UPER: Boosting Multi-Document Summarization with an Unsupervised Prompt-based Extractor

Shangqing Tu[1,2], **Jifan Yu**[1,2], **Fangwei Zhu**[1,2], **Juanzi Li**[1,2], **Lei Hou**[1,2*] and **Jian-Yun Nie**[3]
[1]Dept. of Computer Sci.&Tech., BNRist, Tsinghua University, Beijing 100084, China
[2]KIRC, Institute for Artificial Intelligence, Tsinghua University
[3]Department of Computer Science and Operations Research, University of Montreal, Canada
{tsq22,yujf21,zfw19}@mails.tsinghua.edu.cn
{lijuanzi,houlei}@tsinghua.edu.cn nie@iro.umontreal.ca

Abstract

Multi-Document Summarization (MDS) commonly employs the 2-stage extract-then-abstract paradigm, which first extracts a relatively short meta-document, then feeds it into the deep neural networks to generate an abstract. Previous work usually takes the ROUGE score as the label for training a scoring model to evaluate source documents. However, the trained scoring model is prone to under-fitting for low-resource settings, as it relies on the training data. To extract documents effectively, we construct prompting templates that invoke the underlying knowledge in Pre-trained Language Model (PLM) to calculate the document and keyword's perplexity, which can assess the document's semantic salience. Our unsupervised approach can be applied as a plug-in to boost other metrics for evaluating a document's salience, thus improving the subsequent abstract generation. We get positive results on 2 MDS datasets, 2 data settings, and 2 abstractive backbone models, showing our method's effectiveness. Our code is available at https://github.com/THU-KEG/UPER

1 Introduction

Multi-Document Summarization (MDS) aims to generate a summary from multiple source articles (McKeown and Radev, 1995). The input text in MDS can be overlong and therefore contain much noisy information (Liu et al., 2021c). The goal of MDS is to reduce the long input and extract salient information (Bing et al., 2015; Fan et al., 2019; Song et al., 2022).

Some previous works (Zaheer et al., 2020) utilise *sparse attention* to handle the long input problem in MDS. Many others tackle this problem by an *extract-then-abstract* paradigm (Liu and Lapata, 2019), which first extracts the salient information in source documents to form a meta-document with a preset length then generates the summary. The first stage reduces the input length for the second stage to cut the abstractive model's memory cost. The extractive stage has two main technical lines: (1) the *statistical* method uses the token-level similarity between keywords and source documents to retrieve relevant documents (Liu et al., 2018). (2) the *regressive* method trains a scoring model to predict the document's ROUGE (Lin, 2004) score with the reference summary (Liu and Lapata, 2019; Mao et al., 2021; Zhong et al., 2021; Zhang et al., 2021).

However, these methods for the extractive stage lead to several problems: (1) the *statistical* methods like tf-idf (Ramos et al., 2003) relies on strict matching of keywords, ignoring those documents with relevant semantic context. (2) the *regressive* methods aim to fit some predefined metrics, e.g., ROUGE, but the fitting result seriously depends on the training data, leading to over-fitting or under-fitting. Besides, the predefined metrics may not adequately measure the quality of the selected documents, resulting in the two-stage gap[1] of the extract-then-abstract paradigm.

To tackle the semantic discrepancy of *statistical* methods and the data dependence of *regressive* methods, we intend to find an unsupervised metric that can evaluate a document's contextual relatedness (Zou et al., 2021) with keywords. It is natural to apply the Pre-trained Language Model (PLM) and leverage its inherent ability of calculating the sequence's perplexity to test whether the keyword can appear in a candidate document's context.

In this paper, we propose an Unsupervised Prompt-based ExtractoR (UPER) which utilizes unsupervised prompts that join the keyword with the document to form a new sequence whose perplexity represents the document's semantic salience. Our method is fully unsupervised and can be used as a plug-in to evaluate documents on different datasets or boost other metrics by combining scores.

To test our method, we explore several dimen-

[*] Corresponding author.

[1] See more for a preliminary experiment in section 2.2

sions of the extractive stage in the extract-then-abstract paradigm. We apply our method on 2 multi-document summarization datasets, 4 different domains, 2 data settings, and 2 abstractive backbone models. The experimental results show that our method effectively complements the token-level similarity and significantly boosts the performance of the subsequent abstractive stage.

Overall, our contributions are the following:

- We propose a new unsupervised framework that employs prompt-based methods to measure the lexical and semantic salience in the extractive stage.

- We carry out a series of experiments demonstrating the effectiveness of the proposed approach.

