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

Xinnian Liang\textsuperscript{1}; Jing Li\textsuperscript{2}, Shuangzhi Wu\textsuperscript{3}, Jiali Zeng\textsuperscript{3}, Yufan Jiang\textsuperscript{3}, Mu Li\textsuperscript{3}, Zhoujun Li\textsuperscript{1}\textsuperscript{†}

\textsuperscript{1}State Key Lab of Software Development Environment, Beihang University, Beijing, China
\textsuperscript{2} School of Information Renmin University of China, Beijing, China
\textsuperscript{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 propose 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.\textsuperscript{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 $C_k^m$ times, where $k$ is the number of extracted summary sentences and $m$ is the number of candidate salient sentences.

\textsuperscript{1}Contribution during internship at Tencent Inc.
\textsuperscript{†}Corresponding Author
\textsuperscript{1}https://github.com/xnliang98/c2f-far
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 $D$ 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 $\alpha$ 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 $\mathcal{P}_1 = \{s_{p_1}, \ldots, s_{p_5}\}$, ..., $\mathcal{P}_m = \{s_{p_{m_1}}, \ldots, s_{p_{m_t}}\}$. 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)

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

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.

$$d_i = \max\{\hat{sim}_{i-1} - \hat{sim}_i, 0\}$$

$$+ \max\{\hat{sim}_{i+1} - \hat{sim}_i, 0\}$$

When the similarity of the potential segmentation point is the local minimum value, it will become
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 $\epsilon$ is a threshold and is decided by the mean $\mu$ and standard deviation $\sigma$ of the depth score series. We set $\epsilon = \mu + \lambda \cdot \sigma$, where $\lambda$ is a hyper-parameter to control the granularity of segmentation. The greater the $\lambda$, the segmented block contains more sentences.

Finally, we can segment the whole document into some facet-aware semantic blocks $P_1 = \{s_{p_1}, s_{p_2}, ..., s_{p_{\ell_1}}\}$, ..., $P_m = \{s_{p_{\ell_m}}, s_{p_{\ell_{m+1}}}, ..., s_{p_m}\}$, 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 $P_i$ by computing the average of sentence vectors contained in the block.

$$ p_i = \frac{1}{|P_i|} \sum_{s_i \in P_i} s_i $$

(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.

$$ C(p_i) = \lambda_1 \sum_{j<i} p_i \cdot p_j + \lambda_2 \sum_{j>i} p_i \cdot p_j $$

(4)

After that, we rank all blocks via directed centrality score $C(p_i)$ and only keep top-ranked $\alpha$ percent semantic blocks for the next fine-level stage, where $\alpha$ 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 $p_i$.

$$R(s_j) = \frac{v_j \cdot p_i}{||v_j|| ||p_i||}, s_j \in P_i$$

For each semantic block, we select top-ranked $\beta$ sentences, where $\beta$ 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 $\beta$, 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:

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

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

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

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 $\lambda$ is 1.0 and $\alpha$ is 0.5.

We reported ROUGE-1/2/L scores with ROUGE-1.5.5.pl script$^3$ (Lin, 2004) and BertScore$^4$ (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.

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) $\lambda$ is used to control the granularity of the document segmentation algorithm; 2) $\alpha$ is used to control the ratio of reserved blocks of the coarse-level centrality estimator. We can see the relationship between compression ratio and $\lambda$ in the Tab. 4. The default setting of $\lambda = 1$ has an impressive compression ratio on two datasets.

We fix $\alpha$ and show the change of the ROUGE-1 score while $\lambda$ 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 $\lambda$ to get a faster running speed while ensuring good performance. We set the value range of $\lambda$ between 0.0 and 2.5 because when $\lambda$ is less than 0, the most segmented blocks contain one sentence. Then the
Table 2: Results on Gov-Report and BillSum test set. BS-F refers to $F_1$ of the BertScore.

<table>
<thead>
<tr>
<th>Models</th>
<th>Gov-Report</th>
<th>BillSum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-2</td>
</tr>
<tr>
<td>Oracle</td>
<td>74.87</td>
<td>49.02</td>
</tr>
<tr>
<td>Lead</td>
<td>50.94</td>
<td>19.53</td>
</tr>
<tr>
<td>LexRank</td>
<td>40.16</td>
<td>8.85</td>
</tr>
<tr>
<td>TextRank(TF-IDF)</td>
<td>53.19</td>
<td>23.12</td>
</tr>
<tr>
<td>TextRank(BERT)</td>
<td>56.00</td>
<td>22.42</td>
</tr>
<tr>
<td>PacSum</td>
<td>56.89</td>
<td>26.88</td>
</tr>
<tr>
<td>FAR</td>
<td>57.51</td>
<td>27.54</td>
</tr>
<tr>
<td>C2F-FAR</td>
<td>57.98</td>
<td>27.63</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Models</th>
<th>arXiv</th>
<th>PubMed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-2</td>
</tr>
<tr>
<td>Oracle</td>
<td>53.88</td>
<td>23.05</td>
</tr>
<tr>
<td>Lead</td>
<td>33.66</td>
<td>8.94</td>
</tr>
<tr>
<td>LexRank</td>
<td>33.85</td>
<td>10.73</td>
</tr>
<tr>
<td>TextRank(TF-IDF)</td>
<td>36.59</td>
<td>10.06</td>
</tr>
<tr>
<td>TextRank(BERT)</td>
<td>34.68</td>
<td>8.78</td>
</tr>
<tr>
<td>PacSum</td>
<td>38.58</td>
<td>11.12</td>
</tr>
<tr>
<td>FAR</td>
<td>40.92</td>
<td>13.75</td>
</tr>
<tr>
<td>C2F-FAR</td>
<td>39.32</td>
<td>11.65</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Datasets</th>
<th>BillSum</th>
<th>Gov-Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>$\beta$</td>
<td>Para. Comp.</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>70 41%</td>
</tr>
<tr>
<td>0.5</td>
<td>4</td>
<td>45 27%</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>6 16%</td>
</tr>
<tr>
<td>1.5</td>
<td>11</td>
<td>15 9%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>8 5%</td>
</tr>
<tr>
<td>2.5</td>
<td>36</td>
<td>4 2%</td>
</tr>
</tbody>
</table>

Table 5: Parameters affected by $\lambda$ on two datasets. Para. means the average number of blocks with different hyper-parameters $\lambda$. Comp. means the ratio of the number of blocks to the number of sentences. $\beta$ is the average number of sentences in a block.

4.2 Facets of Extracted Sentences

We can filter them with $\alpha$ 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.

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


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.


