A Multi-Document Summarization Approach based on Hierarchical Clustering of Documents: Extracting both Commonality and Specificity of Documents

Anonymous ACL submission

Abstract

The multi-document summarization task requires the designed summarizer to generate a short text that covers the important information of original documents and satisfies content diversity. This paper proposes a multi-document summarization approach based on hierarchical clustering of documents. It utilizes the constructed class tree of documents to extract both the sentences reflecting the commonality of all documents and the sentences reflecting the specificity of some subclasses of these documents for generating the summary, so as to satisfy the coverage and diversity requirements of multi-document summarization. Comparative experiments on *DUC'2002-2004* datasets prove the effectiveness of considering both the commonality and specificity of documents for multi-document summarization. And the experiments on *DUC'2004* and *Multi-News* datasets show that our approach achieves competitive performance compared to the state-of-the-art unsupervised and supervised approaches.

1 Introduction

Automatic text summarization is becoming much more important because of the exponential growth of digital textual information on the web. Multi-document summarization, which aims to generate a short text containing all important information of original multiple documents, is a challenging focus of NLP research. A well-organized summary of multiple documents needs to cover the main information of all documents comprehensively and simultaneously satisfy content diversity. Extractive summarization approaches, which generate a summary by selecting a few important sentences from original documents, attract much attention because of its simplicity and robustness. This paper focuses on extractive multi-document summarization.

Most extractive multi-document summarization approaches splice all the sentences contained in the original documents into a larger text, and then generate a summary by selecting sentences from the larger text (Lamsiyah et al., 2021; Yang et al., 2014; Erkan and Radev, 2004). However, the task of summarizing multiple documents is more difficult than the task of summarizing single document. Simply transforming multi-document summarization task into summarizing single larger text completely breaks the constraints of documents on their sentences and lacks comparisons between documents, which results in the inability to extract the relevant information between documents, including extracting the common information (commonality) of all documents and the important specific information (specificity) of some subclasses of documents.

The centroid-based summarization approaches focus on the commonality of all documents or all sentences and they select sentences based on the centroid words of all documents (Radev et al., 2004; Rossiello et al., 2017) or the centroid embedding of all sentences (Lamsiyah et al., 2021). The clustering-based summarization approaches divide sentences into multiple groups and select sentences from each group (Yang et al., 2014; Sarkar, 2009). These approaches do not take into account the commonality and specificity of documents simultaneously.

Think about the process of human summarizing multiple documents: we would first describe the common information of all documents and then the important specific information of some subclasses of these documents respectively to satisfy the coverage and diversity requirements of multi-document summarization.

In this paper, inspired by the idea of human summarizing multiple documents, we propose a multi-document summarization approach based on hierarchical clustering of documents. Firstly, our model hierarchically clusters documents from top to bottom to build a class tree of documents. Next, our model traverses each node along the class tree from top to bottom, and selects sentences from

each node according to the similarity of sentences to the centroid embedding of the documents in the node and the dissimilarity to the centroid embedding of the documents not in the node, until the total length of the selected sentences reaches a pre-specified value. The sentence selected from the root node (containing all documents) reflects the commonality of all documents, and the sentence selected from each sub node (subclass) reflects the specificity of the subclass. Finally, all selected sentences are arranged according to the order of their corresponding nodes on the class tree to form a summary.

Experiments are performed on standard datasets, including the *DUC* datasets and the *Multi-News* dataset. Comparative experiments on *DUC'2002-2004* datasets prove that our approach considering both commonality and specificity of documents significantly outperforms the approaches considering only commonality or only specificity; And our approach (based on documents hierarchical clustering) outperforms the comparison approach based on sentences hierarchical clustering; Experiments on *DUC'2004* and *Multi-News* datasets show that our approach outperforms strong baselines and many competitive supervised and unsupervised multi-document summarization approaches, and yields comparable performances to the state-of-the-art supervised and unsupervised approaches.

Our approach is unsupervised and easy to implement, and can be used as a strong baseline for evaluating multi-document summarization systems.

2 Related Work

The related works include centroid-based and clustering-based summarization methods.

The centroid-based methods score each sentence in documents by calculating the similarity between the sentence and the centroid of all documents or all sentences, so as to identify the most central sentences to generate a summary (Radev et al., 2004; Rossiello et al., 2017; Lamsiyah et al., 2021). The centroid-based methods focus on the commonality property of all documents or all sentences. For example, MEAD (Radev et al., 2004) scores each sentence based on the centroid words (the words statistically important to multiple documents) it contains and two other metrics (positional value and first-sentence overlap). Rossiello et al. (2017) improves the original MEAD method, which exploits the word embedding representations to represent the centroid and each sentence, and scores each sentence based on the cosine similarity between the sentence embedding and the centroid embedding. Lamsiyah et al. (2021) exploits sentence embedding model to represent each sentence and the centroid (the mean of all sentence embeddings), and scores each sentence based on the cosine similarity between the sentence embedding and the centroid embedding, and two other metrics (sentence novelty and sentence position).