2 Preliminary

2.1 Problem Formulation

Definition 1 ***Multi-document Summarization*** *is defined as a sequence-to-sequence generation problem, where the input $\mathcal{D}$ consists of n source documents $\{d_1, d_2, \ldots, d_n\}$. The objective is to generate an optimal summary $\mathcal{Y}^*$ according to the conditional distribution, i.e.,*

$$\mathcal{Y}^* = \arg\max_{\mathcal{Y}} P(\mathcal{Y}|\mathcal{D}) \tag{1}$$

However, in automatically collected multi-document summarization datasets (Liu et al., 2018; Fabbri et al., 2019; Gholipour Ghalandari et al., 2020), source documents are usually collected from websites using keywords, e.g., a Wikipedia entity or news title. The target reference summary is usually a description or report of the keyword. It is thus useful to introduce such keywords $\mathcal{K} = \{k_1, k_2, \ldots, k_{n'}\}$ as auxiliary information in Multi-document Summarization, that is,

$$\mathcal{Y}^* = \arg\max_{\mathcal{Y}} P(\mathcal{Y}|\mathcal{K}, \mathcal{D}) \tag{2}$$

Definition 2 *The* ***extractive stage*** *of the extract-then-abstract paradigm first takes n source documents $\mathcal{D}$ as input and selects m candidate documents to form a meta-document $\mathcal{D}'$ which is a subset of $\mathcal{D}$. The* ***abstractive stage*** *trains an end-to-end abstractive model that generates a summary conditioning on the meta-document $\mathcal{D}'$. Therefore, the objective of Multi-document Summarization is re-written as*

$$\mathcal{Y}^* = \arg\max_{\mathcal{D}'} P(\mathcal{D}'|\mathcal{K}, \mathcal{D}) \arg\max_{\mathcal{Y}} P(\mathcal{Y}|\mathcal{D}') \tag{3}$$

While the abstractive stage can be formulated as single document summarization, SOTA transformer architecture is often employed (Hokamp et al., 2020). The goal of extractive stage is to provide the optimal meta-document $\mathcal{D}'$ that can optimize the abstractive stage's output distribution $P(\mathcal{Y}|\mathcal{D}')$. Theoretically, the possible meta-document $\mathcal{D}'$ can be searched within the permutation of documents $A_n^m(\mathcal{D})$, which has the exponential complexity and can't be optimized directly. Prior work performs the extractive stage using either *statistical* method (Liu et al., 2018) or *regressive* method (Liu and Lapata, 2019). However, we observe their weakness in a preliminary experiment described below.

2.2 Preliminary Experiment

We conduct a preliminary experiment on the WikiSum (Liu et al., 2018) dataset to test the statistical and regressive method's performance. We first process the source documents and split them into a fine-grain length; its detail will be described in Table 2. To illustrate the training data dependence of the regressive method, we use a few-shot setting with 1% training data and 100% test data. Following the extract-then-abstract paradigm, we adapt a widely-used Seq2Seq model BART (Lewis et al., 2020) as the abstractive model and change the extractive methods from statistical method tf-idf (Ramos et al., 2003) to regressive method LGB (Ke et al., 2017). Besides, we also introduce ROUGE-1/2/L as oracle extractive methods, which rank the documents by their unigram/bigram/longest sequence overlap with the reference target summary. The highly ranked documents are then sent to BART to generate the final summary. Table 1 is the final summary's ROUGE with the target summary, and figure 1 shows the correlation between the extractive and abstractive stages. We make 2 observations.

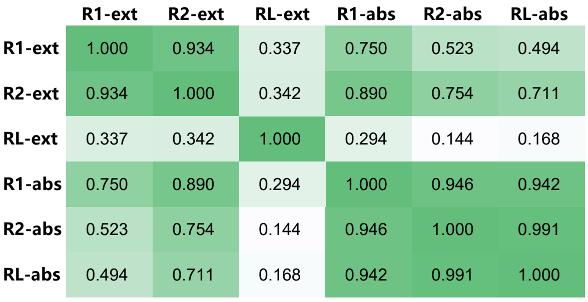
Observation I: The **data dependence** of the trained extractor is demonstrated in Table 1: the supervised regressive method obtains the lowest score because the regressive method is prone to under-fitting for the few-shot setting.

Observation II: The **two-stage gap** is presented in figure 1, the extractive stage's RUOGE scores

Extractive Method	R-1	R-2	R-L
regressive	36.5	17.3	29.3
statistical	37.4	18.6	30.7
Oracle-R1-recall	39.6	19.6	31.4
Oracle-R2-recall	**41.1**	**22.1**	**33.2**
Oracle-RL-recall	40.2	20.5	32.3

Table 1: ROUGE-F1 scores under **few-shot** setting(1% training data) with the same abstractive backbone model and different extractive methods on animal domain of WikiSum dataset. Oracles directly use the corresponding ROUGE-recall scores between the input document and the reference summary to rank documents.

don't completely correlate with the abstractive stage's ROUGE scores, which can be concluded from the 9 boxes in the upper right or the lower left part of the matrix. This suggests that ROUGE score may narrowly model the extractive stage's object function $P(\mathcal{D}'|\mathcal{K},\mathcal{D})$ though many previous regressive methods (Liu and Lapata, 2019) choose ROUGE as their training object.



	R1-ext	R2-ext	RL-ext	R1-abs	R2-abs	RL-abs
R1-ext	1.000	0.934	0.337	0.750	0.523	0.494
R2-ext	0.934	1.000	0.342	0.890	0.754	0.711
RL-ext	0.337	0.342	1.000	0.294	0.144	0.168
R1-abs	0.750	0.890	0.294	1.000	0.946	0.942
R2-abs	0.523	0.754	0.144	0.946	1.000	0.991
RL-abs	0.494	0.711	0.168	0.942	0.991	1.000

Figure 1: **Pearson correlation coefficient** between the three ROUGE metrics {R-1, R-2, R-L} for two stages {extractive, abstractive} with a backbone abstractive model BART and different extractive methods on the animal domain WikiSum dataset. **R-1/2/L** denotes ROUGE-1/2/L, ext/abs is the short for extractive/abstractive stage.

