

Unsupervised Extractive Summarization with Learnable Length Control Strategies

Renlong Jie^{1, 2*}, Xiaojun Meng^{1†}, Xin Jiang¹, Qun Liu¹

¹Huawei Noah's Ark Lab

²Northwestern Polytechnical University, China

jierenlong@nwpu.edu.cn, {xiaojun.meng, Jiang.Xin, qun.liu}@huawei.com

Abstract

Unsupervised extractive summarization is an important technique in information extraction and retrieval. Compared with supervised method, it does not require high-quality human-labelled summaries for training and thus can be easily applied for documents with different types, domains or languages. Most of existing unsupervised methods including TextRank and PACSUM rely on graph-based ranking on sentence centrality. However, this scorer can not be directly applied in end-to-end training, and the positional-related prior assumption is often needed for achieving good summaries. In addition, less attention is paid to length-controllable extractor, where users can decide to summarize texts under particular length constraint. This paper introduces an unsupervised extractive summarization model based on a siamese network, for which we develop a trainable bidirectional prediction objective between the selected summary and the original document. Different from the centrality-based ranking methods, our extractive scorer can be trained in an end-to-end manner, with no other requirement of positional assumption. In addition, we introduce a differentiable length control module by approximating 0-1 knapsack solver for end-to-end length-controllable extracting. Experiments show that our unsupervised method largely outperforms the centrality-based baseline using a same sentence encoder. In terms of length control ability, via our trainable knapsack module, the performance consistently outperforms the strong baseline without utilizing end-to-end training. Human evaluation further evidences that our method performs the best among baselines in terms of relevance and consistency.

Introduction

Text summarization aims at producing a short and clear text that expresses the major ideas and important information of an original article. There are two types of summarization methods: 1) *abstractive summarization* directly generates summary texts as a sequence, while 2) *extractive summarization* extracts the most important sentences from the original article to form a summary. As known to all, recent abstractive models like GPT-style large language models have shown an incredible ability in text generation (Zhang et al. 2023).

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

*Work is done as a postdoctoral research fellow of Noah's Ark Lab, Huawei.

†Corresponding authors.

These LLMs can be used to summarize everything with designing certain prompts. However, it is noticed that compared with abstractive summarization with LLMs, extractive models are in much smaller size and thus have the advantages of faster inference and less demand on computing costs. In addition, factually consistent summaries can not be guaranteed by LLMs (Tam et al. 2023). Oppositely, extractive models can ensure factual consistency to the original article and the extracted text content is more controllable for the safe use of real products.

Extractive summarization includes traditional unsupervised methods like TextRank (Mihalcea 2004; Erkan and Radev 2004), and supervised methods like BertSum (Liu and Lapata 2019). Although being capable of building high-quality summaries, supervised methods rely on high-quality human-labelled datasets. However, it is not realistic to label these datasets across all kinds of domains, document types and user requirements. The mainstream existing unsupervised summarization methods are centrality-based methods, including TextRank and PACSUM (Mihalcea and Tarau 2004; Zheng and Lapata 2019). They are no-differentiable methods and thus can not be trained with end-to-end manner. In addition, some of them like PACSUM and FAR (Zheng and Lapata 2019; Liang et al. 2021) rely on the positional-related prior assumption to achieve good performance.

Building upon existing summarization work, *length-controllable summarization (LCS)* (Yu et al. 2021) has been studied to control the length of output summaries. The LCS approach enables users to control summary length, thus to serve various real industrial scenarios when users read different types of articles, or use different size of screens, on different occasions. However, very few work (Mehdad et al. 2016) targets at the LCS problem of extractive methods, and it is hard to annotate summaries under strict length control. In fact, the nature of extracting summary texts under length constraint is more like a combinatorial optimization problem such as *Knapsack Problem* (Martello and Toth 1990), where we extract the most valuable sentences within the length constraint. This is traditionally solved by dynamic programming (DP), but this solver can not be applied as a differentiable module in end-to-end deep learning architectures.

In this paper, we aim at developing a novel unsupervised extractive summarization method with strong length control ability. We propose an unsupervised summarization archi-

ecture based on Simple Siamese (SimSiam) (Chen and He 2021), and develop a bidirectional prediction target to replace the traditional centrality target. Meanwhile, we model the LCS of extractive methods by a 0-1 Knapsack Problem, and introduce a transformer encoder to approximate the dynamic programming solver for end-to-end training. The main contributions of this paper are listed as follows:

- We propose an end-to-end trainable unsupervised model for extractive summarization, via a transformer-based siamese network. A bidirectional representation prediction objective is introduced for model training, which is more flexible than the traditional centrality by simply summing up the similarities with all other sentences.
- We also introduce a novel transformer model to approximate the knapsack solver with high accuracy, which can be applied as a differentiable module in various deep architectures. By jointed training with the proposed extractive summarization model, the length control strategy can be learned in a differentiable way.
- Experiments show that our proposed unsupervised method can produce high-quality summaries comparable to the state-of-the-art unsupervised methods in automatic evaluation. With the trainable knapsack module, our model performance of length control outperforms related baselines without utilizing end-to-end training, and even outperforms the supervised method in human evaluation.

