EDU-level Extractive Summarization with Varying Summary Lengths

Yuping Wu, Ching-Hsun Tseng, Jiayu Shang, Shengzhong Mao,
Goran Nenadic, Xiao-Jun Zeng*
Department of Computer Science, University of Manchester
{yuping.wu-2, ching-hsun.tseng, jiayu.shang,
shengzhong.mao}@postgrad.manchester.ac.uk
gnenadic, x.zeng@manchester.ac.uk

Abstract

Extractive models usually formulate text summarization as extracting fixed top-k salient sentences from the document as a summary. Few works exploited extracting finer-grained Elementary Discourse Unit (EDU) with little analysis and justification for the extractive unit selection. Further, the selection strategy of the fixed top-k salient sentences fits the summarization need poorly, as the number of salient sentences in different documents varies and therefore a common or best k does not exist in reality. To fill these gaps, this paper first conducts the comparison analysis of oracle summaries based on EDUs and sentences, which provides evidence from both theoretical and experimental perspectives to justify and quantify that EDUs make summaries with higher automatic evaluation scores than sentences. Then, considering this merit of EDUs, this paper further proposes an **EDU**-level extractive model with **V**arying summary **L**engths (EDU-VL[1]) and develops the corresponding learning algorithm. EDU-VL learns to encode and predict probabilities of EDUs in the document, generate multiple candidate summaries with varying lengths based on various k values, and encode and score candidate summaries, in an end-to-end training manner. Finally, EDU-VL is experimented on single and multi-document benchmark datasets and shows improved performances on ROUGE scores in comparison with state-of-the-art extractive models, and further human evaluation suggests that EDU-constituent summaries maintain good grammaticality and readability.

1 Introduction

Automatic text summarization aims at aggregating information in long document(s) into a shorter piece of text while keeping important information. Extractive summarization and abstractive summarization are two categories of it. This paper focuses only on the extractive task which formulates summarization as identifying salient textual segments in document (Lunh, 1958). Under the supervised learning framework, this task is further formulated as a label classification task, i.e., encoding textual segments and predicting labels on the encoded vectors. Recent state-of-the-art models (Liu and Lapata, 2019; Zhong et al., 2020; Liu et al., 2021; Ruan et al., 2022) on this task tend to be Transformer-based since BERT (Devlin et al., 2019) shows significantly better performance than RNN on most natural language understanding tasks.

Document: (...) [The second audio,] [taken from dash cam video from inside a patrol car,] [captures a phone call between Slager and someone] [CNN believes] [is his wife.] (...)
Reference Summary: The second audio captures a phone call between Slager and someone CNN believes is his wife.

Table 1: Example to demonstrate redundant information in sentence. Content within [] indicates an EDU.

Most existing works extract sentences from the document and some works further (Xu and Durrett, 2019) propose post-processing steps to prune the generated summary. The only exception is the few works (Liu and Chen, 2019; Huang and Kurohashi, 2021), which extract finer-grained textual segments, i.e., discourse-level text or EDU, with little justification. The intuition is that a sentence consisting of multiple clauses is inevitable to contain less important information. As demonstrated in Table 1, partially removing a clause in the sentence is conducive to generating a summary. Certainly, such an intuitive explanation does not provide enough evidence and support to justify the use of finer-grained textual segments such as EDU to substitute sentences. Considering such a gap in existing research, the first main motivation of this paper is to propose and conduct the comparison analysis be-

*Corresponding author.
[1] https://github.com/yuping-wu/EDU-VL

Findings of the Association for Computational Linguistics: EACL 2023, pages 1655–1667
May 2-6, 2023

tween sentences and EDUs to disclose and justify whether using EDU is a theoretically advanced and application-advantaged extractive unit.

When selecting textual segments, the top-k strategy with k fixed for all documents is dominant in deciding the length of the generated summary. Some works (Zhong et al., 2020; Chen et al., 2021) manage to output summaries with different lengths, i.e., various numbers of extracted segments, via formulating the problem as deriving a subset of sentences from the combination of top-k sentences. Due to the foreseeing explosion of the combination of sentences to form subsets, these approaches are limited to generating summaries with relatively small values of k. To overcome such a weakness, the second main motivation of this paper is to propose and develop an approach allowing varying lengths for extractive summarization without explicit limitation on the maximum value of k, i.e., the maximum length.

Following the above motivations, the comparison analysis between EDUs and sentences ascertains that EDU is a better text unit for the extractive task because EDU-level summaries achieve higher automatic evaluation scores than sentence-level summaries. This conclusion is justified from two perspectives. Theoretically, a formal theorem about this conclusion could be derived from the property that EDU is essentially part of a sentence. Experimentally, results of comprehensive analysis about oracle summaries of five datasets further quantify this conclusion, i.e., how much the ROUGE scores of EDU-level oracle summary are higher than sentence-level oracle summary.

Based on the aforementioned conclusion and foundation, this paper further proposes and develops an EDU-level extractive model and algorithm, which generates summaries with varying lengths, i.e., EDU-VL. We extend Transformer-based pre-trained language model with an extra classification layer to encode EDUs in a document and predict the corresponding probabilities. Multiple k values are provided to the model to generate a set of candidate summaries under the flexible top-k strategy for the document. Multiple Transformer encoder layers encode the full document and candidate summaries individually. Finally, a similarity score with the encoded document is calculated for each candidate summary and the one with the highest score is the final output of EDU-VL.

Experiments are conducted on five benchmark datasets from different domains and with various writing styles. The experimental results suggest that EDU-VL achieves better performance than all state-of-the-art extractive baselines on single-document summarization datasets CNN/DailyMail, XSum, Reddit, and WikiHow, in terms of three ROUGE metrics. With direct comparison to the multi-document model, EDU-VL still achieves comparable performance on the multi-document summarization dataset Multi-News. Human evaluation is further carried for the summaries generated by EDU-VL to assess the syntax structure of EDU-constituent summaries. The results provide evidence for the good grammaticality and readability of EDU-constituent summaries and therefore justify the applicability.