3 Method

The key challenge of the extractive stage is how to model the objective function $P(\mathcal{D}'|\mathcal{K},\mathcal{D})$, which evaluates each document and retrieves those related to the keywords. This leads to the following question: what is a proper metric for modeling it?

Based on observations in section 2.2, we propose a criteria for modeling $P(\mathcal{D}'|\mathcal{K},\mathcal{D})$: (1) the metric ought to be unsupervised which can avoid the data dependence; (2) the metric can model both lexical and semantic salience; (3) the metric should be effective for improving the abstractive stage to diminish the two-stage gap.

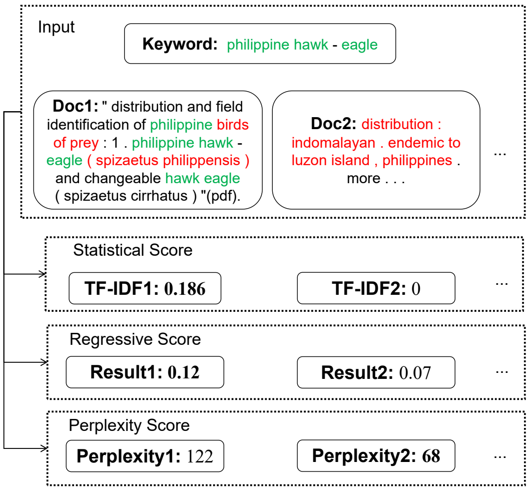

Input
Keyword: philippine hawk - eagle
Doc1: " distribution and field identification of philippine birds of prey : 1 . philippine hawk - eagle (spizaetus philippensis) and changeable hawk eagle (spizaetus cirrhatus) "(pdf).
Doc2: distribution : indomalayan . endemic to luzon island , philippines . more . . .
...
Statistical Score
TF-IDF1: 0.186
TF-IDF2: 0
Regressive Score
Result1: 0.12
Result2: 0.07
Perplexity Score
Perplexity1: 122
Perplexity2: 68

Figure 2: An example of the ranking scores for corresponding extractors. Green words are keywords and red words are effective information for generating the summary. **Bold** score is preferred by the extractor.

3.1 Perplexity

Inspired by the observations and recent success on Pre-trained Language Models, we propose to use perplexity calculated by PLM to model the semantic feature since it can be captured by language model's encoder. Because modeling sequence is the original training task of auto-regressive PLM like GPT (Radford et al., 2019), the perplexity metric can be applied on any datasets without training.

Perplexity is a widely used metric in NLP to evaluate the likelihood of a word sequence $\boldsymbol{x}$ in a language (Jelinek et al., 1977). It is defined as follows:

$$\mathtt{PPL}(\boldsymbol{x}) = \exp(-\frac{1}{T}\sum_{i=1}^{T} p_\theta(x_i|\boldsymbol{x}_{<i})) \qquad (4)$$

where $\boldsymbol{x}_{<i} = [x_0, ..., x_{i-1}]$.

As statistical extractor would prefer the document that has high overlap with the keyword, there are many noisy documents where the keyword occurs but the effective information lacks, e.g., Doc1[2] in figure 2. To filter out the noise input and retain those documents with relevant semantic but contains no keyword like Doc2 in igure 2,Figure 3 shows our overall framework.

[2] Doc1 is a reference book name in https://en.wikipedia.org/wiki/Philippine_hawk-eagle, which can not provide detailed information for summarization.

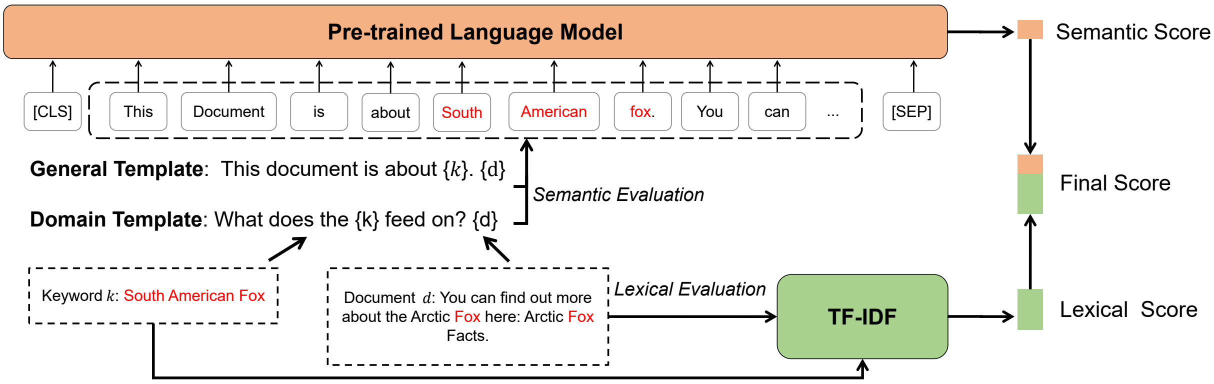

Pre-trained Language Model
Semantic Score
[CLS]
This
Document
is
about
South
American
fox.
You
can
...
[SEP]
General Template: This document is about {k}. {d}
Domain Template: What does the {k} feed on? {d}
Semantic Evaluation
Final Score
Keyword k: South American Fox
Document d: You can find out more about the Arctic Fox here: Arctic Fox Facts.
Lexical Evaluation
TF-IDF
Lexical Score

