SCURank: Ranking Multiple Candidate Summaries with Summary Content Units for Enhanced Summarization

Bo-Jyun Wang[1,2], Ying-Jia Lin[2,3], Hung-Yu Kao[4]
[1]Department of Computer Science and Information Engineering,
National Cheng Kung University,
[2]Artificial Intelligence Research Center, Chang Gung University,
[3]Department of Artificial Intelligence, Chang Gung University,
[4]Department of Computer Science, National Tsing Hua University
bojyun.wang@cgu.edu.tw, yjlin@cgu.edu.tw, hykao@cs.nthu.edu.tw

Abstract

Small language models (SLMs), such as BART, can achieve summarization performance comparable to large language models (LLMs) via distillation. However, existing LLM-based ranking strategies for summary candidates suffer from instability, while classical metrics (e.g., ROUGE) are insufficient to rank high-quality summaries. To address these issues, we introduce **SCURank**, a framework that enhances summarization by leveraging **Summary Content Units (SCUs)**. Instead of relying on unstable comparisons or surface-level overlap, SCURank evaluates summaries based on the richness and semantic importance of information content. We investigate the effectiveness of SCURank in distilling summaries from multiple diverse LLMs. Experimental results demonstrate that SCURank outperforms traditional metrics and LLM-based ranking methods across evaluation measures and datasets. Furthermore, our findings show that incorporating diverse LLM summaries enhances model abstractiveness and overall distilled model performance, validating the benefits of information-centric ranking in multi-LLM distillation. The code for SCURank is available at https://github.com/IKMLab/SCURank.

1 Introduction

Following the advent of ChatGPT (Ouyang et al., 2022), a paradigm shift has occurred in the domain of Natural Language Processing (NLP), marked by substantial advancements, including the field of summarization. In the wake of this development, GPT-4 (OpenAI, 2024a) and numerous Large Language Models (LLMs) have emerged (Google, 2024; Anthropic, 2024; MistralAI, 2024), demonstrating superior performance. These models are easily accessible via APIs, allowing users to obtain responses with minimal effort. However, while these models are powerful, their resource demands, such as the cost of local deployments, are significant (Hsieh et al., 2023) due to their incredibly large model size. This has led to a growing trend of distilling these models into smaller ones, optimized for specific tasks such as summarization (Hinton et al., 2015; Jiang et al., 2024). The distilled models exhibit great resource efficiency without significant performance degradation, and in some cases, they demonstrate a capability to outperform LLMs in the summarization task (Liu et al., 2024).

Previous work by Liu et al. (2024) leverages BRIO (Liu et al., 2022) as a contrastive learning framework for model distillation. In BRIO, positive and negative samples are constructed by ranking candidate summaries. Consequently, the quality of the ranking function is critical for effective contrastive learning. To address this challenge, they introduced GPTRank (Liu et al., 2024), a novel method that ranks high-quality summaries generated by LLMs, ensuring more reliable supervision in BRIO training. However, there are two significant challenges to this approach. First, studies by Shen et al. (2023); Wang et al. (2024) indicate that LLMs remain unreliable and inconsistent in text comparison and candidate ranking. Second, relying on summaries from a single LLM introduces the risk of model-specific bias (e.g. content selection) and limits the diversity of generation patterns. Thus, we explore the use of multiple LLMs to generate candidate summaries for distillation, and investigate effective ranking methods for these summaries.

To bypass the instability of direct LLM ranking, we propose shifting the evaluation focus back to the core goal of summarization: information retention. To capture the information from summaries, we draw upon the concept of Summary Content Units (SCUs) (Nenkova and Passonneau, 2004). Every SCU presents simple, standalone, and unique information in the summary (Shapira et al., 2019). Generally, annotating SCUs requires manual efforts, thus expensive and not reproducible (Zhang and Bansal, 2021). Nawrath et al. (2024)

Findings of the Association for Computational Linguistics: ACL 2026, pages 38980–38997

suggested a new path to extract SCUs, which employed GPT-3.5 and GPT-4 (OpenAI, 2024a) to generate Semantic GPT Units (SGUs). The SGUs are similar to SCUs, capturing the key information in the summaries, and have been proven to be high-quality in the evaluations of Nawrath et al. (2024). Consequently, we adopt the concept of SGUs as SCUs to evaluate the quality of summaries.

In this paper, we introduce **SCURank** (**S**ummary **C**ontent **U**nit **Rank**ing), a ranking framework that evaluates the information richness of candidate summaries by analyzing their SCUs. SCURank operates in three stages. First, it captures the key information in each summary by extracting SCUs. Next, all SCUs are aggregated by clustering to estimate their importance based on their frequency across summaries. Finally, each summary is assigned a score by summing the importance scores of its SCUs, which reflects its overall information richness. This score is further normalized by summary length to mitigate bias toward longer summaries. By focusing on the information content rather than direct comparison or ranking, SCURank provides a robust ranking approach. Specifically, as LLMs are only employed for SCU extraction, the framework avoids the unreliability associated with LLM-based comparison.

Our contributions are as follows: (1) We propose SCURank, a novel method to rank high-quality summaries based on SCUs. (2) We investigate the effect of distilling models from multiple LLMs. (3) We demonstrate that SCURank, combined with contrastive learning, improves distilled model performance.

2 Related Work

SCURank, which ranks summaries by employing Summary Content Units (SCUs), is based on two concepts. The first method is to decompose a summary candidate into SCUs, which are brief and convey a single fact. The second is the ranking. The ranking results for the candidate summaries can be used for contrastive learning to improve the LMs' summarization capabilities.

2.1 Summary Content Units

The concept of SCUs was first introduced by Nenkova and Passonneau (2004). In that study, the authors introduced a reliable, predictive, and diagnostic method for evaluating the summaries. However, due to the high cost and expertise required, Shapira et al. (2019) demonstrated a revised version that is more cost-effective and annotation-friendly. In this approach, they provided the instructions for crowd workers to extract a specific number of the SCU-like statements, eliminating the need to merge and weight SCUs. These efforts reduced the dependency on expertise, making the Pyramid method more efficient.

Nevertheless, the significant cost of hiring workers to extract SCUs remains a significant challenge. In response, researchers have been investigating methods of automating this process in recent years. Zhang and Bansal (2021) proposed a system called $List^3Pyramid$, which employed a semantic role labeling model to extract Summary Triplets Units (STUs). Subsequently, Nawrath et al. (2024) introduced two novel methods. One method leveraged Abstract Meaning Representation to extract Semantic Meaning Units (SMUs), and the other employed an LLM to extract Semantic GPT Units (SGUs). Nawrath et al. (2024) have demonstrated the high quality of the SGUs in a wide range of evaluations. In our work, we adopt the concept of SGUs from Nawrath et al. (2024) for extracting SCUs in SCURank.

2.2 Summarization with Ranking

Contrastive learning is a method of training a model to identify and select superior candidates, while avoiding inferior ones. In summarization, SimCLS (Liu and Liu, 2021) was the first to employ a scoring model for contrastive learning. This scoring model evaluates the quality of the candidate summaries, thereby enhancing overall performance on summarization tasks. Subsequently, BRIO (Liu et al., 2022) integrated these tasks into a single model, which is capable of both generating and evaluating summaries. Both approaches used ROUGE (Lin, 2004) as a metric for evaluating summaries. However, as ROUGE is only concerned with n-gram overlap, it is not accurate enough to evaluate high-quality summaries (Cohan and Goharian, 2016). Recently, Liu et al. (2024) developed GPTRank, which integrates the concept of G-Eval (Liu et al., 2023), to rank the summaries using an LLM. In contrast to GPTScore (Fu et al., 2024), GPTRank does not employ the LLM-predicted probability. Instead, it provides a prompt to request the LLM to rank and offer a concise explanation of the ranking. Nevertheless, Wang et al. (2024) uncovered positional bias in the ranking of LLMs, and Shen et al. (2023) also demonstrated that LLMs'