Related Work

Extractive Summarization

Extractive summarization (Erkan and Radev 2004; Liu and Lapata 2019; Zhong et al. 2020) output text summaries by extracting the most informative segments from the original text. Based on the requirement of labelled data, it can be classified into unsupervised methods and supervised methods. For unsupervised methods, centrality is widely used as a metric for sentence extraction (Mihalcea and Tarau 2004; Zheng and Lapata 2019; Liang et al. 2021). Traditional centrality based method such as TextRank applies similarity-based centrality to determine the importance of each sentence, but it usually can not provide good performance when evaluated with human labelled ground truth. Zheng and Lapata (2019) introduced the prior assumption that the contribution of two nodes' connection to their respective centrality is influenced by their relative position, which lead to a large improvement of the automatic metrics on extractive summarization. Liang et al. (2021) introduced a facet-aware mechanism to handle the case where the document has multiple facets. Xu et al. (2020b) applies a pretrained hierarchical transformer to calculate the probability of containing each sentence in the context, as well as attention based centrality as criteria for scoring sentences. Optimal transport is also applied in as a non-learning-based method (Tang et al. 2022). For supervised methods, bidirectional sequence models like bi-LSTM or transformer encoder can be applied for encoding and scoring the representation of each sentence in the document (Liu and Lapata 2019). Graph-based methods and reinforcement learning have been introduced for further improvement (Wang et al. 2020; Jing et al. 2021; Gu, Ash, and Hahnloser 2022).

Length Control in Summarization

The majority of work in LCS focus on abstractive methods, which aims to stop the generating process at desired length by generating a stop token (*e.g.*, [EOS]) (Rush, Chopra, and Weston 2015; Kikuchi et al. 2016; Liu, Luo, and Zhu 2018; Takase and Okazaki 2019; Yu et al. 2021; Makino et al. 2019; Liu, Jia, and Zhu 2022). Previous studies on length-controllable extractive methods are very limited. Mehdad et al. (2016) shows a comparison of various techniques for extractive summarization under the strict length constraint. They propose a supervised learning method to score each sentence using a combination of pre-computed features, and extract sentences into a final summary in a greedy fashion based on their scores while respecting the length constraint.

Methodology

Problem Formulation

Consider a document D containing a set of n sentences $\{x_1, \dots, x_n\}$ with lengths $\{l_1, \dots, l_n\}$. The basic extractive summarization is to select a subset of sentences to formulate a summary $S = \{x_i\}_{i \in I_{ext}}$, which is an ordered set of selected sentences in D given by indices $I_{ext} \subseteq \{1, \dots, n\}$. Define f as the evaluation function, V is the optimal subset of extracted summary, for general unsupervised extraction **without length control**, the learning objective is given by:

$$V = \operatorname{argmax}_{S \subseteq D} f(S), \text{ s.t. } n(S) \leq M, \quad (1)$$

where M is the maximum number of sentences to be extracted. For the summarization **with the length control** as a requirement, the total length of the extracted summary should not exceed a pre-determined budget C . Thus, Eq.(1) should be subject to $\sum_{i \in S} l_i < C$. Existing extractive approaches such as LexRank (Erkan and Radev 2004) and BertSum (Liu and Lapata 2019) assign a score to each sentence based on measuring the importance of it in the whole document. Assume that the goodness of the summary is represented by the sum of scores of selected sentences, the evaluation function f can be re-written as $f(S) = \sum_{i \in S} v_i$, and the learning objective with length control becomes:

$$V = \operatorname{argmax}_{S \subseteq D} \sum_{i \in S} v_i, \text{ s.t. } \sum_{i \in S} l_i < C, \quad (2)$$

where v_i is the score of the i -th sentence given by a scorer. By now, this length controllable extractive summarization emerges as the classical 0-1 Knapsack Problem.

Unsupervised Method

For the traditional graph-based ranking method on extractive summarization, the degree of centrality for a sentence i is calculated by a simple summation over all its similarities with other sentences as follows:

$$\text{centrality}(x_i) = \sum_{j \in \{1, \dots, i-1, i+1, \dots, n\}} e_{ij}, \quad (3)$$

where e_{ij} represents the similarity between sentence i and sentence j . This text centrality is further used to measure the importance of each sentence.

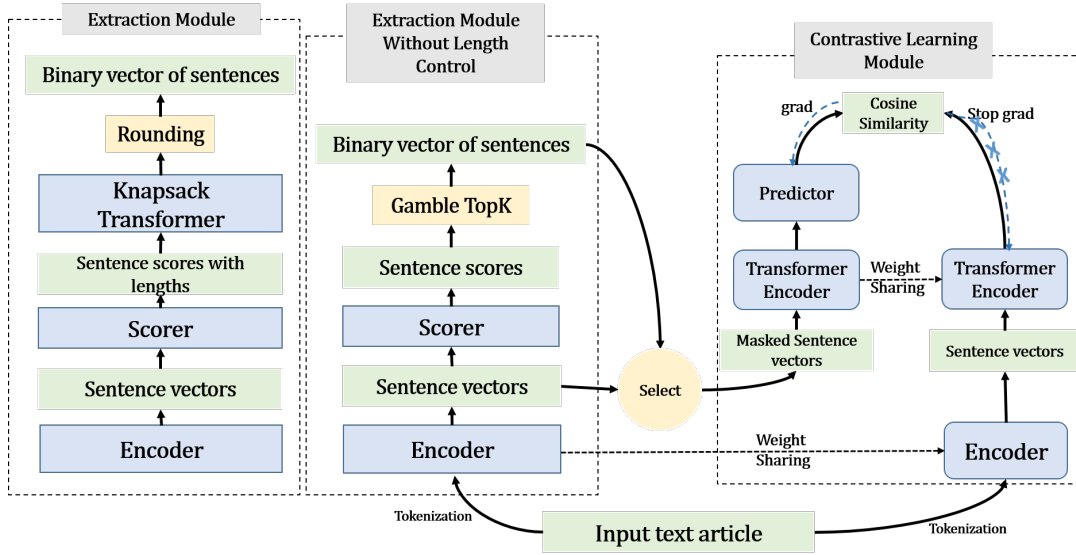


Figure 1: The diagram of the joint siamese model for unsupervised extractive summarization with and without length constraint. For the case with length control, the Knapsack transformer is applied in training time, which takes sentence scores and sentence lengths as input, and outputs the indices of extracted sentences. The unsupervised learning is to train the scorer by maximizing the prediction accuracy of SimSiam-based contrastive learning module with two proposed objectives (**Rest-Ext** and **Ext-Doc**).