Figure 3: The framework of our UPER model. We design general and domain templates to evaluate the document's semantic salience with keywords using PLM and combine it with the tf-idf score to evaluate the lexical salience.

3.2 Prompt Design

To extract high-quality sources for abstractive models, it is important to consider the document's contextual relation with keywords (Liu et al., 2021a). Specifically, prompts have proven to be effective in modeling the contextual relatedness (Zou et al., 2021). Inspired by this, we propose an Unsupervised Prompt-based ExtractoR (UPER). We design several prompting templates $x(k, d)$ to be filled with the keyword k and the document d, then calculate the whole sequence's perplexity. The general template of $x(k, d)$ that puts the keyword in the introduction position is

$$\text{This document is about } \{k\}.\{d\} \tag{5}$$

which tests whether the keyword can appear in the document's introduction. In order to test whether this keyword can appear in the conclusion of the document, we can also design inverse patterns like:

$$\{d\} \text{ This document is about } \{k\}. \tag{6}$$

Other domain-specific prompting templates are listed in appendix A. The perplexity of the $x(k, d)$ sequence can represent the probability of k showing up in the context of d. Therefore, the semantic salience score $g^S(\mathcal{K}, d)$ of a candidate document is calculated as follows:

$$g^S(\mathcal{K}, d) = -\frac{\sum_{k\in\mathcal{K}} \sum_{x\in\mathcal{X}} \texttt{PPL}(x(k, d))}{|\mathcal{K}| \cdot |\mathcal{X}|}, \tag{7}$$

where $x(k, d)$ indicates the sequence built from a prompting template, $\mathcal{X}$ is pre-designed templates, and $\texttt{PPL}$ is the perplexity calculated by GPT2.

3.3 Score Combination

As our semantic salience metric is unsupervised, we can combine our prompt-based method with statistical methods, e.g., tf-idf. For each keyword k in $\mathcal{K}$, its lexical similarity with the document $g^L(k, d)$ can be measured by tf-idf metric:

$$g^L(k, d) = \frac{n_k}{|d|} \cdot log(\frac{n_d}{n_{dk}}) \tag{8}$$

where n_k, $|d|$, n_d and n_{dk} are the count of the keyword in the document, length of the document, total number of documents, and total number of documents containing the keyword.

For multiple keywords, we view the tf-idf as a keyword's occurring probability conditioned on documents (Ramos et al., 2003) so that the tf-idf for multiple keywords is the joint probability:

$$g^L(\mathcal{K}, d) = \prod_{k\in\mathcal{K}} g^L(k, d) \tag{9}$$

To combine the semantic score $g^S(\mathcal{K}, d)$ and the lexical score $g^L(\mathcal{K}, d)$, we need to normalize them to comparable scales first.

$$N(g) = \frac{g - \mu_g}{\sigma_g} \tag{10}$$

where g, μ_g and σ_g are the metric score, its mean value and variance.

The final score $g^F(\mathcal{K}, d)$ of a candidate document aggregates both semantic and lexical scores:

$$g^F(\mathcal{K}, d) = \lambda \cdot N(g^S(\mathcal{K}, d)) + (1-\lambda) \cdot N(g^L(\mathcal{K}, d)) \tag{11}$$

where the λ is a coefficient. Only the top-m documents are selected and passed to the next stage.

4 Experiments

4.1 Experimental Setting

This section introduces the datasets, evaluation, and baselines of our experiments. More implementation details are introduced in appendix B.

Dataset	Domain	#Examples	$\vert\mathcal{D}\vert \times \vert d\vert$	$\vert r\vert$
WikiSum	animal	60,816	180 × 20	92
WikiSum	company	62,545	253 × 25	125
WikiSum	film	59,973	266 × 23	98
WCEP	news	10,200	241 × 16	28

Table 2: Statistics of each domain on the WikiSum and WCEP dataset. $|\mathcal{D}|$, $|d|$, $|r|$ is respectively the average number of source documents, source document tokens and target reference summary tokens.

Dataset. To evaluate the models, we use the **WikiCatSum** dataset (Perez-Beltrachini et al., 2019), a subset of **WikiSum** (Liu et al., 2018), which consists of three different domains in Wikipedia (*Animal*, *Company* and *Film*) and another large-scale multi-document summarization dataset **WCEP** (Gholipour Ghalandari et al., 2020). The statistics are shown in Table 2.

Evaluation. We use ROUGE-F1 (Lin, 2004) to evaluate the generated summary with respect to the reference. For different model settings, we perform corresponding extractive stage on the training and test set, then fine-tune the abstractor on the training set and report its evaluation result on the test set.