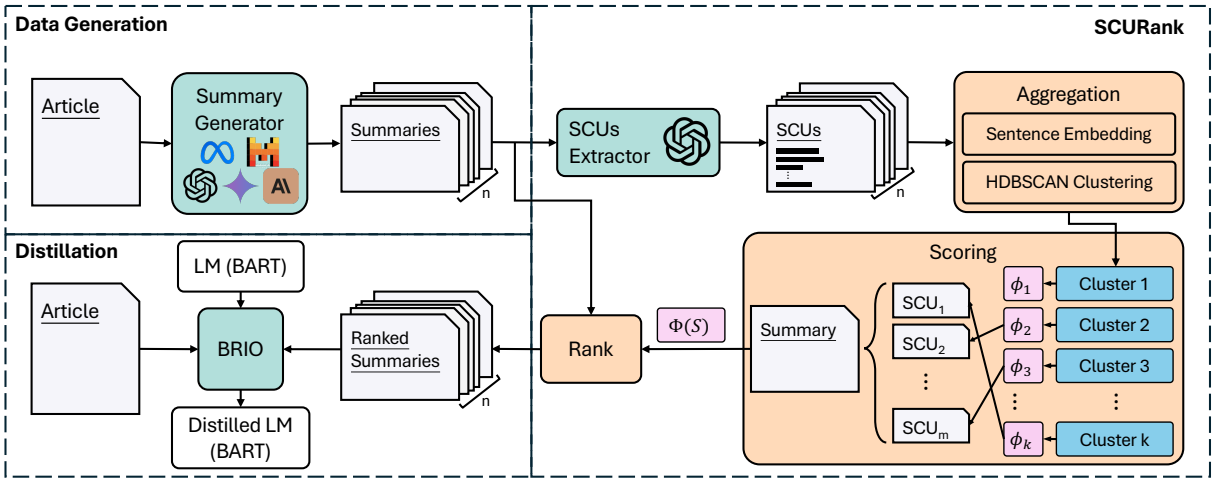

Figure 1: Overview of our training framework. The Data Generation part generates candidate summaries from several LLMs for each article. SCURank, our proposed ranking method, ranks the candidate summaries via three steps: (1) extracting Summary Content Units (SCUs), (2) aggregating SCUs with sentence embeddings and HDBSCAN, and (3) scoring summaries based on SCU cluster distribution. Finally, the ranked summaries are then used to train a distilled model with BRIO.

difficulty in providing consistent text comparisons. These studies demonstrate the need for a more concrete and stable method for training summarization models with ranked candidates (Liu et al., 2024).

3 Methods

3.1 Preliminary

Let $\mathcal{D}$ denote the document being summarized. Assume that there are n summarization models $F = \{f_i\}_{i=1}^{n}$ we want to distill. All models use the same prompt (Appendix A). The set of all summaries produced by these models is given by $S = \{s_i\}_{i=1}^{n}$, where each summary s_i is generated by the corresponding model f_i:

$$s_i = f_i(\mathcal{D}). \quad (1)$$

3.2 SCURank

3.2.1 Overview

The SCURank is designed to rank high-quality summaries based on their information richness and importance. As illustrated in Figure 1, the SCURank process has three key steps: (1) extract SCUs from the candidate summaries, (2) aggregate SCUs across multiple summaries via clustering, and (3) score each summary based on the importance of its SCUs. The details are below.

3.2.2 SCUs Extraction

To extract the SCUs from summaries, we adopt the method based on a large language model from Nawrath et al. (2024). The instruction contains one example from REALSumm (Bhandari et al., 2020) to generate SCUs. We use gpt-4o-mini[1](OpenAI, 2024b) as SCUExt to extract SCUs from the summaries:

$$\mathcal{U}_i = \{u_{i,1}, u_{i,2}, ..., u_{i,m_i}\} = \text{SCUExt}(s_i). \quad (2)$$

Each $\mathcal{U}_i$ contains all the SCUs extracted from s_i. The number of SCUs in s_i, denoted as m_i, varies since the amount of information present in each summary s_i is uncertain.

3.2.3 SCUs Aggregation

After extracting SCUs from the candidate summaries, we aggregate all SCUs by clustering them based on their semantic similarity. Each cluster represents distinct semantic information, and the number of SCUs in a cluster reflects the importance of that information. The more SCUs in a cluster, the more models agree on the information, thus presenting the level of its value.

Sentence Encoder To facilitate clustering, it is essential to translate these SCUs into vectors. We use the all-mpnet-base-v2[2] model, noted as *Encoder*, which is small but effective (Reimers and Gurevych, 2019). Given a set of SCUs $\mathcal{U}_i$, the

[1]We've compared the performance of gpt-4o-mini, gpt-4o, and gpt-3.5-turbo and found that gpt-4o-mini is cost-effective while maintaining comparable performance to gpt-4o. Details and the preliminary results are provided in the Appendix B.

[2]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

corresponding vectors are obtained as follows:

$$\mathcal{V}_i = \{v_{i,1}, v_{i,2}, ..., v_{i,m_i}\} = \text{Encoder}(\mathcal{U}_i). \quad (3)$$

$\mathcal{V}_i$ represents the set of vectorized SCUs for s_i. These embeddings capture semantic meaning, allowing us to cluster the SCUs. We aggregate all embeddings as $\mathcal{V} = \bigcup_{i=1}^{n} \mathcal{V}_i$ for clustering.

HDBSCAN Since the number of distinct semantic information is uncertain, we employ HDBSCAN (Campello et al., 2013) for clustering the SCU embeddings. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), an extension of DBSCAN (Ester et al., 1996), can automatically determine the number of clusters based on the density of the data and identifies outliers as noise (Campello et al., 2013). Unlike DBSCAN, HDBSCAN can handle varying cluster densities, making it more suitable for our task.

Our task is to cluster the SCUs into different groups, with each group representing distinct semantic information. The result of the clustering is a set of clusters, denoted as $\mathcal{C}$.

$$\mathcal{C} = \text{HDBSCAN}(\mathcal{V}) \quad (4)$$

$$\mathcal{C} = \{C^1, C^2, ..., C^K\} \quad (5)$$

Each $C^k (k \in 1, \ldots, K)$ contains the SCUs that belong to the same cluster.

3.2.4 Scoring

Our scoring mechanism is based on the assumption that information selected by more candidate summaries is more important. Since SCUs within the same cluster C^k are semantically equivalent, the cluster size reflects the level of consensus among candidates. We thus define the score of each SCU $u_{i,j}$ as:

$$\phi(u_{i,j}) = \|C^k\|, \text{ where } u_{i,j} \in C^k. \quad (6)$$

For example, if $u_{i,j}$ belongs to a cluster C^k that contains 5 SCUs, then $\phi(u_{i,j}) = 5$.

For each summary candidate s_i, the function Φ returns the sum of the scores of the information associated with its SCUs:

$$\Phi(s_i) = \sum_{j=1}^{m_i} \phi(u_{i,j}). \quad (7)$$

To prevent longer summaries from receiving disproportionately higher scores, we divide all scores by the square root of the number of tokens in each summary. This produces the final adjusted scores with length penalty (lp):

$$\Phi_{\text{lp}}(s_i) = \frac{\Phi(s_i)}{\sqrt{\|s_i\|}}, \quad (8)$$

where $\|s_i\|$ is the number of tokens in the summary s_i. Preliminary results are provided in Appendix C.

3.2.5 Rank

With the adjusted scores, we can rank the summaries based on the richness and importance of their SCUs. The standard ranking is defined as follows:

$$\text{rk} = \text{argsort}(\Phi_{\text{lp}}(s_i) \text{ for } i \in \{1, 2, ..., n\}) \quad (9)$$

so that:

$$\Phi_{\text{lp}}(S_{\text{rk}(1)}) \geq \Phi_{\text{lp}}(S_{\text{rk}(2)}) \geq ... \geq \Phi_{\text{lp}}(S_{\text{rk}(n)}). \quad (10)$$

With this standard ranking result, we can rank the summaries and use them in subsequent training.

