly speaking, *simple alignment* is a good simplification of UOTSumm. However, some details need to be investigated. Besides the way of aligning summary sentences to sections, another difference between *simple alignment* and UOTSumm is: the labels for training ASSC modules are different. *Simple alignment* uses $\boldsymbol{\varphi}^*$ of integer values as labels, while UOTSumm uses $\mathbf{P}^*\mathbf{1}_n$ which allows continuous values. The latter one is more close to the real world. Because it accommodates two situations discussed

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 5. Results on the test set of *GovReport*

Model	ROUGE			BERT-S	Mover-S
	R-1	R-2	R-L		
RSSR (Chen and Bansal, 2018) ♣‡	49.11	19.04	46.85	42.16	24.37
UOTSumm + PG-Net (Our method) ♣‡	57.55	23.99	54.94	45.88	28.35
Pretrain-finetune based Models					
PEGASUS (Zhang et al., 2020a) ‡	51.33	18.69	49.11	43.65	25.43
BigBird + PEGASUS (Zaheer et al., 2020) ♣‡					
finetune `bigbird-pegasus-large-pubmed`	36.64	9.09	35.61	24.82	7.89
finetune `bigbird-pegasus-large-arxiv`	41.65	9.10	37.57	28.77	14.88
finetune `bigbird-pegasus-large-bigpatent`	50.13	15.93	47.48	40.07	23.18
BART (Lewis et al., 2020) ‡	52.24	22.09	49.99	46.36	27.65
Stride Patterns + BART (Child et al., 2019) ♣	54.29	20.80	51.35	–	–
LSH + BART (Kitaev et al., 2019) ♣	54.75	21.36	51.27	–	–
Sinkhorn Attention + BART (Tay et al., 2020) ♣	55.45	21.45	52.48	–	–
Hepos + LSH + BART (Huang et al., 2021) ♣	55.00	21.13	51.67	–	–
Hepos + Sinkhorn Attention + BART (Huang et al., 2021) ♣	56.86	22.62	53.82	–	–
DYLE + RoBERTa + BART (Mao et al., 2022) ♣	**61.01**	**28.83**	57.82	–	–
LED + BART (Beltagy et al., 2020) ♣‡	57.32	25.77	54.78	47.92	29.39
UOTSumm + BART (Our method) ♣‡					
full implementation	60.51	27.22	**58.12**	48.77	**31.02**
simple alignment	60.28	26.88	57.89	48.68	31.00
simple alignment & w/o ASSC	56.76	24.70	54.56	46.91	28.99
w/o trigram blocking	60.39	27.15	58.01	**48.78**	30.95

The boldface type indicates that the model achieves the best performance in terms of the corresponding evaluation metric.

in Section 4.1: one sentence summarizes content from several different sections, and one summary sentence is based on the overlapping information of several different sections. We manually checked some documents from three benchmark datasets. We found that the first situation happens infrequently, while the second situation is rather common. In research papers, it is often the case that the content of one summary sentence appears in the "*introduction*" section, the "*conclusion*" section, and some other sections. Case 2 in Table 2 is one typical example. In this case, the summary sentence contributes close scores (e.g., 0.33, 0.33, and 0.34) to scalar components corresponding to Sections 1–3 in $\mathbf{P}^*\mathbf{1}_n$, In contrast, the summary sentence contributes one-hot scores (e.g., 0.0, 0.0, and 1.0) to scalar components corresponding to Sections 1–3 in $\boldsymbol{\varphi}^*$. Then, the ASSC module of *simple alignment* is trained with inaccurate labels. Because any of three sections can serve as the source of the summary sentence, but only one section is chosen. We conjecture this is the reason why the performance of UOTSumm is slightly better than *simple alignment*. Besides, the formulation of UOTSumm is explainable from the viewpoint of OT, which is rather graceful. To sum up, the full implementation of UOTSumm is more advantageous than *simple alignment*.

5. Consider the second ablation model. When ASSC module is removed, the performance of *simple alignment* degrades obviously, especially on *GovReport*. This observation

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

demonstrates the importance of ASSC module and can be easily explained. Without ASSC module, the same number of tokens are generated for different sections at inference stage. In practice, the lengths of summaries for different sections cannot be always the same. Another obvious advantage of ASSC module is: the number of generated sentences are automatically learned from data, which avoids human's effort to choose a threshold.

