TriSum: Learning Summarization Ability from Large Language Models

Pengcheng Jiang*, **Cao Xiao**†, **Zifeng Wang***, **Jimeng Sun***, **Jiawei Han***

*University of Illinois at Urbana Champaign, † GE Healthcare
*{pj20, zifengw2, jimeng, hanj}@illinois.edu, †danicaxiao@gmail.com

Abstract

The advent of large language models (LLMs) greatly advanced natural language processing tasks, such as text summarization. However, due to their substantial model size, computational demands, and potential privacy concerns when transmitting sensitive data for remote processing, their utility can be limited in resource-constrained environments and applications prioritizing data privacy. To address this, we introduce a framework called TriSum to distill the text summarization capabilities of LLMs into a more compact local model. In the first step, we employ LLMs to probe and extract a collection of *aspect-triple* rationales and summaries. We then refine them by employing a dual-scoring method to identify the most high-quality rationales. Moving to the second step, we train a smaller local model using these carefully organized summarization tasks. Our training strategy employs a curriculum learning approach, gradually progressing from individual tasks to more complex combinations. Our evaluations demonstrate that our TriSum method empowers the local model to outperform baselines by 4.5%, 8.5%, and 7.4% for the abstractive summarization task on CNN/DailyMail, XSum, and ClinicalTrial respectively. Beyond improved performance, our approach also offers insights into the rationale guiding the summarization process, thus enhancing interpretability.

Introduction

Large language models (LLMs), such as GPT-3 (Brown et al. 2020) and its successors (Chowdhery et al. 2022; Touvron et al. 2023; OpenAI 2023), has greatly advanced natural language processing tasks, including machine translation (Brants et al. 2007), question-answering (QA) systems (Yang et al. 2019; Bao et al. 2021), and text summarization (Liu and Lapata 2019). However, due to their substantial model size and computational demands, their utility can be limited in resource-constrained environments (Strubell, Ganesh, and McCallum 2019). Moreover, privacy becomes a major concern when sending proprietary data to external LLM services like ChatGPT.

Among others, text summarization is a crucial task for transforming lengthy texts into concise yet informative summaries (Radev, Hovy, and McKeown 2002). However, many existing methods struggle to generate structured summaries (Brown et al. 2020; Gekhman et al. 2023; Liu et al. 2023). These structured summaries need to encompass essential aspects, key entities and relationships, and a coherent final summary derived from these aspects and rationales. Recent developments have seen the utilization of LLMs to grasp a text's topic structure and core ideas (Vaswani et al. 2017b; Wei et al. 2023), suggesting their potential in generating structured text summaries. While rational distillation from LLMs has been employed for NLP tasks like QA, natural language understanding (NLU), and arithmetic reasoning (Wang et al. 2022; Hsieh et al. 2023; Magister et al. 2023; Ho, Schmid, and Yun 2023), its applicability to abstractive text summarization remains unexplored.

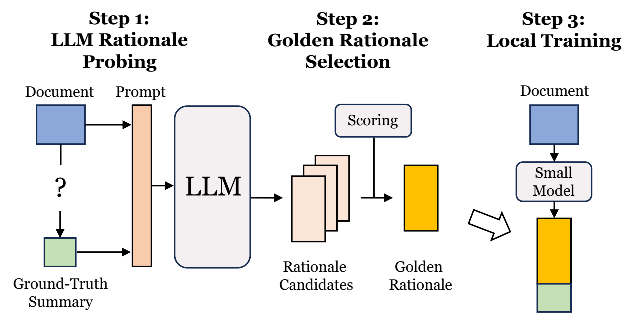


Figure 1: A conceptual demonstration of our three-step framework TriSum that endows local small models with LLM's text summarization capability.

In this study, we aim to distill LLMs' text summarization prowess into a more compact local model. We enhance the transparency and interpretability of this local model by incorporating elicited rationales from LLMs' summarization process as additional guidance. To achieve this, we introduce a three-step framework TriSum (as shown in Figure 1) involving LLM rationale probing, golden rationale selection, and local training:

Step 1: We first prompt vital *aspect-triple* rationales and summaries from the input text using LLMs. This set includes essential aspects, relevant triples extracted from the text, and a concise summary that's tied to these aspects and triples.

Step 2: Next, to ensure quality, we employ a dual-scoring

method for selecting golden (high-quality) rationales to use in the subsequent training. This method evaluates the summary's quality based on semantic similarity and ensures coherent rationales using a topic distribution-based approach.
Step 3: Last, we train our compact local model using a curriculum learning approach (Nagatsuka, Broni-Bediako, and Atsumi 2021; Xu et al. 2020). This method progressively fine-tunes the model by starting with simpler tasks and gradually advancing to more complex ones. This process enables our model to gradually incorporate the rationalized summarization skills acquired from the LLMs.

Our research brings the following contributions.

- We introduce a new approach that distills LLMs' abstractive text summarization power into a small local model.
- We design a scoring mechanism to select high-quality rationales, which serves as a robust base for training the local model.
- Through comprehensive experiments, we validate our framework's effectiveness. Results show that incorporating LLM-generated rationales boosts our local model's summarization performance.
- We enhance model interpretability by analyzing LLM-derived rationales, deepening our insight into their summarization processes.

Overall, our study streamlines powerful summarization models in resource-limited contexts, offering insights into harnessing LLMs' inherent summarization abilities.

Related Work

Text Summarization using LLMs. Transformer-based language models (Vaswani et al. 2017a) have improved the quality of text summarization significantly. These models excel at capturing complex relationships in long texts. Recent research has taken this transformer architecture further for summarization tasks (Liu and Lapata 2019; Lewis et al. 2019; Zhang et al. 2020; Raffel et al. 2020), utilizing LLMs such as ChatGPT, GPT-4, and PaLM (OpenAI 2023; Chowdhery et al. 2022) which have billions of parameters and are trained on vast amounts of text. Their performance can be further enhanced when prompted to execute step-by-step reasoning (Wei et al. 2023).

However, the resource demands of LLMs have limited their widespread use. Concerns over privacy when using LLM-as-a-service APIs have also arisen, especially for sensitive data. This highlights the need for more compact local models that can still capture summarization abilities. To harness the summarization ability of LLMs, Liu et al. (2023) used summaries created by LLMs as benchmarks for training their local models. LLMs were also used to evaluate summary quality during training. However, this approach did not fully transfer the reasoning skills of LLMs to the local models, indicating a partial capture of LLMs' summarization abilities. Also, the uncertainty of labels generated by deep learning models may affect reliability.

Rationale Distillation for Interpretability in LLMs Knowledge distillation, as introduced by Hinton, Vinyals, and Dean (2015), refers to the concept for transferring knowledge from a large model (teacher) to a smaller one (student) to make deep learning models usable in resource-limited environments. This idea has been applied and extended across various fields (Sanh et al. 2019; Tang et al. 2019; Jiao et al. 2019; Chen et al. 2019; Lin et al. 2020; Wang et al. 2023). Notably, Chen et al. (2019) focused on abstractive summarization, while Lin et al. (2020) emphasized extractive summarization. The complexity of deep neural networks has driven research toward making AI models interpretable (Ribeiro, Singh, and Guestrin 2016; Doshi-Velez and Kim 2017). Rationale generation is an emerging technique in interpretability, highlighting a model's key reasoning steps (Zaidan and Eisner 2008; Yu et al. 2020).

In knowledge distillation, rationale generation enhances interpretability, offering insights into the decision-making of LLMs. This informs the development of better knowledge distillation methods. (Wang et al. 2022) developed a smaller model using LLM-generated rationales and questions. Others (Shridhar, Stolfo, and Sachan 2023; Ho, Schmid, and Yun 2023; Magister et al. 2023; Hsieh et al. 2023) used LLM-produced rationales to train models, improving performance and transparency in predictions, primarily for tasks like QA, NLU, arithmetic reasoning, and extractive summarization (Yang et al. 2023). This has left a gap concerning abstractive text summarization. To bridge this gap, we introduce an *aspect-triple* rationale generation approach, aimed at distilling the summarization prowess of LLMs. This method consists of a phased procedure of extracting essential aspects, pinpointing primary entities and relations, and constructing a definitive summary.

Method

Overview of `TriSum`

We introduce `TriSum`, an approach transferring document summarization ability from an LLM ($\geq$100B) to a small LM ($\leq$1B) via rationale probing, golden rationale selection, and curriculum learning. Here, we assume LLM has reasoning ability and can be used for prompting. Before discussing in detail, we define a few key concepts and notations below.