Many clustering-based extractive summarization methods cluster all sentences in documents and then select sentences from each sentence cluster to form a summary (Wang et al., 2008; Mohd et al., 2020; Rouane et al., 2019; Yang et al., 2014). For example, Wang et al. (2008) groups sentences into clusters by sentence-level semantic analysis and symmetric non-negative matrix factorization, and selects the most informative sentences from each sentence cluster. Mohd et al. (2020) represents each sentence as a big-vector using the Word2Vec model, and applies the k-means algorithm to cluster sentences, and then scores sentences in each sentence cluster based on various statistical features (i.e. sentence length, position, etc.). Rouane et al. (2019) uses the k-means algorithm to cluster sentences, and scores each sentence in each cluster based on the frequent itemsets of the cluster contained by the sentence. Yang et al. (2014) proposes a ranking-based sentence clustering framework to generate high quality sentence clusters, and uses a modified MMR-like approach to select highest scored sentences from the descending order ranked sentence clusters to form the summary.

3 Methodology

Our approach takes a set of documents and a pre-given summary length (or compression rate) as input, and outputs a multi-document summary. It consists of three steps: (1) pre-processing of documents, (2) hierarchical clustering of documents, and (3) sentence selection from the generated class tree of documents and summary generation.

3.1 Pre-processing

Pre-trained models are widely used in Natural Language Processing tasks. There are usually two ways to use the pre-trained models: (1) Feature Extraction based approach, which uses the pre-trained model learned from a large amount of textual data to encode texts of arbitrary length into vectors

of fixed length; (2) Fine-Tuning based approach, which trains the downstream tasks by fine-tuning the pre-trained models parameters. In this paper, we adopt the feature extraction based approach, where the pre-trained model is applied on the input documents to obtain the embedding representations of sentences and documents.

Using sentence embedding representations in extractive multi-document summarization has been proven to be effective (Lamsiyah et al., 2021), so our model uses pre-trained sentence embedding model to encode sentences. We can use two ways to obtain document embeddings: one is to directly obtain the document embedding by taking the document as the input of the pre-trained embedding model; the other is to obtain the document embedding based on sentence embeddings, e.g., the document embedding can be obtained by calculating the average of the sentence embeddings of all sentences a document contains.

Formally, given a set of documents D containing n documents $D = \{d_1, d_2, \cdots, d_n\}$. Firstly, our model splits each document $d_i \in D$ into sentences (denoted as $d_i = \{s_1^i, s_2^i, \cdots, s_{|d_i|}^i\}$) using Natural Language Toolkit (NLTK).[1] Next, our model maps each sentence in each document ($s_k^i \in d_i$) to a fixed-length vector (denoted as $\overrightarrow{s_k^i}$) using the pre-trained embedding model, and maps each document ($d_i \in D$) to a vector of the same length (denoted as $\overrightarrow{d_i}$).

3.2 Hierarchical Clustering of Documents

The proposed top-down hierarchical clustering algorithm for constructing the class tree of documents (denoted as H) includes the following steps:

Step 1: Specify the root node of *H*.

All documents in D form the root node. The root node is regarded as the first layer of H. (After *Step 1*, H contains only one layer.)

Step 2: Construct the next layer of *H*.

For each node of the last layer of H, our model uses the k-means algorithm[2] to divide the documents in the node into k sub nodes (k sub classes). All new sub nodes generated in this step constitute the next layer of H.

Step 3: Repeat *Step 2* until one of the following conditions is satisfied.

Condition 1: There is no node in the last layer of H can be divided using the k-means algorithm.

[1] nltk.tokenize

[2] sklearn.cluster.KMeans

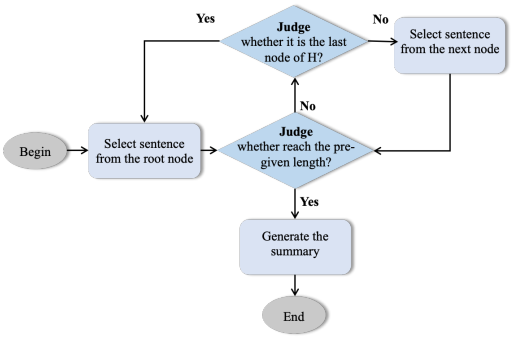

Figure 1: The flow chart of selecting sentences from the class tree.

Condition 2: The total number of nodes on H exceeds the number of sentences required for the summary (specified or estimated according to the pre-given summary length). (Because our model selects sentences from each node of H top-down until the pre-given value is reached.)

3.3 Generation of Summary

After the construction of the class tree (H), our model traverses the nodes on H from top to bottom and selects sentences from the nodes to generate a summary until the summary length reaches the pre-given length.