3.3 Distillation

We integrate SCURank into the BRIO (Liu et al., 2022) framework to enhance contrastive learning-based distillation. BRIO requires a ranking method to differentiate summary quality, and we replace its original ROUGE-based ranking with SCURank to enable more effective training.

3.3.1 Contrastive Learning

BRIO leverages multiple candidate summaries via contrastive learning to help the model distinguish between high-quality and low-quality summaries. Given two summaries s_1, s_2, the model should assign a higher probability to s_1 if it is of higher quality than s_2. Initially, BRIO employed an automatic metric such as ROUGE (Lin, 2004) to rank the summaries. Liu et al. (2024) later introduced GPTRank, which replaced ROUGE with GPT-based ranking, but it suffers from instability and inconsistency. To address these issues, we substitute BRIO's ranking method with SCURank, which offers a more stable and semantically meaningful ranking mechanism.

3.3.2 Maximum Likelihood Estimation

We also train a distilled model using maximum likelihood estimation (MLE) to compare its effectiveness with contrastive learning. Within the BRIO training framework, contrastive learning requires ranked candidate summaries, whereas MLE does

not. Instead, MLE optimizes the likelihood of the reference summaries s^* by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{xent}}(\theta) = -\log(\prod_{l=1}^{|s^*|} p(s_l^*|s_{<l}^*, \mathcal{D}; \theta)). \quad (11)$$

Here s_l^* denotes the lth token in the summary s^*, and p is the probability of s_l^* given the previous tokens $s_{<l}^*$ and the document $\mathcal{D}$. The learnable parameters of the model are represented by the symbol θ.

4 Experiments

We conduct several experiments to evaluate the effectiveness and characteristics of our approach. (1) Distilled model evaluations: We assess the performance of distilled models trained using different ranking methods and automatic metrics. (2) Stability of SCURank: To analyze the ranking consistency of SCURank and GPTRank, we rank the training set multiple times and measured the correlation. (3) LLM-based comparison: We use recent state-of-the-art LLMs to compare summaries generated by the distilled models. (4) Human evaluation: The performance of SCURank is compared against GPTRank using MTurk for pairwise comparison in three dimensions. (5) Writing style: We examine the abstractiveness of summaries generated by the distilled models and investigate the impact of different training datasets. Implementation details are provided in the Appendix D.

4.1 Training Set

We use two datasets to compare distillation from a single LLM versus multiple LLMs: The first dataset, **BASE**, contains summaries generated by GPT-3.5-turbo and candidate summaries from a single, unspecified LLM (Liu et al., 2024). The second dataset, **LLMs-9**, consists of summaries generated by nine different LLMs. The statistical details of both datasets are presented in Appendix E.

4.1.1 BASE

The BASE dataset was derived from Liu et al. (2024), which contains 1,000 articles from CNN/DailyMail (Nallapati et al., 2016). Each article includes a reference generated from gpt-3.5-turbo and nine candidate summaries produced by a single LLM via diverse beam search (Vijayakumar et al., 2018). However, Liu et al. (2024) did not specify the exact model used to generate the summaries. Nevertheless, we consider BASE to be a single-LLM dataset because the summaries are generated by a single, albeit unspecified, LLM.

4.1.2 LLMs-9

For a fair comparison, we generated summaries for the same 1,000 CNN/DailyMail articles as in the BASE dataset, using **nine different LLMs**. We refer to this collection as **LLMs-9**. Details of the adopted models and prompting strategies are provided in the Appendix A.

To further evaluate the effectiveness of SCURank, we extended our experiments to a different summarization dataset, XSum (Narayan et al., 2018). Using the same nine LLMs, we generated summaries for 1,000 articles from the XSum validation set. Note that Liu et al. (2024) did not include XSum in the original BASE dataset.

4.2 Baselines

We compare SCURank against several baselines and metrics: **LLMs Average**, **UnRank** (MLE-only), **GPTRank** (Liu et al., 2024), **ROUGE** (Lin, 2004), **BERTScore** (Zhang* et al., 2020), and **BLANC** (Vasilyev et al., 2020). For GPTRank, we used gpt-4o-mini to ensure a fair comparison with our method.

4.3 Evaluation

4.3.1 Test Set

The reference summaries in the original CNN/DailyMail (Nallapati et al., 2016) and XSum (Narayan et al., 2018) datasets are known to have quality issues (Maynez et al., 2020; Kang and Hashimoto, 2020), which was also confirmed by Zhang et al. (2024). Therefore, we used the human-written references from Zhang et al. (2024) to evaluate the distilled models. This dataset consists of 54 articles from the CNN/DailyMail dataset and 53 articles from the XSum dataset, each of which contains one to three human-written references. If a summary has multiple references, the highest score is selected.

4.3.2 Metrics

We use ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004) as our primary evaluation metrics. To complement these lexical metrics, we additionally include three model-based metrics: BERTScore (Zhang* et al., 2020), BLEURT (Sellam et al., 2020), and BARTScore (Yuan et al., 2021), which

	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	BERTScore	BARTScore
LLMs Results						
GPT-4o	41.2	15.9	26.0	54.7	69.4	-2.78
Gemini-1.5-pro	41.5	16.7	27.2	53.9	69.7	-2.67
Mistral-Large	45.1	18.8	29.4	55.9	71.2	-2.48
LLMs Average	42.0	17.3	27.2	54.5	69.7	-2.59
Dataset: LLMs-9						
UnRank	$43.8_{\pm 0.83}$	$\mathbf{20.5}_{\pm 0.46}$	$30.1_{\pm 0.93}$	$\mathbf{53.3}_{\pm 0.27}$	$\underline{69.9}_{\pm 0.27}$	$-2.59_{\pm 0.47}$
ROUGE	$43.2_{\pm 0.23}$	$\underline{20.4}_{\pm 0.57}$	$29.9_{\pm 0.22}$	$51.4_{\pm 0.40}$	$69.0_{\pm 0.05}$	$-2.58_{\pm 0.02}$
BERTScore	$43.9_{\pm 0.59}$	$20.0_{\pm 0.58}$	$30.1_{\pm 0.81}$	$\underline{52.2}_{\pm 0.50}$	$69.5_{\pm 0.32}$	$-2.46_{\pm 0.06}$
BLANC	$\underline{44.1}_{\pm 0.54}$	$20.3_{\pm 0.47}$	$\underline{30.2}_{\pm 0.35}$	$50.1_{\pm 0.64}$	$69.4_{\pm 0.52}$	$\mathbf{-2.34}_{\pm 0.13}$
GPTRank	$43.8_{\pm 1.12}$	$20.0_{\pm 1.66}$	$30.1_{\pm 1.17}$	$51.4_{\pm 1.36}$	$69.7_{\pm 0.66}$	$\underline{-2.36}_{\pm 0.05}$
SCURank	$\mathbf{44.8}_{\pm 0.28}$	$\mathbf{20.5}_{\pm 0.81}$	$\mathbf{30.6}_{\pm 0.86}$	$51.7_{\pm 0.78}$	$\mathbf{70.0}_{\pm 0.27}$	$\mathbf{-2.34}_{\pm 0.06}$
Dataset: BASE						
UnRank	$42.3_{\pm 0.11}$	$19.4_{\pm 0.09}$	$29.6_{\pm 0.15}$	$49.8_{\pm 0.36}$	$68.6_{\pm 0.08}$	$-2.73_{\pm 0.85}$
ROUGE	$43.3_{\pm 0.56}$	$20.1_{\pm 0.42}$	$30.3_{\pm 0.42}$	$50.9_{\pm 0.37}$	$69.0_{\pm 0.30}$	$-2.62_{\pm 0.02}$
BERTScore	$43.4_{\pm 0.55}$	$20.3_{\pm 0.48}$	$30.1_{\pm 0.76}$	$\underline{51.4}_{\pm 0.30}$	$69.0_{\pm 0.37}$	$-2.59_{\pm 0.02}$
BLANC	$\underline{43.9}_{\pm 0.68}$	$\underline{20.6}_{\pm 0.67}$	$\underline{30.6}_{\pm 0.62}$	$\underline{51.4}_{\pm 0.93}$	$\underline{69.5}_{\pm 0.29}$	$\mathbf{-2.45}_{\pm 0.09}$
GPTRank	$43.0_{\pm 0.35}$	$19.8_{\pm 0.75}$	$29.6_{\pm 0.62}$	$51.2_{\pm 0.81}$	$69.1_{\pm 0.38}$	$\underline{-2.52}_{\pm 0.08}$
SCURank	$\mathbf{44.3}_{\pm 0.87}$	$\mathbf{20.8}_{\pm 0.96}$	$\mathbf{30.9}_{\pm 1.02}$	$\mathbf{51.7}_{\pm 0.67}$	$\mathbf{69.9}_{\pm 0.50}$	$\mathbf{-2.45}_{\pm 0.10}$

Table 1: Distilled model performance on CNN/DailyMail using the LLMs-9 and BASE (Liu et al., 2024) datasets. "LLMs Average" denotes the mean of nine top LLMs. "UnRank" is the MLE-only baseline. Bold/underline indicate best/second-best results (mean ± std over 5 runs).

capture semantic similarity using pre-trained language models.

