EADSum: Element-Aware Distillation for Enhancing Low-Resource Abstractive Summarization

Jia-Liang Lu, Bi-Cheng Yan, Yi-Cheng Wang, Tien-Hong Lo, Hsin-Wei Wang, Li-Ting Pai, Berlin Chen
Department of Computer Science and Information Engineering, National Taiwan Normal University, Taiwan
E-mail: {jialianglu, bicheng, yichengwang, teinhonglo, hsinweiwang, 61147095s, berlin}@ntnu.edu.tw

***Abstract*—Abstractive summarization aims to generate concise summaries from input documents, with notable advancements driven by the advent of large language models (LLMs). However, the substantial computational demands and the sheer size of LLMs pose significant scalability challenges for the corresponding deployment in real-world applications. Existing methods typically adopt large amounts of training data to train smaller, more compact models to achieve performance on par with LLMs, either through fine-tuning with human-annotated labels or distilling rationales from LLM-generated labels. As an appealing alternative for low-resource abstractive text summarization, we propose EADSum (Element-Aware Distillation for Summarization), a novel training framework which aims to generate fine-grained summaries correlated to the human writing mindset while alleviating the heavy requirement of supervised training data. The proposed EADSum approach first guides LLMs to generate element-aware rationales from the input document, drawing attention to crucial elements such as entities, dates, events, and the results of event. These generated rationales then serve as additional supervision for the subsequent training of compact models within a multi-task learning framework. A series of experiments conducted on the CNN/DailyMail benchmark dataset demonstrate the feasibility and effectiveness of our approach.**[1]

I. Introduction

Text summarization is a crucial task in natural language processing which aims to automatically condense lengthy input documents into shorter, coherent, and informative summaries [1][2]. Research endeavors in text summarization generally fall into two modeling paradigms: extractive and abstractive summarization. The former extracts the most representative sentences from the input documents and forms a summary by aggregating these sentences. The latter, in contrast, concentrates more on generating concise, fluent summaries that capture the core ideas of the input documents. Among these paradigms, abstractive summarization has garnered significant attention due to its ability to produce high-quality summaries that are both verbally innovative and capable of integrating external knowledge [3][4].

Early studies on abstractive summarization focused on both sequence-to-sequence neural modeling [5][6][7] and transfer learning techniques coupled with pre-trained large language models [8][9][10]. These approaches typically train or fine-tune pre-trained language models with large amounts of training data to achieve superior performance in terms of lexical overlap (i.e., ROUGE evaluations [11]) compared to golden summaries [12][13][14]. In recent years, the advent of Large Language Models (LLMs), such as GPT-3 [15] and its variants [16][17], has fueled the advancements in language understanding and generation tasks, leading to remarkable breakthroughs and sparking further research in abstractive text summarization. LLMs demonstrated reasoning abilities through chain-of-thought prompting [18]. This approach effectively captures both the topic structures and the core ideas from source texts, suggesting LLMs' promising potential for generating text summaries. As a follow-up study, for example, Wang et al. leveraged chain-of-thought techniques to guide LLMs in focusing on the news elements of input documents. These derived element-aware rationales then act as additional information to prompt the LLMs to generate summaries in a step-by-step manner [18].

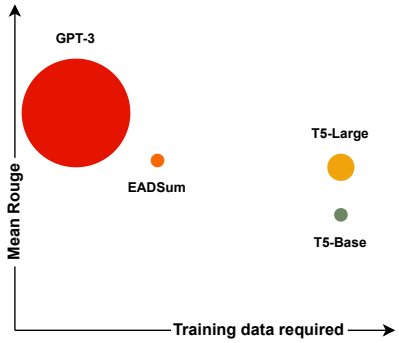

Fig. 1. While GPT-3 offers strong zero-shot performance, it is challenging to implement in practice due to its size and computational requirements. The size of the circles represents the model parameters. Conversely, traditional methods of training smaller task-specific models require large amounts of training data. We propose EADSum, a new paradigm that extracts element-aware rationales from LLMs as additional knowledge to train compact models. This approach reduces both the deployed model size and the amount of data required for training.