Section 3.3.1 introduces the overall flow of traversing the nodes on H, *Section 3.3.2* introduces the details of sentences scoring and selection in each node, and *Section 3.3.3* introduces the process of sorting sentences to form a summary.

3.3.1 Overall Flow of Traversing Class Tree

Fig. 1 displays the overall flow chart of traversing the nodes on the class tree H for selecting sentences.

The order of traversing the nodes on H follows two principles: (1) For different layers of H, traverse the layers from top to bottom; (2) For the nodes on the same layer, traverse the nodes in descending order by the number of documents contained in the nodes. Because under the limitation of the pre-given summary length, our model hopes that the selected sentences can cover as many documents as possible while increasing diversity.

As shown in Fig. 1, if the total length of the selected sentences does not reach the pre-given summary length after selecting sentence from the last node on the last layer of H, our model goes back to the first layer of H (the root node) to start the next iteration of selecting sentences, until the total length of all selected sentences in all iterations reaches the pre-given summary length.

3.3.2 Sentence Scoring and Selection in Each Node

Each node N_t on the class tree consists of multiple documents, denoted as $N_t = \{d_1^t, \cdots, d_{|N_t|}^t\}$. Each document $d_i^t \in N_t$ consists of multiple sentences, denoted as $d_i^t = \{s_1^i, s_2^i, \cdots, s_{|d_i^t|}^i\}$.

(1) Sentences Scoring.

Commonality-Specificity score.

The centroid of all documents in N_t represents the common core of these documents. It is reasonable to think that the sentences that are more similar to the centroid of N_t are more relevant to the documents in N_t, and the sentences that are less similar to the centroid of N_t are less relevant to the documents in N_t. Therefore, the Commonality-Specificity score of each sentence s_k^i in N_t can be calculated as the combination of its similarity to the centroid of N_t and its dissimilarity to the centroid of the documents not in N_t.

Our model builds the centroid embedding vector of N_t (denoted as $\overrightarrow{C_{N_t}}$) as the average of all document embedding vectors in it (as shown in Eq. (1)).

$$\overrightarrow{C_{N_t}} = \frac{1}{|N_t|} \sum_{i=1}^{|N_t|} \overrightarrow{d_i^t} \tag{1}$$

Where $|N_t|$ denotes the number of documents in N_t, and $\overrightarrow{d_i^t}$ is the document embedding vector of the i^{th} document in N_t.

Similarly, the centroid embedding vector of the documents not in N_t (denoted as $\overrightarrow{C_{\overline{N_t}}}$) is built as the average of all document embedding vectors not in N_t (i.e., $\overline{N_t} = \{d \,|\, d \in D \, and \, d \notin N_t\}$).

The Commonality-Specificity score of each sentence s_k^i in N_t is calculated as follows:

$$\begin{aligned} score^{CS}(s_k^i, N_t) = \delta * sim(\overrightarrow{s_k^i}, \overrightarrow{C_{N_t}}) \\ +(1-\delta)*(1-sim(\overrightarrow{s_k^i}, \overrightarrow{C_{\overline{N_t}}})) \end{aligned} \tag{2}$$

The value of $\delta \in [0, 1]$. The larger value of δ illustrates more attention to the relevance with the documents in N_t, and the smaller value of δ illustrates more attention to the irrelevance with the documents not in N_t. When $\delta = 1$, the $score^{CS}$ of each sentence in N_t only focuses on the relevance with the documents in N_t. Our model uses the cosine similarity[3] (denoted as sim) to calculate the similarity between embedding vectors.

[3] sklearn.cosine_similarity

The $score^{CS}$ is bounded in $[0, 1]$, and sentences with higher $score^{CS}$ are considered to be more relevant to the documents in N_k and more irrelevant to the documents not in N_k.

Non-redundant score and ***Position score***.

The Commonality-Specificity score can be used alone or in combination with other scores. We introduce two other scores: the Non-redundant score and the Position score.

To reduce the redundancy of summary, our model would assign lower Non-redundant scores to the sentences that are more similar to the sentences already selected in previous steps. Specifically, we use S^p to represent the collection of sentences already selected in previous steps, the Non-redundant score of each sentence s_k^i in N_t is calculated as the dissimilarity between s_k^i and the sentence most similar to s_k^i in S^p, which is described as follows:

$$score^{NR}(s_k^i, N_t) = 1 - \max_{s_p \in S^p}(sim(\overrightarrow{s_k^i}, \overrightarrow{s_p})) \tag{3}$$

The $score^{NR}$ is bounded in $[0, 1]$, and sentences with higher $score^{NR}$ are considered to be less redundant with the sentences already selected in previous steps. If S^p is Null (i.e., selecting the first sentence from the root node), the $score^{NR}$ of each sentence in the node is 1.

