An Efficient Coarse-to-Fine Facet-Aware Unsupervised Summarization Framework based on Semantic Blocks

Xinnian Liang[1]*, Jing Li[2], Shuangzhi Wu[3], Jiali Zeng[3], Yufan Jiang[3], Mu Li[3], Zhoujun Li[1]†
[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2] School of Information Renmin University of China, Beijing, China
[3]Tencent Cloud Xiaowei, Beijing, China
{xnliang,lizj}@buaa.edu.cn; heylijing@126.com
{frostwu,lemonzeng,garyyfjiang}@tencent.com,limugx@qq.com;

Abstract

Unsupervised summarization methods have achieved remarkable results by incorporating representations from pre-trained language models. However, existing methods fail to consider efficiency and effectiveness at the same time when the input document is extremely long. To tackle this problem, in this paper, we proposed an efficient Coarse-to-Fine Facet-Aware Ranking (C2F-FAR) framework for unsupervised long document summarization, which is based on the semantic block. The semantic block refers to continuous sentences in the document that describe the same facet. Specifically, we address this problem by converting the one-step ranking method into the hierarchical multi-granularity two-stage ranking. In the coarse-level stage, we propose a new segment algorithm to split the document into facet-aware semantic blocks and then filter insignificant blocks. In the fine-level stage, we select salient sentences in each block and then extract the final summary from selected sentences. We evaluate our framework on four long document summarization datasets: Gov-Report, BillSum, arXiv, and PubMed. Our C2F-FAR can achieve new state-of-the-art unsupervised summarization results on Gov-Report and BillSum. In addition, our method speeds up 4-28 times more than previous methods.[1]

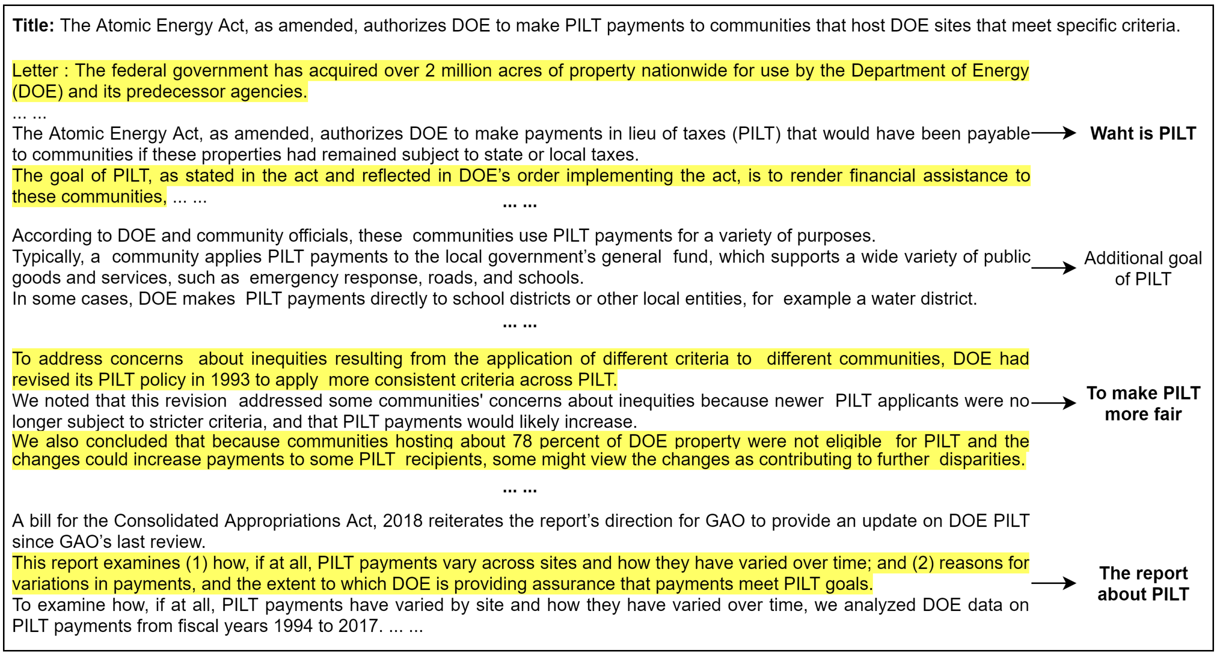
1 Introduction

The text summarization task aims to condense a document or a set of documents into several sentences and keep the primary information. Recent years, both supervised (Liu and Lapata, 2019; Liu and Liu, 2021; Liu et al., 2021b) and unsupervised (Zheng and Lapata, 2019; Dong et al., 2021b; Liang et al., 2021, 2022) methods have made significant improvements over short documents with the development of semantic representations from Pre-trained Language Models (PLMs). Due to the noise and complexity of the increased input and output length, long-form document summarization is still a challenge (Tay et al., 2021; Akiyama et al., 2021; Grail et al., 2021). Compared with supervised one, unsupervised methods do not rely on large amounts of labeled data and have no limitation on input length. In addition, unsupervised methods can be easily adapted to data from different domains, types, and languages. In this paper, we focus on unsupervised extractive methods for long document summarization.