6. In most cases, *trigram blocking* improves scores of evaluation metrics for UOTSumm on three datasets, but the improvements are minute. This observation suggests that sentence repetition is not a severe problem for UOTSumm. One very interesting phenomenon is: on *PubMed* and *GovReport*, *trigram blocking* improves scores of ROUGE and MoverScore, while slightly degrades scores of BERTScore. Currently, only ROUGE scores are reported in most summarization papers. This phenomenon suggests that we should keep alert to the performance gain brought by *trigram blocking*: does it really reduce the semantic repetition, or just improve ROUGE scores?
7. The results of BigBird on *GovReport* are strange, which are explained as follows. The model size of BigBird is too large, it cannot be normally finetuned on one NVIDIA RTX A6000 GPU even when batch size is set to 1. Hence we freeze the encoder, and only tune the decoder. Besides, Huggingface website does not provide a pre-trained model from general domains. It only provide models that are finetuned on three datasets: *arXiv*, *PubMed*, and *BigPatent* (Sharma, Li, and Wang, 2019). We finetuned these three versions of BigBird on *GovReport*, and report their results. The results show that the domain of *GovReport* is closer to *BigPatent*, than *arXiv* or *PubMed*. Besides, if we can get a GPU of larger memory size, the BigBird baseline is expected to get better results on *GovReport*.
8. It should be highlighted that UOTSumm is a universal framework for LDS task, which can be applicable to any existing NATS model. It consistently improves performance when combined with PG-Net or BART. When combined with a more powerful NATS model, UOTSumm is expected to bring some further performance gain. We leave this topic as the future work.

5.4 Case studies and human evaluation

In this part, we investigate some practical cases to show the advantages of UOTSumm at training stage. First, we compare the ability of S2SS alignment between UOTSumm and ROUGE at training stage. For ROUGE-based S2SS alignment, we follow the practice of DANCER (Gidiotis and Tsoumakas, 2020). Specifically, for each sentence $\mathbf{y}$ from the summary $\{\mathbf{y}_j\}_{j=1}^{n}$, we compute the ROUGE-L precision between $\mathbf{y}$ and each document sentence $\mathbf{x}$ from $\left\{\{\mathbf{x}_k\}_{k=1}^{\ell_i}\right\}_{i=1}^{m}$:

$$\text{ROUGE-L}_{\text{precision}}(\mathbf{x},\mathbf{y}) = \frac{\text{LCS}(\mathbf{x},\mathbf{y})}{\text{length}(\mathbf{x})}, \tag{12}$$

where $\text{LCS}(\mathbf{x},\mathbf{y})$ is the length of the longest common sub-sequence (LCS) between $\mathbf{x}$ and $\mathbf{y}$. Then, summary sentence $\mathbf{y}$ is aligned to section $\mathbf{s}_i = \{\mathbf{x}_k\}_{k=1}^{\ell_i}$, which contains the highest scored sentence $\mathbf{x}$. For our method, we utilize the well-trained UOTSumm model. We freeze its model parameters and execute several steps, i.e., from Step (3) to Step (7), of Algorithm 1. Then, each summary sentence $\mathbf{y}$ is aligned to one section $\mathbf{s}_i$ by UOTSumm. We choose three document-summary pairs from the training sets of *arXiv*, *PubMed*, and *GovReport*, and execute the above two S2SS alignment procedures. Limited by space, we select one representative summary sentence for each case. ROUGE-based S2SS alignment method explicitly conducts the sentence-to-sentence alignment, thus we present the aligned document sentences and corresponding scores of ROUGE-L precision. Since UOTSumm directly conducts section-to-sentence alignment, we manually judge the

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 6. Cases to demonstrate the differences in text alignment between UOTSumm and ROUGE-L precision