Definition 1 (Aspect) *An (essential) aspect α is defined as a few words representing a distinct topic in a document.*

- Example: In a document about climate change, an aspect might be "rising sea levels".

Definition 2 (Triple) *A (detailed) triple $\tau = \langle s|r|o \rangle$ is a structure formatting a piece of free-text into a subject s, a relation r, and an object o.*

- Example: For a sentence "Cats eat fish.", "Cats" is the subject, "eat" is the relation, and "fish" is the object. Together, they form a triple $\langle Cats|eat|fish \rangle$.

Task 1 (Aspect Extraction (AE)) *Given a document D, the task of aspect extraction is defined as extracting its essential aspects A (where each $\alpha \in A$ represents an aspect) that approximates the distribution $p(A|D)$.*

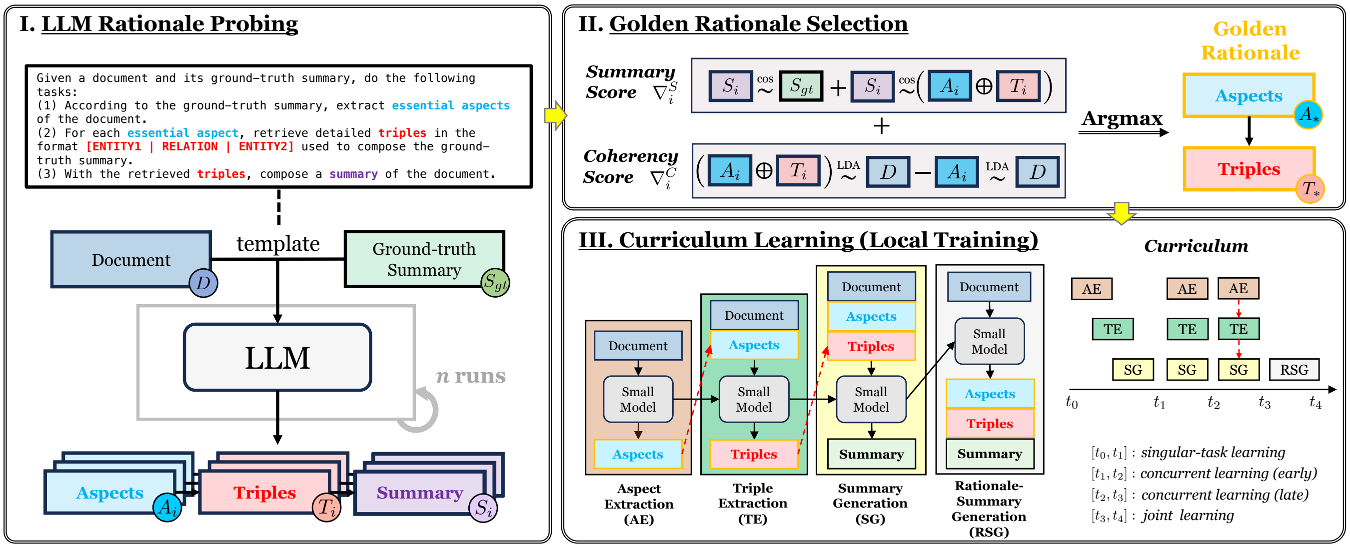


Figure 2: **Distilling text summarization ability from LLM to local model using `TriSum`. Step 1. LLM Rationale Probing:** Employing a template-based prompt incorporating the given document and ground-truth summary, we engage an LLM to generate a set of n step-by-step rationales across n iterations. **Step 2. Golden Rationale Selection:** We leverage summary and coherency scores to meticulously choose high-quality training rationales, enhancing the training dataset. **Step 3. Curriculum Learning:** We implement a curriculum learning strategy to train our compact auto-regressive generation model with rationalized summarization ability from easy to challenging tasks.

Task 2 (Triple Extraction (TE)) *Given a document D and its aspects A, the triple extraction task is defined as extracting triples T (where each $\tau \in T$ represents a triple) from D, aiming to learn the distribution $p(T|D, A)$.*

Task 3 (Summary Generation (SG)) *Given a document D, its aspect A, and the triples T, the task of summary generation is defined as generating a summary S that approximates the distribution $p(S|D, A, T)$.*

Task 4 (Rationale-Summary Generation (RSG)) *Given a document D, the task of rationale-summary generation is defined as generating both rationale and summary that approximates the distribution $p(A, T, S|D)$.*

As illustrated in Figure 2, `TriSum` operates through three key steps: (1) tapping into the LLM for *aspect-triple* rationales in training data; (2) selecting golden (high-quality) rationales based on summary and coherency scores; and (3) training a local model using a curriculum learning approach. We detail each step of `TriSum` as follows.

Step 1: LLM Rationale Probing

Given a set of documents for training, our initial step involves leveraging the LLM to iteratively generate a set of *aspect-triple* rationales alongside their corresponding summaries. The objective is the following: first, to enable the LLM to pinpoint essential aspects, and subsequently, to elaborate on each aspect using detailed triples.

In this process, the auto-regressive LLM generates both the rationale R and the summary S. We denote the length of a sequence by $|\cdot|$. The rationale $R = (A, T)$ is a sequence of tokens $\{r_1, r_2, ..., r_{|R|}\}$, which is composed of aspect tokens $\{a_1, a_2, ..., a_{|A|}\}$ followed by triple tokens $\{t_1, t_2, ..., t_{|T|}\}$, where $|R| = |A| + |T|$. Here, A represents essential aspects, and T provides detailed triples. Each a_i is an individual token in A, and each t_j is an individual token in T. The summary S is defined as $\{s_1, s_2, ..., s_{|S|}\}$. Each token r_i is generated based on the document D, the ground-truth summary S_{gt}, and the tokens previously generated, $R^{<i} = \{r_1, r_2, ..., r_{i-1}\}$. The prediction of s_i is contingent upon the generated rationale R and $S^{<i} = \{s_1, s_2, ..., s_{i-1}\}$:

$$p(R|D, S_{gt}) = \prod_{i=1}^{u} p(r_i|D, S_{gt}, R^{<i}),$$
$$p(S|D, S_{gt}, R) = \prod_{i=1}^{v} p(s_i|D, S_{gt}, R, S^{<i}). \quad (1)$$

where S_{gt} denotes the ground-truth summary corresponding to the document D. To equip our local model with more interpretable and high-quality rationales, we prompt the LLM for n iterations, which results in n pairs of rationale-summary, denoted as $\{R_i, S_i\}_{i=1}^{n}$ for each document. Each pair, where $R_i = (A_i, T_i)$, serves as a candidate for the golden rationale selection described in the subsequent step.

Step 2: Golden Rationale Selection

Given the generated candidate rationales, we then incorporate two types of scores - *Summary Score* and *Latent Dirichlet Allocation (LDA)-based Coherence Score* to select the golden rationales.

Summary Score. For each rationale R_i in the candidates $\{R_i, S_i\}_{i=1}^{n}$, suppose $\hat{R}_i$, $\hat{S}_i$, and $\hat{S}_{gt}$ are the word embeddings of the rationale, LLM-generated summary, and the ground-truth summary respectively, the summary score is a weighted average of two semantic similarity:

$$\nabla_i^S = \mathrm{sim}\langle \hat{S}_i, \hat{S}_{gt}\rangle + \phi_\alpha \cdot \mathrm{sim}\langle \hat{S}_i, \hat{R}_i\rangle, \quad (2)$$

where ϕ_α is a hyper-parameter balancing the importance of two components, and $\mathrm{sim}\langle\cdot\rangle$ is the semantic similarity computation. For example, $\mathrm{sim}\langle x, y\rangle$ can be computed using cosine similarity as $\mathrm{sim}\langle x, y\rangle = \frac{x \cdot y}{||x|| \cdot ||y||}$. The first term in Eq. (2) emphasizes the similarity between the generated summary and the ground-truth summary, while the second term focus on the relevance between the generated summary and the prepended rationale, in avoid scoring high for lazy generation by the LLM (i.e., simply repeat the given ground-truth summary regardless of the generated rationale).