The contributions of this paper are threefold:

1) We justify and quantify that EDU-level achieves higher automatic evaluation scores than sentence-level oracle summary from both theoretical and experimental perspectives, indicating that setting EDU as the extractive text unit is exploitable and superior in applications.

2) We propose a varying summary lengths-enabled extractive model with EDU-level text unit. Such a model and its learning algorithm encodes EDUs in a document and outputs a summary with varying length by making k in the top-k extraction strategy varying.

3) Our proposed model achieves superior performance on four single-document summarization datasets on three ROUGE metrics. Human evaluations show that the generated EDU-constituent summaries maintain good grammaticality and readability.

2 Related Work

2.1 Neural Extractive Summarization

The extractive text summarization task aims at extracting salient textual segments from the original document(s) as a summary. A tendency observed among extractive neural models is that the architecture changes from RNN (Nallapati et al., 2017; Xu and Durrett, 2019) to Transformer-based models, e.g., BERT (Zhang et al., 2019; Liu and Lapata, 2019) and Longformer (Liu et al., 2021; Ruan et al., 2022). GNN also gained extensive attention in recent years and is usually stacked after

an RNN (Wang et al., 2020; Jing et al., 2021) or Transformer-based encoder (Cui et al., 2020; Kwon et al., 2021) to supplement graph-based features. Some research works integrated neural networks with reinforcement learning (Dong et al., 2018; Gu et al., 2022) or unsupervised learning frameworks (Liang et al., 2021). In general, it can be said that taking a pre-trained Transformer-based language model as the starting point to encode textual segments in a document is currently the state-of-the-art approach among neural extractive models. Therefore, the Transformer-based models, i.e., RoBERTa (Liu et al., 2019) and BART (Lewis et al., 2020), are used as the basic building blocks in this paper.

2.2 Sub-sentential Extractive Summarization

Most previous works about the extractive task focused on generating sentence-level summaries, though some of them (Xiao et al., 2020; Cho et al., 2020; Ernst et al., 2022) utilized sub-sentential features. Early works by Marcu (1999); Alonso i Alemany and Fuentes Fort (2003); Yoshida et al. (2014); Li et al. (2016) exploited extracting discourse-level textual segments as the summary but those approaches were tested on small datasets. More recent works by Liu and Chen (2019); Xu et al. (2020); Huang and Kurohashi (2021) were evaluated on relatively larger datasets. However, whether the discourse-level textual segments are a better alternative than sentences as the extractive text unit was not justified in those works. To fill this gap, we provide justification for this research question from both theoretical and experimental perspectives in this paper.

2.3 Flexible Extractive Summarization

Extractive summarization task is usually formulated as extracting the top-k number of salient textual segments from a document. The fixed k value for all documents results in the lack of variety in the length of the generated summary. Few works (Jia et al., 2020; Zhong et al., 2020; Chen et al., 2021) managed to output summaries with varying lengths. However, either it requires extra effort for hyper-parameter searching on validation dataset to find a valid threshold, or formulating the problem as selecting a subset of top-k sentences makes the variety of lengths limited to small lengths due to the explosive nature of combination. In this paper, we propose a model with varying k values but without explicit limitation on the length or the need to do hyper-parameter searching.

3 Oracle Analysis of EDUs and Sentences

Oracle analysis refers to the analysis of oracle summary whose definition is stated in Section 3.1. We conducted oracle analysis from both theoretical and experimental perspectives to justify and quantify that discourse-level summary achieves higher scores on automatic evaluation metrics than sentence-level summary.

3.1 Theoretical Formulation

Elementary Discourse Unit (EDU), the discourse-level textual segment in this paper, refers to the terminal node in the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) tree which describes the discourse structure of a piece of text. EDUs are non-overlapping and adjacent text spans in the piece of text and a single EDU is essentially a segment of a complete sentence, i.e., the sentence itself or a clause in the sentence (Zeldes et al., 2019). Namely, a sentence can always be expressed with multiple EDUs, i.e., for the s-th sentence in a document, there is $sent_s = [edu_{s_1}, \ldots, edu_{s_m}]$. Consequently, a one-way property from sentence to EDU regarding expressiveness is derived.

Expressiveness Property For any given subset of sentences in a document, i.e., $[sent_i, \ldots, sent_j, \ldots, sent_k]$, there is always a subset of EDUs in the document, i.e., $[edu_{i_1}, \ldots, edu_{i_m}, \ldots, edu_{j_1}, \ldots, edu_{j_m}, \ldots, edu_{k_1}, \ldots, edu_{k_m}]$, having identical content.

Oracle Summary The set of salient textual segments that have greedily the highest ROUGE score(s) with the reference summary is the oracle summary for a document. It signifies the upper bound of performance that an extractive summarization model could achieve on ROUGE metrics.

Denote the sentence-level oracle summary as $\mathcal{OS}_{sent}$ and the EDU-level oracle summary as $\mathcal{OS}_{edu}$. Based on the aforementioned property and definition, Theorem 1 can be derived and its detailed proof is provided below.

Theorem 1. *Given a document $\mathcal{D}$ and its reference summary $\mathcal{R}$, for any derived $\mathcal{OS}_{sent}$, there is always an $\mathcal{OS}_{edu}$ having $ROUGE_{F_l}(\mathcal{R}, \mathcal{OS}_{edu}) \geq ROUGE_{F_l}(\mathcal{R}, \mathcal{OS}_{sent})$.*

Proof. For ROUGE-N, let f_n be a function that generates the set of n-grams for the string s and g be a function that calculates the number of overlapping elements between two sets x and y,

i.e.,

$$f_n(s) = n\text{-}gram(s),$$
$$g(x, y) = match(x, y).$$

The recall and precision formulas of the ROUGE-N metric between the reference summary $\mathcal{R}$ and sentence-level oracle summary $\mathcal{OS}_{sent}$ are

$$\text{R-N}_{\text{recall, }\mathcal{OS}_{sent}} = \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{R})|},$$
$$\text{R-N}_{\text{precision, }\mathcal{OS}_{sent}} = \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{OS}_{sent})|}.$$