Most unsupervised extractive methods are graph-based (Zheng and Lapata, 2019; Dong et al., 2021b; Liang et al., 2021, 2022). They represent document sentences as nodes in a graph, where the edge value is the similarity between sentences. Then, they measure the importance of each node via computing the degree centrality (Radev et al., 2000) or running PageRank (Brin and Page, 1998) algorithm. Liang et al. (2021) pointed out that centrality-based methods always tend to select sentences within the same facet (i.e. aspect, sub-topic) and proposed a facet-aware ranking (FAR) method to tackle this problem. FAR forces a centrality-based model to select summary sentences from different facets by incorporating the relevance between the candidate summary and the document. However, this method faces two problems when the document is extremely long: 1) As the input length increases, the document will have more noise and insignificant facets. The relevance computation between the candidate summary and the document may cause the facet-aware ranking to be influenced by insignificant facets. 2) The running time of FAR will rise rapidly as the number of extracted sentences increases. Due to FAR needs to compute the relevance score number of combinations C_m^k times, where k is the number of extracted summary sentences and m is the number of candidate salient sentences.

*Contribution during internship at Tencent Inc.
†Corresponding Author
[1] https://github.com/xnliang98/c2f-far

Proceedings of the 29th International Conference on Computational Linguistics, pages 6415–6425
October 12–17, 2022.



Title: The Atomic Energy Act, as amended, authorizes DOE to make PILT payments to communities that host DOE sites that meet specific criteria.

Letter : The federal government has acquired over 2 million acres of property nationwide for use by the Department of Energy (DOE) and its predecessor agencies.

... ...

The Atomic Energy Act, as amended, authorizes DOE to make payments in lieu of taxes (PILT) that would have been payable to communities if these properties had remained subject to state or local taxes. → **Waht is PILT**

The goal of PILT, as stated in the act and reflected in DOE's order implementing the act, is to render financial assistance to these communities, ...

... ...

According to DOE and community officials, these communities use PILT payments for a variety of purposes.

Typically, a community applies PILT payments to the local government's general fund, which supports a wide variety of public goods and services, such as emergency response, roads, and schools. → Additional goal of PILT

In some cases, DOE makes PILT payments directly to school districts or other local entities, for example a water district.

... ...

To address concerns about inequities resulting from the application of different criteria to different communities, DOE had revised its PILT policy in 1993 to apply more consistent criteria across PILT.

We noted that this revision addressed some communities' concerns about inequities because newer PILT applicants were no longer subject to stricter criteria, and that PILT payments would likely increase. → **To make PILT more fair**

We also concluded that because communities hosting about 78 percent of DOE property were not eligible for PILT and the changes could increase payments to some PILT recipients, some might view the changes as contributing to further disparities.

... ...

A bill for the Consolidated Appropriations Act, 2018 reiterates the report's direction for GAO to provide an update on DOE PILT since GAO's last review.

This report examines (1) how, if at all, PILT payments vary across sites and how they have varied over time; and (2) reasons for variations in payments, and the extent to which DOE is providing assurance that payments meet PILT goals. → **The report about PILT**

To examine how, if at all, PILT payments have varied by site and how they have varied over time, we analyzed DOE data on PILT payments from fiscal years 1994 to 2017. ... ...

Figure 1: An example from the Gov-Report dataset to introduce the process of our method. "..." refers to the omissions of context sentences due to space limitations. Highlight sentences refer to the final extracted summary sentences. The content of the arrow pointed is the facet description of the left semantic block. Bold facets represent vital facet-aware semantic blocks of the final summary.

To tackle these problems, in this paper, we propose a novel Coarse-to-Fine Facet-Aware Ranking (C2F-FAR) Framework based on semantic blocks, which consists of two stages with different granularities: semantic blocks and sentences. The semantic block means continuing sentences that describe the same facet. We use a simple example in Fig. 1 to describe the motivation for building two stages. Fig. 1 shows four facet-aware semantic blocks. Each block contains continuous sentences describing the same facet, which is listed on the right. From the coarse-level view, we should first filter blocks with unimportant facets in the document, e.g. the block related to "additional goal of PILT" in Fig. 1. Then, from the fine-level view, we should select proper sentences in each block, which are more relevant to the block facet. Note that we only show the most relevant sentences with the facet of each semantic block and omit unrelated sentences due to the space limitation. Finally, the highlighted sentences should be selected as the summary.