Motivation. We notice that the similarity e_{ij} here can also be Pearson correlation between each pair of sentence representations, which further indicates that the sentences with higher centrality can be explained or even predicted by the whole set of other sentences with a relatively high possibility. Meanwhile, the representation of the extracted summary should have a strong correlation with the representation of the whole document by the nature of summaries. Motivated by this, we propose a bidirectional prediction criterion to replace the traditional centrality in unsupervised summarization. The first prediction criterion (**Rest-Ext**) is that each of the selected sentences for extractive summary, rather than unselected sentences, can be better predicted by the rest of sentences (selected or unselected) in the document. The second prediction criterion (**Ext-Doc**) is that the whole set of the selected sentences can better predict the representation of the whole document compared to other unselected ones. Overall, with these two objectives, the learning process is to train the sentence scorer and predictor to increase the prediction accuracy, thus extracting the most importance sentences to form as a summary in an unsupervised manner, as in the Figure 1.

Particularly, consider the final sentence representations of the original document is $\mathbf{H}^S = (\mathbf{h}_1^S, \dots, \mathbf{h}_N^S)$, the rest of the document by excluding one randomly sampled sentence from the extracted summary is $\mathbf{H}^{S'} = (\mathbf{h}_1^{S'}, \dots, \mathbf{h}_N^{S'})$, and the extracted summary is $\mathbf{H}^{S''} = (\mathbf{h}_1^{S''}, \dots, \mathbf{h}_N^{S''})$, where we preserve the size of representations N but mask out unrelated information for each case in a lower representation level. We introduce two learning objectives: (a) The first predictive loss (**Rest-Ext**) of using the final representation of the rest of sentences $\mathbf{H}^{S'}$ to predict the representation of a randomly

sampled one from selected sentences.

$$\hat{\mathbf{h}}_i^S = \text{Predictor}(\mathbf{h}_1^{S'}, \dots, \mathbf{h}_N^{S'}), \quad (4)$$

where i is the index sampled from the selected sentences $i \in I_{ext}$. Here we apply the random sampling as we assume each of the selected sentences can be well predicted given other sentences of this document. We select the i -th top vector of the predictor as the predicted representation, where the predictor is a transformer encoder. Meanwhile, to enlarge the relative advantage of selected sentences, and avoid only learning the predictor rather than this selecting mechanism, we also consider a prediction of negative sample $\hat{\mathbf{h}}_{i'}^S$ with the same masking mechanism, where i' is the index of a randomly sampled unselected sentence with low scores. Thus, the predictive loss is to be:

$$L_{Rest-Ext} = -\cos(\hat{\mathbf{h}}_i^S, \mathbf{h}_i^S) + |\cos(\hat{\mathbf{h}}_{i'}^S, \mathbf{h}_i^S)|, \quad (5)$$

where $\hat{\mathbf{h}}_i^S$ is the i -th top sentence vector of the predictor from the extraction side (the left part of contrastive learning module in the Figure 1) and \mathbf{h}_i^S is the i -th top sentence vector of the sentence-level encoder from the whole document context-aware side (similarly, the right part of that contrastive learning module). The cosine similarity is given by $\cos(\mathbf{a}, \mathbf{b}) = \mathbf{a} \cdot \mathbf{b} / (\|\mathbf{a}\| \cdot \|\mathbf{b}\|)$ for any pair of vector \mathbf{a} and \mathbf{b} . Note that minimizing $L_{Rest-Ext}$ is equivalent to optimize a MSE loss between normalized vector \mathbf{a} and \mathbf{b} as $\|\mathbf{a} - \mathbf{b}\|_2^2 = 2 - 2\mathbf{a} \cdot \mathbf{b} / (\|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2) = 2 - 2\cos(\mathbf{a}, \mathbf{b})$. We use the absolute value of prediction similarity from negative samples to achieve low absolute correlation.

(b) The second predictive loss (**Ext-Doc**) of using all the representations of selected sentences to predict the document representation. In our study, we apply the average of sentence

representations as the document representation, thus we have:

$$\hat{\mathbf{h}}^D = \frac{1}{N} \sum_i^M \hat{\mathbf{h}}_i^S = \text{Predictor}(\mathbf{h}_1^{S''}, \dots, \mathbf{h}_N^{S''}) \quad (6)$$

as a prediction. The corresponding loss is:

$$L_{Ext-Doc} = -\cos(\hat{\mathbf{h}}^D, \bar{\mathbf{h}}^S) + |\cos(\hat{\mathbf{h}}^D, \bar{\mathbf{h}}^S)|, \quad (7)$$

where $\bar{\mathbf{h}}^S = \frac{1}{N} \sum_{i=1}^M (\mathbf{h}_i^S)$ is the average sentence representation of the full document, and $\hat{\mathbf{h}}^D$ is the predicted representation given by the set of unselected sentence representations. Overall, the total loss of training objective is:

$$L = \lambda_1 L_{Rest-Ext} + \lambda_2 L_{Ext-Doc}, \quad (8)$$

where λ_1 and λ_2 are two hyper-parameters.