There is always an EDU-level summary $\mathcal{S}_{edu}$ having $\mathcal{S}_{edu} = \mathcal{OS}_{sent}$. Let $\mathcal{S}^{sub}_{edu}$ be the subset of EDUs in $\mathcal{S}_{edu}$ having equivalent number of overlapping n-grams as $\mathcal{S}_{edu}$, i.e.,

$$\mathcal{S}^{sub}_{edu} \subseteq \mathcal{S}_{edu} = \mathcal{OS}_{sent}$$

and

$$g(f_n(\mathcal{R}), f_n(\mathcal{S}^{sub}_{edu})) = g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent})).$$

The number of words in $\mathcal{S}^{sub}_{edu}$ is smaller than or equal to the number of words in $\mathcal{OS}_{sent}$, i.e.,

$$|\mathcal{S}^{sub}_{edu}| \leq |\mathcal{OS}_{sent}|,$$

and consequently, the number of n-grams is correspondingly smaller or equal, i.e.,

$$|f_n(\mathcal{S}^{sub}_{edu})| \leq |f_n(\mathcal{OS}_{sent})|.$$

Therefore, the precision score for $\mathcal{S}^{sub}_{edu}$ is larger than or equal to $\mathcal{OS}_{sent}$ and their recall scores are the same, i.e.,

$$\text{R-N}_{\text{precision, }\mathcal{S}^{sub}_{edu}} = \frac{g(f_n(\mathcal{R}), f_n(\mathcal{S}^{sub}_{edu}))}{|f_n(\mathcal{S}^{sub}_{edu})|} \geq \text{R-N}_{\text{precision, }\mathcal{OS}_{sent}} = \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{OS}_{sent})|}$$

and

$$\text{R-N}_{\text{recall, }\mathcal{S}^{sub}_{edu}} = \text{R-N}_{\text{recall, }\mathcal{OS}_{sent}}$$

Therefore, the EDU-level subset of $\mathcal{OS}_{sent}$, i.e., $\mathcal{S}^{sub}_{edu}$, is found to have higher or equal F1-scores on ROUGE-N metrics than $\mathcal{OS}_{sent}$, i.e.,

$$\text{R-N}_{\text{F}_1,\mathcal{S}^{sub}_{edu}} \geq \text{R-N}_{\text{F}_1,\mathcal{OS}_{sent}}$$

That is to say, it is guaranteed to have an EDU-level summary having higher or equal R-N scores than $\mathcal{OS}_{sent}$. By taking this $\mathcal{S}^{sub}_{edu}$ as $\mathcal{OS}_{edu}$, we have $\text{R-N}_{\text{F}_1,\mathcal{OS}_{edu}} \geq \text{R-N}_{\text{F}_1,\mathcal{OS}_{sent}}$.

A similar proof process can be conducted on ROUGE-L. Therefore, for any $\mathcal{OS}_{sent}$, there is always an $\mathcal{OS}_{edu}$ having $\text{ROUGE}_{\text{F}_1}(\mathcal{R}, \mathcal{OS}_{edu}) \geq \text{ROUGE}_{\text{F}_1}(\mathcal{R}, \mathcal{OS}_{sent})$.

□

Text Unit	R-1	R-2	R-L
CNN/DailyMail			
Sentence	53.33	31.09	49.67
EDU	**61.02**	**37.16**	**58.63**
XSum			
Sentence	29.13	8.70	22.32
EDU	**36.07**	**11.74**	**30.95**
WikiHow			
Sentence	37.98	13.76	35.18
EDU	**44.28**	**17.94**	**42.56**
Reddit			
Sentence	30.58	10.95	24.57
EDU	**40.62**	**16.01**	**35.95**
Multi-News			
Sentence	49.65	22.20	44.99
EDU	**51.35**	**23.99**	**48.70**

Table 2: ROUGE F1-scores of sentence-level and EDU-level oracle summaries on training datasets.

3.2 Empirical Justification

Five datasets from different domains were analyzed from the experimental perspective and experimental settings are listed in Appendix A. Table 2 presents the ROUGE scores of $\mathcal{OS}_{sent}$ and $\mathcal{OS}_{edu}$ on training datasets. $\mathcal{OS}_{edu}$ gains significantly higher ROUGE scores on all datasets. Larger improvements are observed on ROUGE-1 (6.3-10.04) and ROUGE-L (7.38-11.38) on the majority of datasets, and improvement on ROUGE-2 is smaller but there is still an increase.

Figure 1 shows the comparison of breakdown ROUGE scores between two text units on the CNN/DailyMail training dataset and details about other datasets could be found in Appendix B. Recall scores on all three metrics are approximately equal between the two text units, suggesting that the amount of salient information in both is equal. However, precision scores are observed with a significantly higher value on $\mathcal{OS}_{edu}$, suggesting the length of $\mathcal{OS}_{edu}$ is smaller.

The experimental results quantify the potential gains that EDU-level oracle summary could achieve on five datasets and the breakdown scores indicate that EDU-level oracle summary is less redundant than sentence-level oracle summary.

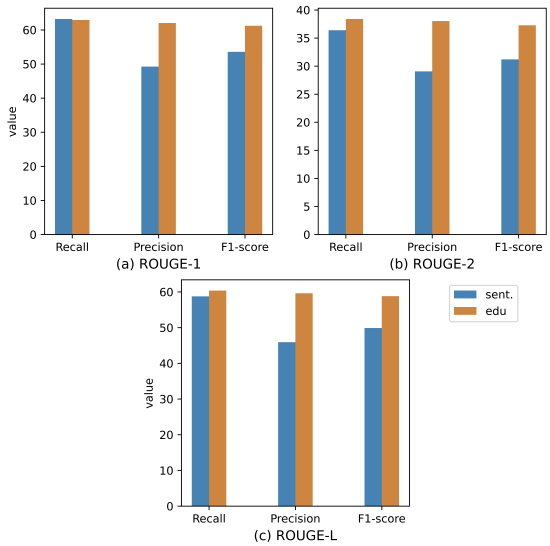

Figure 1: Breakdown ROUGE scores of sentence/EDU-level oracle summaries on CNN/DM training dataset.