Following this intuitive process, we designed our framework with a coarse-level stage and a fine-level stage. The coarse-level stage aims to select several salient facet-aware semantic blocks for the fine-level stage. We first segment the document into facet-aware semantic blocks by our proposed new document segmentation algorithm, which is inspired by TextTiling (Hearst, 1997). Then, we filter insignificant facets via a coarse-level centrality estimator to measure the salience of blocks. The fine-level stage aims to select final summary sentences from previously selected blocks. We first select candidate sentences in each block to represent its facet by simply computing relevance between sentences and the block. Finally, we extract the final summary from candidate sentences by sentence-level centrality-based estimator. Overall, the coarse-level stage can identify all facets of the document effectively and filter insignificant ones. The fine-level stage can reduce the influence of facets with many sentences by only selecting several related sentences for the final ranking. This framework with a hierarchical coarse-to-fine structure can guarantee effective and efficient long document summarization.

We evaluate the effectiveness and efficiency of our C2F-FAR on four long-document summarization datasets with two different metrics. Our method achieves new state-of-the-art performance on Gov-Report and BillSum. It is comparable to strong baselines on arXiv and PubMed. Besides, our method can achieve a speedup of 4-28 times more than two strong baselines.

2 Methodology

We show the workflow of our proposed coarse-to-fine facet-aware ranking (C2F-FAR) framework in Fig. 2. After encoding the document into sentence embeddings, the workflow contains two main stages and each stage contains two steps.

(1) In the coarse-level stage, we first employ a document segmentation algorithm to split the document into coarse-level semantic blocks and we call them facet-aware semantic blocks. Then, we score all blocks via the centrality estimator and select top-ranked blocks for the next fine-level stage.

(2) In the fine-level stage, we first select several sentences of each facet-aware semantic block, which can cover the main facet of each block. Then, we employ a sentence-level centrality estimator to score selected sentences and extract the final summary.

We describe the details of each step in the following sections.

2.1 Sentence Embeddings

Formally, let $\mathcal{D}$ indicate a long document containing n sentences $\{s_1, \ldots, s_n\}$. In this paper, we employ pre-trained language model to obtain the sentence embeddings $\{v_1, \ldots, v_n\}$. Specifically, we employ an improved BERT (Devlin et al., 2019a) from previous work PacSum (Zheng and Lapata, 2019) to represent each sentence s_i with the hidden state v_i of "[CLS]" token. This improved BERT can obtain better sentence semantic representation.

2.2 The Coarse-Level Stage

The coarse-level stage contains two steps: document segmentation and coarse-level centrality estimator. The document segmentation splits the document into semantic blocks. The coarse-level centrality estimator employs a directed centrality score to measure the importance of each facet-aware semantic block. After the coarse-level stage, we only keep top-ranked $\alpha \times m$ semantic blocks of the whole document, where m is the number of facet-aware semantic blocks and α is a hyper-parameter used to control the ratio of reserved important blocks (default $\alpha = 0.5$).

2.2.1 Document Segmentation Algorithm

We propose a simple but effective document segmentation algorithm to split the input document into facet-aware semantic blocks. This algorithm is based on the assumption that when sentences with adjacent positions are semantically similar, they focus on the same facet (Skorochod'ko, 1971). As shown in Fig. 3, the algorithm aims to select some potential segmentation points to segment the document into several facet-aware semantic blocks $\mathcal{P}_1 = \{s_{p_1^s}, ..., s_{p_1^e}\}$, ..., $\mathcal{P}_m = \{s_{p_m^s}, ..., s_{p_m^e}\}$. Our proposed document segmentation algorithm is inspired by TextTiling (Hearst, 1997). It contains two steps: similarity measure and segmentation point identification.

In the similarity measure step, we compute the similarity of sentences on both sides of the potential segmentation point g_i. Each side select w sentences and apply mean operation method over their vectors to obtain global representations $b_i^l = \frac{1}{w}\sum_{j=i-w+1}^{i} v_j$ and $b_i^r = \frac{1}{w}\sum_{j=i+1}^{i+w} v_j$, where b_i^l and b_i^r refer to the left and right side block with w sentences, respectively. The similarity of the sentence on both sides of the potential segmentation point g_i is computed by cosine similarity $sim_i = \frac{b_i^l \cdot b_i^r}{||b_i^l|| ||b_i^r||}$.

Then, we apply the moving average on the similarity list of potential segmentation points $\{sim_1, ..., sim_{n-1}\}$ to get a smooth similarity list with Equ. (1)

$$\hat{sim}_i = \frac{1}{2\hat{w}+1} \sum_{j=i-\hat{w}}^{i+\hat{w}} sim_j \tag{1}$$

where the $\hat{w}$ is the window size used for moving average operation and the similarity list is refactored as $\{\hat{sim}_1, ..., \hat{sim}_{n-1}\}$. In this paper, the window size w and $\hat{w}$ are all set as 2.

The segmentation point identification step is based on the smooth similarity list. We show an intuitive similarity curve in Fig. 4. If the value of $\hat{sim}_i$ is low, the facets in the left and right blocks are different. So we should segment them with the point g_i. We can see that segmentation points g_3 and g_5 are the local minimum value of the curve in Fig. 4, which are suitable to segment the document.