Baselines. We select several typical baselines in related tasks, including:

• **TF-S2S** (Liu et al., 2018). A method that views the documents as a long sequence and uses a transformer decoder to generate the summary.

• **C2T** (Perez-Beltrachini et al., 2019) uses a CNN encoder and two structured decoders with topic-aware information discovered by LDA.

• **TWAG** (Zhu et al., 2021) is a recent wikipedia abstractor which explicitly considers topics on WikiCatSum dataset using topic classifiers.

• **Noisysumm** (Liu et al., 2021c) uses self-distillation to improve the abstractor's ability to handle noisy input. It can be applied to other abstractors like UniLMv2 (Bao et al., 2020).

• **BART** (Lewis et al., 2020) is a sequence-to-sequence model with bidirectional and auto-regressive transformers that accomplished state-of-the-art results on single-document summarization. It has a length limit of 1024.

• **Longformer Encoder Decoder (LED)** (Beltagy et al., 2020) is a transformer-based sequence-to-sequence model which utilizes a sparse attention mechanism to achieve the linear complexity with respect to the input length. Its length limit is 16384.

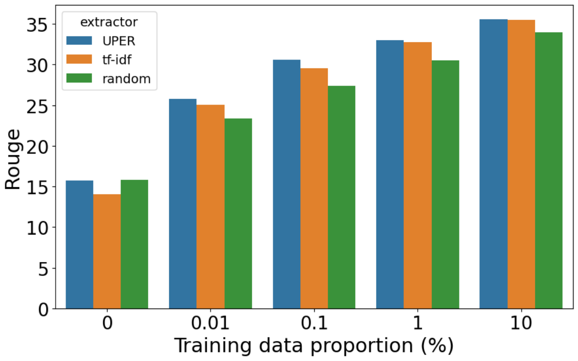


Figure 4: ROUGE-F1 scores under different **few-shot** training data proportions on the film domain of WikiSum dataset. random, `tf-idf` and `UPER` are three LED models trained with corresponding extractor.

Extractive Method	R-1	R-2	R-L
Random	34.8	12.5	26.8
tf-idf	35.0	12.6	27.3
UPER(ours)	**35.1**	**12.8**	**27.4**

Table 3: ROUGE-F1 scores under **few-shot** setting (1% training data) with the same backbone abstractive model LED and different extractive methods on WCEP dataset.

4.2 Low-resource Setup Results

To test our model's performance under a low-resource setting, we conduct experiments with different proportions of training data for fine-tuning abstractive models. As shown in figure 4, UPER outperforms both tf-idf and random order with the data scale changed from 0 to 10%. Note that UPER uses a default $\lambda = 0.75$, which means it combines with tf-idf metric and steadily boosts tf-idf's performance across different data scales. In addition to WikiCatSum dataset, UPER also boosts the performance on WCEP dataset in Table 3, demonstrating UPER's generalization ability across datasets as it benefits from the unsupervised prompts.

4.3 Full-data Setup Results

Besides the low-resource setting, we also conduct experiments under the full-data setup. Table 4 and Table 5 show the overall results. We first reproduce the abstractors without the extractive stage. Then UPER is applied on BART and LED to extract the input, which helps them achieve SOTA ROUGE scores on WikiCatSum dataset.

From the experimental results, we have several observations: (1) UPER can boost both BART and LED's performance, indicating our model's gen-

Model	Company			Film			Animal		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
TF-S2S (Liu et al., 2018)	26.0	9.5	20.4	36.5	18.8	31.0	44.0	28.8	40.0
C2T (Perez-Beltrachini et al., 2019)	27.5	10.6	21.4	38.0	21.2	32.3	42.7	27.9	37.9
TWAG (Zhu et al., 2021)	34.1	11.9	31.6	40.8	21.2	34.3	43.1	24.4	40.9
UniLMv2 (Liu et al., 2021c)	33.3	14.4	25.4	42.5	25.9	36.5	45.5	31.7	40.9
Noisysumm (Liu et al., 2021c)	33.5	**15.0**	25.9	42.7	26.1	36.8	45.9	**32.2**	41.4
BART	33.2	10.5	30.4	37.6	17.5	35.4	42.8	24.3	40.6
BART + UPER	36.7	14.3	33.8	43.3	24.6	41.0	**46.4**	29.1	**44.4**
LED	36.4	13.7	33.4	43.9	24.7	41.4	43.9	25.5	41.7
LED + UPER	**37.0**	14.5	**33.9**	**44.7**	**26.2**	**42.4**	**46.4**	26.5	**44.4**

Table 4: ROUGE-F1 scores of different models on three domains (*Company*, *Film* and *Animal*) of WikiCatSum dataset under the **full-data** setting (100% training data for fine-tuning abstractive models).

Extractive Method	R-1	R-2	R-L
Random	39.1	16.4	31.2
tf-idf	39.8	17.0	32.0
UPER(ours)	**41.4**	**18.7**	**33.8**

Table 5: ROUGE-F1 scores under the **full-data** setting (100% training data for abstractive models) with the same backbone abstractive model LED and different extractive methods on WCEP dataset.

eralization ability for different abstractors; (2) On both WikiCatSum and WCEP datasets, UPER can improve abstractive models' performance, showing our model's robustness for different datasets; (3) UPER brings more improvements to BART than LED, in that LED's input length is 15 times more than BART's, which means our extractive methods only re-rank the documents for LED other than extracting a shorter document for BART.