Sentence position is one of the most effective heuristics for selecting sentences to generate summaries, especially for news articles (Edmundson, 1969; Ouyang et al., 2010). We adopt the sentence position relevance metric (as Eq. (4)) introduced by Joshi et al. (2019) to calculate the Position score of each sentence in each document.

$$score^{P}(s_k^i) = \max(0.5, \exp(\frac{-\mathcal{P}(s_k^i)}{\sqrt[3]{|d_i|}})) \tag{4}$$

$\mathcal{P}(s_k^i)$ denotes the relative position of the k^{th} sentence s_k^i in the document d_i (starting by 1). The $score^{P}$ is bounded in $[0.5, 1]$. The first sentence in each document obtains the highest $score^{P}$. The $score^{P}$ of sentences decrease as their distances from the beginning of documents increase, and remain stable at a value of 0.5 after several sentences.

(2) Sentence Selection.

The final score of each sentence s_k^i in N_t can be defined as a linear combination of the three scores (as Eq. (5)).

$$\begin{aligned} score^{final}(s_k^i, N_t) = \alpha * score^{CS}(s_k^i, N_t) \\ +\beta * score^{NR}(s_k^i, N_t) + \gamma * score^{P}(s_k^i) \end{aligned} \tag{5}$$

Where $\alpha + \beta + \gamma = 1$, and $\alpha, \beta, \gamma \in [0, 1]$. Different values of α, β and γ represent different emphases on different scoring metrics, e.g., (*section 4.3 (1) and (2)*) uses only the Commonality-Specificity score (i.e. $\alpha = 1$); (*section 4.3 (3)*) uses the combination of the three scores.

Our model selects the sentence with the highest final score that have not been selected in previous steps from N_t. Only one sentence is selected from each node in each iteration because our model wants to traverse as many nodes as possible under the limitation of the pre-given summary length to increase the diversity of the generated summaries.

3.3.3 Summary Generation

After the process of selecting sentences from *H*, our model sorts the selected sentences to form a summary: (1) For the sentences selected from different nodes, our model sorts these sentences according to the traversal order of the nodes on *H* (following the two principles introduced in *Section 3.3.1*); (2) For multiple sentences selected from the same node (i.e., sentences selected in the first iteration are not enough), our model sorts the sentences according to the order in which they are selected.

The sentences selected from the root node express the commonality of all documents, and the sentences selected from each sub node express the specificity of the subclass. The way of sorting sentences forms a summary with a total-sub structure.

4 Experiment

4.1 Datasets and Evaluation Metrics

We evaluate the proposed approach on the standard multi-document summarization datasets, including *DUC'2002-2004* datasets[4] and the recently released *Multi-News* dataset[5]. Table 1 describes the details of these datasets. Each news set of *DUC* datasets contains approximately 10 documents on the same topic. Each news set of *Multi-News* dataset contains a different number of documents (from 1 to 10) on the same topic (Fabbri et al., 2019).

ROUGE is a standard evaluation metric for automatic document summarization (Lin, 2004), including *ROUGE-1*, *ROUGE-2*, *ROUGE-L* and *ROUGE-SU4* (denoted as *R-1*, *R-2*, *R-L* and *R-SU4* respectively). We use the ROUGE toolkit (version *1.5.5*), and adopt the same ROUGE settings[6] that are commonly used on the *DUC* datasets and *Multi-News* dataset for multi-document summarization. Guided by the state-of-the-art approaches, we report *ROUGE recall* on *DUC* datasets and *ROUGE F1-score* on the *Multi-News* dataset, respectively.

4.2 Experimental Settings

(1) Selection of Pre-trained Model

Lamsiyah et al. (2021) has studied multi-document summarization based on nine different pre-trained sentence embedding models, which verifies the effectiveness of sentence embedding representations for multi-document summarization, and shows that using different sentence embedding models would affect the performance of summarization. Among them, the *USE-DAN* model (Cer et al., 2018) is one of the best performing models.

In order to focus on our proposed approach and not affected by different embedding models, we use the *USE-DAN* model[7] to encode sentences. In order to unify the expression of sentences and documents and preserve the relationship between documents and sentences, we obtain the embedding vector of each document $d_i \in D$ by calculating the average of the sentence embedding vectors of all sentences it contains.

(2) Determination of Hyperparameters

Different values of hyperparameters affect the results of the proposed approach, and we determined their values both theoretically and experimentally.

Estimation of the hyperparameter *k* in k-means algorithm. Our model needs to select sentences not only from the root node of the class tree, but also from as many sub nodes as possible, to extract both commonality and specificity information of the input documents. Thus, when generating the sub nodes of the second layer of *H*, the value of *k* in k-means clustering should not be set too large. Otherwise, under the limitation of the pre-given summary length, the sub nodes participating in sentences selection cannot cover all input documents, resulting in the generated summary cannot contain the specificity information of some subclasses of the input documents.