We convert the similarity list of the potential segmentation point into depth score series $\{d_i\}_{i=1}^{n-1}$ by Equ. 2 to select proper segmentation points.

$$\begin{aligned} d_i = &\max\{(\hat{sim}_{i-1} - \hat{sim}_i), 0\} \\ &+ \max\{(\hat{sim}_{i+1} - \hat{sim}_i), 0\} \end{aligned} \tag{2}$$

When the similarity of the potential segmentation point is the local minimum value, it will become

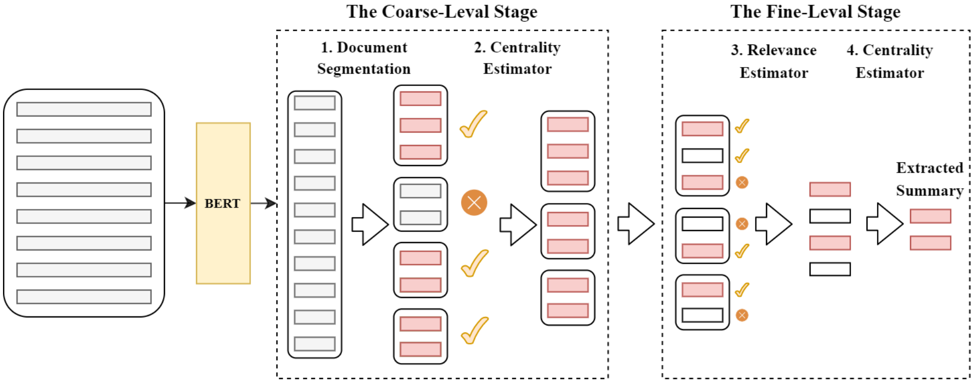


Figure 2: The workflow of our proposed coarse-to-fine facet-aware ranking framework.

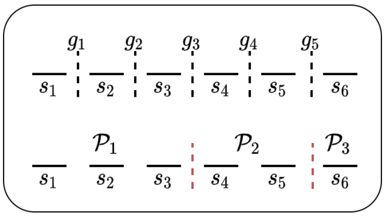


Figure 3: A diagram for document segmentation.

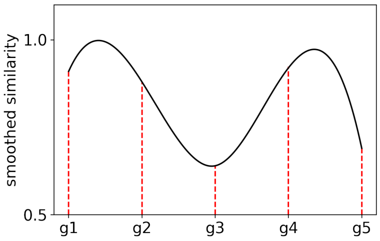


Figure 4: The smooth similarity curve.

the local maximum value after being converted into a depth score. If $d_i > \epsilon$, we choose the potential segmentation point g_i as the segmentation point. The ϵ is a threshold and is decided by the mean μ and standard deviation σ of the depth score series. We set $\epsilon = \mu + \lambda \cdot \sigma$, where λ is a hyper-parameter to control the granularity of segmentation. The greater the λ, the segmented block contains more sentences.

Finally, we can segment the whole document into some facet-aware semantic blocks $\mathcal{P}_1 = \{s_{p_1^s}, ..., s_{p_1^e}\}$, ..., $\mathcal{P}_m = \{s_{p_m^s}, ..., s_{p_m^e}\}$, like examples in the Fig. 3.

2.2.2 Coarse-Level Centrality Estimator

We introduce the coarse-level centrality estimator for filtering unimportant facet-aware semantic blocks in this section. We represent the semantic information of each block $\mathcal{P}_i$ by computing the average of sentence vectors contained in the block.

$$p_i = \frac{1}{|\mathcal{P}_i|} \sum_{s_i \in \mathcal{P}_i} (s_i) \tag{3}$$

The representations of blocks are $\{p_1, \ldots, p_m\}$. Then, we employ directed centrality (Zheng and Lapata, 2019) to score each block based on the assumption that the contribution of any two nodes' connection to their respective centrality is influenced by their relative position.

$$\mathcal{C}(p_i) = \lambda_1 \sum_{j<i}^{n} p_i \cdot p_j + \lambda_2 \sum_{j>i}^{n} p_i \cdot p_j \tag{4}$$

After that, we rank all blocks via directed centrality score $\mathcal{C}(p_i)$ and only keep top-ranked α percent semantic blocks for the next fine-level stage, where α is a hyper-parameter to control the ratio of reserved blocks.

2.3 The Fine-Level Stage

The fine-level stage contains two steps: relevance estimator and fine-level centrality estimator. The relevance estimator is used to select some sentences in each facet-aware semantic block, which can retain the main information of the block. The fine-level centrality estimator is applied to sentences from the previous relevance estimator and also employs the directed centrality score to extract the final summary.