Human Evaluation. Aiming to examine the factual correctness of generated abstracts, we follow (Liu et al., 2021c) to conduct a human evaluation by asking crowdworkers to annotate which model generates better results. For each domain, 20 examples are randomly sampled for 3 participants' opinions. We report the proportion of systems preferred by participants in figure 5. Results show that our model improves the quality of the abstracts generated by BART in all three domains. Compared with other domains, the improvement in *Film* appeared marginal, in that humans are more familiar with films and sensitive to film errors. Meanwhile, UPER still managed to improve the performance.

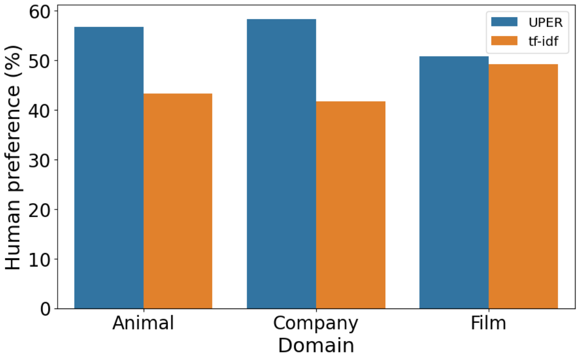


Figure 5: Human evaluation on WikiCatSum test set. `tf-idf` and `UPER` is two BART trained with corresponding extractor.

4.4 Ablation and Case study

4.4.1 Ablation Study

#Templates	R1	R2	RL
0	34.3	15.5	32.5
1	35.4	16.7	33.5
5	**35.6**	**17.2**	**33.6**
10	35.3	16.4	33.4

Table 6: ROUGE-F1 scores using different number of templates in salience estimation under few-shot setting. Here we just adopt $\lambda = 1$.

Use multiple prompts or not? We design the general prompting template and many special templates for the specific domain. One may argue that these special domain prompts hamper the generality of our model. So we conduct experiments on the number of prompt templates. Table 6 shows the ablation study on the animal domain of WikiCatSum with 1% training data for the abstractive model BART. The result shows that if we don't use prompting templates, the performance of UPER will be undermined seriously. However, increasing the number of additional templates does not necessarily improve the performance. So we use one general prompting template in other experiments.

λ	R1	R2	RL
0	45.1	27.1	43.0
0.25	45.6	28.4	43.6
0.5	46.0	**28.8**	43.9
0.75	**46.1**	**28.8**	**44.0**
1	42.0	23.8	40.0

Table 7: ROUGE-F1 scores of different λ for BART on the animal domain of WikiCatSum with full data.

Is λ important? We make grid search in {0,0.25,0.5,0.75,1}. In fact, when $\lambda = 0$, the extractor $g^F(\mathcal{K}, d)$ will become the statistical extractor. Moreover, when $\lambda = 1$, the extractor $g^F(\mathcal{K}, d)$ will become the GPT extractor which only evaluates semantic salience. Table 7 shows the ablation study results on the animal domain when the number of max input tokens is 500. We find that 0.75 is a reasonable number of λ while the statistical extractor ($\lambda = 0$) and the GPT extractor ($\lambda = 1$) are not able to outperform any ensemble extractor ($0 < \lambda < 1$). The success of ensemble extractors demonstrates that applying our semantic evaluation model as a plug-in to other metrics can boost the performance. Therefore, it is necessary to consider both the lexical and the semantic salience when designing the extractor.

#Tokens	Extractor	R1	R2	RL
500	tf-idf	45.1	27.1	43.0
500	UPER	46.1	28.8	44.0
1000	tf-idf	45.9	28.1	43.8
1000	UPER	**46.4**	**29.1**	**44.4**

Table 8: ROUGE-F1 scores of different max input tokens and extractor for BART on the animal domain of WikiCatSum with the full-data setting.

The number of input tokens for abstractors. As the input length limit of the abstractor is different for BART and LED. The top K documents we feed into the abstractor will be truncated to the number of max input tokens. We tried 500 and 1000 input tokens for BART, whose max input length is 1024. Table 8 shows the ablation study on the animal domain. UPER outperforms the tf-idf with two different input token numbers. Furthermore, the fewer input tokens, the larger advantage UPER has over tf-idf.

#Tokens	Extractor	R1	R2	RL
2048	random	36.7	17.0	34.7
2048	tf-idf	41.3	22.3	39.1
2048	UPER	41.3	22.4	39.1
4096	random	39.0	19.1	36.8
4096	tf-idf	41.6	23.0	39.4
4096	UPER	41.8	22.8	39.6
16384	random	40.9	21.5	38.7
16384	tf-idf	42.3	**23.6**	40.1
16384	UPER	**42.5**	**23.6**	**40.3**

Table 9: ROUGE-F1 scores of different max input tokens and extractor for LED trained on the film domain of WikiCatSum with the full-data and early-stopping setting.