LDA-based Coherence Score. We also want to evaluate how the aspects and rationale align with the latent topics of the document. Here, we employ a Latent Dirichlet Allocation (LDA) model (Blei, Ng, and Jordan 2003), an algorithm that represents each document as a blend of a certain number of topics. To be specific, we represent each document as a distribution over the entire lexicon. Given a document D, a rationale R_i, and aspects $A_i \in R_i$, we initially train an LDA model on the corpus (all documents in the dataset) to identify latent topics with our specified number of topics k. It is important to clarify that the topics identified by LDA are based on the entire corpus, in contrast to the aspects which are specific to individual documents. From this model, we derive the topic distributions p^D_{LDA}, $p^A_{i,\mathrm{LDA}}$, and $p^R_{i,\mathrm{LDA}}$ for the document, the i-th aspects, and the i-th rationale, respectively. The coherence score ∇_i^C is calculated as the KL-divergence between these distributions:

$$\begin{aligned}\nabla_i^C = (1+\phi_\beta) \cdot KL(p^D_{\mathrm{LDA}} || p^R_{i,\mathrm{LDA}}) \\ - KL(p^D_{\mathrm{LDA}} || p^A_{i,\mathrm{LDA}}),\end{aligned} \quad (3)$$

where ϕ_β is a parameter that manages the weight of the $KL(p^D_{\mathrm{LDA}} || p^R_{i,\mathrm{LDA}})$ term itself, and $KL(\cdot||\cdot)$ symbolizes the KL-divergence computation:

$$KL(p||q) = \sum_j p(j) \log \frac{p(j)}{q(j)}, \quad (4)$$

where p and q are the probability distributions compare. The score ∇_i^C in Eq. (3) fosters two primary objectives:

- $\phi_\beta \cdot KL(p^D_{\mathrm{LDA}} || p^R_{i,\mathrm{LDA}})$, an term that enhances the topical coherence between the document and rationale.
- $KL(p^D_{\mathrm{LDA}} || p^R_{i,\mathrm{LDA}}) - KL(p^D_{\mathrm{LDA}} || p^A_{i,\mathrm{LDA}})$, a term which encourages the triples ($T_i \in R_i$) to refine this coherence beyond what is achieved by aspects alone.

The final selection of optimal rationales, denoted as $R_* = (A_*, T_*)$, is based on those that yield the highest combined score of Eq. (2) and Eq. (3), and given by Eq. (5),

$$R_* = \mathrm{argmax}_i(\nabla_i^S + \lambda_{cs} \cdot \nabla_i^C), \quad (5)$$

where λ_{cs} is a balancing hyperparameter that manages the relative contributions of the two scores. We then use the gold rationales as the supervision to train our local lightweight language model in the following step.

Step 3: Curriculum Learning

To train the student Seq2Seq language model with the selected golden rationales for rationalized text summarization, we introduce an approach reminiscent of curriculum learning (Bengio et al. 2009; Hacohen and Weinshall 2019; Nagatsuka, Broni-Bediako, and Atsumi 2021; Xu et al. 2020), which facilitates learning in stages of increasing complexity. This strategy consists of the following phases: (1) Singular-task learning, (2) Concurrent learning, and (3) Joint learning.

For the first two phases, we focus on the tasks of *aspect extraction*, *triple extraction*, and *summary generation*, distinguished by prefix tokens $\langle$`AspExt`$\rangle$, $\langle$`TriExt`$\rangle$, and $\langle$`SumGen`$\rangle$, respectively. Additionally, we use prefix tokens $\langle$`article`$\rangle$, $\langle$`aspects`$\rangle$, $\langle$`triples`$\rangle$, $\langle$`summary`$\rangle$ to specify D, A, T, and S, respectively.

Singular-task learning Initially, we train the model on each task separately, aiding the model in developing a baseline understanding and ability to handle each task individually. For instance, in *aspect extraction*, we aim to train a model that minimizes the loss $\mathcal{L}_A$ given the document D:

$$\mathcal{L}_A = -\sum_{D \in \mathcal{D}} \log p(A_* | D; \theta_s), \quad (6)$$

where $\mathcal{D}$ is the training set of documents, $p(A|D) = \prod_{j=1}^m p(a_j | D, A^{<j})$, with m the length of the aspects in the rationale, a_j the j-th token of the aspects, and $A^{<j}$ the previous generated aspect tokens. The model follows a similar procedure for *triple extraction* and *summary generation*, focusing on minimizing losses $\mathcal{L}_T$ and $\mathcal{L}_S$, respectively:

$$\mathcal{L}_T = -\sum_{D \in \mathcal{D}} \log p(T_* | D, A_*; \theta_s), \quad (7)$$

$$\mathcal{L}_S = -\sum_{D \in \mathcal{D}} \log p(S_{gt} | D, A_*, T_*; \theta_s). \quad (8)$$

Concurrent Learning Once the model has become proficient in performing individual tasks, we advance to the concurrent learning phase where the model simultaneously learns the tasks. This phase allows for task interplay and reciprocal reinforcement of learning. To facilitate a smooth transition, we further bifurcate this phase into early-stage and late-stage.

Early Stage: LLM-guided Training. In the early phase, we use the aspects A_* and triples T_* from the best rationale R_*, along with the document D, as the supervisory signal for each task. The model is trained to minimize the loss:

$$\begin{aligned}\mathcal{L}_{\mathrm{concurrent-early}} = -\sum_{D \in \mathcal{D}} \Big[\log p(A_* | D; \theta_c) \\ + \log p(T_* | D, A_*; \theta_c) + \log p(S_{gt} | D, R_*; \theta_c) \Big],\end{aligned} \quad (9)$$

Using the LLM's output as a form of teacher forcing (Bengio et al. 2015) allows the model to focus on learning the structured (aspect-triple-summary) summarization in the early stage, without its own flawed prediction distracting it.

Late Stage: Self-guided Training. As we transition to the later stages, our focus pivots to training the model using its

Dataset	# Samples Train	Valid	Test	# Words Doc.	Sum.
CNN/DailyMail	287,113	13,368	11,490	766.6	54.8
XSum	204,045	11,332	11,334	414.5	23.0
ClinicalTrial	163,088	20,386	20,386	181.4	45.2

Table 1: **Statistics of abstractive summarization datasets.**

own predictions as inputs for subsequent tasks. This strategy is characterized by a cascading training approach: the model begins with aspect extraction, progresses to triple extraction, and ultimately leads to summary generation. The benefit of this approach stems from its sequential information flow, where the outcome of one task informs the next. However, a challenge emerges due to the computational overhead of decoding intermediate results, such as aspects and triples. To mitigate this, while maintaining the sequential integrity, we employ greedy decoding. This method accelerates the process by selecting the most likely token at each step, eliminating the need for full-blown generation at every juncture. Based on this, Eq. (9) is changed to:

$$\mathcal{L}_{\text{concurrent}-\text{late}} = - \sum_{D \in \mathcal{D}} \Big[\log p(A_* | D; \theta_c) + \log p(T_* | D, \tilde{A}; \theta_c) + \log p(S_{gt} | D, \tilde{A}, \tilde{T}; \theta_c) \Big] \quad (10)$$

where $\tilde{A}$ and $\tilde{T}$ represent the intermediate aspects and triples obtained generated through greedy decoding by the model itself. The primary aim of this phase is twofold: (1) to diminish the model's dependency on LLM-provided rationales and, (2) to augment the model's capability for autonomous learning, with the overarching aspiration of enabling it to generate its own rationales and summaries.

Joint Learning In the final phase, we enhance the model's ability to concurrently generate both the rationale and the summary from a given document with the *rationale-summary generation* task. Different from the late stage of concurrent learning, this stage streamlines the process by collapsing three pairs of encode-decode processes into a single pair. We use the optimal rationale from the LLM and the ground-truth summary as the labels. We introduce the prefix token $\langle \texttt{RatGen} \rangle$ for this task. The model aims to minimize the following loss function:

$$\mathcal{L}_{\text{joint}} = - \sum_{D \in \mathcal{D}} \Big[\lambda_R \log p(R_* | D; \theta_r) + \lambda_S \log p(S_{gt} | D, \tilde{R}; \theta_r) \Big] \quad (11)$$

where S_{gt} is the human-annotated ground-truth summary in the dataset, $\tilde{R}$ is the generated rationale via greedy decoding, and λ_R and λ_S are hyperparameters that balance the importance of rationale and summary generations.

Through our strategically designed curriculum learning process, the model progressively gains the capability to generate accurate and succinct rationales and summaries.

Experiments

Implementation

Data Source Our evaluation of `TriSum` is carried out using three datasets: CNN/Daily Mail (CNNDM) v3.0.0 (Nallapati et al. 2016), XSum (Narayan, Cohen, and Lapata 1808), and a bespoke dataset we have developed from Clinical Trial[1]. The comprehensive statistics of these datasets can be found in Table 1. To construct the ClinicalTrial dataset, we treat the "detailed description" from Clinical Trial as the document and the "brief summary" as its corresponding ground-truth summary. From an original total of 305,591 samples, we have selected 203,860 (with a splitting ratio of 8:1:1), filtering out entries where documents exceed 1,024 tokens or where summaries surpass 256 tokens.