To get the final representations \mathbf{H}^S , $\mathbf{H}^{S'}$ and $\mathbf{H}^{S''}$ for bidirectional prediction with a trainable selector, we develop a SimSiam architecture as shown in the Figure 1. In general, it consists of a sentence extraction module, and a contrastive learning module. Sentence extraction module takes a tokenized document $\{x_1, \dots, x_n\}$ as the input, in which a sentence-level encoder such as BERT or SimCSE processes sentences and outputs the first-level sentence representation $\mathbf{E} = (\mathbf{e}_1, \dots, \mathbf{e}_n)$ by taking the top vectors of [CLS] token. A transformer-based scorer provides importance scores for each sentence. These scores are used for selecting the sentence representations in a differentiable manner, which will be discussed later. The selected sentence vectors \mathbf{E}'' , sentence vectors excluding a randomly sampled one \mathbf{E}' and all sentence vectors \mathbf{E} are passed through another transformer-based encoder in the contrastive learning module, to get the second-level sentence representations $\mathbf{H}^{S''}$, $\mathbf{H}^{S'}$ and \mathbf{H}^S for later cosine similarity. Note that our so-called first-level and second-level sentence representations are separated, as they serve different purposes in the training. The first-level one is used for scoring and selecting sentences, while the second-level is mainly learned to serve our designed bidirectional prediction criterion. The ‘‘Weight Sharing’’ in Figure 1 denotes that the weight in each group of encoders is shared. In inference, only the encoder and scorer is needed.

Trainable Length Controller

As discussed in Problem Formulation, length controller extractive summarization can be modelled by the Knapsack Problem (Martello and Toth 1990). The classical Knapsack Problem is often solved by adopting the dynamic programming algorithm, however which is non-differentiable. Recent studies have explored neural knapsack models (Nomer et al. 2020; Hertrich and Skutella 2021; Li et al. 2021). Xu et al. (2020a) use NNs to enhance dynamic programming and Hertrich and Skutella (2021) present a detailed mathematical study to investigate the expressivity of NNs, and find that a class of RNNs with feedforward RELU compute provably good solutions to the NP-hard Knapsack Problem. In addition, Yildiz (2022) uses reinforcement learning to achieve near-optimum solutions in a much faster way. Previous work of neural knapsack motivates us to rethink the LCS problem and address it with a trainable end-to-end neural network.

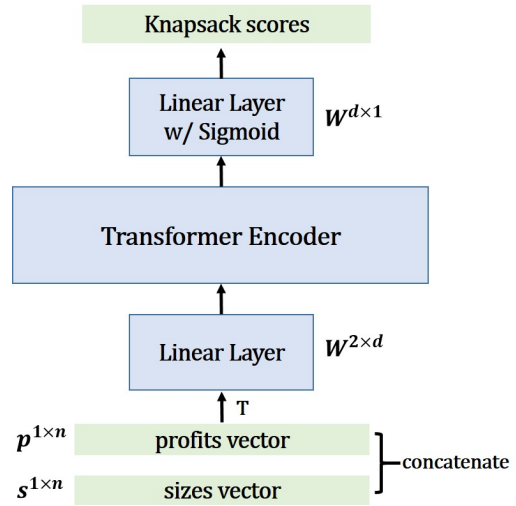


Figure 2: The proposed knapsack transformer. A linear layer is added to convert the feature dimension of the concatenated input to the model dimension. In our LCS problem, the profit vector represents each sentence score and the size vector is its length. ‘‘T’’ denotes to transpose the matrix.

In this paper, we introduce a novel knapsack transformer to approximate the dynamic programming solver. We apply BERT-like transformer encoder (Vaswani et al. 2017; Devlin et al. 2018) as the approximator, and simulate data for training it. The structure of knapsack transformer is presented in Figure 2. For each training sample, the profits and sizes of items are concatenated into a 2 by n matrix. Prior to the original transformer encoder, a linear layer is applied to map the input feature dimension of 2 to the hidden embedding dimension d . After the transformer module, another linear layer is applied to map the hidden states from dimension d back to single dimension, followed by a Sigmoid activation. By now, the knapsack score of each sentence is outputted. The training objective is to minimize a binary cross entropy loss between the model output and the binary vector of the ground truth. To make the trained model generalize well, the profits for each sample are normalized and sum up to 1, while the sizes are normalized by dividing the target total size. In the inference, these knapsack scores are rounded to 0 or 1 corresponding to not select or select the item.

Enabling Differentiable End-to-End Training

Figure 1 demonstrates our joint SimSiam model for unsupervised extractive summarization and the extractive module with or without length-controllable ability (*i.e.*, using a pre-trained knapsack transformer). We apply gumble softmax and straight-through trick to enable the end-to-end training with a differentiable process of sentence representation selection. For unsupervised model **without knapsack transformer**, we use topK sampling to get the indices of selected sentences in inference. However, in training, Gumbel-soft topK sampling is applied for better robustness (Jang, Gu, and Poole 2017;

Kool, Van Hoof, and Welling 2019) as follows:

$$y_i = \text{GumbleSoftmax}(\pi_i, T), i = 1, 2, \dots, M$$

$$\mathbf{s}_{tk} = \text{TopK}(y),$$

where π_i is the score of sentence i given by BertSum module, \mathbf{s}_{tk} is the binary vector showing the position of topK indices in \mathbf{y} . Then we can use a straight through trick given by:

$$\hat{\mathbf{s}} = (\mathbf{s}_{tk} - \mathbf{y})_f + \mathbf{y},$$

to get a derivable binary vector $\hat{\mathbf{s}}$, where $(\cdot)_f$ means we do not calculate gradients within the parentheses in backward propagation. In forward propagation, modules using this topK can only see a binary vector $\hat{\mathbf{s}} \in \mathbb{B}^n$ containing a binary value per sentence, which means to select (*i.e.*, rounding to 1) or not select (*i.e.*, rounding to 0) this sentence to form the resulting summary. In backward propagation, the gradients are calculated with the continuously differentiate value of \mathbf{y} .