Case 1	Source dataset: *PubMed*.
summary sentence **y**:	
One patient was admitted to the intensive care unit and received a platelet transfusion.	
source document sentence aligned by ROUGE-L precision:	
he was admitted and treated with prophylactic antibiotics and g csf.	(ROUGE-L precision: 27.27)
heading of the aligned section: Case 2	
source document section aligned by UOTSumm:	
... A 62-year-old woman, previously diagnosed with hypertension, was admitted to the emergency department due to desquamation of the hands and feet. ... After 4 pints of platelet infusion and continuous g-csf infusion, her blood cell count improved without any other infective, renal, or cardiovascular complications. ...	
heading of the aligned section: Case 1	
Case 2	Source dataset: *arXiv*.
summary sentence **y**:	
We identify the impact of network characteristics and website infrastructure on spdy s potential page loading benefits, finding that these factors are decisive for spdy and its optimal deployment strategy.	
source document sentence aligned by ROUGE-L precision:	
Finally, we inspect the impact of packet loss on spdy s performance.	(ROUGE-L precision: 58.33)
heading of the aligned section: Effect of network performance	
source document section aligned by UOTSumm:	
... Motivated by this, we perform a large body of controlled experiments in our local testbed to understand the reasons behind these performance variations. ... We identify the website types and network characteristics that spdy thrives under, as well as how these benefits vary based on provider-side infrastructural decisions. ...	
heading of the aligned section: Introduction	
Case 3	Source dataset: *GovReport*.
summary sentence **y**:	
To identify considerations for developing a physician feedback system, GAO reviewed the literature and interviewed officials from health plans and specialty societies.	
source document sentence aligned by ROUGE-L precision:	
($\mathbf{x}_{ROUGE}$:) CMS found the attention in our report to considerations for developing a physician feedback system to be particularly helpful.	(ROUGE-L precision: 42.11)
heading of the aligned section: CMS Comments	
source document section aligned by UOTSumm:	
... ($\mathbf{x}_{UOTSumm}$:) Through our review of selected literature and interviews with officials of health insurance companies, specialty societies, and profiling experts, we identified several key considerations in developing reports to provide feedback to physicians on their performance, including their per capita resource use. ...	(ROUGE-L precision: 14.63)
heading of the aligned section: Research literature, health insurers, and specialists identified considerations in developing physician feedback reports on resource use	

We use the yellow background to highlight the longest common sub-sequence of the summary sentence and the sentence aligned by ROUGE-L precision.

semantically related sentences from the aligned section and present them. For both methods, we also present the headings of the aligned sections. The results are presented in Table 6, from which we can draw the following conclusions.

For all the cases, UOTSumm and ROUGE-based method align the summary sentence to the different sections. UOTSumm correctly conducts S2SS alignment, while the ROUGE-based

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 7. Section types and corresponding common keywords

Section	Keywords
introduction	introduction, case
literature	background, literature, related
methods	method(s), techniques, methodology
results	result(s), experimental, experiments
conclusion	conclusion(s), concluding, discussion, limitations

The contents of this table are taken from Gidiotis and Tsoumakas (2020).

method aligns the summary sentence to a wrong source sentence. This observation suggests that ROUGE-based text alignment does not correlate with human judgment in some situations. For ROUGE-L precision in Formula (12), the denominator is the length of sentence $\mathbf{x}$. Hence, the ROUGE-based method is prone to select a shorter sentence as long as it has a common sub-sequence with the summary sentence. It cannot detect two pieces of texts adopting different words while preserving the same meanings. In contrast, as shown in Formula (9), UOTSumm relies on the neural architectures for text alignment, which is good at understanding literally different paraphrases and disturbed word order. In the next, we discuss Case 3 in more detail. We use $\mathbf{x}_{\text{ROUGE}}$ and $\mathbf{x}_{\text{UOTSumm}}$ to denote the sentences aligned by ROUGE-based method and by UOTSumm, respectively. The correctly aligned sentence $\mathbf{x}_{\text{UOTSumm}}$ gets a lower score: $\text{ROUGE-L}_{\text{precision}}(\mathbf{x}_{\text{UOTSumm}}, \mathbf{y}) = 14.63$. In contrast, the wrongly aligned sentence $\mathbf{x}_{\text{ROUGE}}$ gets a higher score: $\text{ROUGE-L}_{\text{precision}}(\mathbf{x}_{\text{ROUGE}}, \mathbf{y}) = 42.11$. This is because $\mathbf{x}_{\text{ROUGE}}$ and $\mathbf{y}$ share a long common sub-sequence, although the rest parts of two sentences are totally irrelevant. For $\mathbf{x}_{\text{UOTSumm}}$ and $\mathbf{y}$, we use blue and purple fonts to highlight the clauses that are semantically equivalent. Apparently, $\mathbf{x}_{\text{UOTSumm}}$ and $\mathbf{y}$ swap two primary clauses. Swapping the order of two clauses while preserving the sentence meaning is a very common linguistic phenomenon. However, it greatly degrades the ROUGE-L precision which strictly relies on the word order. To sum up, these cases show that correctly conducting S2SS alignment is one reason that UOTSumm outperforms DANCER, since UOTSumm is supervised by a less-biased target.