Despite the impressive performance of abstractive text summarization achieved by LLMs, deploying these models in real-world applications is confronted with at least two intrinsic challenges. On one hand, the substantial computational demands and large model size of LLMs greatly limit their utility in resource-constrained environments. On the other hand, training a smaller, compact model to achieve performance comparable to LLMs is a data-hungry process that

[1]Source code is available at: https://github.com/VitasLu/EADSum

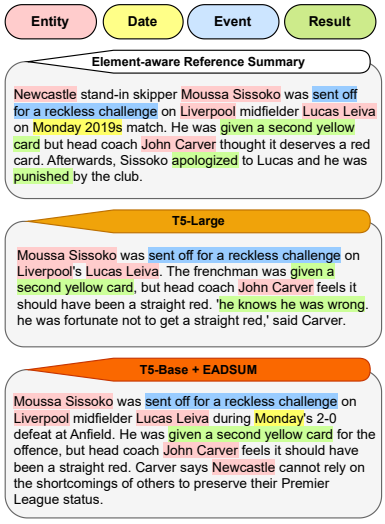

Fig. 2. Comparison of summarization outputs from different models and the Element-aware Reference Summary.

heavily relies on enormous amounts of training data, either by fine-tuning the model with human-labeled data or distilling knowledge from a given LLM. In light of this, we propose EADSum (Element-Aware Distillation for Summarization), a novel training framework for low-resource abstractive text summarization managed to generate high-quality summaries closely aligned with human assessment criteria, such as coherence, consistency, and relevance [19]. Specifically, our method first employs chain-of-thought reasoning [20][21] to guide LLMs in extracting element-aware rationales from the input document, including detailed information of entities, dates, events, and the results of event, as shown in Fig. 2. These extracted summary rationales then act as additional supervision to train compact models through a multi-task training setup with both summary and rationale generation tasks. A series of experiments conducted on the CNN/DailyMail benchmark dataset demonstrate the feasibility and effectiveness of our approach. Experimental results show that our method surpasses the performance of fine-tuning pre-trained language models and is competitive with the zero-shot summarization capabilities of GPT-3 on element-aware reference summary. It is worth noting that the proposed method provides a practical solution tailered for deploying advanced summarization techniques in resource-constrained environments by distilling the capabilities of LLMs into a compact model.

II. METHODOLOGY

We propose a novel method called EADSum (Element-Aware Distillation Summarization) that leverages the capability of Large Language Models (LLMs) to identify the core elements of a document, enabling the training of compact models in a data-efficient manner. Our comprehensive framework, as illustrated in Fig. 3, comprises two main steps designed for low-resource scenarios: (A) rationales extraction, and (B)

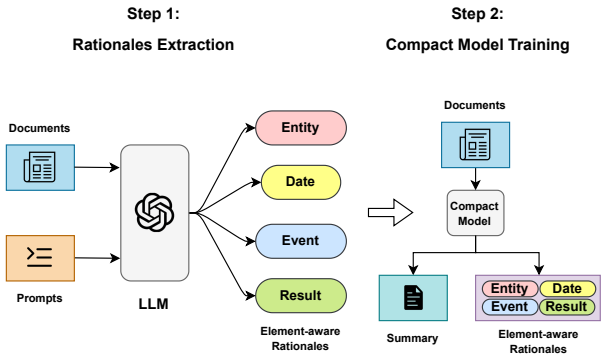

Fig. 3. A conceptual demonstration of our two-step framework, EADSum, which endows compact models with LLM's text summarization capabilities.

compact model training. In the first step, we extract element-aware rationales from LLMs. In the second step, we train a compact model through a multi-task training setup.

A. Rationales Extraction

The extraction of element-aware rationales from Large Language Models (LLMs) draws inspiration from Chain-of-Thought prompting research [20]. This line of inquiry has demonstrated that LLMs are capable of providing step-by-step explanations, or rationales, for their outputs. This capability has been successfully applied to various tasks, including summarization [18]. Our Element-Aware Distillation Summarization (EADSum) method further leverages this approach, guiding LLMs to identify and articulate key elements such as entities, dates, events, and the results of event.

To extract element-aware rationales from Large Language Models (LLMs), we have designed a series of guiding questions. These questions prompt the LLM to identify and extract four core elements—entities, dates, events, and the results of event—from the input document. For each element i, we formulate a guiding question q_i. For instance, the prompt for the entities element $q_{entities}$ can be "What are the important entities in the following article?", and so on.

We then combine these guiding questions for the four key elements into a single prompt set $Q = [q_{entities}, q_{dates}, q_{events}, q_{results}]$. Subsequently, we input both the source document S and the question prompt set Q to the LLM, represented as $[S; Q]$. This process generates a rationale R, which serves as crucial role for the next stages of our method.