For our model **with knapsack transformer** shown in Figure 2, the concatenated input of score vector and length vector is fed into the pre-trained knapsack transformer for selecting sentences as the summary. Note that there is a step-wise rounding operation on the output score of knapsack transformer, where we similarly apply the straight through trick to enable back propagation:

$$\hat{\mathbf{s}} = (\mathbf{s}_{\text{round}} - \mathbf{s})_f + \mathbf{s},$$

where \mathbf{s} is the output score vector of the knapsack transformer, and $\mathbf{s}_{\text{round}}$ is its rounded vector with elements as either 0 or 1. Similar to the above-mentioned topK, the $\hat{\mathbf{s}}$ is differentiable given that the \mathbf{s} is differentiable. To optimize $L_{Ext-Doc}$, the binary vector $\hat{\mathbf{s}}$ representing the selected sentences is multiplied with the representation of that sentence with a row-wise scalar multiplications by the following masking:

$$\mathbf{e}_i'' = \hat{s}_i \mathbf{e}_i, \quad \hat{s}_i \in \{0, 1\}, i \in \{1, \dots, n\}, \quad (9)$$

where \hat{s}_i , \mathbf{e}_i and \mathbf{e}_i'' are the i -th element in $\hat{\mathbf{s}}$, the vector for i -th sentence in \mathbf{E} , and the new masked vector for i -th sentence in \mathbf{E}'' after multiplication. It means that only the embedding of selected sentences are kept. Then the vectors of extracted summary \mathbf{E}'' are passed to the contrastive learning module. To optimize $L_{Rest-Ext}$, each time we mask the vector of a randomly sampled selected sentence indexed by k to get the rest representations for the predictor through another row-wise multiplication:

$$\mathbf{e}_i' = (1 - \tilde{s}_i) \mathbf{e}_i, \quad \tilde{s}_i \in \{0, 1\}, i \in \{1, \dots, n\}, \quad (10)$$

where $\tilde{s}_i = (\delta_{ik} - s_i)_f + s_i$, and $k \in I_{ext}$ is a single index sampled from the set of selected sentence indices by Gumble-topK or rounding. δ_{ik} equals to 1 only when $i = k$ otherwise it equals to 0. Thus \mathbf{e}_i' is the new masked vector for the i -th sentence in \mathbf{E}' , where the k -th representation is masked by a 0 vector. Finally, we are able to get the input representations for two training objectives in contrastive learning, and the score-based selecting process is fully differentiable.

Experiment

We conduct experiments on CNNDM dataset (Hermann et al. 2015), NYT dataset (Durrett, Berg-Kirkpatrick, and Klein 2016) and CNewSum dataset (Wang et al. 2021) for news summarization in both English and Chinese.

Method	Distribution	Error Rate	Matched Rate
FPTAS-NN	CNewSum	2.7%	73.4%
Neural Knapsack	Pre-defined		75.3%
Reinforcement Learning	Pre-defined		92.7%
KT (DP)	CNewSum	0.7%	93.8%
KT (DP)	CNNDM	1.7%	82.1%
KT (Density Greedy)*	CNewSum	0.4%	96.8%
KT (Density Greedy)*	CNNDM	0.5%	93.2%

Table 1: Comparison of Neural Knapsack Methods. The second column shows whether the data distribution is predefined with a fixed number of items, or sampled based on the distribution of the given dataset. KT denotes to our Knapsack Transformer. The asterisk mark denotes the method approximates sub-optimal solution given by the greedy algorithm.

Training of Knapsack Network

We simulate 6 million training samples including profits and sizes of each item according to the empirical distribution of lengths and scores in the context of extractive summarization. As there is a large margin between average sentence lengths of CNewSum and CNNDM/NYT, we separately create training data for each of them. We use *Sent-512* to denote the average number of sentences within the first 512 tokens. Firstly, we sample the number of sentences within the same range as *Sent-512* of each dataset, shown in Table 1 in supplementary materials. In particular, we use a Poisson distribution $Poisson(\lambda)$ to sample sentence count for each article, where λ equals to *Sent-512* of each dataset. Secondly, a gamma distribution $\Gamma(\alpha, \beta)$ is applied to sample the sentence length l_i for each sentence sample, where $\alpha = 2$ and $\beta = l/2$. Finally, we sample scores for each sentence via a uniform distribution $U(0, 1)$. We normalize the lengths on a set of target sizes of $\{75, 100, 125, 150, 175, 200, 225\}$ for CNewSum, $\{300, 350, 400, 450, 500, 550, 600\}$ for CNNDM, and normalize the sentence scores to sum up to 1. By now, we successfully simulate the profit (*i.e.*, sentence score) and the size (*i.e.*, sentence length) values to train a knapsack network.

Greedy method & DP solver. For the simulated data, we obtain the ground-truth indices of selected items using either dynamic programming (DP) as a solver on values, also named as profits (*i.e.*, sentence scores), or the greedy method as a solver on value densities (*i.e.*, the scores divided by the length of sentences) for a sub-optimal solution. In another word, we use the DP solver or the greed method to produce ground-truth labels for training. After labelling indices, 95% of samples are used as a train set and the rest as a validation set. We use an 8-layer knapsack transformer with 8 heads and the model dimension of 768.