4 EDU-level Extractive Model with Varying Summary Lengths

4.1 Problem Formulation

Suppose a document $\mathcal{D}$ consists of m EDUs, i.e., $\mathcal{D} = [edu_1, \ldots, edu_m]$, the i-th EDU consists of n_i words, i.e., $edu_i = [w_{i1}, \ldots, w_{in_i}]$, and the reference summary wrote by human is denoted as $\mathcal{R}$. The set of ground truth labels for each EDU could be derived from $\mathcal{R}$, i.e., $L = [l_1, \ldots, l_m]$, via a greedy algorithm as previous works did. Our proposed model aims to generate a summary via selecting one summary from the set of candidate summaries $\mathcal{C}$ where $\mathcal{C} = [cand_1, \ldots, cand_c]$ and $cand_j$ consists of EDUs with top-k_j probabilities that are also predicted by the proposed model.

4.2 Model

Figure 2 illustrates the architecture of our proposed model. From bottom to top, firstly, the EDU-level block generates a representation vector and probability for each EDU in a document. Secondly, the candidate summary generator aggregates EDU representation vectors to generate several candidate summaries with varying lengths by specifying different k values. Different from the previous top-k strategy where k is a fixed value, multiple k values are provided to the proposed model, allowing different numbers of EDUs being extracted to form different candidate summaries with varying lengths for the same document. Lastly, the document-level block encodes each candidate summary and selects one of the candidate summaries as the final model output. In this way, the proposed model decides the most suitable summary length, i.e., k, for each document.

EDU-level Block Given input document $\mathcal{D} = [w_{11}, \ldots, w_{mn_m}]$ where w_{ij} denotes j-th word in i-th EDU, [CLS] and [SEP] tokens are inserted into $\mathcal{D}$ at the start and end of each EDU. We adapt the pre-trained Transformer-based language model (PLM) as the EDU encoder, e.g., RoBERTa. The hidden states of [CLS] tokens derived from the PLM are taken as EDU representations, i.e., $\mathbf{edu^E}$ in Equation (1). A classification layer is further applied on EDU representations to predict probabilities, i.e., $\mathbf{P}$ in Equation (2).

$$[edu_1^E, \ldots, edu_m^E] = \mathbf{PLM}_\theta(\mathcal{D}) \quad (1)$$

$$P_i(y_i = 1) = \sigma(\mathbf{W}^c edu_i^E + \mathbf{b}^c), \quad (2)$$

where θ is the set of all trainable parameters in PLM; $\mathbf{W}^c$ and $\mathbf{b}^c$ are trainable parameters in classification layer, and $\sigma(\cdot)$ denotes sigmoid function.

Candidate Summary Generator Given a predefined extraction lengths set $\mathcal{K} = [k_1, \ldots, k_c]$, the s-th candidate summary, $cand_s$, consists of EDUs whose probabilities are in top-$k_s(\mathbf{P})$, i.e., $[edu_{i_1}, \ldots, edu_{i_j}, \ldots, edu_{i_k}]$ where $i_j \leq m$ and $P_{i_j} \in$ top-$k_s(\mathbf{P}), j = 1, 2, ..., k_s$. The initial representation vector, $cand_s^C$, for $cand_s$ is the concatenation of representation vectors of EDUs in it. The initial document representation vector, $\mathcal{D}^C$, is aggregated from the representation vectors of all EDUs.

Document-level Block Multiple Transformer encoder layers (MTL) are stacked to encode document-level information for document $\mathcal{D}^C$, and all candidate summaries, e.g., $cand_s^C$, separately, and generate $\mathcal{D}^D$ and $cand_s^D$ in Equation (3). Then cosine similarity, i.e., sim_s in Equation (4), is computed between the encoded document representation and the encoded s-th candidate summary representation. The candidate summary with the highest similarity with the document is taken as the final model-generated summary.

$$[\mathcal{D}^D, cand_s^D] = [\mathbf{MTL}_\eta(\mathcal{D}^C), \mathbf{MTL}_\eta(cand_s^C)] \quad (3)$$

$$sim_s = \mathbf{cosine}(\mathcal{D}^D, cand_s^D), \quad (4)$$

where η is the set of trainable parameters in MTL.

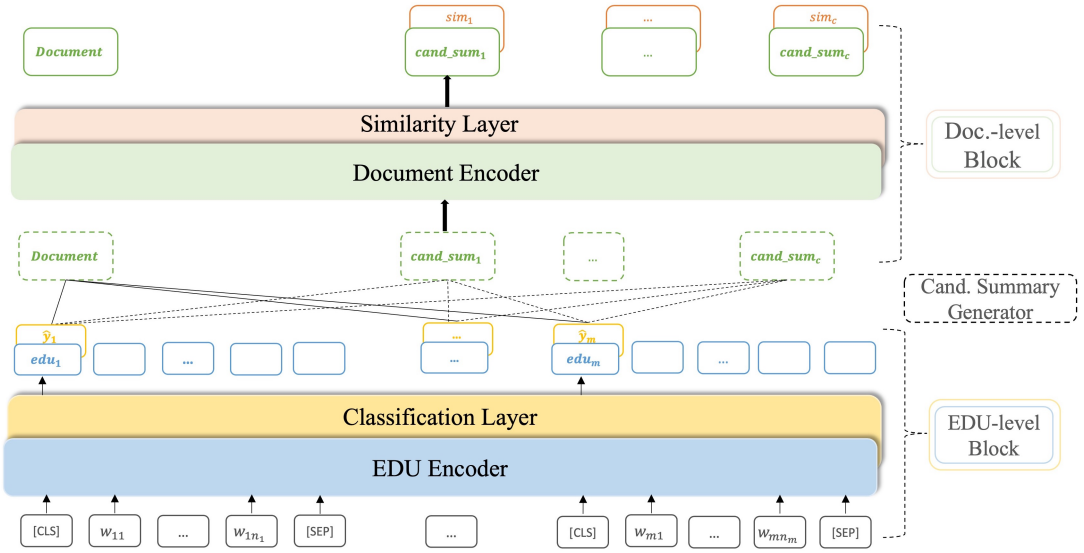

Figure 2: Model architecture. The EDU-level block encodes and predicts a probability value for each EDU in the input document; The candidate summary generator generates a set of candidate summaries based on the predicted probability values; The document-level block encodes the whole document and candidate summaries and generates similarity values between them. The final output is the candidate summary with the highest similarity score.