B. Compact Model Training

We begin by outlining the existing framework for training task-specific models. Building upon this foundation, we then expand this framework to incorporate extracted rationales from step 1 into the training process. Our dataset is formally represented as $\mathcal{D} = (x_i, y_i)_{i=1}^{n}$, where n denotes the size of the dataset, x_i represents the source document, and y_i represents

its corresponding reference summary. In our experiments, we constrain both x_i and y_i to be natural language.

Standard fine-tuning and task-specific knowledge distillation. The predominant approach for developing task-specific models involves fine-tuning a pre-trained model using supervised data [22]. When human-annotated labels are unavailable, researchers often employ task-specific distillation techniques [23]. In this method, Large Language Models (LLMs) serve as teachers, generating pseudo-noisy training labels [24][25].

In the context of text summarization, we train the compact model f to minimize the summary prediction loss in both fine-tuning and distillation scenarios, defined as:

$$\mathcal{L}_{summary} = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i), \tag{1}$$

where ℓ represents the cross-entropy loss between the generated and reference summaries.

Multi-task learning with element-aware rationales. To create a more explicit connection between the source document x_i and the reference summary y_i, we use rationales r_i extracted from step 1 as additional supervision.

A key innovation in our approach is how we utilize these rationales. Instead of using them as additional input to the model, we frame the learning process as a multi-task problem. This decision is driven by two important factors. First, our compact model has fewer parameters compared to large language models (LLMs), making it challenging to effectively generate rationales using chain-of-thought reasoning [20]. Second, considering deployment challenges, we reduce the need for LLM involvement during inference, making our method more efficient and scalable.

Specifically, we train the model $f(x_i) \rightarrow (\hat{y_i}, \hat{r_i})$ to generate not only the text summaries $\hat{y_i}$ but also the corresponding element-aware rationales $\hat{r_i}$ given the text input:

$$\mathcal{L} = \alpha\mathcal{L}_{summary} + (1-\alpha)\mathcal{L}_{rationale}, \tag{2}$$

where α is a balancing parameter, $\mathcal{L}_{summary}$ is the summary prediction loss in Eq. 1, and $\mathcal{L}_{rationale}$ is the rationale generation loss:

$$\mathcal{L}_{rationale} = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), r_i). \tag{3}$$

This rationale generation loss enables the model to learn to generate the intermediate reasoning steps for the summary, potentially guiding the model to better generate the resultant summary and this is our proposed EADSum distillation framework.

III. EXPERIMENTS

A. Experimental Settings

Evaluation Dataset. Our experiments are conducted on the CNN/DailyMail dataset, a widely-used benchmark in news summarization tasks. This large-scale corpus comprises 287,113 training, 13,368 validation, and 11,490 test examples, each consisting of a news article paired with a multi-sentence summary. The articles are sourced from CNN and Daily Mail websites, while the summaries are created by concatenating the bullet points accompanying each article.

To investigate the efficacy of our approach in low-resource scenarios, we deliberately constrain our experimental setup. Instead of utilizing the full dataset, we employ only 70,835 examples (approximately 24.7% of the full training set) for model training, and 7,871 examples for validation. Furthermore, we evaluate our model on a carefully selected subset of 200 examples from the test set. This resource-constrained configuration allows us to simulate real-world situations where annotated data may be limited, assess our model's performance under data scarcity conditions, and explore the potential of our approach for domains or languages with restricted resources.

Compared Baselines. To evaluate the effectiveness of our proposed approach, we compare it against several strong baseline models:

- PEGASUS-Large[2]: A state-of-the-art abstractive summarization model pre-trained on massive text corpora using gap-sentence generation objectives.
- T5-base[3] and T5-Large[4]: Two variants of the Text-to-Text Transfer Transformer (T5) model, which have demonstrated impressive performance across various natural language processing tasks, including summarization.
- BART-base[5] and BART-Large[6]: Two versions of the BART (Bidirectional and Auto-Regressive Transformers) model, specifically designed for sequence-to-sequence tasks like text summarization.
- GPT-3[7]: The largest language model in our baseline set, known for its remarkable zero-shot learning capabilities across a wide range of language tasks.

For the pre-trained language models (PLMs), we utilize the officially models fine-tuned on CNN/DailyMail dataset (BART-Base[8], BART-Large[9], PEGASUS-Large[10], T5-Base[11], and T5-Large[12]) released on the Huggingface platform for generation tasks. In the case of zero-shot prompting for GPT-3, we use the standard prompt p = "Summarize the above article:" for the CNN/DailyMail dataset by approach of [26].