Result. As in the Table 1, We compare our *Knapsack Transformer* to a set of existing neural knapsack methods including *FPTAS-NN* (Hertrich and Skutella 2021), *Neural Knapsack* (Nomer et al. 2020), and *Reinforcement Learning* (Yildiz 2022). As the original *FPTAS-NN* can not directly generate

Method	CNNDM			NYT			CNewSum		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
Oracle	54.70	30.40	50.80	61.90	41.70	58.30	46.84	30.54	40.08
Lead-3	40.50	17.70	36.70	35.50	17.20	32.00	30.43	17.26	25.33
TextRank (tf-idf)	33.20	11.80	29.60	33.20	13.10	29.00	28.43	13.51	23.13
TextRank (BERT)	30.80	9.60	27.40	29.70	9.00	25.30	29.35	15.56	24.17
TextRank (SimCSE)	33.36	12.33	30.33	31.90	14.62	27.48	28.26	14.18	23.40
Adv-RF (Wang and Lee 2018)	35.51	9.38	20.98	-	-	-	-	-	-
NeuSum (Zhou et al. 2018)	-	-	-	-	-	-	30.61	17.36	25.66
TED (Wang and Lee 2018)	38.73	16.84	35.40	37.78	17.63	34.33	-	-	-
PACSUM (Zheng and Lapata 2019)	40.70	17.80	36.90	41.40	21.70	37.50	-	-	-
STAS (Xu et al. 2020b)	40.90	18.02	37.21	41.46	21.80	37.57	-	-	-
FAR (Liang et al. 2021)	40.83	17.85	36.91	41.61	21.88	37.59	-	-	-
PMI (Padmakumar and He 2021)	36.60	14.52	23.30	-	-	-	-	-	-
Ours (Ext-Doc)	38.10	16.65	34.56	39.01	19.60	34.63	32.56	18.47	27.71
Ours (Rest-Ext)	40.71	17.79	37.06	41.49	21.65	37.33	31.34	17.77	26.08
Ours (both)	40.92	17.88	37.27	41.55	21.86	37.62	32.82	18.62	27.84

Table 2: Comparison of unsupervised methods. For existing methods, the results of TextRank (SimCSE) is re-conducted by us, while other results are reported in existing papers (Zheng and Lapata 2019; Xu et al. 2020b; Wang et al. 2021).

the optimal selection output, we apply an additional transformer to learn the mapping from its output vectors to the binary vectors of optimal selection in an end-to-end manner. For the other two alternatives, we list their reported results.

Results show that our proposed *Knapsack Transformer* (KT) outperforms baselines under the same data distribution, such as CNewSum. In particular, the per item validation error (i.e., *error rate* in the table header) of *Transformer Knapsack* decreases to **0.7%**, while the proportion of matching the optimal results (i.e., *matched rate* in the table header) given by DP is about **93.8%**. Our *Transformer Knapsack* trained with greedy algorithm can even reach **96.8%**, but note that this matched rate is compared to labels given by the greedy method, which is sub-optimal to knapsack problem.

Although the distributions of simulated data may differ in different work (Hertrich and Skutella 2021; Nomer et al. 2020; Yildiz 2022), we believe our *Transformer Knapsack* has shown strong power as a trainable module to approximate the DP solver of knapsack problem.

Training of Joint Summarization Model

As in Figure 1, for the encoder in extraction module and right-bottom siamese encoder, we apply the pretrained SimCSE model (Gao, Yao, and Chen 2021) from Huggingface. We load the pre-trained knapsack transformer in our study. During the training, the parameters of knapsack transformer and encoder are not updated. The scorer is a 2-layer transformer with 4 attention heads and 768 hidden size, and dimension of FFNN layer is 2048. For the transformer encoder and predictor in contrastive learning module, we use 4-layer transformers with 8 heads and 768 hidden size. The seed is set to be 42. We apply two Adam optimizers with $\beta_1 = 0.9$, $\beta_2 = 0.999$ for training randomly initialized parameters and loaded pre-trained parameters, respectively. We set the main learning rate $lr = 3e - 6$ for CNNDM and NYT and $lr = 1e - 6$ for CNewSum after a grid search from set $\{3e-8, 1e-7, 3e-7, 1e-6, 3e-6, 1e-5\}$, and the batch size is 64. We apply $\lambda_1 = 1$

Method	CNNDM			NYT			CNewSum		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
Default	40.9	17.9	37.3	41.6	21.9	37.6	32.8	18.6	27.8
w/o Ext-Doc	40.7	17.8	37.1	41.5	21.7	37.3	31.3	17.8	26.1
w/o Rest-Ext	38.1	16.6	34.6	39.0	19.6	34.6	32.6	18.5	27.7
w/o Predictor	26.2	11.5	23.4	33.6	14.8	29.1	30.4	15.4	25.3
w/o Sent. Enc.	37.9	16.5	34.2	39.6	20.0	35.2	30.6	15.5	25.5
w/o Neg. Sam.	38.5	16.8	35.3	35.1	16.4	30.8	31.2	16.2	36.1

Table 3: Ablation study of on CNNDM, NYT and CNewSum.

and $\lambda_2 = 0.3$ for CNNDM/NYT and $\lambda_1 = 1$ and $\lambda_2 = 1$ for CNewSum as in Eq. (8). We compare the performance of a set of candidate models for summarization with the target length limits of 100, 150, 200 characters for CNewSum, and 400, 450, 500 letters for CNNDM.

Baselines. We compare our method with: (a) Lead-based algorithm; (b) Textrank with different types of sentence representations (Erkan and Radev 2004); These methods provide explicit sentence scores that can be used by DP. We also compare the proposed models with a set of existing models in the case without length control, including Adv-RF (Wang and Lee 2018), TED (Yang et al. 2020), PACSUM (Zheng and Lapata 2019), STAS (Xu et al. 2020b), PMI (Padmakumar and He 2021), NeuSum (Zhou et al. 2018), etc.