In this part, we investigate one case to show the advantages of UOTSumm at inference stage. We choose one document from the test set of *PubMed*, and utilize the trained UOTSumm model to generate its summary. In Table 8, we present the section headings of this document, one summary sentence generated by UOTSumm, and the related ground-truth sentence. In brackets after the heading names, we list the numbers of generated sentences for the corresponding sections, which is computed by Step (5) of Algorithm 3. The sentence generated by UOTSumm well captures the meaning of one ground-truth sentence, while it is generated from the section with heading "web page." As mentioned in Section 4.2, at inference stage, DANCER relies on a heuristic of heading matching to decide which section should be adopted for summary generation. As a reference, we replicate their heuristic matching method in Table 7. Except the heading "introduction," the other headings of this document cannot be matched with any section type in Table 7. Hence, the section with heading "web page" will never be adopted by DANCER for summary generation. This case demonstrates that since UOTSumm does not rely on the heading but directly learns the number of generated sentences for each section from the content, it can utilize any document section for summary generation at inference stage.

In this part, we analyze Case 1 in Table 2, in which UOTSumm does not work very well. We use UOTSumm-based method to conduct S2SS alignment for this case. Sections 1–3 get alignment scores of 0.04, 0.92, and 0.03, respectively. All the other sections get an alignment score of 0.01 in total. UOTSumm-based method successfully aligns the summary sentence to Sections 1–3, which are indeed the source of this summary sentence. However, the distribution of alignment

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Table 8. A case of text generation by UOTSumm

section headings of the document:
introduction (2), automated update of patterns (1), new functional prediction tool (1), web page (1), improvement of the profile method construction (2), how to obtain prosite (0)
one summary sentence generated by UOTSumm from the source section with the heading "web page":
The prosite website was redesigned and new predictive tools were implemented to assign more detailed functional information to the scanned proteins.
the ground-truth sentence:
During the last 2 years, the documentation and the scan prosite web pages were redesigned to add more functionalities.

scores obviously violates human judgment. We infer the reason as follows. For UOTSumm-based method, S2SS alignment is mainly decided by loss values of a NATS model. Since the word sequences "theoretical status" and "phenomenological applications" are both very short when compared with length of the summary sentence, current NATS models are prone to predict the summary sentence with some large loss values.

The above cases qualitatively demonstrate the advantages and weaknesses of UOTSumm. To quantitatively investigate the quality of S2SS alignment learned by UOTSumm, we design a human evaluation to compare it with the alignment decided by ROUGE. We recruit two annotators to judge S2SS alignment. Documents are randomly chosen from three benchmark datasets for annotation. If one summary sentence can be aligned to more than one sections, only the most suitable alignment is recorded. If two annotators disagree on certain alignment, then this document is discarded. Finally, we get 10 annotated documents from each of *PubMed*, *arXiv*, and *GovReport*. We treat the S2SS alignment annotated by humans as the ground truth. We use UOTSumm-based method and ROUGE-based method to produce S2SS alignments for these documents. Then, for each method, we count the number of overlapping alignment pairs with the ground truth and compute the precision of correctly predicted alignment pairs. ROUGE-based method gets the precision scores of 84.5%, 86.7%, and 87.6%, on *PubMed*, *arXiv*, and *GovReport*, respectively. UOTSumm-based method gets the precision scores of 88.7%, 91.6%, and 90.6%, on *PubMed*, *arXiv*, and *GovReport*, respectively. This human evaluation quantitatively shows that UOTSumm-based method achieves better S2SS alignment than ROUGE-based method.