The approximate number of sentences need to be selected for generating summaries can be estimated by *(average length of target summaries)* ÷

[4] DUC datasets

[5] Multi-News dataset

[6] ROUGE-1.5.5 with parameters "-n 2 -2 4 -u -m -r 1000 -f A -p 0.5" and "-l 100" for DUC'2002 and DUC'2003; "-b 665" for DUC'2004; "-l 264" for Multi-News.

[7] universal-sentence-encoder

Dataset	# news sets	# docs	# references (of each news set)	# words (of each sentence in docs)
DUC'2002	59	567	2	22.86
DUC'2003	30	298	4	25.43
DUC'2004	50	500	4	25.38
Multi-News	5622	15326	1	22.24

Table 1: Description of each dataset, including the number of news sets, the total number of documents (news), the number of reference summaries for each news set and the average length of sentences in source documents.

(average length of sentences in source documents), i.e., 4.65 for *DUC'2004*, 3.93 for *DUC'2003*, 4.37 for *DUC'2002*. Based on the estimation, when generating the sub nodes of the second layer, the hyperparameter *k* of the k-means clustering is set to be within the range of $[2, 4]$ (*minimum number of sentences estimated for different datasets*-1). For simplicity, when generating the sub nodes of the third layer and subsequent layers, we set *k* in k-means clustering to 2.

Estimation of the hyperparameters δ and α, β, γ in sentences scoring equation. The hyperparameter δ in $score^{CS}$ illustrates the concern for the relevance of each sentence to the documents in its own node, so theoretically it cannot be set too small. The hyperparameters α, β and γ in $score^{final}$ illustrate different degrees of attention to the three scores. Theoretically α cannot be set too small because our approach focuses on extracting both commonality and specificity of documents for multi-document summarization.

In order to determine the exact values of these hyperparameters, we employed a procedure similar to that used by Lamsiyah et al. (2021) and Joshi et al. (2019). We built a small held-out set by randomly sampling 25 clusters with different length of reference summaries from the validation set of the *Multi-News* dataset. Then we performed a grid search for these hyperparameters: δ, α, β, $\gamma \in [0, 1]$ with constant step of 0.1 under the condition $\alpha + \beta + \gamma = 1$, $k \in [2, 4]$ with constant step of 1. Finally, the obtained values of the hyperparameters are 3, 0.9, 0.8, 0.1, 0.1 for k, δ, α, β and γ respectively, which are consistent with the theoretical analysis of these hyperparameters.

4.3 Evaluations

We evaluate the proposed approach by three parts: (1) verify the effectiveness of considering both commonality and specificity of documents for multi-document summarization; (2) verify the effectiveness of using documents hierarchical clustering for multi-document summarization; (3) verify the effectiveness of the proposed multi-document summarization approach. Due to the randomness of k-means, each experiment has been run three times to get the intermediate results. (We have uploaded the generated summaries for each experiment as supplementary material.)

(1) Verify the effectiveness of considering both commonality and specificity of documents.

In order to not affected by other factors, our model uses only the *Commonality-Specificity score* to score sentences, denoted as ***Ours*$_{\mathbf{CS}}$** (i.e., $\alpha = 1, \beta = 0, \gamma = 0$). The comparison experiments are designed as follows:

***Comp*$_\mathbf{1}$**: score sentences by calculating the cosine similarity between sentence embeddings and the centroid embedding of all documents, and select the top-ranked sentences.

***Comp*$_\mathbf{2}$**: use k-means algorithm to cluster documents, then use the *Commonality-Specificity score* to score sentences in each sub cluster and select the highest scored sentence from each sub cluster.

***Comp*$_\mathbf{3}$**: same as *Comp*$_2$ but score sentences in each sub cluster by calculating the cosine similarity between each sentence embedding and the centroid embedding of the sub cluster.

Table 2 displays the results on three *DUC* datasets. *Comp*$_1$ focuses on the commonality of all documents (similarity to the centroid of all documents); *Comp*$_2$ focuses on the specificity of each subcluster (the combination of similarity to the centroid of the subcluster and dissimilarity to the centroid of documents not in the subcluster); *Comp*$_3$ generates summaries based on the similarity to the centroid of each subcluster. Our approach outperforms all comparison approaches on all metrics. Because our approach first selects sentence based on the commonality of all documents, and then selects sentences based on the specificity of different subclasses, which is in line with the way of human summarizing multiple documents. Appendix A shows an example of the summaries generated

	Method	R-1	R-2	R-L	R-SU4
DUC'2004	**Ours$_{CS}$**	**38.36**	**8.66**	**33.94**	**13.30**
	Comp$_1$	36.44	8.03	32.08	12.61
	Comp$_2$	36.38	7.91	31.78	12.28
	Comp$_3$	36.85	8.18	32.25	12.44
DUC'2003	**Ours$_{CS}$**	**37.89**	**8.27**	**32.37**	**12.85**
	Comp$_1$	36.77	8.22	31.29	12.7
	Comp$_2$	35.31	7.05	30.24	11.39
	Comp$_3$	35.03	7.02	30.07	11.29
DUC'2002	**Ours$_{CS}$**	**34.81**	**7.32**	**30.11**	**11.48**
	Comp$_1$	33.23	6.95	28.78	10.99
	Comp$_2$	32.90	6.20	28.37	10.37
	Comp$_3$	32.86	6.12	28.46	10.31

Table 2: Comparison results of different approaches regarding whether or not the commonality and specificity of documents are considered on *DUC* datasets.