2.3.1 Relevance Estimator

The relevance estimator simply computes the relevance between sentences and the block to select sentences to represent the facet in semantic blocks. This step is based on the assumption that each facet-aware semantic block only contains one facet. We employ cosine similarity to measure the relevance between sentence representation v_j and block representation p_i.

$$\mathcal{R}(s_j) = \frac{v_j \cdot p_i}{||v_j||||p_i||}, s_j \in \mathcal{P}_i \tag{5}$$

For each semantic block, we select top-ranked β sentences, where β is the average number of semantic block sentences, which is determined by the granularity of document segmentation. If the number of sentences in a block is lower than β, we keep all sentences. Then, we can get t candidate sentences $\{\hat{s}_1, \ldots, \hat{s}_t\}$ for the final summary selection.

2.3.2 Fine-Level Centrality Estimator

The final fine-level centrality estimator aims to select the final summary sentences from previous candidate sentences. The final fine-level centrality estimator measures the importance of each candidate sentence as follows:

$$\mathcal{C}(s_i) = \lambda_1 \sum_{j<i}^{t} v_i \cdot v_j + \lambda_2 \sum_{j>i}^{t} v_i \cdot v_j \tag{6}$$

where $s_i, s_j \in \{\hat{s}_1, \ldots, \hat{s}_t\}$. We select top-ranked k sentences as the final summary, where k is the average number of sentences of different datasets.

3 Experiments

3.1 Datasets

Datasets	#docs	document		summary	
		words	sen.	words	sen.
Gov-Report	973	9,409	304	657	23
BillSum	3,269	2,148	169	209	10
arXiv	6,440	4,938	206	220	10
PubMed	6,658	3,016	107	203	8

Table 1: Statistics information of Gov-Report, BillSum, arXiv, and PubMed datasets. We compute the average document and summary length in terms of words and sentences, respectively.

We evaluate our C2F-FAR on 4 datasets. The statistics information of them is shown in Tab. 1.

Gov-Report (Huang et al., 2021) is a large-scale long document summarization dataset containing 19,466 long reports published by U.S. Government Accountability Office (GAO) and Congressional Research Service (CRS). Documents and summaries in Gov-Report are significantly longer than other datasets.

BillSum (Kornilova and Eidelman, 2019) contains US Congressional bills and human-written references from the 103rd-115th (1993-2018) sessions of Congress. We found that previous works have some errors in the sentence segmentation of the dataset. We re-segmented this dataset with the StanfordNLP toolkit and conducted experiments on the basis of the new sentence segmentation.

arXiv and **PubMed** (Cohan et al., 2018) are two long scientific document summarization datasets from scientific papers.

3.2 Settings and Metrics

We employ sentence-BERT[2] from (Zheng and Lapata, 2019) to encode sentences in the document, which converts each sentence into a vector with 768 elements. The window size of the document segmentation algorithm is 2. The default setting of λ is 1.0 and α is 0.5.

We reported ROUGE-1/2/L scores with `ROUGE-1.5.5.pl` script[3] (Lin, 2004) and BertScore (Zhang* et al., 2020) of baselines and our methods. The ROUGE score is the lexical level metric to measure the similarity between extracted summary and gold summary. The BertScore[4] measures the semantic level similarity between the extracted summary and gold reference.

3.3 Baselines

We compare our method with recent strong unsupervised extractive summarization models.

Lead, which selects the first k tokens as a summary.

Oracle, which is the upper bound of extractive summarization methods. It selects sentences by computing ROUGE scores with the gold summary.

TextRank (Mihalcea and Tarau, 2004) and **LexRank** (Erkan and Radev, 2004), which are two traditional unsupervised ranking method based on TF-IDF and PageRank algorithm to select salient sentences.

TextRank(BERT), which employs embeddings from improved BERT to compute the edge weight of TextRank.

FAR (Liang et al., 2021), which defined the facet bias problem and proposed a facet-aware centrality method to tackle the bias problem.

[2] https://github.com/huggingface/transformers

[3] https://github.com/andersjo/pyrouge

[4] https://github.com/Tiiiger/bert_score

3.4 Evaluation of Summary Quality and Inference Time

We report the results of automatic and human evaluation of all systems to measure the extracted summary quality of our C2F-FAR. Besides, we also compare the inference time of our method with two strong baselines to prove the high efficiency of our method.

The automatic evaluation results of ROUGE score and BertScore are shown in the Tab. 2 and Tab. 3. These two scores measure the lexical and semantic level similarity between extracted summary and gold reference, respectively. All reported results of our C2F-FAR framework employed the default hyper-parameters $\lambda = 1$ and $\alpha = 0.5$. We can see that our C2F-FAR achieved new state-of-the-art results on Gov-Report and BillSum in unsupervised methods. The performance of our method also is better than PacSum and comparable to FAR on the other two datasets: arXiv and PubMed. We will analyze the reason for the results on arXiv and PubMed in the discussion section. Interestingly, there is no big difference between the two versions of TextRank. We guess that the iterative algorithm based on PageRank is not sensitive to the similarity measure methods.