Training Algorithm 1 summarizes the model learning procedure. The model encodes EDUs in the document and predicts the probability for each EDU (lines 1-2), generates indices of EDUs for candidate summaries with different lengths which are derived from different k values (lines 3-4), encodes the whole document and candidate summaries and calculates similarity scores (lines 7-10), and selects the best candidate summary (line 16) in an end-to-end manner. Inspired by Zhong et al. (2020) that the candidate summary having a higher ROUGE score with the reference summary is expected to have a higher similarity score with the whole document, during training, ROUGE scores for each $(\mathcal{R}, cand_s)$ pair are calculated and used to sort the set $\mathcal{C}$ in descending order (lines 5-6) to align with the loss function in Equation (7). Besides, to better emphasize those important EDUs, the EDU-level oracle summary, denoted as $cand_{gt}$ here, is introduced to the training process and assumed to have the highest ROUGE score (lines 12-13).

Algorithm 1 Model Learning Algorithm

Input: $\mathcal{D}|_1^m, \mathcal{K}|_1^c, L|_1^m$
Output: candSumIdx

1: eduRep$|_1^m \leftarrow PLM_\theta(\mathcal{D})$
2: $P|_1^m \leftarrow classification_{w,b}$(eduRep$|_1^m$)
3: **for** $i \leftarrow 1$ to c **do**
4: selIdx$_i \leftarrow$ indices of top-$\mathcal{K}_i(P|_1^m)$
5: **if** training **then**
6: selIdx$|_1^c \leftarrow$ sort based on ROUGE scores
7: docRep $\leftarrow MTL_\eta$(eduRep$|_1^m$)
8: **for** $j \leftarrow 1$ to c **do**
9: candRep$_j \leftarrow MTL_\eta$(eduRep$_{\in \text{selIdx}_j}$)
10: sim$_j \leftarrow cosine$(docRep, candRep$_j$)
11: **if** training **then**
12: gtIdx $\leftarrow$ indices of 1 in $L|_1^m$
13: sim$_{gt} \leftarrow$ repeat 9-10
14: $\mathcal{L} \leftarrow$ loss from $P|_1^m, L|_1^m, sim|_1^c, sim_{gt}$
15: $\theta, w, b, \eta \leftarrow$ parameters updated by $\mathcal{L}$
16: candSumIdx $\leftarrow$ selIdx$_{\text{index_}max(\text{sim}|_1^c)}$
17: **return** candSumIdx

4.3 Objective Function

Binary cross entropy is calculated on the outputs of the classification layer in the EDU-level block, as in Equation (6). Contrastive learning loss is calculated on the outputs of the similarity layer in the document-level block, as in Equations (7-9). The final training loss $\mathcal{L}$ in Equation (5) is calculated as a weighted summation between them.

$$\mathcal{L} = \mathcal{L}_{bce} + \rho * \mathcal{L}_{con}, \quad (5)$$

where

$$\mathcal{L}_{bce} = -\sum_{i=1}^{m}(l_i log(P_i) + (1-l_i) log(1-P_i)) \quad (6)$$

$$\mathcal{L}_{con} = \mathcal{L}_1 + \mathcal{L}_2, \quad (7)$$

where

$$\mathcal{L}_1 = \sum_{s=1}^{c} max(0, sim_s - sim_{gt} + \gamma_1) \quad (8)$$

$$\mathcal{L}_2 = \sum_{i<j}^{c} max(0, sim_j - sim_i + (j-i) * \gamma_2) \quad (9)$$

5 Experiments

5.1 Datasets

CNN/DailyMail (Hermann et al., 2015) is the most commonly used news dataset for the extractive task with human-written highlights as reference summary. The non-anonymized version was used in our experiments. **XSum** (Narayan et al., 2018) is another news dataset with the first introductory sentence in the article as the reference summary. **Reddit** (Kim et al., 2019) is a dataset crawled from the social media forum with the content in the section TL;DR as the reference summary. Experiments were conducted on the TIFU-long version. **WikiHow** (Koupaee and Wang, 2018) is a dataset crawled from the question-answering website with the first sentence in each paragraph as the reference summary. **Multi-News** (Fabbri et al., 2019) is a multi-document dataset with one summary for a cluster of documents. We follow Zhong et al.'s (2020) setting to split Reddit and Multi-News datasets and concatenate multiple documents into one single document. The detailed statistics of the five datasets in our experiments can be found in Appendix C.

5.2 Baselines

Various extractive models are selected as baselines. **HETFORMER** (Liu et al., 2021) modifies Longformer with longer input lengths to implement multi-granularity attention and selects sentences. Among models generating summaries with varying lengths, **MATCHSUM** (Zhong et al., 2020) selects among a set of candidate summaries derived from a trained sentence-level extractive model; **HAHSUM** (Jia et al., 2020) transforms a document into a heterogeneous hierarchical graph and flexibly selects sentences based on a threshold. Among models with sub-sentential segments as input, the **Proposed** model by Huang and Kurohashi (2021) is another Longformer-based model but extracts EDUs based on the constructed heterogeneous graph; **DISCOBERT** (Xu et al., 2020) and **D-SUM** (Liu and Chen, 2019) are models extracting discourse-level textual segments but they differ in whether integrating GNN into the model. **SGSUM** (Chen et al., 2021) is a multi-document model by encoding all documents within one cluster individually and selecting the best sub-graph. **FAR** (Liang et al., 2021) is an unsupervised ranking model considering facet-specific information.