Model and Parameters For the rationale generation and the summarization process, we employ GPT-3.5 (specifically, the gpt-3.5-turbo[2]) as the LLM. In the LLM rationale probing phase, we prompt the LLM differently for each dataset: $n = \{15, 8, 8\}$ times for CNNDM, XSum, and ClinicalTrial respectively. This generates a diverse set of potential rationale candidates. The parameters for the golden rationale selection are set as follows: $\phi_\alpha = 0.6$, $\phi_\beta = 1.3$, and $\lambda_{cs} = 1.5$. We use cosine similarity to calculate the summary score with the embeddings retrieved from text-davinci-003 (a GPT-3.5 model that provides embedding). LDA latent topics are specified at 200, 500, and 300 for CNNDM, XSum, and ClinicalTrial respectively. For the joint learning phase, the parameters are fixed at $\lambda_R = 0.8$ and $\lambda_S = 1.2$.

Training For both CNNDM and XSum datasets, we utilize the BART-Large (Lewis et al. 2019) checkpoints[3] that have been fine-tuned specifically for these datasets, as the backbone models. In the case of ClinicalTrial, we fine-tune the BART-Large CNNDM checkpoint using only the summary to create a backbone model. All models, including the baselines, undergo fine-tuning for three epochs, with an early stopping mechanism in place to optimize performance.

Baselines We compare `TriSum` to baseline abstractive summarization models including BERTSumAbs (Liu 2019), T5 (Raffel et al. 2020), BART (Lewis et al. 2019), PEGASUS (Zhang et al. 2020), GSum (Dou et al. 2021), BigBird (Zaheer et al. 2021), SimCLS (Liu and Liu 2021), SeqCo (Xu et al. 2022), GLM (Du et al. 2022), and GPT-3.5.

Evaluation We use the following evaluation metrics:

- ROUGE-F1: measures the overlap of n-grams between the generated summary and the reference summary. We measure ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L).
- BERTScore and BARTScore: measure the semantic similarity between the generated summary and the reference summary using pre-trained language models $\text{RoBERTa}_{\text{Large}}$ and $\text{BART}_{\text{Large}}$, respectively.

[1]https://clinicaltrials.gov/

[2]We use the checkpoint gpt-3.5-turbo-0613, available at https://platform.openai.com/docs/models/gpt-3-5

[3]Accessible at https://huggingface.co/facebook/bart-large-cnn and https://huggingface.co/facebook/bart-large-xsum

	CNN/DailyMail				XSum				ClinicalTrial			
Model	**R-1**	**R-2**	**R-L**	**Δ**	**R-1**	**R-2**	**R-L**	**Δ**	**R-1**	**R-2**	**R-L**	**Δ**
Baselines												
BERTSumAbs (Liu and Lapata 2019)	41.2	18.7	37.2	+13.6%	38.8	16.5	31.0	+28.3%	39.2	19.3	29.6	+19.3%
$T5_{Large}$ (Raffel et al. 2020)	42.4	20.8	39.9	+7.0%	40.1	17.2	32.3	+23.5%	41.3	22.1	32.5	+9.6%
$BART_{Large}$ (Lewis et al. 2019)	44.0	21.1	40.6	+4.4%	45.4	22.3	37.3	+5.4%	43.5	23.3	33.7	+4.6%
PEGASUS (Zhang et al. 2020)	44.2	21.6	41.3	+3.0%	**46.7**	**24.4**	38.9	+0.6%	41.8	22.9	31.7	+9.0%
GSum (Dou et al. 2021)	45.5	22.3	**42.1**	+0.4%	45.1	21.5	36.6	+7.3%	43.5	23.1	32.8	+5.7%
$BigBird_{Large}$ (Zaheer et al. 2021)	43.8	21.1	40.7	+4.5%	47.1	24.1	38.8	+0.6%	**44.2**	23.8	**34.5**	+2.5%
SimCLS (Liu and Liu 2021)	45.6	21.9	41.0	+1.7%	46.6	**24.2**	**39.1**	+0.7%	43.8	23.3	34.1	+3.9%
SeqCo (Xu et al. 2022)	45.0	21.8	41.8	+1.6%	45.6	22.4	37.0	+5.4%	42.8	22.5	33.2	+6.7%
$GLM_{RoBERTa}$ (Du et al. 2022)	43.8	21.0	40.5	+4.7%	45.5	23.5	37.3	+4.1%	43.3	23.0	33.9	+4.9%
GPT-$3.5_{zero\text{-}shot}$	37.4	13.8	29.1	+37.4%	26.6	6.7	18.8	+112.5%	34.8	12.8	23.5	+47.8%
Our Method												
GPT-3.5 w/ `TriSum` rationale	**46.7**	**23.5**	40.7	−0.5%	34.4	12.6	28.4	+46.8%	**44.6**	**24.5**	30.4	+5.6%
`TriSum-S`	**45.9**	**22.8**	**42.3**	−0.6%	**47.4**	**24.8**	**39.4**	−1.0%	**45.3**	**24.8**	**35.0**	+0.0%
`TriSum-C`	45.5	22.3	41.2	+1.2%	46.5	24.0	38.7	+1.1%	**44.2**	23.7	34.4	+2.7%
`TriSum-J`	**45.7**	**22.7**	**41.9**	—	**47.3**	**24.4**	**39.0**	—	**45.3**	**24.6**	**35.2**	—

Table 2: **Performance comparison of ROUGE Scores across CNN/DailyMail, XSum, and ClinicalTrial datasets**. The labels `TriSum-S`, `TriSum-C`, and `TriSum-J` signify model checkpoints at the end of singular-task, concurrent, and joint learning stages, respectively. For `TriSum-S`, distinct optimal checkpoints, each tailored for a specific task, are used in a pipeline of three Seq2Seq models. The symbol Δ signifies the percentage improvement in the aggregate ROUGE scores achieved by `TriSum-J`. The top-3 results are **highlighted**. Our backbone model $BART_{Large}$ is shaded for reference.

	CNN/DailyMail		XSum		ClinicalTrial	
Model	**BS**	**BAS**	**BS**	**BAS**	**BS**	**BAS**
Baselines						
BERTSumAbs	85.76	-3.81	87.23	-3.66	85.41	-3.79
$T5_{Large}$	87.22	-3.71	90.73	-2.70	87.76	-2.89
$BART_{Large}$	87.98	-3.45	91.62	-2.50	88.30	-2.79
PEGASUS	87.37	-3.64	91.90	-2.44	87.62	-2.80
GSum	87.83	-3.54	91.23	-2.57	88.41	-2.75
$BigBird_{Large}$	88.03	-3.38	**91.97**	**-2.40**	**89.45**	-2.67
SimCLS	88.28	-3.39	90.78	-2.93	87.85	-3.15
SeqCo	87.47	-3.56	91.35	-2.56	88.06	-2.93
$GLM_{RoBERTa}$	87.33	-3.69	91.87	-2.51	88.55	-2.84
GPT-$3.5_{zero\text{-}shot}$	87.70	-3.36	87.67	-2.80	87.08	-3.01
Our Method						
GPT-3.5_{TriSum}	**89.20**	**-3.14**	89.25	-2.58	89.20	**-2.55**
`TriSum-S`	**88.48**	**-3.22**	**91.95**	**-2.38**	**90.05**	**-2.47**
`TriSum-C`	87.21	-3.76	90.88	-2.84	89.40	-2.59
`TriSum-J`	**88.50**	**-3.25**	**92.17**	**-2.33**	**89.97**	**-2.53**

Table 3: **Pre-trained language model-evaluated semantic similarity scores**. "*" indicate the inference with `TriSum`-generated rationale. "BS" and "BAS" are BERTScore and BARTScore, respectively. Top-3 results are **highlighted**.

Performance Analysis

Tables 2 and 3 provide an in-depth look at how our `TriSum` approach performs compared to various baseline models. The results include both ROUGE scores and semantic similarity metrics across different datasets, from general news sources to specialized domain-specific collections. Our analysis reveals several key insights:

Consistent Edge Over Baselines The `TriSum` approach consistently outperforms many state-of-the-art models across different datasets, highlighting its strength and adaptability. Statistically, in terms of overall ROUGE scores, `TriSum-J` outperforms fine-tuned models (excluding GPT-3.5) by 4.5% on CNNDM, 8.5% on XSum, and 7.4% on ClinicalTrial.