6. Conclusion

In this paper, we propose UOTSumm, a novel framework for LDS task. UOTSumm belongs to the DANCER approach, which summarizes each section of a long document separately. UOTSumm includes a joint training objective, which is formulated as a UOT problem. Under a unified framework, UOTSumm jointly learns the optimal S2SS alignment, a section-level NATS summarizer, and the number of aligned summary sentences for each section. UOTSumm is universal enough and can be easily combined with most existing NATS models. We implement UOTSumm with two popular NATS models: PG-Net and BART and evaluate them on three public LDS benchmarks: *PubMed*, *arXiv*, and *GovReport*. UOTSumm outperforms its counterparts that utilize ROUGE for text alignment. This finding validates that although ROUGE is a long-standing workhorse of text alignment at the training stage, directly learning S2SS alignment from data brings a remarkable performance gain. When combined with UOTSumm, the improved PG-Net and BART also outperforms their respective vanilla models by a large margin. Besides, different from the related baseline which relies on the section heading at the inference stage, UOTSumm can be directly applied to any type of long documents as long as they are organized in paragraphs. Since UOT

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

demonstrates its effectiveness in learning text alignment from data directly, as the future work, we will explore UOT formulations in the other NLP settings that involve text alignment.

Acknowledgements. The work described in this paper is substantially supported by a grant (LOGITSCO) from the Asian Institute of Supply Chains and Logistics, the Chinese University of Hong Kong. Shumin Ma acknowledges the support from: Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, BNU-HKBU United International College (2022B1212010006), Guangdong Higher Education Upgrading Plan (2021–2025) of "Rushing to the Top, Making Up Shortcomings and Strengthening Special Features" with UIC research grant (R0400001-22) and UIC (UICR0700019-22).

Competing interests. The authors declare none.

References

Alvarez-Melis D. and Jaakkola T. (2018). *Gromov-Wasserstein alignment of word embedding spaces*. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium: Association for Computational Linguistics, pp. 1881–1890.

Bao J., **Basu P.**, **Dean M.**, **Partridge C.**, **Swami A.**, **Leland W. and Hendler J. A.** (2011). Towards a theory of semantic communication, *2011 IEEE Network Science Workshop*. IEEE, pp. 110–117.

Beltagy I., **Peters M. E. and Cohan A.** (2020). Longformer: The long-document transformer.

Bezdek J. C. and Hathaway R. J. (2002). *Some notes on alternating optimization*. AFSS International Conference on Fuzzy Systems, Springer, pp. 288–300.

Chen L., **Bai K.**, **Tao C.**, **Zhang Y.**, **Wang G.**, **Wang W.**, **Henao R. and Carin L.** (2020a). *Sequence generation with optimal-transport-enhanced reinforcement learning*. Proceedings of the AAAI Conference on Artificial Intelligence, **34**, pp. 7512–7520.

Chen L., **Zhang Y.**, **Zhang R.**, **Tao C.**, **Gan Z.**, **Zhang H.**, **Li B.**, **Shen D.**, **Chen C. and Carin L.** (2019). *Improving sequence-to-sequence learning via optimal transport*. 7th International Conference on Learning Representations, New Orleans, LA, USA: OpenReview.net.

Chen Y., **Lan Y.**, **Xiong R.**, **Pang L.**, **Ma Z. and Cheng X.** (2020b). *Evaluating natural language generation via unbalanced optimal transport*. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, pp. 3730–3736,

Chen Y.-C. and Bansal M. (2018). *Fast abstractive summarization with reinforce-selected sentence rewriting*. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia: Association for Computational Linguistics, pp. 675–686.

Child R., **Gray S.**, **Radford A. and Sutskever I.** (2019). Generating long sequences with sparse transformers, arXiv preprint arXiv: 1904.10509.

Chizat L., **Peyré G.**, **Schmitzer B. and Vialard F.-X.** (2015). Unbalanced optimal transport: Geometry and kantorovich formulation, arXiv preprint arXiv: 1508.05216.

Chizat L., **Peyré G.**, **Schmitzer B. and Vialard F.-X.** (2018). Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation* **87**(314), 2563–2609.

Chopra S., **Auli M. and Rush A. M.** (2016). *Abstractive sentence summarization with attentive recurrent neural networks*. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California: Association for Computational Linguistics, pp. 93–98.

Clark E., **Celikyilmaz A. and Smith N. A.** (2019). *Sentence mover's similarity: Automatic evaluation for multi-sentence texts*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, pp. 2748–2760.

Cohan A., **Dernoncourt F.**, **Kim D. S.**, **Bui T.**, **Kim S.**, **Chang W. and Goharian N.** (2018). *A discourse-aware attention model for abstractive summarization of long documents*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA: Association for Computational Linguistics, pp. 615–621.

Cuturi M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In Burges C., Bottou L., Welling M., Ghahramani Z. and Weinberger K., (eds), *Advances in Neural Information Processing Systems*, **26**, Curran Associates, Inc.