5 Results

5.1 Main Results

5.1.1 SCURank vs. Baselines

Table 1 shows the performance of the distilled models trained using different ranking methods on CNN/DailyMail. The distilled model trained with SCURank outperforms those trained with the other ranking methods, achieving the highest scores in all metrics, except BLEURT in the LLMs-9 dataset. To further evaluate the effectiveness of SCURank, we test the distilled model on the XSum version of LLMs-9. Different from CNN/DailyMail, we conduct 10 runs on the XSum dataset (instead of 5) to ensure more robust evaluation. In Table 2, the model trained with SCURank achieves the best scores in ROUGE-1, ROUGE-2, and BLEURT, and the second-highest scores in ROUGE-L and BERTScore. Interestingly, the model trained with the ROUGE-based ranking method also performs well on XSum, achieving the highest scores in ROUGE-L, BERTScore, and BARTScore; however, its scores on other metrics are notably lower than those of the SCURank model. Overall, the results on both datasets suggest that SCURank is a more robust and effective ranking method than GPTRank or traditional automatic metrics. The significance test results between SCURank and GPTRank are provided in Appendix F.

5.1.2 MLE vs. Contrastive Learning

We also evaluate the UnRank model, which is trained using maximum likelihood estimation (MLE) on summaries generated by LLMs. In Table 1, the UnRank model trained on the LLMs-9 dataset achieves the highest score in ROUGE-2 and BLEURT, and the second-highest in BERTScore. However, the UnRank model's performance is comparatively lower when trained on the BASE dataset, suggesting that the outputs of multiple LLMs can have a positive effect through MLE training. Additionally, the UnRank model trained on the XSum dataset performs worse across all metrics compared to the distilled model trained with contrastive learning (SCURank), as shown in Table 2. These find-

	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	BERTScore	BARTScore
LLMs Results						
GPT-4o	43.2	16.1	29.3	55.1	70.3	-2.85
Gemini-1.5-pro	43.9	16.1	30.5	55.6	71.3	-2.65
Mistral-Large	42.8	15.8	29.1	54.1	69.7	-2.82
LLMs Average	42.5	15.9	29.4	52.9	69.6	-2.91
Dataset: LLMs-9						
UnRank	45.3$_{\pm 0.11}$	19.2$_{\pm 0.23}$	31.5$_{\pm 0.24}$	54.2$_{\pm 0.14}$	70.6$_{\pm 0.09}$	-2.71$_{\pm 0.01}$
ROUGE	43.9$_{\pm 1.05}$	18.9$_{\pm 0.72}$	**32.5**$_{\pm 0.49}$	54.4$_{\pm 0.55}$	**71.1**$_{\pm 0.14}$	**-2.47**$_{\pm 0.04}$
BERTScore	44.0$_{\pm 1.04}$	18.5$_{\pm 0.46}$	31.8$_{\pm 0.31}$	53.6$_{\pm 0.66}$	70.5$_{\pm 0.17}$	-2.52$_{\pm 0.06}$
BLANC	44.3$_{\pm 0.52}$	18.8$_{\pm 0.39}$	31.7$_{\pm 0.32}$	54.1$_{\pm 0.92}$	70.3$_{\pm 0.22}$	-2.55$_{\pm 0.02}$
GPTRank	44.2$_{\pm 1.67}$	18.4$_{\pm 1.17}$	31.1$_{\pm 1.16}$	54.2$_{\pm 1.52}$	70.3$_{\pm 0.58}$	-2.57$_{\pm 0.04}$
SCURank	**45.4**$_{\pm 0.26}$	**19.3**$_{\pm 0.23}$	32.0$_{\pm 0.26}$	**55.4**$_{\pm \mathbf{0.44}}$	70.8$_{\pm 0.25}$	-2.54$_{\pm 0.02}$

Table 2: Distilled model performance on XSum using the LLMs-9 datasets. "LLMs Average" denotes the mean of nine top LLMs. "UnRank" is the MLE-only baseline. Bold/underline indicate best/second-best results (mean ± std over 10 runs).

ings indicate that contrastive learning, especially with SCURank, offers more effective supervision than MLE alone. The case studies in Appendix H.1 further illustrate the advantage of SCURank over MLE-only training.

5.1.3 Comparison with LLM Outputs

Both Table 1 and Table 2 show that the distilled models consistently outperform the LLMs average in all metrics except BLEURT. However, certain individual LLMs, such as Mistral-Large, still achieve higher scores than the distilled model in ROUGE-1, BLEURT, and BERTScore. These results suggest that the performance of the distilled models have reached a performance level comparable to top-performing LLMs, indicating the effectiveness of the distillation process with ranking-based contrastive learning. More scores of individual LLMs are provided in the Appendix I.

5.2 Stability of Ranking

To assess the consistency of ranking approaches, we evaluate SCURank and GPTRank on the LLMs-9 dataset (CNN/DailyMail). We perform five independent ranking runs on 1,000 samples.

For each method, we adopt a representative-ranking strategy: for each sample, we select the run with the highest average correlation with the other runs as the representative, reporting its mean correlation. Rankings are evaluated using Kendall's τ, Spearman's ρ, and Pearson's r, along with Krippendorff's α to measure inter-run agreement, where values above 0.8 are generally considered to indicate reliable agreement.

	τ	ρ	r	α
GPTRank	76.8	78.4	78.4	96.4
GPTRank*	16.7	22.4	22.4	3.0
SCURank	**66.1**	**72.0**	**72.0**	**84.6**

Table 3: Stability evaluation of ranking methods. * indicates that summaries were shuffled before ranking (SCURank is order-invariant). Metrics: Kendall's τ, Spearman's ρ, Pearson's r, and Krippendorff's α.

As shown in Table 3, GPTRank exhibits high correlation when the input summary order is fixed, but its performance degrades substantially under randomized ordering. This behavior is consistent with prior findings that LLM-based ranking methods are sensitive to input order.

In contrast, SCURank achieves consistently high reliability across runs, with a Krippendorff's α of 0.846, exceeding the commonly accepted threshold of 0.8. This robustness stems from using LLMs solely for SCU extraction rather than direct ranking, thereby avoiding order sensitivity. Overall, these results demonstrate that SCURank is more stable and robust than GPTRank.

5.3 Human Evaluation

To ensure our results align with human perception, we conducted a human evaluation to compare the summaries generated by distilled models

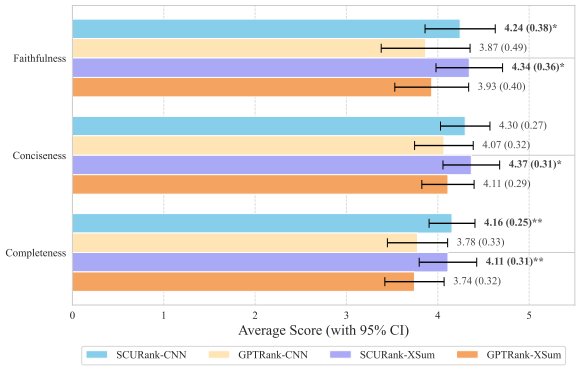

Figure 2: Human evaluation results comparing distilled models trained with SCURank and GPTRank. Each summary was scored by 3 annotators. Significance (paired t-test): * $p < 0.05$, ** $p < 0.01$.