To evaluate the ability of our C2F-FAR in reducing facet bias and improving the quality of extracted summaries, we asked 3 human annotators to evaluate the extracted summaries of C2F-FAR and FAR with the gold reference summary. Three annotators were given extracted and gold summary. Then they were asked to give 0-2 scores for facets coverage (whether the extracted summary contains most primary facets) and quality (the comprehensive feelings of the extracted summary) of 20 random sampled examples from test sets of BillSum and 20 random sampled examples from test sets of Gov-Report (0-bad, 1-normal, 2-good). The results of FAR in terms of facets coverage is 1.16 and quality is 1.03. Our C2F-FAR performs significantly better ($p < 0.05$ with Mann-Whitney U tests) than FAR whose facets coverage is **1.38** and quality is **1.15**.

To test the inference time of our method, we randomly select 100 examples from the test set of each dataset and ensure that the average input length of these 100 examples is the same as the average length of the test set. Then, we run each method 10 times and report the average inference time of them on four datasets. We can see Fig. 5 and find that our method is far ahead of the other two methods in inference time, and this advantage becomes more obvious as the length of the input document increases.

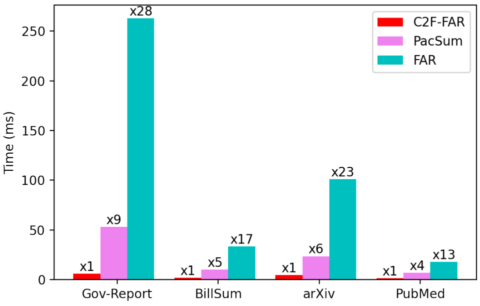


Figure 5: The inference time of each system. Each time is the average of multiple runs (10 times). ”$\times N$“ means the running time is N times (rounded up) of our method.

Overall, compared with other methods, our method takes into account both efficiency and effectiveness. In addition, our framework also can adjust the specific ranking method in each step for datasets with different types and domains, which makes it flexible.

4 Analysis

In this section, we first analyze the parameter sensitivity of our C2F-FAR and then discuss the reason why our method is inferior to the FAR on arXiv and PubMed via facets analysis of extracted sentences.

4.1 Impact of Hyper-parameters

In this section, we will analyze the parameter sensitivity of two hyper-parameters in our C2F-FAR framework: 1) λ is used to control the granularity of the document segmentation algorithm; 2) α is used to control the ratio of reserved blocks of the coarse-level centrality estimator. We can see the relationship between compression ratio and λ in the Tab. 4. The default setting of $\lambda = 1$ has an impressive compression ratio on two datasets.

We fix α and show the change of the ROUGE-1 score while λ changes in Fig. 6. We can find that the performance is best when $\lambda = 1.0$, and there is little change when $\lambda \in [0, 2]$. This shows that our algorithm is stable. You can set a larger λ to get a faster running speed while ensuring good performance. We set the value range of λ between 0.0 and 2.5 because when λ is less than 0, the most segmented blocks contain one sentence. Then the

Models	Gov-Report				BillSum			
	R-1	R-2	R-L	BS-F	R-1	R-2	R-L	BS-F
Oracle	74.87	49.02	72.48	88.83	65.24	47.09	58.81	86.29
Lead	50.94	19.53	48.45	83.47	40.53	18.28	34.15	80.24
LexRank	40.16	8.85	37.65	82.48	34.39	10.05	28.93	79.76
TextRank(TF-IDF)	53.19	23.12	49.86	84.83	40.04	16.12	32.64	80.81
TextRank(BERT)	56.00	22.42	52.86	85.10	38.05	12.99	31.46	80.02
PacSum	56.89	26.88	54.33	85.02	41.11	17.24	34.54	81.33
FAR	57.51	27.54	54.94	85.38	41.53	17.44	34.84	81.21
C2F-FAR	**57.98**	**27.63**	**55.33**	**86.62**	**42.53**	**17.85**	**35.58**	**81.57**

Table 2: Results on Gov-Report and BillSum test set. BS-F refers to F_1 of the BertScore.

Models	arXiv				PubMed			
	R-1	R-2	R-L	BS-F	R-1	R-2	R-L	BS-F
Oracle	53.88	23.05	34.9	87.06	55.05	27.48	38.66	87.05
Lead	33.66	8.94	22.19	82.97	35.63	12.28	25.17	80.43
LexRank	33.85	10.73	28.99	80.42	39.19	15.87	34.53	83.21
TextRank(TF-IDF)	36.59	10.06	30.29	82.49	38.66	15.87	34.53	82.43
TextRank(BERT)	34.68	8.78	30.05	81.19	39.43	12.89	34.66	83.39
PacSum	38.58	11.12	33.5	81.78	39.79	14.00	36.09	83.43
FAR	**40.92**	**13.75**	**35.56**	**83.74**	**41.98**	**16.74**	**37.58**	**83.89**
C2F-FAR	39.32	11.65	34.28	82.04	40.12	14.79	36.91	83.50

Table 3: Results on arXiv and PubMed test set. BS-F refers to F_1 of the BertScore.