Dai Z., **Yang Z.**, **Yang Y.**, **Carbonell J.**, **Le Q. and Salakhutdinov R.** (2019). *Transformer-XL: Attentive language models beyond a fixed-length context*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, pp. 2978–2988.

Devlin J., **Chang M.-W.**, **Lee K. and Toutanova K.** (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN: Association for Computational Linguistics, pp. 4171–4186.

Fabbri A. R., Kryściński W., McCann B., Xiong C., Socher R. and Radev D. (2021). SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* **9**, 391–409.

Frogner C., Zhang C., Mobahi H., Araya M. and Poggio T. A. (2015). Learning with a wasserstein loss. In Cortes C., Lawrence N., Lee D., Sugiyama M. and Garnett R., (eds), *Advances in Neural Information Processing Systems*, **28**, Curran Associates, Inc, pp. 2053–2061.

Gambhir M. and Gupta V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review* **47**(1), 1–66.

Gehrmann S., Deng Y. and Rush A. (2018). *Bottom-up abstractive summarization*. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium: Association for Computational Linguistics, pp. 4098–4109.

Gidiotis A. and Tsoumakas G. (2020). A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 3029–3040.

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

Gu J., Lu Z., Li H. and Li V. O. (2016). *Incorporating copying mechanism in sequence-to-sequence learning*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany: Association for Computational Linguistics, pp. 1631–1640.

Hermann K. M., Kocisky T., Grefenstette E., Espeholt L., Kay W., Suleyman M. and Blunsom P. (2015). Teaching machines to read and comprehend, *Advances in Neural Information Processing Systems*. **28**, Curran Associates, Inc, pp. 1693–1701.

Hochreiter S. and Schmidhuber J. (1997). Long short-term memory. *Neural Computation* **9**(8), 1735–1780.

Huang L., Cao S., Parulian N., Ji H. and Wang L. (2021). *Efficient attentions for long document summarization*. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 1419–1436.

Jing H. and McKeown K. R. (1999). *The decomposition of human-written summary sentences*. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99, New York, NY, USA: Association for Computing Machinery, pp. 129–136.

Kantorovich L. V. (1942). On the translocation of masses, *Doklady Akademii Naukokl USSR (NS)*, **37**, 199, 201, (in Russian).

Kantorovich L. V. (2006). On the translocation of masses. *Journal of Mathematical Sciences* **133**(4), 1381–1382 (English translation).

Khandelwal U., He H., Qi P. and Jurafsky D. (2018). *Sharp nearby, fuzzy far away: How neural language models use context*. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia: Association for Computational Linguistics, pp. 284–294.

Kingma D. P. and Ba J. (2014). Adam: A method for stochastic optimization.

Kitaev N., Kaiser L. and Levskaya A. (2019). *Reformer: The efficient transformer*. 8th International Conference on Learning Representations, Addis Ababa, Ethiopia: OpenReview.net.

Koh H. Y., Ju J., Liu M. and Pan S. (2022). An empirical survey on long document summarization: Datasets, models and metrics. *ACM Computing Surveys* **55**(8), pp.1–35.

Kolouri S., Park S. R., Thorpe M., Slepcev D. and Rohde G. K. (2017). Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine* **34**(4), 43–59.

Kornilova A. and Eidelman V. (2019). *BillSum: A corpus for automatic summarization of US legislation*. Proceedings of the 2nd Workshop on New Frontiers in Summarization, Hong Kong, China: Association for Computational Linguistics, pp. 48–56.

Kryscinski W., Keskar N. S., McCann B., Xiong C. and Socher R. (2019). *Neural text summarization: A critical evaluation*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics, pp. 540–551.

Kusner M., Sun Y., Kolkin N. and Weinberger K. (2015). *From word embeddings to document distances*. Proceedings of the 32nd International Conference on Machine Learning, Lille, France: PMLR, pp. 957–966.

Lewis M., Liu Y., Goyal N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V. and Zettlemoyer L. (2020). *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 7871–7880.

Liero M., Mielke A. and Savaré G. (2018). Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones Mathematicae* **211**(3), 969–1117.

https://doi.org/10.1017/S1351324923000177 Published online by Cambridge University Press

Lin C.-Y. (2004). Rouge: A package for automatic evaluation of summaries, *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, pp. 74–81.