Implement Detail. We employ GPT-3.5 as our Large Language Model (LLM) to extract element-aware rationales from the input articles. To elicit these rationales, we used the following series of prompts: 'What are the important entities in the following article?', 'What are the important dates?', 'What events are happening?', and 'What is the result of

[2]https://huggingface.co/google/pegasus-large
[3]https://huggingface.co/google-t5/t5-base
[4]https://huggingface.co/google-t5/t5-large
[5]https://huggingface.co/facebook/bart-base
[6]https://huggingface.co/facebook/bart-large
[7]https://openai.com/api/
[8]https://huggingface.co/ainize/bart-base-cnn
[9]https://huggingface.co/facebook/bart-large-cnn
[10]https://huggingface.co/google/pegasus-cnn_dailymail
[11]https://huggingface.co/flax-community/t5-base-cnn-dm
[12]https://huggingface.co/ksstevenn/T5-large-cnndm

TABLE I
PERFORMANCE COMPARISON OF LARGER MODELS (BART-LARGE, PEGASUS-LARGE, T5-LARGE, GPT-3) AND SMALLER MODELS (BART-BASE, T5-BASE) IN TERMS OF ROUGE-1, ROUGE-2, ROUGE-L, MEAN-ROUGE, AND BERTSCORE ON ORIGINAL REFERENCE SUMMARIES

Model	Training Size	# Params	Original Reference Summary				
			Rouge-1	Rouge-2	Rouge-L	Mean-Rouge	BERTScore
BART-Large	-	406M	39.11	16.56	36.43	30.70	0.8708
BART-Large	100%	406M	43.44	20.51	40.52	34.82	0.8828
PEGASUS-Large	-	568M	36.96	15.10	33.28	28.45	0.8637
PEGASUS-Large	100%	568M	40.34	18.65	38.26	32.42	0.8793
T5-Large	-	770M	41.24	18.66	38.57	32.82	0.8795
T5-Large	100%	770M	41.84	19.28	39.34	33.49	0.8829
GPT-3	Zero-Shot	175B	31.44	10.86	29.25	23.85	0.8655
BART-Base	-	139M	40.80	17.82	38.14	32.25	0.8728
BART-Base	100%	139M	41.86	19.75	39.00	33.53	0.8801
T5-Base	100%	220M	41.06	19.08	38.44	32.86	0.8805
T5-Base + EADSum	24.7%	220M	41.29	18.63	38.39	32.77	0.8780

TABLE II
PERFORMANCE COMPARISON OF LARGER MODELS (BART-LARGE, PEGASUS-LARGE, T5-LARGE, GPT-3) AND SMALLER MODELS (BART-BASE, T5-BASE) IN TERMS OF ROUGE-1, ROUGE-2, ROUGE-L, MEAN-ROUGE, AND BERTSCORE ON ELEMENT-AWARE REFERENCE SUMMARIES

Model	Training Size	# Params	Element-aware Reference Summary				
			Rouge-1	Rouge-2	Rouge-L	Mean-Rouge	BERTScore
BART-Large	-	406M	31.78	12.81	28.47	24.35	0.8737
BART-Large	100%	406M	32.57	12.41	29.21	24.73	0.8804
PEGASUS-Large	-	568M	28.87	9.16	24.34	20.79	0.8645
PEGASUS-Large	100%	568M	31.53	11.82	28.97	24.11	0.8786
T5-Large	-	770M	30.68	10.54	27.47	22.90	0.8762
T5-Large	100%	770M	32.55	12.22	29.21	24.66	0.8813
GPT-3	Zero-Shot	175B	37.52	14.59	33.71	28.61	0.8922
BART-Base	-	139M	32.97	13.52	29.69	25.39	0.8744
BART-Base	100%	139M	33.18	12.44	29.74	25.12	0.8809
T5-Base	100%	220M	31.41	11.84	27.99	23.75	0.8784
T5-Base + EADSum	24.7%	220M	33.73	13.57	30.19	25.83	0.8805

these events?'. This approach helps us obtain comprehensive information about the article's key elements.

For the training stage, we utilize T5-Base, a model with 220 million parameters, as our compact model. We implement a multi-task setup by prepending task prefixes to the input examples. Specifically, we use [summary] to indicate summarization tasks and [rationale] for rationale generation tasks. This approach, inspired by [9], allows us to train the compact model to produce summaries when the [summary] prefix is provided and to generate rationales when the [rationale] prefix is used. We set the balancing parameter α to 0.5 for this multi-task training.