	Method	R-1	R-2	R-L	R-SU4
DUC'2004	**Ours$_{CS}$**	**38.36**	**8.66**	**33.94**	**13.30**
	Comp$_4$	37.17	7.65	32.93	12.45
DUC'2003	**Ours$_{CS}$**	**37.89**	**8.27**	**32.37**	**12.85**
	Comp$_4$	36.63	7.88	31.48	12.31
DUC'2002	**Ours$_{CS}$**	**34.81**	**7.32**	**30.11**	**11.48**
	Comp$_4$	33.50	6.48	29.09	10.63

Table 3: Comparison results of documents hierarchical clustering-based approach and sentences hierarchical clustering-based approach on *DUC* datasets.

by different approaches on one news set.

(2) Verify the effectiveness of using documents hierarchical clustering.

In order to not affected by other factors, our model also uses only the *Commonality-Specificity score* to score sentences. For a fair comparison, the comparison experiment is designed as follows:

Comp$_4$: same as the proposed approach (*Ours$_{CS}$*) but use all sentences in the documents as input to hierarchically cluster all sentences and select sentences from the class tree of sentences.

Table 3 displays the results on three *DUC* datasets. Our approach based on documents hierarchical clustering outperforms the comparison approach that based on sentences hierarchical clustering on all metrics. Because the sentences hierarchical clustering-based approach lacks comparisons between documents, thus resulting in the inability to discover the relationships between documents, which are important for multi-document summarization.

(3) Compare the proposed approach with other multi-document summarization approaches.

Method	R-1	R-2	R-L	R-SU4
Unsupervised methods				
Lead	32.37	6.38	28.68	10.29
LexRank	37.32	7.84	33.18	12.53
Centroid$_{BOW}$	37.03	8.19	32.48	12.68
GreedyKL	37.99	8.54	33.03	13.02
CLASSY 04	37.63	8.98	32.40	13.04
DPP	39.15	9.86*	30.88	13.83*
ICSISumm	37.43	9.56	32.89	13.31
OCCAMS_V	37.51	9.42	33.69	13.12
Ranking-clustering	37.87	9.35	-	13.25
SummPip	36.30	8.47	-	11.55
Centroid$^{Run1}_{embedding}$	36.92	8.20	32.53	12.72
Centroid$^{Run4}_{embedding}$	38.12	9.07	34.15	13.44
Supervised methods				
PG-MMR	36.42	9.36	-	13.23
CopyTransformer	28.54	6.38	-	7.22
Hi-MAP	35.78	8.90	-	11.43
BART-Long-Graph	34.72	7.97	-	11.04
Primera	35.1	7.2	17.9	-
Ours$_{final}$	**39.28***	**9.31**	**35.02***	**13.75**

Table 4: ROUGE scores of different approaches on *DUC'2004* dataset. The best performing approach for each metric is indicated by *.

Our model uses the $score^{final}$ to score sentences, denoted as ***Ours$_{final}$***. We compare the performance of the proposed approach with existing state-of-the-art multi-document summarization approaches.

Table 4 and Table 5 display the results on *DUC'2004* and *Multi-News* datasets respectively. We compare our approach with both unsupervised approaches and supervised deep learning-based approaches.

The unsupervised approaches includes *Lead* (Brandow et al., 1995), *LexRank* (Erkan and Radev, 2004), *Centroid$_{BOW}$* (Radev et al., 2004), *CLASSY 04* (Conroy et al., 2004), *ICSISumm* (Gillick et al., 2008), *DPP* (Kulesza and Taskar, 2012), *OCCAMS_V* (Davis et al., 2012), *GreedyKL* (Hong et al., 2014), *Ranking-clustering* (Yang et al., 2014), *SummPip* (Zhao et al., 2020), and *Centroid$_{embedding}$* (Lamsiyah et al., 2021), which are competitive baselines or state-of-the-art approaches for extractive multi-document summarization. We have reproduced *Run 1* and *Run 4* of *Centroid$_{embedding}$* using the *USE-DAN* sentence embedding model (Lamsiyah et al., 2021) and list their results on *DUC'2004*, and the results of other approaches are directly taken from their original

articles (Hong et al., 2014) or published materials[8].