Datasets	BillSum			Gov-Report		
λ	β	Para.	Comp.	β	Para.	Comp.
0	3	70	41%	3	120	39%
0.5	4	45	27%	5	74	24%
1	6	27	16%	10	44	14%
1.5	11	15	9%	15	26	9%
2	20	8	5%	20	15	5%
2.5	36	4	2%	36	4	1%

Table 4: Parameters affected by λ on two datasets. Para. means the average number of blocks with different hyper-parameters λ. Comp. means the ratio of the number of blocks to the number of sentences. β is the average number of sentences in a block.

following algorithms are equivalent to acting on the sentence-level structure.

We also fix λ and show the change of the ROUGE-1 score while α changes in Fig. 6. We can see that the second half of the curve is almost flat. This shows that the low centrality score of the segmented segment does not contribute to the final summary quality. The facets contained in these blocks are not important to the whole document. We can filter them with α in the coarse-level step and achieve a faster running speed.

The analysis of the two hyper-parameters proves that our C2F-FAR framework can employ simple hyper-parameter settings to improve the running speed of the algorithm while ensuring the quality of the summary.

4.2 Facets of Extracted Sentences

	Gov-Report	BillSum	arXiv	PubMed
#fac.	11.1	7.0	3.8	3.2
#sen.	20	10	10	7
#sen./#fac.	1.80	1.42	2.63	2.19

Table 5: #fac. refers to the average number of facet-aware semantic blocks, which contain extracted sentences. #sen. refers to the number of extracted sentences. #sen./#fac. refers to the average number of sentences from each block. Extracted sentences are from the Oracle system.

In Tab. 5, we employ the extracted sentences from the Oracle system to analyze the character-

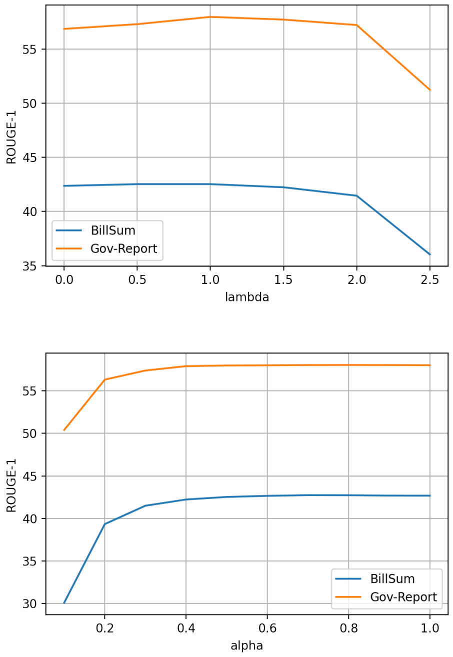


Figure 6: Impact of hyper-parameters λ and α.

istics of four datasets. The granularity of the document segmentation algorithm is $\lambda = 1$. We can see that selected summary sentences in arXiv and PubMed datasets distribute in fewer facet-aware semantic blocks than those in Gov-Report and BillSum. Our model tends to select summary sentences from more blocks, thus achieving better performance in Gov-Report and BillSum datasets.

By observing the extracted summary sentences from the Oracle system and combining the results in Tab. 5, we can roughly get the reason why our model is not as good as FAR on these datasets: the contents of the document and the summary is more concentrated on 3-4 facets of the document. Besides, the extracted sentences of them are mainly distribute at the start or end part (introduction and conclusion) of the document (Dong et al., 2021b). However, our method is more inclined to select summary sentences from more blocks and select many sentences in the middle part of the document. This leads to our method not performing so well on these two datasets.

5 Related Work

5.1 Long Document Summarization

Thanks to the development of Transformer-based (Vaswani et al., 2017) Pre-trained Language Models (PLMs), such as BERT (Devlin et al., 2019b), recent summarization models (Liu and Lapata, 2019; Zhang et al., 2019a; Li et al., 2020; Lewis et al., 2020; Zhong et al., 2020; Liu and Liu, 2021; Liu et al., 2021b) achieved excellent performance in short document summarization. However, these models can not be simply transferred to long document summarization due to both salient and noise content increasing according to the increase of the input text. How to summarize the long-form document, including books (Mihalcea and Ceylan, 2007), patents (Sharma et al., 2019), scientific publications (Qazvinian and Radev, 2008; Cohan et al., 2018), etc., is an important and long-standing challenge.