Evaluation Metrics. We employ automatic metrics to evaluate the quality of the generated summaries. Specifically, we utilize the following evaluation methods:

- ROUGE Score [11]: We report the F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L. These metrics measure the overlap between the generated summaries and the reference summaries in terms of unigrams, bigrams, and the longest common subsequence, respectively.
- Mean-ROUGE [11]: This metric is calculated by taking the average of ROUGE-1, ROUGE-2, and ROUGE-L scores. It provides a single, comprehensive measure of the overall quality of the generated summaries, balancing the evaluation across different levels of text granularity.
- BERTScore [27]: This metric evaluates the semantic similarity between the generated summaries and the reference summaries. It operates by computing the cosine similarity between the contextualized embeddings of tokens in the summaries, providing a more nuanced assessment of semantic equivalence.

B. Main Results

In our study, we evaluate the effectiveness of our proposed EADSum training method using two distinct test sets, each designed to assess different aspects of summarization quality. The first test set utilizes the original reference summaries from the CNN/DailyMail dataset, a widely recognized benchmark in text summarization. These summaries provide a general assessment of the models' ability to capture the main points of news articles. The second test set employs element-aware reference summaries [18], meticulously crafted by three news experts following a comprehensive writing protocol. This protocol encompasses micro demands, which focus on incorporating core news elements such as entities, dates, events, and the results of event, and macro demands, which ensure professional quality in terms of fluency, coherence, consistency, and relevance. By utilizing these complementary test sets, we aim to provide a more comprehensive evaluation of our EADSum method, assessing its performance not only in general summarization tasks but also in capturing nuanced, element-specific information that aligns with professional news writing standards.

Based on the evaluation using original reference summaries, our proposed EADSum training method demonstrates perfor-

TABLE III
THE ABLATION STUDY COMPARES THREE SCENARIOS: T5-BASE WITHOUT TRAINING, T5-BASE TRAINED ON 24.7% OF THE DATASET, AND T5-BASE WITH EADSUM TRAINED ON 24.7% OF THE DATASET.

Model	Training Size	Original Reference Summary				Element-aware Reference Summary			
		Rouge-1	Rouge-2	Rouge-L	BERTScore	Rouge-1	Rouge-2	Rouge-L	BERTScore
T5-Base	-	41.55	18.75	38.88	0.8783	31.10	11.00	27.56	0.8760
T5-Base	24.7%	41.29	18.87	38.57	0.8784	33.54	12.78	29.87	0.8798
T5-Base + EADSum	24.7%	41.29	18.63	38.39	0.8780	33.73	13.57	30.19	0.8805

mance comparable to, or even surpassing, some larger models presented in TABLE I. Notably, the T5-Base model with our EADSum method on just 24.7% of the data outperforms the non-finetuned T5-Large model in Mean-ROUGE scores. It also surpasses the PEGASUS-Large model trained on 100% of the data in ROUGE-1 and ROUGE-L scores. Furthermore, the EADSum method exhibits competitive performance when compared to BART-Base and T5-Base models trained on the full dataset.

In the second evaluation using element-aware reference summaries, as presented in TABLE II, our EADSum method demonstrates remarkable performance in Mean-ROUGE scores, ranking second only to GPT-3. While GPT-3, with its 175 billion parameters, can produce more informative summaries, it also demands substantial computational resources for deployment. Notably, our EADSum method achieves a Mean-ROUGE score of 25.83, surpassing most larger models and smaller models trained on the full dataset. This result signifies that our method generates summaries with greater overlap with element-aware reference summaries, showcasing superior performance in identifying and summarizing critical information elements. Moreover, the BERTScore of our EADSum method is competitive with those of BART-Base and T5-Large trained on the full dataset, as well as GPT-3 in a zero-shot setting. This performance is particularly impressive considering that our model uses only a fraction of the training data yet can generate summaries with high semantic similarity to expert-crafted, element-aware reference summaries.

These findings highlight the effectiveness of the EADSum approach in producing high-quality summaries that capture essential information elements while utilizing minimal training data.

C. Ablation Studies

To evaluate the effectiveness of our proposed EADSum method, we conduct ablation studies using the T5-Base model. We compared three configurations: T5-Base without training, T5-Base trained on 24.7% of the dataset, and T5-Base with EADSum trained on 24.7% of the dataset. This setup allows us to highlight the impact of EADSum and assess its performance in low-resource scenarios.