Gains Over Backbone We use BART as the backbone model, which is already known for its performance in summarization tasks. The noticeable overall improvement across all datasets (+4.8% ROUGE score and +1.0% BERTScore, and +7.3% BARTScore) when using the `TriSum` approach over BART is significant. This shows the effectiveness of including the LLM-generated rationales as the additional supervision and indicates the potential of our method to be scaled for the enhancement of other summarization models as well. Notably, `TriSum-S` consistently excels in performance. This heightened effectiveness is rooted in its modular design, which encompasses three checkpoints, each optimized for a unique task. Therefore, the improved results may be attributed to its thrice-enlarged parameter set, when compared to `TriSum-C` or `TriSum-J`.

Optimized Rationale for LLM Interestingly, the rationales generated by `TriSum` can significantly improve the performance of GPT-3.5 within the dataset (+40.9% ROUGE Score, +2.0% BERTScore, and +9.9% BARTScore compared to GPT-$3.5_{zero\text{-}shot}$). For example, in our tests with the CNNDM dataset, the LLM, guided by the `TriSum`'s rationale and without any fine-tuning, outperform all the other fine-tuned models in terms of ROUGE-1 score. This suggests that users can use fine-tuned `TriSum` to guide the LLM in creating high-quality summaries.

Effect of Curriculum Learning Figure 4 shows the benefits of curriculum learning on the model's task performance. Two key comparisons are evident: the raw model versus one trained with singular-task learning in the early concurrent learning stage, and the raw model versus one trained through

Article:
(CNN) Four workers died in a massive oil rig fire that raged for hours off the coast of Mexico Wednesday. Mexican state oil company Pemex said 45 workers were injured in the blaze, which began early Wednesday morning. Two of them are in serious condition, the company said. Authorities evacuated about 300 people from the Abkatun Permanente platform after the fire started, Pemex said. At least 10 boats worked to battle the blaze for hours. The fire had been extinguished by Wednesday night, Pemex said in a Twitter post. The company denied rumors that the platform had collapsed and said there was no oil spill as a result of the fire. The state oil company hasn't said what caused the fire on the platform, which is located in the Gulf of Mexico's Campeche Sound. The fire began in the platform's dehydration and pumping area, Pemex said. CNN's Mayra Cuevas contributed to this report.

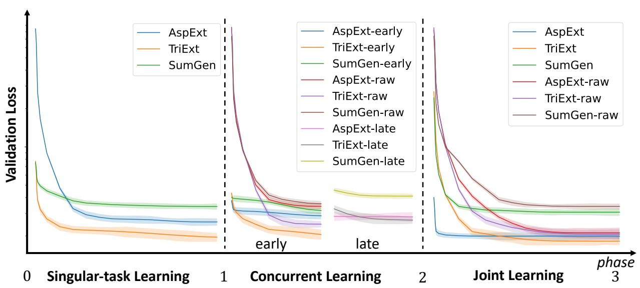
Ground truth summary:
The fire on a platform in the Gulf of Mexico has been extinguished, Pemex says . 45 workers were injured in the blaze, according to the state oil company . Four workers were killed in the oil rig fire, which started early Wednesday .

BART summary:
About 300 people were evacuated from the Abkatun Permanente platform. The fire began in the platform's dehydration and pumping area, Pemex says. The company denies rumors that the platform had collapsed and said there was no oil spill as a result of the fire.

TriSum rationale:
<rationale> <aspects> Massive oil rig fire off the coast of Mexico. Four workers died. 45 workers injured. Two workers in serious condition. 300 people evacuated. Fire extinguished by Wednesday night. No oil spill. **<triples>** [Four workers | were died in | oil rig fire], [45 workers | were injured in | the blaze], [Two workers | are in | serious condition], [300 people | were evacuated from | the platform], [The fire | had been extinguished by | Wednesday night], [The fire | did not result in | oil spill].

TriSum summary:
Four workers were died in an oil rig fire. 45 workers were injured, two of them in a serious condition, state oil company Pemex says. About 300 people were evacuated after the fire started early Wednesday. The fire has been extinguished and it did not result in oil spill, the company says.

Figure 3: **An example of abstractive summarization on CNN/DailyMail dataset.** We compare the summary generated by our `TriSum` approach to the ground-truth summary and the one generated by BART. We use different colors to show the distinct topics in the article and summary.



Singular	Concurrent - Early	Concurrent - Late	Joint	R-1	R-2	R-L
✓	✓	✓	✓	45.7	22.7	41.9
✓	✓	×	✓	45.3 (↓0.4)	22.2 (↓0.5)	41.0 (↓0.9)
✓	×	×	✓	44.4 (↓1.3)	21.3 (↓1.4)	40.4 (↓1.5)
×	×	×	✓	42.3 (↓3.4)	20.5 (↓2.2)	38.4 (↓3.5)

Figure 4: **Validation loss by training steps and ablation study for curriculum learning on CNN/DailyMail.** *AspExt*, *TriExt*, and *SumGen* denote aspect extraction, triple extraction, and summary generation tasks, respectively. *-early* / *-late* denote the early/late stage of concurrent learning. *-raw* denotes training the model from scratch.

the previous two learning stages. The ablation study further reveals a step-wise performance improvement. Notably, when trained solely on joint learning from scratch, the model underperforms the original BART. This emphasizes the indispensable role of foundational tasks, without which BART struggles with the rationale-summary generation task.

Effect of Golden Rationale Selection Figure 5 demonstrates the impact of our golden rationale selection. The performance of the trained model drops significantly when the number of latent topics is either too low (e.g., 50) or high (e.g., 5000). On the other hand, choosing an appropriate number of topics (e.g., 200) leads to improved outcomes. This underscores the importance of the quality of rationales; poor-quality rationales can negatively impact the model, emphasizing the value of our rationale selection strategy.

Case Study on CNNDM Figure 3 compares summaries created from a CNN article discussing an oil rig fire in

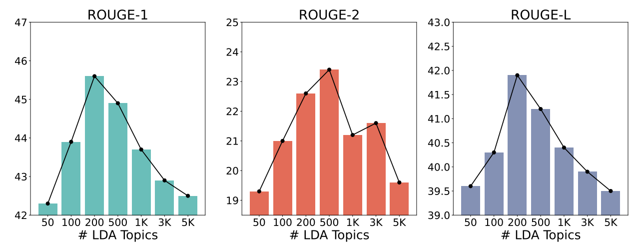


Figure 5: **Performance by different numbers of LDA latent topics specified in golden rationale selection.** We compare the ROUGE scores of the summaries generated by `TriSum-R` on CNN/DailyMail dataset.

Mexico. The ground truth summary adeptly encapsulates the main events, emphasizing the aftermath in terms of fatalities, injuries, and containment. BART's rendition, while detailed about the evacuation and fire's origin, misses out on pivotal information like the death toll and injury scale. On the other hand, `TriSum`'s rationale begins by itemizing the essential aspects of the incident. These aspects present a high-level overview of the events and their aftermath. Following these aspects, the triples zoom into the specifics, elucidating the relations between the entities involved. This technique used by `TriSum` ensures a comprehensive summary and improves clarity. Readers can follow the summary's content back to its main aspects and detailed triples, gaining a deeper understanding of how the summarization process works. This transparency is a key feature of `TriSum`, allowing users to grasp the reasoning behind the summarized content. We provide more examples in the Appendix.

Conclusion

In this study, we introduced `TriSum`, an approach aimed at distilling summarization capabilities from a large language model to a small local model. Extensive experiments verified its superior performance over state-of-the-art models across diverse datasets on the abstractive summarization task. Our work underscores the potential of leveraging large model insights for efficient and nuanced text summarization.

References

Bao, S.; He, H.; Wang, F.; Wu, H.; Wang, H.; Wu, W.; Guo, Z.; Liu, Z.; and Xu, X. 2021. PLATO-2: Towards Building an Open-Domain Chatbot via Curriculum Learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2513–2525. Online: Association for Computational Linguistics.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022.

Brants, T.; Popat, A. C.; Xu, P.; Och, F. J.; and Dean, J. 2007. Large language models in machine translation.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, Y.-C.; Gan, Z.; Cheng, Y.; Liu, J.; and Liu, J. 2019. Distilling knowledge learned in BERT for text generation. *arXiv preprint arXiv:1911.03829*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dou, Z.-Y.; Liu, P.; Hayashi, H.; Jiang, Z.; and Neubig, G. 2021. GSum: A General Framework for Guided Neural Abstractive Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4830–4842. Online: Association for Computational Linguistics.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 320–335. Dublin, Ireland: Association for Computational Linguistics.