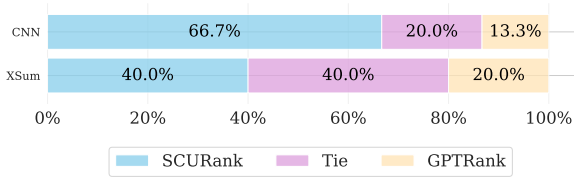

Figure 3: Human preference evaluation results between distilled models trained with SCURank and GPTRank.

trained with SCURank and GPTRank. We randomly sampled 30 articles from the test sets of CNN/DailyMail and XSum, respectively. For each article, we presented the source text and pairs of summaries generated by the two models to three independent annotators recruited via Amazon Mechanical Turk (MTurk). Annotators were asked to assess the summaries in two ways: (1) scoring them on a Likert scale (1-5) across three dimensions: *Faithfulness*, *Conciseness*, and *Completeness*; and (2) providing an overall preference (Win/Tie/Loss). The details of the quality of annotators and evaluation criteria are provided in the Appendix G.

Figure 2 presents the average scores for each metric. The distilled model trained with SCURank consistently outperforms the GPTRank baseline across all dimensions on both datasets. Notably, SCURank demonstrates highly significant improvements ($p < 0.01$) in *Completeness* for both CNN/DailyMail and XSum, suggesting that our SCU-based approach effectively encourages the model to retain more key information.

Figure 3 illustrates the pairwise preference results. On both datasets, SCURank exhibits a dominant advantage over GPTRank. On the XSum dataset, while the percentage of ties increases to 40.0% (likely due to the shorter length of XSum summaries making distinct differentiation harder), SCURank still achieves a 40.0% win rate, double that of GPTRank (20.0%). These results indicate that SCURank produces more faithful, concise, complete, and overall better summaries under the human evaluations. To further illustrate the qualitative differences, we provide the sample summaries in Appendix H.2.

5.4 LLM-based Evaluation

In order to further validate the effectiveness of SCURank, we evaluate the distilled models trained with candidate summaries ranked by SCURank and GPTRank, using three of the latest LLMs: Grok-4 (xAI, 2025), Claude-4-sonnet (Anthropic, 2025), and Gemini-2.5-pro (Google, 2024). For each test sample, we present the summaries generated by the two distilled models (trained using SCURank and GPTRank rankings, respectively) in the prompt of an LLM, asking it to decide which summary is better. To reduce the positional bias, each pairwise comparison is conducted twice, with the order of the summaries reversed in the second round. A model receives a point if it wins two rounds, or if it wins one round and ties in the other.

The results are shown in Figure 4. The distilled model trained with SCURank consistently outperforms the one trained with GPTRank across both datasets. The results further confirm that the ranking superiority of SCURank as a ranking method for training summarization models.

5.5 Writing Style

To investigate the impact of distillation from multiple LLMs, we analyze the abstractiveness of summaries generated by distilled models trained on BASE and LLMs-9.

Following Grusky et al. (2018), we measure abstractiveness using coverage and density, where lower values indicate more abstractive summaries. Coverage measures the proportion of words in the summary that appear in the source document, while density measures the average length of contiguous spans shared between them.

Table 4 compares distilled models with human-written summaries (Zhang et al., 2024) and LLM-generated summaries. The distilled model trained on LLMs-9 consistently achieves lower coverage and density than the one trained on BASE, indicating improved abstractiveness.

Although distilled models still exhibit higher coverage and density than both LLMs and human-

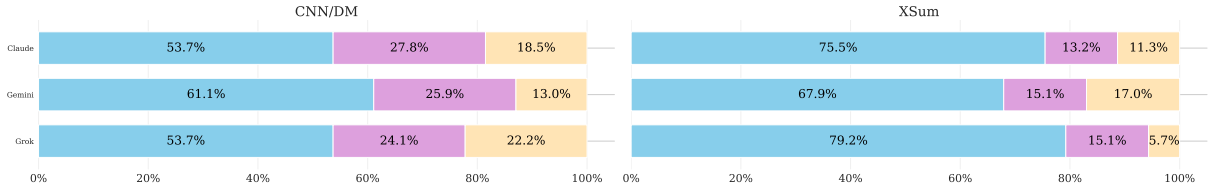

Figure 4: Win-tie-loss rates between distilled models trained with SCURank and GPTRank, evaluated by three LLM judges: Claude-4-sonnet, Gemini-2.5-pro, and Grok 4.

written summaries, the gap is substantially reduced. These results suggest that distillation from diverse LLM-generated summaries enhances abstractiveness and moves the model closer to human writing style. This improvement likely stems from the increased diversity of training summaries, since different LLMs often express similar content using different phrasings. Learning from multiple LLMs therefore encourages the model to capture semantic content rather than copying spans from the source text.

	Coverage		Density	
	BASE	LLMs-9	BASE	LLMs-9
Human	–	0.81	–	2.07
LLMs	–	0.83	–	2.70
SCURank	0.95	0.94	10.34	6.73
ROUGE	0.98	0.97	18.46	16.34
BERTScore	0.97	0.95	17.31	9.99
BLANC	0.96	0.94	11.82	7.56
GPTRank	0.96	0.93	14.80	5.97
Average	0.96	0.95	14.54	9.32

Table 4: Coverage and density of summaries generated by models trained on the BASE and LLMs-9 datasets on CNN/DailyMail. For reference, we also report scores for human-written summaries (Zhang et al., 2024) and LLM-generated candidates (LLMs). **Lower** scores indicate higher abstractiveness.

6 Conclusions

We propose SCURank, an information-based ranking method that leverages Summary Content Units (SCUs) to provide more stable and semantically meaningful ranks than existing approaches. Experimental results demonstrate that SCURank improves contrastive learning performance, outperforming both GPTRank and traditional automatic metrics. Moreover, training with diverse summaries generated by multiple LLMs has been shown to enhance abstractiveness while maintaining overall performance. These findings highlight SCURank as an effective solution to rank high-quality summaries and demonstrate the potential of multi-LLM distillation for enhanced abstractive summarization.

Limitations

In SCUs extraction, we used the gpt-4o-mini with 1-shot examples from REALSumm due to budget constraints. The results of the intrinsic evaluation showed that the SCUs extracted by gpt-4o with 3-shot examples from REALSumm achieved the highest score. Moreover, if the total number of SCUs is too small, the clustering may change the parameters of the HDBSCAN.

The SCURank should be more competitive with more candidate summaries, but this study targets the comparison with GPTRank (Liu et al., 2024), which uses one reference and eight candidate summaries for training with contrastive learning.

Acknowledgements

This work was supported by the National Science and Technology Council, Taiwan, under Grants NSTC 114-2223-E-007-011 and NSTC 114-2222-E-182-001-MY2. We would like to express our sincere gratitude to the reviewers for their thoughtful comments and valuable feedback.

References

AI@Meta. 2024. Llama 3 model card. Accessed: 2025-02-01.

Anthropic. 2024. Claude 3.5 sonnet: Latest model release. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2025-02-01.

Anthropic. 2025. Claude 4: Latest model release. https://www.anthropic.com/news/claude-4. Accessed: 2025-08-01.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 9347–9359, Online. Association for Computational Linguistics.

Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 806–813, Portorož, Slovenia. European Language Resources Association (ELRA).

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and 1 others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Google. 2024. Gemini models. https://deepmind.google/technologies/gemini/. Accessed: 2025-02-01.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Pengcheng Jiang, Cao Xiao, Zifeng Wang, Parminder Bhatia, Jimeng Sun, and Jiawei Han. 2024. TriSum: Learning summarization ability from large language models with structured rationale. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2805–2819, Mexico City, Mexico. Association for Computational Linguistics.

Daniel Kang and Tatsunori B. Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. 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.