TABLE III presents an intriguing result: our EADSum method performs slightly lower than T5-Base (both without training and when trained on 24.7% of the dataset) in terms of ROUGE scores and BERTScores when evaluated against the original reference summaries. However, EADSum outperforms T5-Base in both scenarios when evaluated against element-aware reference summaries using the same metrics. We attribute this phenomenon to the structural differences between the original and element-aware reference summaries. Specifically, when a generated summary achieves a higher score compared to one type of reference summary, it indicates greater similarity in terms of word occurrence and semantic meaning to that particular reference summary style. Consequently, this may result in less similarity to the other type of reference summary.

These results demonstrate that EADSum not only compensates for the reduced training data but also enhances the model's ability to generate summaries that align more closely with expert-written, element-focused references.

IV. CONCLUSIONS

In this paper, we propose EADSum, a novel distillation framework comprising two steps: extracting element-aware rationales from Large Language Models (LLMs) and training compact models with the extracted rationales. Our method successfully distills the capabilities of LLMs into a compact model. Experimental results on the CNN/DailyMail dataset demonstrate that EADSum, when applied to the T5-Base model, outperforms some fully fine-tuned models on original reference summaries. Moreover, it surpasses most larger and smaller models trained on the full dataset when evaluated on element-aware reference summaries, while requiring significantly fewer computational resources and less training data. This approach not only advances the field of abstractive summarization but also provides a practical solution for real-world applications where computational resources are limited. Future work could explore the applicability of EADSum to other domains and languages, as well as investigate ways to further enhance the element-aware rationale training process.

REFERENCES

[1] M. F. Salchner and A. Jatowt, "A survey of automatic text summarization using graph neural networks," in *Proceedings of the 29th International Conference on Computational Linguistics*, N. Calzolari, C.-R. Huang, H. Kim, *et al.*, Eds., Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 6139–6150. [Online]. Available: https://aclanthology.org/2022.coling-1.536.

[2] A. P. Wibawa, F. Kurniawan, *et al.*, "A survey of text summarization: Techniques, evaluation and challenges," *Natural Language Processing Journal*, vol. 7, p. 100 070, 2024.

[3] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Neural abstractive text summarization with sequence-to-sequence models," *ACM Transactions on Data Science*, vol. 2, no. 1, pp. 1–37, 2021.

[4] Y. Zhang, Y. Liu, Z. Yang, *et al.*, "Macsum: Controllable summarization with mixed attributes," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 787–803, 2023.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[6] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

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

[8] M. Lewis, Y. Liu, N. Goyal, *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[9] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[10] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[11] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[12] S. Narayan, Y. Zhao, J. Maynez, G. Simões, V. Nikolaev, and R. McDonald, "Planning with learned entity prompts for abstractive summarization," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1475–1492, 2021.

[13] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[14] M. Zaheer, G. Guruganesh, K. A. Dubey, *et al.*, "Big bird: Transformers for longer sequences," *Advances in neural information processing systems*, vol. 33, pp. 17 283–17 297, 2020.

[15] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[16] P. Kavehzadeh, M. Valipour, M. Tahaei, A. Ghodsi, B. Chen, and M. Rezagholizadeh, "Sorted LLaMA: Unlocking the potential of intermediate layers of large language models for dynamic inference," in *Findings of the Association for Computational Linguistics: EACL 2024*, Y. Graham and M. Purver, Eds., St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 2129–2145. [Online]. Available: https://aclanthology.org/2024.findings-eacl.141.

[17] J. Achiam, S. Adler, S. Agarwal, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[18] Y. Wang, Z. Zhang, and R. Wang, "Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 8640–8665. DOI: 10.18653/v1/2023.acl-long.482. [Online]. Available: https://aclanthology.org/2023.acl-long.482.

[19] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "Summeval: Re-evaluating summarization evaluation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 391–409, 2021.

[20] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[21] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[22] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

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

[24] F. Iliopoulos, V. Kontonis, C. Baykal, G. Menghani, K. Trinh, and E. Vee, "Weighted distillation with unlabeled examples," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7024–7037, 2022.

[25] S. Arora, A. Narayan, M. F. Chen, *et al.*, "Ask me anything: A simple strategy for prompting language models," in *The Eleventh International Conference on Learning Representations*, 2022.

[26] V. Sanh, A. Webson, C. Raffel, *et al.*, "Multitask prompted training enables zero-shot task generalization," *arXiv preprint arXiv:2110.08207*, 2021.

[27] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.