Automatic Evaluation. For automatic evaluation, model checkpoints are saved and evaluated on the validation set every 1,000 steps. The top-3 checkpoints are then selected based on the validation loss, and the averaged results of checkpoints on the test set are reported.

The automatic evaluation is reported in Table 2. In general, the performance of our proposed method is very competitive with strong state-of-the-art models for unsupervised summarization. On all three datasets, our model using both two objectives largely outperform the most related baseline by

Target	Method	R1	R2	RL	Ave. Len
400	TR (tf-idf)+ DP	31.7	10.5	28.8	392 (13)
	TR (SimCSE)+ DP	30.4	9.6	27.5	381 (20)
	Ours + DP	38.6	16.9	35.0	380 (19)
	Ours + TFDP + DP	39.0	17.1	35.4	381 (18)
	Ours + TFDEN + DP	39.2	17.1	35.5	380 (9)
450	TR (tf-idf)+DP	32.3	10.9	29.5	443 (15)
	TR (SimCSE)+ DP	31.4	10.4	28.5	428 (21)
	Ours + DP	39.5	17.3	35.9	430 (19)
	Ours + KTDP + DP	40.0	17.5	36.4	433 (17)
	Ours + KTDEN + DP	39.2	17.1	35.8	431 (11)
500	TR (tf-idf)+DP	32.5	11.1	29.8	494 (13)
	TR (SimCSE)+ DP	31.7	10.5	28.7	480 (23)
	Ours + DP	39.8	17.4	36.3	480 (20)
	Ours + KTDP + DP	40.5	17.7	37.0	481 (18)
	Ours + KTDEN + DP	40.4	17.7	37.0	480 (11)

Table 4: Comparison of Length control methods on CNNDM. We provide the average and standard deviation (in parentheses) of the summary lengths in the last column.

Target	Method	R1	R2	RL	Ave. Len
100	TR(tf-idf)+DP	28.0	13.2	22.3	89 (17)
	TR(SimCSE)+DP	28.1	13.6	22.2	88 (18)
	Ours+DP	33.0	18.7	27.7	89 (15)
	Ours+KTDP+DP	33.5	19.0	28.2	89 (12)
	Ours+KTDEN+DP	33.2	18.8	28.0	88 (15)
150	TR(tf-idf)+DP	28.5	13.5	22.8	136 (25)
	TR(SimCSE)+ DP	28.6	14.3	23.3	135 (25)
	Ours+DP	32.7	18.6	27.5	136 (22)
	Ours+KTDP+DP	33.1	18.8	28.0	136 (23)
	Ours+KTEN+DP	33.1	18.7	27.9	136 (21)
200	TR(tf-idf)+DP	28.1	13.5	22.9	180 (35)
	TR (SimCSE)+DP	28.4	14.1	23.3	180 (34)
	Ours+DP	30.6	17.4	25.7	181 (31)
	Ours+KTDP+DP	30.9	17.5	25.9	180 (30)
	Ours+KTDEN+DP	31.9	18.1	27.2	181 (30)

Table 5: Length control methods on CNewSum.

using graph-based TextRank with SimCSE. We also notice that for CNNDM and NYT, the Rest-Ext objective alone can already bring quite good performance. It evidences the effectiveness of our proposed prediction objectives.

An ablation study is given in Table 3, which shows the effect of different learning objectives and model components. Although the training flow can work without the predictor and second-level encoder, results show that they contribute significantly. Without negative sampling, it is likely that learning will collapse to trivial states as the selector can not be well-trained as discussed in Method section.

Length Control. Results with the length control for CNNDM and CNewSum are provided in Table 4 and Table 5, respectively, where “TR” denotes TextRank, “KTDP” and “KTDEN” denote our Knapsack Transformer trained to approximate DP solver and greedy density solver, respectively.

Model	Rele.	Coh.	Cons.
Bertsum + DP	3.268	3.248	3.306
Ours + DP	3.293	3.256	3.347
Ours + KTDP + DP	3.329	3.217	3.393

Table 6: Human evaluation of different model outputs on CNNDM. We provide the average scores judged by human annotators. The improvements for relevance and consistency are significant by two-sample statistical tests ($p < 0.05$).

In the inference, we apply dynamic programming (DP) for 0-1 knapsack problem on the pre-trained scorers using different methods, which means that the final selecting of sentences is not solely based on score ranking thus probably with a performance drop. We can see by using the end-to-end training with our Knapsack Transformer, the performance of the scorer is consistently increased in a variety of target lengths, in terms of both automatic metrics and the distance to target lengths. Results show that our proposed length control method brings more potentials towards length-controllable summarization.

Human Evaluation. We further conduct a manual evaluation on the CNNDM dataset with a target length of 500. We randomly sample 1000 entries from the test set and generate summaries with three candidate models. Each sample is scored between 1 and 5 points by three participants from Amazon Mechanical Turk to answer three questions: (a) Whether the summaries capture the key points of the source document (Relevance or Rele.); (b) the sentence-level coherence of the summaries (Coherence or Cohe.); (c) the factual consistency of the summaries (Consistency or Cons.). Results in Table 6 show that our proposed unsupervised model using knapsack transformer achieves the best score in terms of relevance and consistency. Meanwhile, we find that our unsupervised models perform even better than the supervised Bertsum model (Liu and Lapata 2019) according to judge scores. It indicates that rouge scores might not capture the real quality of the extracted summaries, while our proposed unsupervised learning methods could outperform supervised ones from the aspect of human readers.