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.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. 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.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Yixin Liu, Kejian Shi, Katherine He, Longtian Ye, Alexander Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2024. On learning to summarize with large language models as references. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8647–8664, Mexico City, Mexico. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

MistralAI. 2024. Mistral large 2: A new generation of frontier ai. https://mistral.ai/news/mistral-large-2407/. Accessed: 2025-02-01.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Marcel Nawrath, Agnieszka Nowak, Tristan Ratz, Danilo Walenta, Juri Opitz, Leonardo Ribeiro, João Sedoc, Daniel Deutsch, Simon Mille, Yixin Liu, Sebastian Gehrmann, Lining Zhang, Saad Mahamood, Miruna Clinciu, Khyathi Chandu, and Yufang Hou. 2024. On the role of summary content units in text summarization evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 272–281, Mexico City, Mexico. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

OpenAI. 2024a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024b. Gpt-4o mini. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2025-02-01.

OpenAI. 2024c. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. 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 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4215–4233, Singapore. Association for Computational Linguistics.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

xAI. 2025. Grok 4 api reference: grok-4-0709. https://docs.x.ai/docs/models/grok-4-0709. Accessed: 2025-08-01.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

A Details of LLM-generated Summaries

We adopt the following nine LLMs to generate summaries for both CNN/DailyMail and XSum: GPT-4 (OpenAI, 2024a), GPT-4o (OpenAI, 2024c), GPT-4o-mini (OpenAI, 2024b), Gemini-1.5-flash, Gemini-1.5-pro (Google, 2024), Llama-3.1-instruct-70b, Llama-3.1-instruct-405b (AI@Meta, 2024), Claude-3.5-sonnet (Anthropic, 2024), and Mistral-Large-2407 (MistralAI, 2024).

The prompt is designed to generate brief summaries for the CNN/DailyMail-type datasets (BASE and LLMs-9, as detailed in the primary paper). The feature of the CNN/DailyMail dataset, in which the summary is constructed in three sentences, is maintained.

> Summarize the main content of the following news article in three sentences.

XSum Summary Generation The following prompt is used for the summary generation task for XSum-type datasets. Although XSum summaries are typically one sentence long, we modify the prompt to request a three-sentence summary. This design choice ensures that the distillation process yields a model capable of generating summaries comparable in length to those in the test set.

> Summarize the following article in three sentences. Ensure the summary is concise, with a total word count between 40 and 50 words.

B Details of Summary Content Units

B.1 Summary Content Units Extraction

Below is the prompt used for the SCU extraction task.

> You split the provided input in small sentences separated by an #. The split sentences represent subsentences of the original sentences.
> Example inputs:
> Anuradha Koirala and 425 young women and girls have been sleeping outdoors because of aftershocks. Pushpa Basnet and 45 children she cares for were forced to evacuate their residence. Seven other CNN Heroes and their organizations now assisting in relief efforts.
> Example outputs:
> Anuradha Koirala has been sleeping outdoors. # 425/many young women and girls have been sleeping outdoors. # Many people have been sleeping outdoors because of aftershocks. # Pushpa Basnet was forced to evacuate her residence. # Pushpa Basnet cares for 45 children. # The children were forced to evacuate their residence. # Anuradha Koirala was a CNN Hero. # Pushpa Basnet was a CNN Hero. # Seven other CNN Heros were now assisting relief efforts. # The organizations of CNN Heros were now assisting relief efforts.

B.2 SCUs Intrinsic Evaluation

Since we changed both the LLM and the instruction for SCU extraction, we conducted an intrinsic evaluation to assess the quality of the extracted SCUs. This intrinsic evaluation follows the methodology of Nawrath et al. (2024), and is performed on the REALSumm dataset (Bhandari et al., 2020) and the PyrXSum dataset (Zhang and Bansal, 2021). The evaluation score is built by iterating over each pair of human-annotated SCUs and the llm-extracted SCUs, averaging the maximum ROUGE-1-F1 score obtained for each human-annotated SCU. As this metric is recall-biased, we additionally compute the score in the reverse direction, following Nawrath et al. (2024).

Table 5 presents SCUs extraction quality, with scores generally increasing as the number of examples increases. Across both REALSumm and PyrXSum, GPT-4o and GPT-4o-mini outperform GPT-3.5-turbo. On REALSumm, GPT-4o achieves the highest score (0.79), slightly higher than GPT-4o-mini (0.76). On PyrXSum, both GPT-4o and GPT-4o-mini reach similar, high quality (0.75). All models exhibit substantial improvement when using a 1-shot example compared to zero-shot setting.

R represent the maximum ROUGE-1-F1 score found for each human-annotated SCU, while **P** represent the maximum ROUGE-1-F1 score found for each llm-extracted SCU.

C Analysis of Scoring and Length Penalty

To mitigate potential length bias in SCURank, we evaluate the relationship between summary length

	RealSumm		PyrXSum	
	R	P	R	P
SGUs_3.5	.58	.67	.58	.63
SGUs_4	.61	.69	.61	.66
gpt-3.5-turbo				
0-shot	.54	.66	.55	.62
1-shot	.62	.74	.65	.73
3-shot	.68	.78	.70	.74
gpt-4o-mini				
0-shot	.55	.67	.58	.67
1-shot	.70	.76	.72	.72
3-shot	.76	.79	.75	.77
gpt-4o				
0-shot	.51	.59	.60	.67
1-shot	.73	.78	.71	.76
3-shot	.77	.79	.75	.77

Table 5: Intrinsic evaluation results on REALSumm and PyrXSum. Results for SGUs_3.5 and SGUs_4 are taken from the original paper. Each block reports results for a specific LLM and number of shots. **R** and **P** denote the maximum ROUGE-1 F1 scores in the two evaluation directions.

($||s_i||$) and various scoring configurations. Our goal is to minimize the correlation between the final rank and word count to ensure the model prioritizes information richness over verbosity.

C.1 Evaluation of Length Penalty

We use Kendall's τ to measure the correlation between summary scores and lengths. As shown in Table 6, raw SCU counts ($|c|$) without normalization (N/A) exhibit a strong positive correlation with length across both CNN/DailyMail and XSum datasets. While a linear penalty (len(s)) leads to over-penalization (negative correlation), the **square root penalty** ($\sqrt{\textbf{len}(s)}$) consistently achieves a correlation closest to zero, effectively decorrelating the score from summary length.

C.2 Justification of Linear Sum

We further compared the linear sum against square root and logarithmic transformations of the raw SCU scores. The results indicate that more complex scoring functions do not significantly improve length decorrelation. In the CNN/DM dataset, the simple linear sum ($|c|$) paired with a square root penalty achieves the lowest correlation (0.1316). Following the principle of simplicity, we adopt the linear sum as the primary scoring mechanism defined in Eq. 6.

Dataset	Score Type	N/A	len(s)	$\sqrt{\textbf{len}(s)}$	$\log(\textbf{len}+1)$
CNN/DM	Sum ($\|c\|$)	0.3666	-0.1394	**0.1316**	0.2931
	$\sqrt{\|c\|}$	0.4230	-0.1609	0.1412	0.3022
	$\log(\|c\|+1)$	0.4293	-0.1595	0.1546	0.3202
XSUM	Sum ($\|c\|$)	0.2091	-0.2272	**-0.0346**	0.0768
	$\sqrt{\|c\|}$	0.2761	-0.2707	-0.0135	0.1044
	$\log(\|c\|+1)$	0.2705	-0.2752	-0.0047	0.1070

Table 6: Kendall's τ correlation between various scoring functions and summary word count. Values closer to zero indicate superior mitigation of length bias. Bold values denote the settings used in SCURank.

D Implementation Details

D.1 Distillation in CNN/DailyMail

We employed the BART model (Lewis et al., 2020) as the target model, initialized with the checkpoint of facebook/bart-large-cnn[3]. Prior to the fine-tuning stage described in Liu et al. (2024), a warm-up training using maximum likelihood estimation (MLE) was conducted with 10,000 GPT-3.5 summaries, provided by Liu et al. (2024). Then, the process continued with contrastive learning, as detailed in Section: **Contrastive Learning** of our main paper. To ensure reliability, each experiment was repeated five times, and the average performance was reported.