Our UPER can also improve the abstractor's performance for the abstractive model without input limits like LED. As shown in Table 9, from limited 2048 input tokens to the unlimited 16384 tokens, UPER can steadily optimize the final abstract against randomly ordered input documents. However, tf-idf and UPER almost achieve equal success in improving the unlimited input length model LED compared to the significant promotion brought by UPER on the limited input length model BART. This phenomenon reflects that the salient information in the extracted meta-document is more influential for the limited input length models because they can only accept a small part of the long input. While the unlimited input length model can receive the entire input, the extractor only plays the role of re-ranking documents.

4.4.2 Case Study

We sample an example from the animal domain to analyze. As shown in Table 10, the gold abstract is mainly composed of information from three topics: *taxonomy*, *distribution*, and *description*.

The abstract generated by BART+tf-idf and BART+UPER both covers all three topics in the gold abstract. But BART+tf-idf contains more factual errors on the wingspan information. It also outputs unrelated words about the larvae, which

Gold Abstract: syndemis musculana is a moth of the family tortricidae . it is found in europe , china (heilongjiang , jilin , inner mongolia) , the korean peninsula , japan , russia (amur) and north america . the wingspan is 15 - 22 mm . the adults fly from april to july in the temperate parts of their range , such as belgium and the netherlands . the caterpillars feed on oaks (quercus) , birches (betula) , spruces (picea) , ragworts (senecio) and rubus (brambles and allies) . less usually , they have been recorded to eat plant refuse and dry leaves .
BART+tf-idf: syndemis musculana is a moth of the family tortricidae . it is found in europe , china (heilongjiang , jilin , inner mongolia) , the korean peninsula , japan , russia (amur) and north america . the wingspan is 16 - 21 mm . adults are on wing from july to september . there is one generation per year . in the north , the larvae feed on archips similis . larvae can be found from june to july .
BART+UPER: syndemis musculana is a moth of the family tortricidae . it is found in europe , china (heilongjiang , jilin , inner mongolia) , the korean peninsula , japan , russia (amur) and north america . the wingspan is 15 - 22 mm . adults are on wing from april to july in the temperate parts of their range , such as belgium and the netherlands . they are active from july to october , overwintering as a full .

Table 10: Comparison between abstracts generated by BART with different extractors (tf-idf and UPER) about the animal *syndemis musculana* on WikiCatSum dataset.

are not mentioned in the gold abstract. Therefore, our perplexity-based extractor UPER can provide the abstractive model with a reliable information source, thus avoiding generating wrong or redundant words.

5 Related Work

Multi-document summarization (MDS) aims to generate an abstract for the related documents collected from referred websites or search engines. This task is commonly regarded as a two-stage problem (Liu et al., 2018; Zhu et al., 2021). The extractive stage selects high-quality relevant texts as sources, and the abstractive stage summarizes them into an abstract of required length. Prior work mostly focuses on improving the latter abstrative stage with various techniques like topic information (Perez-Beltrachini et al., 2019; Zhu et al., 2021), graph representation (Li et al., 2020) and attention (Perez-Beltrachini and Lapata, 2021). The exploration of the extractive stage is limited to a few methods like tf-idf (Liu et al., 2018) and ROUGE scorer (Liu et al., 2019), and they both focus on the token-level lexical similarity, while we take the semantic salience into consideration, which suppresses the intrinsic noise in the corpus.

Automatic evaluation metrics are vital for the extractive stage of multi-document summarization task, which can select source documents. They can be divided into two classes: referenced and reference-free. Referenced metrics usually focus on the lexical overlap (Papineni et al., 2002; Banerjee and Lavie, 2005) or embedding similarity (Zhao et al., 2019; Liu et al., 2021b) between the document and the reference summary. While reference-free metrics usually evaluate the document without reference summary using perplexity (Brown et al., 1992) or aspects evaluation (Ke et al., 2022). Our method is inspired by a referenced metric proposed by Bajaj et al. (2021). They train a scoring model to predict the perplexity of the sequence formed by concatenating the reference summary with the input document. In Section 3.2, we find that the pretrained language model like GPT can be utilized to evaluate documents' semantic salience without training, so we propose our reference-free metric UPER.

6 Conclusion

This work investigates the extractive stage of the Multi-document Summarization task. We propose a simple but effective approach UPER to model the semantic contextual salience and combine the lexical token-level similarity to extract the input documents. UPER utilizes unsupervised prompts to take advantage of prior knowledge distributed in PLMs so that we can convert our extraction task to PLM's original perplexity calculation task. In our future work, we will extend our framework to single-document summarization and explore the application of prompt-based methods in the supervised learning scenario.

Acknowledgement

This work is supported by the New Generation Artificial Intelligence of China (2020AAA0106501), the NSFC Youth Project (62006136) and a grant from the Institute for Guo Qiang, Tsinghua University (2019GQB0003).

References

Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew McCallum. 2021. Long document summarization in a low resource setting using pretrained language models. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 71–80, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pages 65–72.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 642–652. PMLR.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1587–1597, Beijing, China. Association for Computational Linguistics.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. 1992. An estimate of an upper bound for the entropy of english. Computational Linguistics, 18(1):31–40.