Conclusion

We introduce a novel unsupervised extractive summarization model based on siamese network and a bidirectional prediction objective. Meanwhile, by introducing the knapsack transformer that approximates the DP solver of knapsack problem, the length control strategy can be trained in an end-to-end manner. Experiments demonstrate that our method largely outperforms related centrality-based baselines and comparable with the state-of-the-art unsupervised methods. For the length control, our method outperforms baselines in a set of target lengths on both English and Chinese datasets, and the use of end-to-end length control training consistently provides a performance gain. Moreover, our method achieves higher average rating scores in human evaluation even compared with supervised BERTsum. We believe the model performance can be further improved by using better sentence and document representations for unsupervised training.

References

- Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.
- Durrett, G.; Berg-Kirkpatrick, T.; and Klein, D. 2016. Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Erkan, G.; and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research (JAIR)*, 22: 457–479.
- Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 6894–6910.
- Gu, N.; Ash, E.; and Hahnloser, R. 2022. MemSum: Extractive Summarization of Long Documents Using Multi-Step Episodic Markov Decision Processes. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- 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 (NeurIPS)*, 28.
- Hertrich, C.; and Skutella, M. 2021. Provably good solutions to the knapsack problem via neural networks of bounded size. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7685–7693.
- Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparametrization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR)*.
- Jing, B.; You, Z.; Yang, T.; Fan, W.; and Tong, H. 2021. Multiplex Graph Neural Network for Extractive Text Summarization. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Kikuchi, Y.; Neubig, G.; Sasano, R.; Takamura, H.; and Okumura, M. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Kool, W.; Van Hoof, H.; and Welling, M. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning (ICML)*.
- Li, D.; Liu, J.; Lee, D.; Seyedmazloom, A.; Kaushik, G.; Lee, K.; and Park, N. 2021. A Novel Method to Solve Neural Knapsack Problems. In *International Conference on Machine Learning (ICML)*.
- Liang, X.; Wu, S.; Li, M.; and Li, Z. 2021. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1685–1697.
- Liu, Y.; Jia, Q.; and Zhu, K. 2022. Length Control in Abstractive Summarization by Pretraining Information Selection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Liu, Y.; and Lapata, M. 2019. Text Summarization with Pre-trained Encoders. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Liu, Y.; Luo, Z.; and Zhu, K. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Makino, T.; Iwakura, T.; Takamura, H.; and Okumura, M. 2019. Global Optimization under Length Constraint for Neural Text Summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Martello, S.; and Toth, P. 1990. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.
- Mehdad, Y.; Stent, A.; Thadani, K.; Radev, D.; Billawala, Y.; and Buchner, K. 2016. Extractive summarization under strict length constraints. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 3089–3093.
- Mihalcea, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL interactive poster and demonstration sessions*, 170–173.
- Mihalcea, R.; and Tarau, P. 2004. Textrank: Bringing order into text. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Nomer, H. A.; Alnowibet, K. A.; Elsayed, A.; and Mohamed, A. W. 2020. Neural knapsack: A neural network based solver for the knapsack problem. *IEEE Access*, 8: 224200–224210.
- Padmakumar, V.; and He, H. 2021. Unsupervised Extractive Summarization using Pointwise Mutual Information. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Rush, A. M.; Chopra, S.; and Weston, J. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Takase, S.; and Okazaki, N. 2019. Positional Encoding to Control Output Sequence Length. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Tam, D.; Mascarenhas, A.; Zhang, S.; Kwan, S.; Bansal, M.; and Raffel, C. 2023. Evaluating the Factual Consistency of Large Language Models Through News Summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, 5220–5255. Toronto, Canada: Association for Computational Linguistics.
- Tang, P.; Hu, K.; Yan, R.; Zhang, L.; Gao, J.; and Wang, Z. 2022. OTextSum: Extractive Text Summarisation with Optimal Transport. In *Findings of NAACL*, 1128–1141.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, D.; Chen, J.; Wu, X.; Zhou, H.; and Li, L. 2021. CNew-Sum: A Large-scale Chinese News Summarization Dataset with Human-annotated Adequacy and Deducibility Level. *arXiv:2110.10874*.

Wang, D.; Liu, P.; Zheng, Y.; Qiu, X.; and Huang, X.-J. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wang, Y.; and Lee, H.-Y. 2018. Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Xu, S.; Panwar, S. S.; Kodialam, M.; and Lakshman, T. 2020a. Deep neural network approximated dynamic programming for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1684–1691.

Xu, S.; Zhang, X.; Wu, Y.; Wei, F.; and Zhou, M. 2020b. Unsupervised Extractive Summarization by Pre-training Hierarchical Transformers. In *Findings of Empirical Methods in Natural Language Processing (EMNLP)*.

Yang, Z.; Zhu, C.; Gmyr, R.; Zeng, M.; Huang, X.; and Darve, E. 2020. TED: A Pretrained Unsupervised Summarization Model with Theme Modeling and Denoising. In *Findings of the Empirical Methods in Natural Language Processing (EMNLP)*.

Yildiz, B. 2022. Reinforcement learning using fully connected, attention, and transformer models in knapsack problem solving. *Concurrency and Computation: Practice and Experience*, 34(9): e6509.

Yu, Z.; Wu, Z.; Zheng, H.; XuanYuan, Z.; Fong, J.; and Su, W. 2021. LenAtten: An Effective Length Controlling Unit For Text Summarization. In *Findings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 363–370.

Zhang, T.; Ladhak, F.; Durmus, E.; Liang, P.; McKeown, K.; and Hashimoto, T. B. 2023. Benchmarking Large Language Models for News Summarization. *arXiv:2301.13848*.

Zheng, H.; and Lapata, M. 2019. Sentence Centrality Revisited for Unsupervised Summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 6236–6247.

Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X.-J. 2020. Extractive Summarization as Text Matching. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; and Zhao, T. 2018. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.