Gekhman, Z.; Herzig, J.; Aharoni, R.; Elkind, C.; and Szpektor, I. 2023. TrueTeacher: Learning Factual Consistency Evaluation with Large Language Models. arXiv:2305.11171.

Hacohen, G.; and Weinshall, D. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, 2535–2544. PMLR.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ho, N.; Schmid, L.; and Yun, S.-Y. 2023. Large Language Models Are Reasoning Teachers. arXiv:2212.10071.

Hsieh, C.-Y.; Li, C.-L.; Yeh, C.-K.; Nakhost, H.; Fujii, Y.; Ratner, A.; Krishna, R.; Lee, C.-Y.; and Pfister, T. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. arXiv:2305.02301.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lin, Y.-J.; Tan, D.; Chou, T.-H.; Kao, H.-Y.; and Wang, H.-Y. 2020. Knowledge Distillation on Extractive Summarization. In *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 71–76. IEEE.

Liu, Y. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3730–3740. Hong Kong, China: Association for Computational Linguistics.

Liu, Y.; Fabbri, A. R.; Liu, P.; Radev, D.; and Cohan, A. 2023. On Learning to Summarize with Large Language Models as References. *arXiv preprint arXiv:2305.14239*.

Liu, Y.; and Lapata, M. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Liu, Y.; and Liu, P. 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)*, 1065–1072. Online: Association for Computational Linguistics.

Magister, L. C.; Mallinson, J.; Adamek, J.; Malmi, E.; and Severyn, A. 2023. Teaching Small Language Models to Reason. arXiv:2212.08410.

Nagatsuka, K.; Broni-Bediako, C.; and Atsumi, M. 2021. Pre-training a BERT with Curriculum Learning by Increasing Block-Size of Input Text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 989–996. Held Online: INCOMA Ltd.

Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Narayan, S.; Cohen, S. B.; and Lapata, M. 1808. Don't Give Me the Details, Just the Summary! *Topic-Aware Convolutional Neural Networks for Extreme Summarization. ArXiv, abs*.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Radev, D. R.; Hovy, E.; and McKeown, K. 2002. Introduction to the Special Issue on Summarization. *Computational Linguistics*, 28(4): 399–408.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shridhar, K.; Stolfo, A.; and Sachan, M. 2023. Distilling Reasoning Capabilities into Smaller Language Models. arXiv:2212.00193.

Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. Florence, Italy: Association for Computational Linguistics.

Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; and Lin, J. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017a. Attention is all you need. *Advances in neural information processing systems*, 30.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017b. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, P.; Chan, A.; Ilievski, F.; Chen, M.; and Ren, X. 2022. PINTO: Faithful Language Reasoning Using Prompt-Generated Rationales. *arXiv preprint arXiv:2211.01562*.

Wang, Z.; Gao, C.; Xiao, C.; and Sun, J. 2023. AnyPredict: Foundation Model for Tabular Prediction. *arXiv preprint arXiv:2305.12081*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Xu, B.; Zhang, L.; Mao, Z.; Wang, Q.; Xie, H.; and Zhang, Y. 2020. Curriculum Learning for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6095–6104. Online: Association for Computational Linguistics.

Xu, S.; Zhang, X.; Wu, Y.; and Wei, F. 2022. Sequence Level Contrastive Learning for Text Summarization. arXiv:2109.03481.

Yang, W.; Xie, Y.; Lin, A.; Li, X.; Tan, L.; Xiong, K.; Li, M.; and Lin, J. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 72–77. Minneapolis, Minnesota: Association for Computational Linguistics.

Yang, X.; Li, Y.; Zhang, X.; Chen, H.; and Cheng, W. 2023. Exploring the Limits of ChatGPT for Query or Aspect-based Text Summarization. arXiv:2302.08081.

Yu, W.; Jiang, Z.; Dong, Y.; and Feng, J. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. *arXiv preprint arXiv:2002.04326*.

Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; and Ahmed, A. 2021. Big Bird: Transformers for Longer Sequences. arXiv:2007.14062.

Zaidan, O.; and Eisner, J. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, 31–40.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, 11328–11339. PMLR.

Ethics, Limitations, and Risks

Ethics

Data Privacy and Source: All datasets used in this research, namely CNN/DailyMail, XSum, and ClinicalTrial, are publicly available[4][5][6]. This transparency minimizes ethical concerns related to data sourcing and usage.

Interpretability: The transparency and interpretability of AI models are ethical imperatives in many applications. `TriSum` not only improves summarization performance but also enhances the interpretability of the summarization process, making it more trustworthy.

Limitations

Dependence on LLMs: `TriSum`'s effectiveness is contingent on the quality and capabilities of the LLMs it distills from. If the LLM has biases or inaccuracies, these could potentially be transferred to the local model.

Scope of Rationales: The *aspect-triple* rationales, while enhancing interpretability, might not capture all nuances of the original text. Some information might be lost or oversimplified during the distillation process.

Risks

Overfitting: There's a potential risk that the local model might overfit to the rationales and summaries derived from the LLM, leading to reduced generalization on unseen data.

Misinterpretation: Enhanced interpretability can sometimes lead users to place undue trust in the model's outputs. Users should be cautious and consider the model's outputs as one of many tools in decision-making processes.

Ethical Misuse: Like all summarization tools, there's a risk that users might misuse `TriSum` to misrepresent complex information, leading to misinformation.

Templates Used for Prompting LLM

In this section, we showcase the templates we used for prompting the large language model for different purposes.

Figure 6 shows the template we use for **Step 1** (LLM Rationale Probing). It instructs the LLM to (1) generate essential aspects of the document with respect to the ground-truth summary; (2) extract triples from the document that elaborate on these key aspects; (3) generate a summary referring to both the retrieved triples and the ground-truth summary. The template then instructs the LLM to generate in a specific format, to reduce the randomness of the LLM's output. The document and the ground-truth summary are input to the placeholders to finalize the prompting request.

Figures 7 and 8 show the templates we use for testing the LLM's summarization ability in a zero-shot setting and with `TriSum`-generated rationales, respectively.

[4]https://github.com/abisee/cnn-dailymail

[5]https://github.com/EdinburghNLP/XSum

[6]https://clinicaltrials.gov/

```
Given a document and its ground-truth summary, do
the following tasks:
(1) According to the ground-truth summary, extract
essential aspects of the document.
(2) For each essential aspect, retrieve detailed
triples in the format [ENTITY1 | RELATION |
ENTITY2] used to compose the ground-truth summary.
(3) With the retrieved triples, compose a summary.

The essential aspects, triples, and composed
summary should be in the same response, separated
by a new line.

All triples [ENTITY1 | RELATION | ENTITY2] should
be in length 3 (separated by "|").

Example:
================Example=================
Prompt:
[Document]: [document]
[Ground-truth Summary]: [ground-truth summary]

Update:
Essential Aspects:
[aspects]

Triples:
- [ENTITY1_1 | RELATION_1 | ENTITY1_2]
- [ENTITY2_1 | RELATION_2 | ENTITY2_2]
- [ENTITY3_1 | RELATION_3 | ENTITY3_2]
- ...

Generated Summary:
[summary]
========================================

Prompt:
[Document]: {doc}
[Ground-truth Summary]: {gt_summary}

Update:
```

Figure 6: Template used for prompting rationale and summary from LLM

```
Given a document, summarize the document in
one sentence:                          for XSum

Given a document, summarize the document in
three sentence:   for CNNDM & ClinicalTrial

Document: {doc}

Summary:
```

Figure 7: Template used for prompting summary from LLM in zero-shot setting.

Dataset Description

CNN/DailyMail The CNN/DailyMail dataset is one of the most popular datasets for extractive and abstractive summarization tasks. Originating from online news stories, the dataset comprises articles from CNN and DailyMail websites. The overview of this dataset is described as follows:

```
Given a document and the rationale for
summarization, summarize the document in one
sentence.

The rationale contains (1) the essential
aspects of the document; (2) triples of
entities and relations in the document that
compose the summary, in the format of
[ENTITY1 | RELATION | ENTITY2].
We use the prefixes <aspects> and <triples> to
indicate the start of the rationale for
aspects and triples, respectively.

The generated summary should not longer than
one sentence.                          for XSum

The generated summary should not longer than
three sentence.   for CNNDM & ClinicalTrial

Example:
================Example=================
Prompt:
[Document]: [document]
[Rationale]: <aspects> + [aspects] +
<triples> + [triples]

Update:
Summary:
[summary]

========================================

Prompt:
[Document]: {doc}
[Rationale]: {aspects} {triples}

Update:
```

Figure 8: Template used for prompting summary from LLM given TriSum-generated rationale (GPT-3.5$_{\text{TriSum}}$).