Liu P. J., **Saleh M.**, **Pot E.**, **Goodrich B.**, **Sepassi R.**, **Kaiser L. and Shazeer N.** (2018). *Generating wikipedia by summarizing long sequences*. 6th International Conference on Learning Representations, Vancouver, BC, Canada: OpenReview.net.

Liu Y. and Lapata M. (2019). *Text summarization with pretrained encoders*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics, pp. 3730–3740.

Liu Y., **Ott M.**, **Goyal N.**, **Du J.**, **Joshi M.**, **Chen D.**, **Levy O.**, **Lewis M.**, **Zettlemoyer L. and Stoyanov V.** (2019). Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692.

Manakul P. and Gales M. (2021). *Long-span summarization via local attention and content selection*. 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), Association for Computational Linguistics, pp. 6026–6041.

Mani I. (1999). *Advances in Automatic Text Summarization*. Cambridge, Massachusetts, USA: MIT Press.

Mao Z., **Wu C. H.**, **Ni A.**, **Zhang Y.**, **Zhang R.**, **Yu T.**, **Deb B.**, **Zhu C.**, **Awadallah A. and Radev D.** (2022). *DYLE: Dynamic latent extraction for abstractive long-input summarization*. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland: Association for Computational Linguistics, pp. 1687–1698.

Monge G. (1781). Mémoire sur la théorie des déblais et des remblais, *Histoire de l'Académie Royale des Sciences de Paris, Paris, France, Imprimerie royale.*

Nallapati R., **Zhai F. and Zhou B.** (2017). *Summarunner: a recurrent neural network based sequence model for extractive summarization of documents*. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, pp. 3075–3081.

Nallapati R., **Zhou B.**, **dos Santos C.**, **Gulçehre Ç. and Xiang B.** (2016). *Abstractive text summarization using sequence-to-sequence RNNs and beyond*. Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany: Association for Computational Linguistics, pp. 280–290.

Narayan S., **Cohen S. B. and Lapata M.** (2018). *Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization*. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium: Association for Computational Linguistics, pp. 1797–1807.

Noreen E. W. (1989). *Computer-Intensive Methods for Testing Hypotheses*, New York, Wiley.

Paszke A., **Gross S.**, **Massa F.**, **Lerer A.**, **Bradbury J.**, **Chanan G.**, **Killeen T.**, **Lin Z.**, **Gimelshein N.**, **Antiga L.**, **Desmaison A.**, **Kopf A.**, **Yang E.**, **DeVito Z.**, **Raison M.**, **Tejani A.**, **Chilamkurthy S.**, **Steiner B.**, **Fang L.**, **Bai J. and Chintala S.** (2019). Pytorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems*, **32**, Curran Associates, Inc, pp. 8026–8037.

Paulus R., **Xiong C. and Socher R.** (2018). *A deep reinforced model for abstractive summarization*. 6th International Conference on Learning Representations, Vancouver, BC, Canada: OpenReview.net.

Peyrard M. (2019). *A simple theoretical model of importance for summarization*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, pp. 1059–1073.

Peyré G. and Cuturi M. (2019). Computational optimal transport. *Foundations and Trends in Machine Learning* **11**(5-6), 355–607.

Pilault J., **Li R.**, **Subramanian S. and Pal C.** (2020). *On extractive and abstractive neural document summarization with transformer language models*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 9308–9319.

Raffel C., **Shazeer N.**, **Roberts A.**, **Lee K.**, **Narang S.**, **Matena M.**, **Zhou Y.**, **Li W. and Liu P. J.** (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* **21**, 1–67.

Rush A. M., **Chopra S. and Weston J.** (2015). *A neural attention model for abstractive sentence summarization*. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal: Association for Computational Linguistics, pp. 379–389.

Schmitzer B. (2019). Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing* **41**(3), A1443–A1481.

See A., **Liu P. J. and Manning C. D.** (2017). *Get to the point: Summarization with pointer-generator networks*. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada: Association for Computational Linguistics, pp. 1073–1083.

Shannon C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal* **27**(3), 379–423.

Sharma E., **Li C. and Wang L.** (2019). *BIGPATENT: A large-scale dataset for abstractive and coherent summarization*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, pp. 2204–2213.

Shi T., **Keneshloo Y.**, **Ramakrishnan N. and Reddy C. K.** (2021). Neural abstractive text summarization with sequence-to-sequence models. *ACM/IMS Transactions on Data Science* **2**(1), 1–37.