Most recent works for long-form document summarization are supervised and mainly tackle this problem through two angles. The first angle tends to design more efficient self-attention mechanisms to reduce the complexity. (Child et al., 2019; Kitaev et al., 2020; Beltagy et al., 2020; Zaheer et al., 2020; Huang et al., 2021; Tay et al., 2021; Dong et al., 2021a) The other angle employed the condense-then-generate paradigm (Cohan et al., 2018; Xu and Durrett, 2019; Zhang et al., 2019b; Lebanoff et al., 2019; Zhu et al., 2020; Akiyama et al., 2021; Grail et al., 2021). This paradigm first employs sentence/discourse-level structure to select salient sentences and then generates the summary based on them. This paradigm is intuitive and similar to the behavior of humans summarizing a long document. Our method also borrows some ideas from it.

5.2 Unsupervised Summarization

Most traditional unsupervised summarization methods are graph-based and extractive (Radev et al., 2000; Mihalcea and Tarau, 2004; Radev et al., 2000; Erkan and Radev, 2004; Wan, 2008). They represent the document as a graph, where each sentence is a node with a weighted edge which is the similarity between nodes. They rank sentences via computing centrality with node degree or PageRank algorithm (Brin and Page, 1998). Recently, many unsupervised works (Chu and Liu, 2019; Zhou and Rush, 2019; Zheng and Lapata, 2019; Yang et al., 2020; Xu et al., 2020; Liu et al., 2021a; Dong et al., 2021b; Liang et al., 2021) combined traditional methods with PLMs and achieved fantastic performance.

Zheng and Lapata (2019) first employed BERT to enhance similarity measure for graph-based ranking and proposed a directed degree centrality computation method. Dong et al. (2021b) pointed out that the previous method is not suitable for long scientific papers and proposed a hierarchical discourse-based unsupervised ranking method. Liang et al. (2021) found that they all ignored the facet-bias problem (Mao et al., 2020), which is ubiquitous in unsupervised methods and proposed a facet-aware ranking method FAR. However, as the document length increases, they cannot extract proper sentences which cover vital facets of the document, from rapidly increased insignificant facets.

6 Conclusion

In this paper, we focus on unsupervised long document summarization tasks, which is a vital and long-standing challenge in text summarization. To obtain summary sentences efficiently and effectively, we proposed a novel coarse-to-fine facet-aware ranking framework. Our method can achieve new state-of-the-art results on two datasets. Experiments show that our approach is effective and efficient for the long document summarization task. In future work, we will investigate how to refactor this process into an end-to-end paradigm.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62276017, U1636211, 61672081), the 2022 Tencent Big Travel Rhino-Bird Special Research Program, and the Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2021ZX-18).

References

Kazuki Akiyama, Akihiro Tamura, and Takashi Ninomiya. 2021. Hie-BART: Document summarization with hierarchical BART. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 159–165, Online. Association for Computational Linguistics.

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

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30:107–117.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proc. ICLR2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232, Long Beach, California, USA.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL*, pages 615–621, New Orleans, Louisiana.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2021a. A survey of natural language generation. *arXiv preprint arXiv:2112.11739*.

Yue Dong, Andrei Mircea, and Jackie Chi Kit Cheung. 2021b. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, Online. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Quentin Grail, Julien Perez, and Eric Gaussier. 2021. Globalizing BERT-based transformer architectures for long document summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1792–1810, Online. Association for Computational Linguistics.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2175–2189, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Xinnian Liang, Jing Li, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2022. Improving unsupervised extractive summarization by jointly modeling facet and redundancy. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1546–1557.

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

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

Jingzhou Liu, Dominic J. D. Hughes, and Yiming Yang. 2021a. Unsupervised extractive text summarization with distance-augmented sentence graphs. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2313–2317, New York, NY, USA. Association for Computing Machinery.

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

Yixin Liu, Zi-Yi Dou, and Pengfei Liu. 2021b. RefSum: Refactoring neural summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1448, Online. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Yuning Mao, Liyuan Liu, Qi Zhu, Xiang Ren, and Jiawei Han. 2020. Facet-aware evaluation for extractive summarization. In *Proc. 58th Annu. Meet. Assoc. Comput. Linguist.*, pages 4941–4957, Online.

Rada Mihalcea and Hakan Ceylan. 2007. Explorations in automatic book summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 380–389, Prague, Czech Republic. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 689–696, Manchester, UK. Coling 2008 Organizing Committee.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

E. F. Skorochod'ko. 1971. Adaptive method of automatic abstracting and indexing. In *Proceedings of the IFIP Congress 71*, volume 2, pages 1179–1182.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xiaojun Wan. 2008. An exploration of document impact on graph-based multi-document summarization. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 755–762, Honolulu, Hawaii. Association for Computational Linguistics.

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

Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020. Unsupervised extractive summarization by pre-training hierarchical transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1784–1795, Online.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020. TED: A pretrained unsupervised summarization model with theme modeling and denoising. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1865–1874, Online.

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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019b. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL*, pages 5059–5069, Florence, Italy.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *ACL*, pages 6236–6247, Florence, Italy.

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

Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *ACL*, pages 5101–5106, Florence, Italy.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.