- **Size**: It contains 287,113 training examples, 13,368 validation examples, and 11,490 test examples.
- **Content**: Each example in the dataset consists of a news article and several accompanying highlight points, which, when combined, form a coherent summary of the main article.
- **Nature of Summaries**: The highlights, crafted to engage a reader's attention, effectively form summaries. Typically, a summary consists of 2 to 3 sentences. They can be approached either extractively or abstractively by summarization models.
- **Usage**: Due to its substantial size and real-world data, CNN/DailyMail has been a benchmark for several state-of-the-art summarization models, enabling researchers to compare performances and strategies across diverse methods.

XSum XSum (Extreme Summarization) dataset provides a more challenging scenario for abstractive summarization. The overview of this dataset is described as follows:

- **Size**: It contains 204,045 training examples, 11,332 validation examples, and 11,334 test examples, which are the articles collected from the BBC (British Broadcasting Corporation).
- **Content**: Unlike CNN/DailyMail where summaries are constructed from highlights, each article in the XSum dataset is paired with a single-sentence summary, often written in a style that is not present in the article body.
- **Nature of Summaries**: The summaries in XSum are more abstractive in nature and are not simply extractive snippets from the articles. This demands models to truly understand the content and generate a unique summarizing sentence, making it a challenging dataset for abstractive summarization.
- **Usage**: XSum's distinctive nature has made it a preferred choice for researchers focusing on advanced abstractive methods in summarization. Its summaries, being creatively crafted and not directly extracted from the text, test the genuine abstracting capabilities of models.

ClinicalTrial We collected the clinical trial protocol documents from clinicaltrials.gov where there are over 400K registered clinical trials across the world. The overview of this dataset is described as follows:

- **Size**: We downloaded the static copy of the whole clinical trial database which is with around 460K clinical trial documents. 203,860 were selected out of all based on the standard (a) they are interventional clinical trials, (b) missing or duplicate titles, (c) missing the brief summary section. To fit the context window of used language models, we further exclude documents that have more than 1024 tokens or the target summaries are with more than 256 tokens.
- **Content**: The clinical trial document describes the proposal for testing the effectiveness and the safety of a new treatment, e.g., a drug. The researchers need to list all the main elements required for FDA regulation, such as the title, proposed treatment, target condition, primary outcome measurements, eligibility criteria, etc.
- **Nature of Summaries**: An effective summary of clinical trials need to deliver the main message about the motivation of the study as well as the route planning to reach the target. To make a good summary of clinical trials, the model needs a comprehensive view of the whole documents and maintain the key information.
- **Usage**: We will use the "brief summary" section written by human experts provided in the raw clinical trial documents as the target for all models.

Interpretability of `TriSum`

Interpretability is paramount in understanding and trusting AI systems, especially in tasks like abstractive summarization where the derivation of conclusions isn't always overtly apparent. The workflow of TriSum, illustrated in Figure 9, is designed with this transparency in mind.

Starting with a given document, TriSum identifies its essential aspects. This step offers a clear insight into what the model perceives as the primary themes or topics within

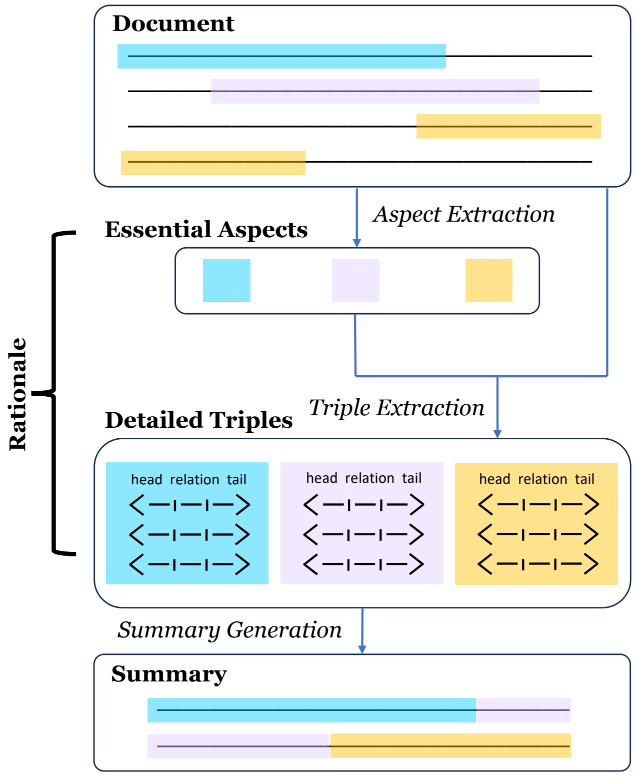


Figure 9: **Abstractive summarization with TriSum**. Different colors indicate different essential aspects covered by the document. We showcase how an *aspect-triple* rationale is extracted and contribute to the final summary generation.

the document. Subsequently, using these aspects as anchors, `TriSum` revisits the document to meticulously extract triples, structured as ⟨head | relation | tail⟩, for each aspect. These triples provide a structured, detailed representation, offering granular insights into the model's understanding of the relationships and entities in the text. Finally, `TriSum` fuses these extracted aspects and triples to produce a summary. By correlating the final summary with the previously identified aspects and triples, users can trace back the origins of particular summary fragments, gaining a clear understanding of how `TriSum` processes and abstracts information.

This step-by-step elucidation of the summarization process significantly enhances the model's transparency, making its decision-making rationale more discernible and hence fostering trust among its users.

Hyperparameter Tuning

Table 4 shows our comprehensive hyperparameter study to select the optimal values for `TriSum`.

Case Studies

In addition to Figure 3, Figure 10 shows other two examples comparing our `TriSum`'s performance with our backbone model BART on XSum and ClinicalTrial datasets. We can draw the following findings:

Hyperparameter	Values
Golden Rationale Selection	
ϕ_α	{0.2, 0.4, **0.6**, 0.8, 1.0, 1.2}
ϕ_β	{0.4, 0.6, 0.8, 1.0, **1.3**, 1.5, 2.0}
λ_{cs}	{0.5, 1.0, **1.5**, 2.0 }
LDA latent topics	{50, 100, **200**, **300**, **500**, 1000, 3000, 5000}
Rationale Learning	
(λ_R, λ_S)	{(1.0, 1.0), **(0.8, 1.2)**, (0.5, 1.5), (0.3, 1.7)}

Table 4: **Hyperparameters of TriSum we tuned.** We highlightthe optimal ones based on our experiments in **bold**.

Case Study on XSum

In the given example, we juxtapose the performance of our approach, `TriSum`, with BART, our backbone model. Upon scrutinizing the sourced article detailing a research study on job discrimination against women with Turkish names and those wearing Islamic headscarves in Germany, we discern distinct nuances in the summaries rendered by both methods.

BART's summary encapsulates a broad understanding, highlighting that women wearing headscarves in Germany are at a disadvantage during job applications. While it successfully conveys a salient point, it omits the specific discrimination against women with Turkish names.

`TriSum`, on the other hand, demonstrates its prowess through a more holistic, nuanced, and detailed summary. It distinctly notes both aspects of the discrimination: one against women with Turkish names and the other against those donning an Islamic headscarf. `TriSum`'s rationale section further accentuates its strength by explicitly presenting the core aspects and triples that delineate the focus points of the summary. This methodical extraction and representation ensure that no vital information is sidestepped.

Moreover, `TriSum`'s summary doesn't merely report the findings but emphasizes the intensification of discrimination when both factors - a Turkish name and an Islamic headscarf - are combined. Such a layered insight is invaluable, especially in sensitive subjects such as discrimination, where capturing the entire scope of the issue is crucial.

In essence, while BART gives a generalized overview, `TriSum` offers a richer, more comprehensive narrative that mirrors the depth and breadth of the original article, underscoring the strength and precision of our approach.

Case Study on ClinicalTrial

In this case study centered around adult tonsillectomies, it is evident that the BART primarily grasped the core goal of the study but missed out on essential details, particularly the varied fluid intake groups and post-operative data recording. Meanwhile, the ground truth summary offers a comprehensive view, but it remains relatively generalized.