Sutskever I., **Vinyals O. and Le Q. V.** (2014). Sequence to sequence learning with neural networks. In Ghahramani Z., Welling M., Cortes C., Lawrence N. and Weinberger K., (eds), *Advances in Neural Information Processing Systems*, **27**, Curran Associates, Inc.

Swanson K., **Yu L. and Lei T.** (2020). *Rationalizing text matching: Learning sparse alignments via optimal transport*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 5609–5626.

Tay Y., **Bahri D.**, **Yang L.**, **Metzler D. and Juan D.-C.** (2020). Sparse Sinkhorn attention. In III H. D. and Singh A., (eds), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of Proceedings of Machine Learning Research, PMLR, 9438, 9447.

Tay Y., **Dehghani M.**, **Bahri D. and Metzler D.** (2022). Efficient transformers: A survey. *ACM Computing Surveys* **55**(6), pp. 1–28.

Tu Z., **Lu Z.**, **Liu Y.**, **Liu X. and Li H.** (2016). *Modeling coverage for neural machine translation*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany: Association for Computational Linguistics, pp. 76–85.

Vaswani A., **Shazeer N.**, **Parmar N.**, **Uszkoreit J.**, **Jones L.**, **Gomez A. N.**, **Kaiser L.u and Polosukhin I.** (2017). Attention is all you need. In Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S. and Garnett R., (eds), *Advances in Neural Information Processing Systems*, **30**, Curran Associates, Inc.

Villani C. (2008). *Optimal Transport: Old and New*, **338**. Berlin, Germany: Springer Science & Business Media.

Webber B. and Joshi A. (2012). *Discourse structure and computation: Past, present and future*. Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, Jeju Island, Korea, Association for Computational Linguistics, pp. 42–54.

Xu J., **Zhou H.**, **Gan C.**, **Zheng Z. and Li L.** (2021). *Vocabulary learning via optimal transport for neural machine translation*. 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), Association for Computational Linguistics, pp. 7361–7373.

Yokoi S., **Takahashi R.**, **Akama R.**, **Suzuki J. and Inui K.** (2020). *Word rotator's distance*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 2944–2960.

Zaheer M., **Guruganesh G.**, **Dubey K. A.**, **Ainslie J.**, **Alberti C.**, **Ontanon S.**, **Pham P.**, **Ravula A.**, **Wang Q.**, **Yang L. and Ahmed A.** (2020). Big bird: Transformers for longer sequences. In Larochelle H., Ranzato M., Hadsell R., Balcan M. F. and Lin H., (eds), *Advances in Neural Information Processing Systems*, **33**, Curran Associates, Inc, pp. 17283–17297.

Zhang H., **Gong Y.**, **Shen Y.**, **Li W.**, **Lv J.**, **Duan N. and Chen W.** (2021). Poolingformer: Long document modeling with pooling attention. In Meila M. and Zhang T., (eds), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual Event*, volume 139 of Proceedings of Machine Learning Research, PMLR, pp. 12437–12446,

Zhang J., **Zhao Y.**, **Saleh M. and Liu P.** (2020a). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In III H. D. and Singh A., (eds), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of Proceedings of Machine Learning Research, PMLR, pp. 11328–11339.

Zhang M., **Liu Y.**, **Luan H. and Sun M.** (2017). *Earth mover's distance minimization for unsupervised bilingual lexicon induction*. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark: Association for Computational Linguistics, pp. 1934–1945.

Zhang T., **Kishore V.**, **Wu F.**, **Weinberger K. Q. and Artzi Y.** (2020b). *Bertscore: Evaluating text generation with bert*. 8th International Conference on Learning Representations, Addis Ababa, Ethiopia: OpenReview.net.

Zhao W., **Peyrard M.**, **Liu F.**, **Gao Y.**, **Meyer C. M. and Eger S.** (2019). *MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics, pp. 563–578.

Zhao Y., **Saleh M. and Liu P. J.** (2020). Seal: Segment-wise extractive-abstractive long-form text summarization.

Cite this article: Shen X, Lam W, Ma S and Wang H (2024). Joint learning of text alignment and abstractive summarization for long documents via unbalanced optimal transport. *Natural Language Engineering* **30**, 525–553. https://doi.org/10.1017/S1351324923000177