D.1.1 Explanation to the UnRank Baseline

To evaluate the effectiveness of contrastive learning, we conducted an additional experiment using MLE training only after the warm-up stage. In this setting, the distilled model was trained on all pairs of summaries in the dataset, processing nine different summaries per document. We then compared its performance with that of the model trained with contrastive learning. In our experiments, we referred to the model trained with only MLE as "UnRank."

D.2 Distillation in XSum

The training process for CNN/DailyMail and XSum follows the same procedure. The only difference is the training size of LLMs-9 for the XSum dataset. We used 9,000 gpt-4o-mini summaries to fine-tune the BART checkpoint for the original XSum dataset[4]. The number of summaries is fewer

[3]https://huggingface.co/facebook/bart-large-cnn
[4]https://huggingface.co/facebook/bart-large-xsum

than 10,000 due to the limited size of the original XSum dataset.

D.3 HDBSCAN Clustering

For each SCUs clustering, since the number of samples is relatively small, we set the minimum cluster size and the minimum samples size jointly to **2**, which allows us to capture even small clusters of semantically similar SCUs. Furthermore, the cluster selection epsilon is set to **0.15** to stabilize the clustering results, reducing over-fragmentation caused by small local density variations. To ensure that all SCUs are included in the ranking, we treat 'noise' outliers as individual clusters.

E Dataset Statistics

Two datasets, BASE (Liu et al., 2024), LLMs-9, were employed to train the distilled model. Table 7 shows the statistics of these datasets.

Dataset	#Examples		Avg. Words	
	Train	Valid	Doc.	Sum.
BASE (CNN/DM)	1k	100	601.7	75.7
LLMs-9 (CNN/DM)	1k	100	601.7	85.7
LLMs-9 (XSum)	1k	100	423.5	52.4

Table 7: Dataset statistics used in this study. The #Example and Avg. Words columns report the number of examples and average number of words in documents and summaries, respectively. The BASE dataset is provided by Liu et al. (2024), where the articles drawn from CNN/DailyMail (CNN/DM).

F Significant Test Evaluation

F.1 Paired Bootstrap Test

To evaluate the statistical significance of the performance difference between SCURank and GPTRank, we conducted a paired bootstrap test on the evaluation metrics reported in Table 1 and Table 2. For each metric, we collected all scores for the samples in the evaluation set in each run, resulting in two sets of scores for SCURank and GPTRank. We then performed 10,000 bootstrap resamples of these score pairs, calculating the mean difference between SCURank and GPTRank for each resample. The p-value was computed as twice the proportion of bootstrap samples on the minority side of zero (two-tailed test). A p-value less than 0.05 was considered statistically significant, and is denoted with an asterisk (*) in the results.

Data Type	R_1	R_2	R_L	BLEU	BScore	BaScore
BASE	0.012*	0.035*	0.013*	0.186	0.017*	0.195
LLMs-9	0.043*	0.317	0.311	0.448	0.255	0.699
LLMs-9 (XSum)	0.000*	0.002*	0.000*	0.000*	0.001*	0.000*

Table 8: Paired bootstrap test p-values. * denotes SCURank significantly outperforms GPTRank ($p < 0.05$). Metrics are abbreviated (R: ROUGE, BScore: BERTScore, BaScore: BARTScore).

F.2 Analysis of Significance Results

Table 8 reveals distinct patterns across datasets. On XSum (LLMs-9), SCURank achieves highly significant improvements ($p < 0.01$) across all automatic metrics, demonstrating its effectiveness in highly abstractive summarization scenarios. On CNN/DM (BASE), significant improvements ($p < 0.05$) are observed in ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore, further confirming the superiority of our method. For CNN/DM (LLMs-9), although SCURank consistently achieves higher mean scores than the baseline across several metrics, the differences do not reach statistical significance. This does not imply that SCURank is less effective in this setting; when model outputs are already of high quality, automatic metrics are known to have limited sensitivity in detecting subtle but meaningful improvements.

This is further supported by human evaluation (Section 5.3), where SCURank shows significant gains in Faithfulness ($p < 0.05$) and Completeness ($p < 0.01$), consistent with our LLM-based evaluation (Section 5.4).

G Human Evaluation

G.1 Guidelines

The task was hosted on Amazon Mechanical Turk (MTurk). Each task included an article and two summaries generated by the distilled models trained with SCURank and GPTRank. To mitigate positional bias, the summaries were randomly ordered. Three independent annotators were recruited for each task to ensure reliable evaluation.

Under the previous qualification criteria, annotators were required to complete a minimum of 500 HITs, maintain an approval rate of at least 90%, and use English as their primary language. Each annotator was provided with detailed guidelines explaining the evaluation criteria and scoring system. The guidelines are defined in Figure 5.

G.2 Evaluation Criteria

Each summary is evaluated based on three dimensions: Faithfulness, Conciseness, and Completeness. Then, annotators rate each criterion on a five-point Likert scale, where one indicates poor quality and five indicates excellent quality. Finally, annotators are asked to compare the two summaries and select which one they prefer overall. A summary needs to receive at least two votes to be considered the preferred summary. If both summaries receive one vote each and one tie, it is considered a tie.

H Sample Summaries

H.1 SCURank vs. MLE

To evaluate the performance of the SCURank distilled model and the MLE distilled model, we provide sample summaries generated by both models. Table 9 presents these summaries alongside the human-written summaries. The SCURank distilled model explicitly mentions "amid population growth and environmental pressures" as the primary cause of the Hainan gibbons' decline. Similarly, human-written summary 1 conveys the same idea but with different wording, "due to deforestation and human population growth." This differentiate in phrasing results in a lower ROUGE-1 score for the SCURank distilled model in this case. In contrast, the MLE distilled model focuses primarily on the role of historical Chinese documents, which is similar to the human-written summary 3. This similarity leads to a higher ROUGE-1 score for the MLE distilled model. This result indicates that, while the SCURank distilled model provides more comprehensive information, its summary may sometimes yield a lower ROUGE-1 score due to differences in content focus. This suggests that the ROUGE-n scores reflect only content similarity rather than overall summary quality.

H.2 SCURank vs. GPTRank

To further demonstrate the capability of SCURank in capturing diverse information, we present a qualitative comparison between SCURank and GPTRank in Table 10. In this example, the source article describes a controversy involving the partner of a political leader.

As shown in the table, the **GPTRank** model tends to focus heavily on sensational and high-frequency keywords found in the lead paragraph (e.g. specific quotes about "rape fantasies" and "complex human beings"). While these details are salient, the summary misses the narrative context of their relationship.

In contrast, the **SCURank** model captures a unique semantic unit: the irony of their meeting origin ("Bennett contacted him to correct something he had written about her"). This specific detail, while not containing high-frequency controversial terms, adds significant narrative completeness to the summary. This observation supports our quantitative findings that SCURank, by leveraging Summary Content Units (SCUs), effectively identifies and retains distinct semantic information that might be overlooked by ranking methods relying solely on surface-level overlap or LLM-based preference, which can be biased towards sensational content.

I LLMs Result

I.1 LLMs generated summaries

We used the same LLMs as in the LLMs-9 dataset to generate summaries for the test dataset and evaluate their performance. Table 11 presents the performance of various LLMs on both of the CNN/DailyMail and XSum datasets using multiple evaluation metrics and extractiveness measures.

I.2 LLM-based Summary Comparison Prompt

The following prompt was used to instruct each LLM to compare two summaries based on their information richness and importance.