5.3 Experimental Setting

EDU segmentation of sentences in the document is conducted by NeuralEDUSegmentation[2] (Wang et al., 2018). To facilitate the training process, the calculation of ROUGE scores is avoided by pre-selecting the set of candidate summaries based on the predicted probabilities by the fine-tuned RoBERTa on the extractive task for each dataset. The pre-trained "roberta-base" or "bart-base" is adapted as the EDU encoder and enlarged to handle the first 768 BPEs of each document. The number of Transformer encoder layers is 4 by default. Following Liu and Lapata (2019), a similar greedy algorithm is applied to generate ground truth labels for EDUs (also for oracle summaries in Section 3.2) and the pseudo-code is in Appendix D. The trigram strategy is applied when forming the final EDU-constituent summary during validating and testing.

We follow Zhong et al.'s (2020) setting to set up $\gamma_1 = 0$ and $\gamma_2 = 0.01$. ρ is set as 100 based on our observation during training. Adam optimizer is used. The batch size is 5 to fit the GPU memory limit during training and 60 during validating or testing. Every 6k steps are defined as one epoch; the training process could take up to 100 epochs and early stopping is activated with patience as 10 epochs and R-2 as the metric. Experiments are conducted on a single Nvidia-v100-16GB GPU. The F1-scores of ROUGE-1/2/L[3] (Lin, 2004) are taken as the automatic evaluation metrics. More details are provided in Appendix E.

5.4 Experimental Results

CNN/DailyMail Table 3 shows the results. The top section includes F1-scores of oracle summaries and the Lead-3 method. The second section presents the F1-scores reported in the original papers of all baselines. The last section lists the F1-scores of our proposed model.

Our proposed model outperforms the unsupervised baseline, FAR, by a large margin, aligning with the observation from other supervised

[2] https://github.com/PKU-TANGENT/NeuralEDUSeg
[3] https://github.com/bheinzerling/pyrouge

Model	R-1	R-2	R-L
ORACLE (EDU)	62.50	38.67	60.16
ORACLE (sentence)	55.31	32.73	51.63
LEAD-3 (sentence)	39.96	17.39	36.27
D-SUM (Liu and Chen, 2019)	42.78	20.23	-
DISCOBERT (Xu et al., 2020)	43.77	20.85	40.67
Proposed (Huang and Kurohashi, 2021)	43.61	20.81	41.12
HAHSUM (Jia et al., 2020)	44.68	21.30	40.75
MATCHSUM (Zhong et al., 2020)	44.41	20.86	40.55
HETFORMER (Liu et al., 2021)	44.55	20.82	40.37
FAR (Liang et al., 2021)	40.83	17.85	36.91
EDU-VL$_{\text{RoBERTa}}$	**44.80**	**21.66**	**42.56**
EUD-VL$_{\text{BART}}$	44.70	21.63	42.46

Table 3: F1-scores on CNN/DailyMail test dataset.

Model	R-1	R-2	R-L
XSum			
ORACLE (EDU)	36.16	11.74	31.02
ORACLE (sentence)	29.11	8.66	22.29
LEAD-3 (sentence)	19.41	2.65	15.05
MATCHSUM (Zhong et al., 2020)	24.86	4.66	18.41
EDU-VL$_{\text{RoBERTa}}$	**26.48**	5.74	22.33
EDU-VL$_{\text{BART}}$	26.43	**5.78**	**22.35**
Reddit			
ORACLE (EDU)	44.49	18.53	38.87
ORACLE (sentence)	34.36	12.97	26.98
LEAD-3 (sentence)	18.39	3.01	14.12
MATCHSUM (Zhong et al., 2020)	25.09	6.17	20.13
EUD-VL$_{\text{RoBERTa}}$	**27.04**	6.87	22.64
EDU-VL$_{\text{BART}}$	27.01	**7.06**	**22.70**

Table 4: F1-score results on test dataset of XSum and Reddit. The number of Transformer encoder layers in BART version of XSum is 6 and 2 for both versions of Reddit.

baselines. Compared with discourse-level baselines, i.e., D-SUM, DISCOBERT and Proposed, our proposed model achieves an improvement of at least 1.03/0.81/1.44 on R-1/2/L. When compared against other two varying lengths-enabled models, i.e., HAHSUM and MATCHSUM, our proposed model achieves better R-1 result on a small scale (0.12) and R-2/L on a larger scale (0.8/1.81). Our proposed model also beats HETFORMER which allows longer input length by a similar scale pattern. It is observed that the RoBERTa version of our proposed model performs slightly better than the BART version. The experimental results suggest that our proposed model achieves better performance than all baselines on the R-1/2/L.

XSum and Reddit The results in Table 4 show that our proposed model outperforms the baseline model, MATCHSUM, by a large margin on all three metrics (1.57/1.12/3.94 and 1.92/0.89/2.57 on R-1/2/L for XSum and Reddit, respectively). The RoBERTa version of our model only achieves slightly better result on R-1 than the BART version.

Model	R-1	R-2	R-L
WikiHow			
ORACLE (EDU)	44.13	17.90	42.38
ORACLE (sentence)	37.89	13.80	35.13
LEAD-3 (sentence)	23.97	5.37	22.22
FAR (Liang et al., 2021)	27.54	6.17	25.46
MATCHSUM (Zhong et al., 2020)	31.85	8.98	29.58
EDU-VL$_{\text{RoBERTa}}$	33.94	10.31	32.55
EDU-VL$_{\text{BART}}$	**34.01**	**10.45**	**32.66**
Multi-News			
ORACLE (EDU)	51.60	24.24	48.92
ORACLE (sentence)	49.87	22.43	45.18
LEAD-3 (sentence)	28.40	8.63	24.93
HETFORMER (Liu et al., 2021)	46.21	17.49	42.43
SGSUM (Chen et al., 2021)	47.53	**18.75**	43.31
FAR (Liang et al., 2021)	43.48	16.87	44.00
MATCHSUM (Zhong et al., 2020)	46.20	16.51	41.89
EDU-VL$_{\text{RoBERTa}}$	46.82	17.05	44.36
EDU-VL$_{\text{BART}}$	**47.56**	17.64	**45.05**

Table 5: F1-score results on test dataset of WikiHow and Multi-News.