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.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1302–1308, Online. Association for Computational Linguistics.

Chris Hokamp, Demian Gholipour Ghalandari, Nghia The Pham, and John Glover. 2020. Dyne: Dynamic ensemble decoding for multi-document summarization. arXiv preprint arXiv:2006.08748.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. The Journal of the Acoustical Society of America, 62(S1):S63–S63.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 3146–3154.

Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. 2022. Ctrleval: An unsupervised reference-free metric for evaluating controlled text generation. arXiv preprint arXiv:2204.00862.

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.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6232–6243, 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.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ArXiv, abs/2107.13586.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net.

Ruibo Liu, Jason Wei, and Soroush Vosoughi. 2021b. Language model augmented relevance score. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6677–6690, Online. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021c. Noisy self-knowledge distillation for text summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 692–703, Online. 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, abs/1907.11692.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed H Awadallah, and Dragomir Radev. 2021. Dyle: Dynamic latent extraction for abstractive long-input summarization. arXiv preprint arXiv:2110.08168.

Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pages 74–82.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Multi-document summarization with determinantal point process attention. Journal of Artificial Intelligence Research, 71:371–399.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5107–5116, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning, 242(1):29–48.

Yun-Zhu Song, Yi-Syuan Chen, and Hong-Han Shuai. 2022. Improving multi-document summarization through referenced flexible extraction with credit-awareness. arXiv preprint arXiv:2205.01889.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems, 33:17283–17297.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H Awadallah, Dragomir Radev, and Rui Zhang. 2021. Summˆ n: A multi-stage summarization framework for long input dialogues and documents. arXiv preprint arXiv:2110.10150.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies, pages 5905–5921, Online. Association for Computational Linguistics.

Fangwei Zhu, Shangqing Tu, Jiaxin Shi, Juanzi Li, Lei Hou, and Tong Cui. 2021. TWAG: A topic-guided Wikipedia abstract generator. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4623–4635, Online. Association for Computational Linguistics.

Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. arXiv preprint arXiv:2103.10685.

A Template details for salience estimation

For each source document, we assign at least one basic template's perplexity score. The basic prompting template should fit the context of all documents. So we use a general sentence: "This document is about k" to insert into the introduction or conclusion position of the document.

If a conclusion about the keyword's attribute can inferred from the document or the document contains salient information to answer a question about the keyword, then we assume that the document is salient. Therefore, we design two domain-specific template types for the extractive stage of Multi-document Summarization task: *conclusion* and *question* in Table 11. Besides, we also tried using no prompting template, we call it `none_prompt` .

Besides basic general prompting templates, we also design special templates for each domain to create more features for calculating the perplexity. Take animal domain for example, the $\texttt{conclusion}_1$ to $\texttt{conclusion}_4$ and $\texttt{question}_0$ to $\texttt{question}_9$ in Table 11 are the additional special templates.

B Implementation details

In the extractive stage, we load GPT2 checkpoint from *transformers* library[3] to calculate the perplexity of each filled prompting template. It costs about 15 hours to finish scoring one domain's documents on a single NVIDIA GeForce RTX 2080. For tf-idf, we use the *nltk* library[4] to count the term frequency and iverse document frequency. For random order, we shuffle the input documents randomly and send them into abstractive models.

In the abstractive stage, since the max input length of LED is 16384 which is much larger than BART's 1024 limit, their memory and time cost is quite different for fine-tuning. We train BART on a single NVIDIA GeForce RTX 2080 for 1 day (16 epochs) and train LED on a single NVIDIA GeForce RTX 3090 for 5 days (5 epochs). The learning rate is 1e-4 for the first epoch and decays to 1e-5 for other epochs. During interface, we use beam search decoding strategy with a beam size of 16, a minimum decoding length of 55, and a maximum decoding length of 120.

Note that we conduct the ablation study on the input token numbers of LED using early-stopping setting, where the training process will stop at the second epoch. Because the training LED until fitting costs too much GPU time and our GPU resource is limited, we have to compare the LED model under low-resource setup in the ablation study.

[3] https://huggingface.co/docs/transformers/index

[4] https://www.nltk.org/

Type	Template
none_prompt	d
conclusion$_0$	d This document is about k.
conclusion$_1$	d k's distribution is mentioned in above sentences.
conclusion$_2$	d This document introduces subspecies of k.
conclusion$_3$	d This document describes k.
conclusion$_4$	d This document introduces conservation status of k.
question$_0$	Where does k live? d
question$_1$	What is the Taxonomy of k? d
question$_2$	What are the Species of k? d
question$_3$	What are the Subspecies of k? d
question$_4$	What does the k feed on? d
question$_5$	Where does k live? d
question$_6$	What is the Diet of k? d
question$_7$	What is the Behaviour of k? d
question$_8$	What is the Breeding of k? d
question$_9$	What is the Conservation Status of k? d

Table 11: The general and special domain templates on the animal domain of WikiCatSum dataset. Notice that none_prompt is used as a backbone which represents the document without adding any prompts. And the conclusion$_0$ is the general template. Other templates are additional templates designed especially for the animal domain.