> Compare the following summaries based on **information richness** (how much relevant detail is preserved) and **importance** (focus on the most significant points).
>
> For each summary, evaluate:
> - Information richness: Comprehensiveness, specific details, coverage of key topics
> - Importance: Focus on high-impact information, critical insights, actionable content
>
> Provide a brief analysis and rank the summaries from best to worst, explaining your reasoning.
>
> Input format:
> Article: [article text]
> Summary 1: [summary 1 text]
> Summary 2: [summary 2 text]
>
> Output format:
> Summary 1 Analysis: [Your analysis here]
> Summary 2 Analysis: [Your analysis here]
> Winner: [1 or 2 or "tie" if they are equally good]

Article
Historical Chinese documents have helped scientists to track the decline of the world's rarest primates. Today, China has between 26 and 28 Hainan gibbons left, but government records that date back to the 17th Century show that gibbons were once widespread across half of the country. The apes began to disappear from the documents about 150 years ago, corresponding with population growth. The study is published in the Proceedings of the Royal Society B. Hainan gibbons are now limited to a few isolated patches of forest in the south west of China. They live in just four social groups, one of which was only discovered a few weeks ago. ... Dr Sam Turvey, from the Zoological Society of London, said: "China is one of the few places in the world that has a very very rich, long historical record. ... These included records of animals, including gibbons, he said. "We looked at the pattern of disappearance of gibbons through time and how that varied from place to place and the different environmental conditions and human pressures that were also present in these places." The archives show that gibbons were a common sight in about 20 provinces in China well into the 17th and 18th Century. However, Dr Turvey said it was "a stark contrast to their very imperilled position today". "We see a really steep increase in population decline and real population collapse across China about 100-150 years ago," he added. ... The researchers said a better understanding of the animals' decline would help them to establish a conservation plan for the country's last few Hainan gibbons. Dr Turvey said: "It is an incredible privilege to be able to see gibbons in China in the wild. "The Hainan gibbon is such as rare species, but knowing that this species is still hanging on there gives you hope that conservation will be able to bring that population back from the brink." Follow Rebecca on Twitter

SCURank distilled model's Output
Researchers have used historical Chinese documents to identify the decline of Hainan gibbons, which are now limited to isolated areas in China, amid population growth and environmental pressures.

MLE distilled model's Output
Historical Chinese documents have helped scientists understand the decline of the Hainan gibbon, revealing that gibbons were once widespread across half of the country, but are now limited to a few isolated patches.

Human-written Summary 1
Thanks to China's extensive historical records, scientists have been able to track the population decline of the rarest primate in the world. The Hainan gibbon was once widespread across the country, but due to deforestation and human population growth, there are now between 26 to 28 of the animals left.
ROUGE-1: SCURank: 0.405 | MLE: 0.452

Human-written Summary 2
Hainan gibbons in China have been decreasing precipitously in population ever since the 1800s, new records show. The decline accelerated in the 20th century with massive human population growth and deforestation. Today China has roughly 28 of the gibbons left.
ROUGE-1: SCURank: 0.324 | MLE: 0.247

Human-written Summary 3
Historical Chinese government documents show that although the world's rarest primates were once widespread across half of the country, their numbers are now declining. The remaining primates, Hainan Gibbons, are now isolated to small patches of forest in China's southwest region. Researchers continue to study the decline in the species.
ROUGE-1: SCURank: 0.400 | MLE: 0.541

Table 9: Sample summaries generated by the SCURank- and MLE-distilled models, together with the corresponding human-written summaries. ROUGE-1 scores for the SCURank and MLE models are also reported.

Article Green party leader Natalie Bennett has been forced to distance herself from a bizarre online site run by her boyfriend. Jim Jepps used a blog called The Daily Maybe to defend 'rape fantasies', describe paedophiles as 'complex human beings' and question why teachers who have relationships with pupils are put on the sex offenders register. **The couple met five years ago when Ms Bennett contacted him to correct something he had written about her**, but now the Green party stresses they do not 'want to be associated' with his internet rants. ... (Context omitted for brevity) ... Ms Bennett has been dating Mr Jepps for five years, after meeting online. ...
SCURank Distilled Model Natalie Bennett, the leader of the UK's Green party, has distanced herself from a website run by her boyfriend, Jim Jepps, who wrote controversial blog posts about sex, rape, and paedophiles. The couple met five years ago and **Bennett contacted him to correct something he had written about her**. The Green party has said that they do not want to be associated with his internet rants.
GPTRank Distilled Model Natalie Bennett, the leader of the Green party, has distanced herself from a blog run by her boyfriend, Jim Jepps, who used the site to defend 'rape fantasies' and describe paedophiles as 'complex human beings'. The Green party has stressed that they do not want to be associated with his internet rants. Bennett has been dating Jepps for five years, but little is known about her private life.

Table 10: Qualitative comparison of summaries generated by SCURank and GPTRank. **Bold text** highlights a specific Semantic Content Unit (SCU) regarding the origin of the couple's relationship. SCURank successfully captures this unique semantic detail, whereas GPTRank focuses primarily on high-frequency controversial keywords.

	R-1	R-2	R-L	BLEURT	BS	BaS	Coverage	Density
CNN/DailyMail								
GPT-4o-mini	38.3	15.7	24.8	54.6	68.7	-2.71	0.81	2.32
GPT-4o	41.2	15.9	26.0	54.7	69.4	-2.78	0.81	2.47
GPT-4-turbo	41.2	15.9	26.0	54.7	69.4	-2.78	0.81	2.47
Gemini-1.5-flash	43.1	17.9	28.0	55.5	70.4	-2.40	0.83	2.53
Gemini-1.5-pro	41.5	16.7	27.2	53.9	69.7	-2.67	0.79	2.26
Llama-3.1-70b	42.7	18.4	28.0	54.4	69.7	-2.47	0.86	3.24
Llama-3.1-402b	42.7	19.0	28.1	55.2	69.9	-2.39	0.86	3.25
Mistral-Large	45.1	18.8	29.4	55.9	71.2	-2.48	0.85	2.77
Claude-3.5-sonnet	42.3	16.9	27.2	51.5	69.5	-2.65	0.83	3.01
XSum								
GPT-4o-mini	37.1	14.3	24.9	53.2	67.7	-2.87	0.80	2.53
GPT-4o	38.6	14.4	24.8	50.5	67.6	-2.93	0.78	2.38
GPT-4-turbo	38.6	14.4	24.8	50.5	67.6	-2.93	0.78	2.38
Gemini-1.5-flash	42.7	18.2	29.4	55.0	69.9	-2.60	0.81	2.73
Gemini-1.5-pro	41.3	15.3	27.7	53.8	69.1	-2.75	0.78	2.24
Llama-3.1-70b	40.4	14.7	25.9	53.1	68.7	-2.63	0.83	2.97
Llama-3.1-402b	42.4	17.8	28.8	54.8	69.6	-2.51	0.86	3.42
Mistral-Large	44.4	17.7	29.8	54.5	69.8	-2.64	0.83	2.89
Claude-3.5-sonnet	42.5	17.1	28.1	53.2	69.6	-2.69	0.83	2.97

Table 11: Automatic evaluation results of summaries generated by various LLMs on CNN/DailyMail and XSum datasets. Evaluation metrics include R-1(ROUGE-1), R-2(ROUGE-2), R-L(ROUGE-L), BLEURT, BS(BERTScore), BaS(BartScore), Coverage, and Density.

Summary Evaluation and Comparison Task

Task Overview

You will be evaluating and comparing two different summaries of the same article. Your task is to assess the quality of each summary based on three key criteria, and then make an overall comparison between them.

What You'll Do

- Read the original article carefully
- Read Summary 1 and evaluate it on three dimensions
- Read Summary 2 and evaluate it on three dimensions
- Compare the two summaries and indicate which is better overall

Evaluation Criteria

You will rate each summary on:

- **Completeness:** How well the summary captures the core messages and important information from the article
- **Conciseness:** How efficiently the summary conveys information without unnecessary details or redundancy
- **Faithfulness:** How accurate the summary is - whether it contains any errors, misrepresentations, or hallucinated information

STEP 1

Read the Original Article

Please read the following article carefully. You may refer back to it at any time during the evaluation.

${article}

STEP 2

Summary Evaluation

Please rate each summary on the following three dimensions:

Summary 1

${summary1}

Summary 2

${summary2}

Figure 5: Annotation guidelines provided to annotators for human evaluation on MTurk.