Model	R-1	R-2	R-L
EDU-VL$_{\text{RoBERTa}}$	**44.80**	**21.66**	**42.56**
w/o EDU	43.89	20.79	40.18
w/o VL	44.32	21.38	42.12

Table 6: Ablation analysis on test dataset of CNN/DM.

WikiHow and Multi-News As shown in Table 5, our proposed model achieves significantly better performance on WikiHow dataset, beating both MATCHSUM and FAR by at least 2.16/1.47/3.08 on R-1/2/L. For the Multi-News dataset, our proposed model outperforms HETFORMER, MATCHSUM and FAR. It is noteworthy that SGSUM is initially designed to incorporate multiple documents, meaning that its input document is more complete than ours. Though our proposed model underperforms SGSUM on R-2, our proposed model achieves comparable result on R-1 and better result on R-L. The BART version of our proposed model outperforms the RoBERTa version on all three metrics on both datasets. To sum up, our proposed model performs better on WikiHow dataset and comparably on Multi-News dataset when compared against the corresponding state-of-the-art baselines.

5.5 Analysis

Ablation Analysis We further conduct ablation analysis by removing specific characteristics in our model and the result is presented in Table 6. Both letting the model extract sentences under the same

architecture and removing the document-block to disable the varying lengths characteristic reduce model performance on all three metrics. A larger decrease is observed in the sentence-level model.

Human Evaluation We randomly sample 50 summaries generated by our model from the CNN/DailyMail test dataset and conduct detailed qualitative analysis. For each summary, we combine EDUs from the same sentence together as one textual segment. Then referring to the dependency tree of the corresponding sentence, we evaluate the syntactical completeness of the extracted textual segment. Out of 221 extracted textual segments in all 50 summaries, 68% are syntactically complete and 32% are not. It is noteworthy that about half of those incomplete ones are subordinate clauses, whose syntax structure is close to being complete. Out of these complete ones, 44.7% are the whole sentence itself because all EDUs in that sentence are extracted; 55.3% maintain complete syntax structure after dropping some EDU(s) in that sentence (as the example shown in Table 7). Therefore, it is safe to believe that even sentences split into multiple EDUs, the model is capable to maintain the syntax structure by choosing multiple EDUs in a sentence and in some cases, filtering out some redundant information without breaking the completeness of the syntax.

Generated Summary Examples Table 7 provides an example of a summary generated by our proposed model, which illustrates that the model manages to selectively drop redundant information in sentences by operating on the EDU-level while maintaining an informative and readable summary.

Document: (...) [*Arnold Breitenbach of St. George wanted to get 'CIB-69' put on a license plate,*]$_{21}$ [the Spectrum newspaper of St. George reported.]$_{22}$ [*That would have commemorated both Breitenbach getting the Purple Heart in 1969 and his Combat Infantryman's Badge,*]$_{31}$ [according to the newspaper.]$_{32}$ (...) [*The Utah DMV denied his request,*]$_{51}$ [*citing state regulations*]$_{52}$ [*prohibiting the use of the number 69*]$_{53}$ [*because of its sexual connotations*]$_{54}$ (...)
Reference Summary: Arnold Breitenbach of St. George, Utah, wanted to get 'CIB-69' put on a license plate. That would have commemorated both Breitenbach getting the Purple Heart in 1969 and his Combat Infantryman's Badge. The Utah DMV denied his request, citing state regulations prohibiting the use of the number 69 because of its sexual connotations.

Table 7: Example from model-generated summary. Content within [] represents an EDU and subscript number ij indicates it is the j-th EDU in the i-th sentence in the document. Each color represents information in a sentence in reference summary. Italic denotes content selected by *our proposed model*.

6 Conclusion

In this paper, we verify and quantify the argument that the EDU-level summary achieves higher automatic evaluation scores than sentence-level summary from both theoretical and experimental perspectives. We further propose an EDU-level extractive summarization model and develop its learning algorithm, which generates summaries with different lengths for different documents. The experimental results demonstrate that our model achieves superior performance on four single-document summarization datasets and comparable performance for multi-document summarization with direct comparison with the multi-document model. In the future, we will explore integrating the EDU-level summary generated by our model into the abstractive summarization model.

Limitations

Though EDU is defined as a clause in a sentence, current EDU segmenters are still underdeveloped due to the limited training dataset and usually split a sentence into consecutive EDUs, which breaks the syntactic structure. Occasionally some extracted EDUs from a sentence fail to recover a complete syntactic structure. Therefore, a more sophisticated segmenter could further improve the segmentation, or some post-processing treatments could be developed to address such a potential issue specifically.

Acknowledgements

We would like to acknowledge the assistance given by Research IT and the use of the Computational Shared Facility at The University of Manchester. We thank the anonymous reviewers for their helpful comments.

References

Laura Alonso i Alemany and Maria Fuentes Fort. 2003. Integrating cohesion and coherence for automatic summarization. In *Proceedings of EACL2003*, page 1.

Moye Chen, Wei Li, Jiachen Liu, Xinyan Xiao, Hua Wu, and Haifeng Wang. 2021. SgSum:transforming multi-document summarization into sub-graph selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4063–4074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sangwoo Cho, Kaiqiang Song, Chen Li, Dong Yu, Hassan Foroosh, and Fei Liu. 2020. Better highlighting: Creating sub-sentence summary highlights. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6282–6300, Online. Association for Computational Linguistics.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.

Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2022. Proposition-Level Clustering for Multi-Document Summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1765–1779, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems*, 2015-Janua:1693–1701.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052, Online. Association for Computational Linguistics.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online. Association for Computational Linguistics.

Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. 2021. Multiplex graph neural network for extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 2519–2531.

Mahnaz Koupaee and William Yang Wang. 2018. WikiHow: A Large Scale Text Summarization Dataset. In *arXiv preprint arXiv:1810.09305*, pages 1–5.

Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, and Manabu Okumura. 2021. Considering nested tree structure in sentence extractive summarization with pre-trained transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4039–4044, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,

pages 7871–7880, Online. Association for Computational Linguistics.

Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. The role of discourse units in near-extractive summarization. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147, Los Angeles. Association for Computational Linguistics.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1685–1697, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Ye Liu, Jianguo Zhang, Yao Wan, Congying Xia, Lifang He, and Philip Yu. 2021. HETFORMER: Heterogeneous transformer with sparse attention for long-text extractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 146–154, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Zhengyuan Liu and Nancy Chen. 2019. Exploiting discourse-level segmentation for extractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 116–121, Hong Kong, China. Association for Computational Linguistics.

H. P. Lunh. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2(2):159–165.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Daniel Marcu. 1999. Discourse trees are good indicators of importance in text. In *Advances in automatic text summarization*, pages 123–136.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 3075–3081.

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

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. HiStruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland. Association for Computational Linguistics.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium. Association for Computational Linguistics.

Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. Do we really need that many parameters in transformer for extractive summarization? discourse can help ! In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134, Online. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839, Doha, Qatar. Association for Computational Linguistics.

Amir Zeldes, Debopam Das, Erick Galani Maziero, Juliano Antonio, and Mikel Iruskieta. 2019. The DISRPT 2019 shared task on elementary discourse unit segmentation and connective detection. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 97–104, Minneapolis, MN. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Parameters for Oracle Summaries

Table 8 presents parameters for oracle summaries.

Dataset	# Sentences	# EDUs
CNN/DM	5	8
XSum	5	8
Reddit	5	8
WikiHow	5	8
Multi-News	15	30

Table 8: Maximum number of textual segments allowed to be extracted in oracle summaries.

B Breakdown Comparison on ROUGE scores

Table 9 presents the breakdown ROUGE scores of other four datasets.

C Statistics of Datasets

Table 10 presents the statistics of the five datasets.

D Greedy Selection Algorithm

Algorithm 2 presents the pseudo-code of the algorithm of selecting salient textual segments, which is used to generate oracle summary and ground truth labels.

E Supplementary Experimental Settings and Results

Table 11 and Table 12 present detailed experimental settings and results, respectively.

	Sentence		EDU	
Metric	recall	precision	recall	precision
XSum				
R-1	40.18	25.77	40.16	36.54
R-2	11.70	7.95	12.86	12.26
R-L	30.68	19.79	34.31	31.44
WikiHow				
R-1	45.28	36.90	44.41	49.25
R-2	16.45	13.44	18.01	19.99
R-L	41.96	34.17	42.71	47.29
Reddit				
R-1	44.70	26.71	45.40	40.39
R-2	15.63	10.02	17.62	16.48
R-L	35.86	21.56	40.19	35.75
Multi-News				
R-1	45.09	58.87	42.45	68.35
R-2	19.96	26.72	19.86	31.79
R-L	40.77	53.44	40.24	64.86

Table 9: Breakdown ROUGE scores of sentence/EDU-level oracle summaries on XSum, WikiHow, Reddit, and Multi-News training datasets.

Dataset	# word	# EDU	# sent.	# EDU/sent.
CNN/DM	733.98	94.25	36.23	2.67
XSum	431.12	52.02	19.76	2.63
Reddit	443.46	65.28	23.44	3.01
WikiHow	581.15	75.72	29.42	2.58
Multi-News	503.33	58.33	18.13	3.35

Table 10: Statistics of datasets. #word, #EDU and #sent. refer to the average number of words, EDUs and sentences, respectively, of documents in the dataset. #EDU/sent. refers to the average number of EDUs per sentence.

Model Statistics		
model	#params	runtime per epoch
EDU-VL$_{\text{RoBERTa}}$	147M	1h 20min
EDU-VL$_{\text{BART}}$	161M	1h 30min
Pre-processing Setting		
dataset	#min	#max
CNN/DM	6	10
XSum	3	7
Reddit	4	8
WikiHow	6	10
Multi-News	27	31

Table 11: Supplementary information of experimental settings. #params refers to the total number of trainable parameters in the model (here both versions are calculated with 4 MTLs). #min and #max refer to the range of lengths (k values in the top-k strategy) of candidate summaries generated by the model, respectively.

Algorithm 2 Greedy Selection Algorithm

Input: Doc, Ref, k ▷ k: # of selections
Output: sel_idx ▷ selected indices

$sel\_idx \leftarrow [\,]$ ▷ empty list
$C \leftarrow [\,]$ ▷ candidate: empty list
while $k \geq 0$ **do**
    end $\leftarrow$ TRUE
    for $i \leftarrow 0$ to $len(Doc)$ **do**
        $tmp\_C \leftarrow C + [Doc_i]$
        $score \leftarrow ROUGE(tmp\_C, Ref)$
        if $score$ increases **then**
            $sel\_idx \leftarrow sel\_idx + [i]$
            $C \leftarrow tmp\_C$
            $k \leftarrow k - 1$
            end $\leftarrow$ FALSE
            break
    if end **then**
        break
return sel_idx

Model	R-1	R-2	R-L
CNN/DM			
EDU-VL$_{\text{RoBERTa}}$	45.45	22.10	43.23
EDU-VL$_{\text{BART}}$	45.29	22.08	41.11
XSum			
EDU-VL$_{\text{RoBERTa}}$	26.58	5.83	22.34
EDU-VL$_{\text{BART}}$	26.66	5.97	22.51
Reddit			
EDU-VL$_{\text{RoBERTa}}$	28.20	7.84	23.58
EDU-VL$_{\text{BART}}$	28.40	7.81	23.89
WikiHow			
EDU-VL$_{\text{RoBERTa}}$	33.90	10.19	32.53
EDU-VL$_{\text{BART}}$	33.95	10.31	32.59
Multi-News			
EDU-VL$_{\text{RoBERTa}}$	46.58	17.00	44.14
EDU-VL$_{\text{BART}}$	47.29	17.49	44.82

Table 12: Experimental results of ROUGE F1-scores on the corresponding validation datasets.