The supervised approaches includes *PG-MMR* (Lebanoff et al., 2018), *CopyTransformer* (Gehrmann et al., 2018), *Hi-MAP* (Fabbri et al., 2019), *DynE* (Hokamp et al., 2020), *MatchSum* (Zhong et al., 2020), *MGSum* (Jin et al., 2020), *BART-Long-Graph* (Pasunuru et al., 2021) and *Primera* (Xiao et al., 2022), which are first trained on large datasets, such as *CNN*, *DailyMail* and *Multi-News*, and then tested on *DUC'2004* and *Multi-News* datasets. The results are directly taken from their original articles.

As shown in Table 4, for *R-1* measure, our approach significantly outperforms all unsupervised and supervised approaches, and it achieves comparable results with *DPP*, which is considered as the best performing approach on *DUC'2004*. For *R-L* measure, our approach significantly outperforms all unsupervised and supervised approaches. For *R-2* and *R-SU4* measures, our approach achieves comparable result with the state-of-the-art approaches. The supervised approaches yield worse results on *DUC'2004* than most unsupervised approaches because these deep learning-based approaches are trained on other datasets and tested directly on *DUC'2004*.

As shown in Table 5, our approach significantly outperforms all unsupervised approaches on all metrics. By comparing with the supervised deep learning-based approaches that are trained and tested on *Multi-News* dataset, our approach still achieves significantly better *R-1*, *R-L* and *R-SU4* scores than *PG-MMR*, *CopyTransformer*, *Hi-MAP*, *DynE* and *Primera*$^{zero\_shot}$. For *R-L* measure, our approach significantly outperforms the *Primera*^{fully} approach, which is considered as the state-of-the-art supervised multi-document summarization model.

Overall, as an unsupervised and easy-to-implement approach, our model achieves competitive performances compared to the state-of-the-art unsupervised and supervised multi-document summarization approaches. Moreover, comparative experiments prove the effectiveness of considering both the commonality and specificity of documents for multi-document summarization.

Method	R-1	R-2	R-L	R-SU4
Unsupervised methods				
Lead	39.41	11.77	-	14.51
LexRank	38.27	12.70	-	13.20
TextRank	38.44	13.10	-	13.50
MMR	38.77	11.98	-	12.91
SummPip	42.32	13.28	-	16.20
Centroid$^{Run4}_{embedding}$	42.93	14.04	27.7	17.27
Supervised methods				
PG-MMR	40.55	12.36	-	15.87
CopyTransformer	43.57	14.03	-	17.37
Hi-MAP	43.47	14.89	-	17.41
DynE	43.9	15.8	22.2	-
MatchSum	46.20	16.51	41.89*	-
MGSum	44.75	15.75	-	19.30*
Primera^{zero_shot}	42.0	13.6	20.8	-
Primera^{fully}	49.9*	21.1*	25.9	-
Ours$_{\text{final}}$	**44.04**	**14.15**	**39.74**	**18.19**

Table 5: ROUGE scores of different approaches on *Multi-News* dataset. The best performing approach for each metric is indicated by *.

5 Conclusion and Future Work

In this paper, we propose a multi-document summarization approach based on hierarchical clustering of documents, which makes use of the generated class tree of documents to extract both the commonality of all documents and the important specificity of some subclasses of documents, so as to generate the summary in line with human summarizing multiple documents. In the experiments, we show that our approach significantly outperforms the comparison approaches considering only commonality or only specificity, and the comparison approach based on sentences hierarchical clustering. Furthermore, as an easy-to-implement unsupervised approach, our approach is superior to many competitive supervised and unsupervised multi-document summarization approaches, and yields comparable performances to the state-of-the-art supervised approaches.

Documents hierarchical clustering has been proven to be effective for multi-document summarization in this paper. In future work, we plan to explore the best k of k-means in hierarchical clustering or other effective hierarchical clustering approaches for multi-document summarization. Additionally, we will compare different document embedding methods for hierarchical clustering and multi-document summarization to explore the suitable embedding representation of documents.

[8] github/duc2004-results

References

Ronald Brandow, Karl Mitze, and Lisa F Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675–685.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.

John M Conroy, Judith D Schlesinger, Jade Goldstein, and Dianne P O'leary. 2004. Left-brain/right-brain multi-document summarization. In *Proceedings of the Document Understanding Conference (DUC 2004)*.

Sashka T Davis, John M Conroy, and Judith D Schlesinger. 2012. Occams–an optimal combinatorial covering algorithm for multi-document summarization. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 454–463. IEEE.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

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.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Daniel Gillick, Benoit Favre, and Dilek Hakkani-Tür. 2008. The icsi summarization system at tac 2008. In *Tac*.

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*.

Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1608–1616.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6244–6254, Online. Association for Computational Linguistics.