The strength of our approach, the *aspect-triple* rationaled summarization (`TriSum`), is significantly highlighted when we delve into the details and the rationale-driven structure it adheres to. `TriSum` operates by identifying essential aspects of the text, followed by extracting and constructing

Article:
A university researcher sent 1,500 identical CVs to German firms - except that some bore the name Meryem Ozturk and others the name Sandra Bauer. In 18.8% of cases Sandra Bauer was invited for interview, whereas the figure for Meryem was just 13.5%. When the photo of Meryem showed her in a headscarf only 4.2% invited her. The study was published by the Institute for the Study of Labour, in Bonn. The researcher was Doris Weichselbaumer from the University of Linz, in Austria. The findings are especially significant in light of Germany's current efforts to integrate record numbers of Muslim migrants, many of them refugees from the conflicts in Syria, Iraq and Afghanistan. More than a million asylum seekers reached Germany last year, and there has been a backlash from nationalist groups, especially Pegida and Alternative for Germany (AfD). With an estimated three million people of Turkish origin, Germany is home to the largest Turkish diaspora. In her study, Ms Weichselbaumer said that "a very modern binding of the headscarf was chosen to signal that the applicant was a young, modern woman who could easily fit into a secular environment". It is normal in Germany to attach a photo to a job application, she pointed out. The result of the experiment "implies that the candidate with the headscarf had to send 4.5 times as many applications as an identical applicant with a German name and no headscarf to receive the same number of callbacks for interview", she said. ...

Ground truth summary:
An experiment in Germany has found evidence of job discrimination against women with Turkish names - and even more if they wear an Islamic headscarf.

BART summary:
Women who apply for jobs in Germany wearing headscarves are much less likely to be invited for interview, a study suggests.

TriSum rationale:
<rationale> <aspects> Job discrimination against women with Turkish names in Germany. Job discrimination against women wearing an Islamic headscarf in Germany. **<triples>** [women with Turkish names | face discrimination in | job applications in Germany], [women wearing an Islamic headscarf | face even more discrimination in | job applications in Germany].

TriSum summary:
Women with Turkish names in Germany face discrimination in job applications, and this discrimination is intensified for women wearing an Islamic headscarf.

Article:
Common indications for adult tonsillectomy include tonsillar hypertrophy, obstructive sleep apnea, and chronic tonsillitis. Although there are many studies looking into how best to prevent and treat post-operative pain and nausea, there are very few that give recommendations to the patients pre-operatively. This study's aim is to evaluate if pre-operative oral hydration has an impact in reducing post-operative pain, nausea and vomiting after adult tonsillectomies. If pre-operative hydration can reduce the pain associated with the procedure, then this could also be an adjunct to help reduce the amount of narcotic pain medication that is required.~The goal of this study is to evaluate if the amount of pre-operative oral hydration can reduce the amount of pain and nausea/vomiting in patients undergoing adult tonsillectomies. This study will be measuring post-operative pain and nausea scores at several points after the adult tonsillectomy procedure for participants that have consumed different amounts of clear fluid in the 24 hours leading up to their nothing by mouth (NPO) status. Participants will be randomized to one of three categories of preoperative fluid: 0.5 - 1.5 Liters (L), 1.5 - 3 L, and 3 - 4.5 L, which will correlate to the low, medium and high fluid intake groups, respectively. Participants will be given a 1000 mL measuring water bottle that they can use to accurately record their fluid consumption in the 24 hours prior to their pre-operative NPO status. The fluid consumption that they record will include all liquids (i.e., water, flavored water, juice, coffee, tea, soda, milk, alcohol). They will be asked to keep a record of the volume of caffeinated beverage versus un-caffeinated beverages.~After surgery the participants will be given a form to record their pain and nausea scores. Participants will record their pain and nausea by using two separate 10-point scales at 7 different points after the procedure. These scores will be measured immediately after surgery, and at 4 hours, 8 hours, 24 hours, 72 hours, 5 days and 7 days postoperative. This form, along with a count of opioid medication used, will be retuned to the physician at the first post-operative follow-up visit.

Ground truth summary:
Tonsillectomy is commonly described as one of the most painful procedures that an adult can undergo. This study's aim is to evaluate if pre-operative oral hydration has an impact in reducing post-operative pain, nausea and vomiting, and opioid medication use after adult tonsillectomies. Participants will be randomized to a low, medium and high fluid intake groups and will record fluid intake on the day before surgery. After surgery, participants will record their pain and nausea at timepoints up to 7 days post-operative. Medication use will be recorded up to the first post-operative follow-up visit with their physician.

BART summary:
The goal of this study is to evaluate if the amount of pre-operative oral hydration can reduce the amount of pain and nausea/vomiting in patients undergoing adult tonsillectomies.

TriSum rationale:
<rationale> <aspects> Purpose of the study. Concern about tonsillectomy pain. Role of pre-operative oral hydration. Randomization of participants into fluid intake groups. Recording of fluid intake prior to surgery. Post-operative recording of pain and nausea. Recording of opioid medication use. **<triples>** [Study | aims to| evaluate impact of pre-operative hydration on post-operative outcomes], [Tonsillectomy | is described as| painful procedure for adults], [Participants | will be randomized to | fluid intake groups: low, medium, high], [Participants | will record | fluid intake before surgery], [Participants | will record | pain and nausea post-operatively], [Participants | will record | opioid medication use until first follow-up].

TriSum summary:
This study focuses on the possible influence of pre-operative oral hydration on post-operative pain, nausea, vomiting, and the use of opioid medication. Individuals taking part in the study will be divided into low, medium, and high fluid intake categories. On the day before their procedure, they will note down their fluid consumption. After undergoing the tonsillectomy, these participants will monitor and record their pain, nausea for up to seven days, and also their opioid medication intake until their initial post-operative check-up with the doctor.

Figure 10: **Examples of abstractive summarization on XSum (above) and ClinicalTrial (below) datasets.** We compare the summary generated by our `TriSum` approach to the ground-truth summary and the one generated by BART. We use different colors to show the distinct topics in the article and summary.

triples that map the relationships in the content.

- **Aspect-Driven Understanding**: `TriSum`'s rationale points out the key aspects such as the purpose of the study, concerns related to tonsillectomy pain, the role of pre-operative hydration, among others. By capturing these aspects, the model sets the stage for a summary that does not miss out on the diverse elements of the original text.
- **Triple-Based Detail Extraction**: The aspect-driven approach is further enriched by the triples `TriSum` generates. These triples, such as [Participants | will record | pain and nausea post-operatively], ensure that the summary remains faithful to the article by capturing nuanced relationships. It does not just reiterate what the study does, but also how it goes about it, ensuring the reader understands the methodology.
- **Precision and Brevity**: The `TriSum` summary captures all the key points—right from the study's focus, the categorization of participants, to the post-operative documentation—without becoming verbose. It offers a condensed yet comprehensive view of the article, ensuring that readers can quickly grasp the core concepts without getting overwhelmed.

Additional Evaluation on ClinicalTrial-Base

In addition to the ClinicalTrial (Large) dataset, we also constructed a simpler version - ClinicalTrial-Base where we consider the article-summary pairs included in this dataset to be those with a BARTScore higher than -2.0. The statistics for this dataset are in Table 5 shown as follows.

Dataset	**# Samples** Train	Valid	Test	**# Words** Doc.	Sum.
ClinicalTrial-Base	62,012	7,752	7,752	277.7	76.1

Table 5: **Statistics of ClinicalTrial-Base.**

Our evaluation results are shown in Table 6 below.

ClinicalTrial-Base						
Model	**R-1**	**R-2**	**R-L**	Δ	**BS**	**BAS**
Baselines						
T5_{Large}	**53.9**	**41.7**	**47.2**	-2.0%	90.49	-1.91
$\text{BART}_{\text{Large}}$	51.8	38.6	43.6	$+4.4\%$	89.61	-1.99
PEGASUS	51.8	40.7	44.8	$+1.9\%$	90.16	-1.61
$\text{GPT-3.5}_{\text{zero-shot}}$	45.4	23.8	32.5	$+37.6\%$	89.00	-2.44
Our Method						
$\text{GPT-3.5}_{\text{TriSum}}$	**54.1**	37.6	42.2	$+4.5\%$	90.84	-1.52
`TriSum-S`	**53.6**	**42.2**	**46.6**	-1.8%	90.67	-1.66
`TriSum-C`	50.3	37.2	42.8	$+7.4\%$	89.25	-2.14
`TriSum-J`	52.9	**41.8**	**45.2**	—	90.81	-1.64

Table 6: **Performance comparison of ROUGE Scores and semantic similarity scores on ClinicalTrial-Base Dataset**.The top-3 results are **highlighted**. Our backbone model, $\text{BART}_{\text{Large}}$, is shadowed for reference.