Akanksha Joshi, Eduardo Fidalgo, Enrique Alegre, and Laura Fernández-Robles. 2019. Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129:200–215.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Salima Lamsiyah, Abdelkader El Mahdaouy, Bernard Espinasse, and Saïd El Alaoui Ouatik. 2021. An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Systems with Applications*, 167:114152.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. 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.

Mudasir Mohd, Rafiya Jan, and Muzaffar Shah. 2020. Text document summarization using word embedding. *Expert Systems with Applications*, 143:112958.

You Ouyang, Wenjie Li, Qin Lu, and Renxian Zhang. 2010. A study on position information in document summarization. In *Coling 2010: Posters*, pages 919–927, Beijing, China. Coling 2010 Organizing Committee.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, Online. Association for Computational Linguistics.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, Valencia, Spain. Association for Computational Linguistics.

Oussama Rouane, Hacene Belhadef, and Mustapha Bouakkaz. 2019. Combine clustering and frequent itemsets mining to enhance biomedical text summarization. *Expert Systems with Applications*, 135:362–373.

Kamal Sarkar. 2009. Sentence clustering-based summarization of multiple text documents. *TECHNIA–International Journal of Computing Science and Communication Technologies*, 2(1):325–335.

Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. 2008. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Libin Yang, Xiaoyan Cai, Yang Zhang, and Peng Shi. 2014. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information sciences*, 260:37–50.

Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. 2020. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1949–1952.

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

A An Example of Summaries Generated by Different Comparison Approaches

Figure 2 shows an example of the summaries generated by *Ours*$_{CS}$, *Comp*$_1$ (only commonality of all documents), and *Comp*$_2$ (only specificity of subclasses of documents) on one news set (*d30036*) of *DUC'2004*.

In this news set, six news articles talk about the Nobel Prize for literature, mainly describing Jose Saramago, the first Portuguese Nobel Prize winner in literature, his imaginative novels and Portugal's response to it. Three news articles talk about the Nobel Prize in medicine won by three American researchers and their research. And one news article talks about the Nobel Peace Prize. The generated summary of *Ours*$_{CS}$ mainly describes the stories about Jose Saramago, and describes the other two events respectively. However, the generated summary of *Comp*$_1$ only describes the stories about Jose Saramago, and the generated summary of *Comp*$_2$ describes the three events respectively, but does not highlight the main contents.

Ours$_{CS}$:

- A day after winning the Nobel Prize for literature, Portuguese novelist Jose Saramago insisted that while he was delighted to win the award, it could just as easily have gone to many other Portuguese writers.
- Jose Saramago, a 75-year-old Portuguese writer who took up literature relatively late in life and whose richly imaginative novels soon won him a following of loyal readers across Europe and vocal admirers in the United States, was awarded this year's Nobel Prize in Literature Thursday by the Swedish Academy in Stockholm.
- Three American researchers on Monday won the Nobel Medicine Prize for discovering how nitric oxide acts as a signal molecule in the cardiovascular system, a breakthrough with applications ranging from hardening of the arteries to impotence.
- The Camp David accords were not signed until Sept. 17, 1978, about five weeks before that year's peace prize was annouCkuld not give him the prize, the Norwegian committee recognized in the 1978 awards citation ``the positive initiative taken by President Jimmy Carter."

Comp$_1$:

- A day after winning the Nobel Prize for literature, Portuguese novelist Jose Saramago insisted that while he was delighted to win the award, it could just as easily have gone to many other Portuguese writers.
- Jose Saramago became the first writer in Portuguese to win the Nobel Prize for Literature on Thursday.
- Portuguese novelist Jose Saramago, whose capricious vision includes a section of Europe breaking off and floating out to sea, on Thursday was named the winner of the 1998 Nobel Literature Prize.
- He said that he knew that previous Nobel prize winners had nominated him for the award, but that he still was ``somewhat surprised" to receive it.
- Among other well-known names to receive the prize is David Baltimore, although he shared the prize in 1975, long before becoming one of the world's most visible AIDS researchers.

Comp$_2$:

- There's room for a few more names on a 20th century honor roll of writers, and one will be added this week when the Swedish Academy announces the latest Nobel Literature laureate.
- Three American researchers on Monday won the Nobel Medicine Prize for discovering how nitric oxide acts as a signal molecule in the cardiovascular system, a breakthrough with applications ranging from hardening of the arteries to impotence.
- The Camp David accords were not signed until Sept. 17, 1978, about five weeks before that year's peace prize was annouCkuld not give him the prize, the Norwegian committee recognized in the 1978 awards citation ``the positive initiative taken by President Jimmy Carter.''
- Jose Saramago, a 75-year-old Portuguese writer who took up literature relatively late in life and whose richly imaginative novels soon won him a following of loyal readers across Europe and vocal admirers in the United States, was awarded this year's Nobel Prize in Literature Thursday by the Swedish Academy in Stockholm.

Figure 2: The examples of generated summaries on one news set.