DSQA-LLM: Domain-Specific Intelligent Question Answering Based on Large Language Model

Dengrong Huang[1], Zizhong Wei[1], Aizhen Yue[1], Xuan Zhao[3], Zhaoliang Chen[2], Rui Li[1], Kai Jiang[1], Bingxin Chang[1], Qilai Zhang[1], Sijia Zhang[1], and Zheng Zhang[1](✉)

[1] Inspur Academy of Science and Technology, Jinan, Shandong, China
jiangkai@inspur.com
[2] Inspur Software Co., Ltd., Jinan, Shandong, China
[3] China International Information Center, Beijing, China

Abstract. Question Answering (QA) is crucial for humans to access vast knowledge bases, but there is a lack of attention towards representing raw, unstructured questions and answers in specific fields. Additionally, the efficiency of finding candidate questions based on the trigger question and the generation of reasonable answers have been neglected. In this paper, we introduce Domain Specific Question Answering Language Model (DSQA-LLM), a framework that delivers informative answers within a specific domain. We utilize techniques like question classification, information retrieval, and answer generation. We enhance efficiency and accuracy through the integration of XLNET for question classification and a novel similarity searching method using Sentence-T5. Furthermore, the powerful GPT-3.5-turbo is employed for generating coherent answers. We implemented DSQA-LLM and curated a dataset of 127,840 question-answer pairs. Empirical experiments conducted on real-world questions confirm the effectiveness of our QA system.

Keywords: Question Answering · LLM · XLNET · Sententce-T5 · deep learning · natural language processing

1 Introduction

With the expanding volume of domain-specific knowledge bases (KBs), there is a growing interest in accessing these valuable resources effectively. Domain-specific knowledge base-based question answering (DSKB-QA) has gained prominence as a user-friendly solution, utilizing natural language as the query language. The objective of DSKB-QA is to automatically retrieve accurate results and aggregate them based on relevance to user queries. This paper focuses on DSKB-QA in the digital government domain, where each data sample consists of a 4-tuple

D. Huang, Z. Wei and A. Yue—Equal contribution.

F. Zhao and D. Miao (Eds.): AIGC 2023, CCIS 1946, pp. 170–180, 2024.
https://doi.org/10.1007/978-981-99-7587-7_14

(type, question, document, and answer). Specifically, our paper addresses the domain-specific extractive QA task, extracting answers from contextual information based on given questions as input.

Question classification is essential for QA systems in NLP, as it assigns labels to questions and narrows down the search range in large datasets. This helps accurately locate and verify answers. Transformer-based models like XLNet have gained attention for their ability to learn global semantic representation and handle large-scale datasets without relying on sequential information. They have significantly improved NLP tasks, including text classification. In this paper, we fine-tune XLNet using our question-type dataset to enhance question classification accuracy.

Similarity query processing is essential in domains like databases and machine learning. Deep learning techniques, including embedding and pre-trained models, have significantly improved similarity query processing for high-dimensional data. Question embedding plays a crucial role in retrieving similar questions, and a recent approach involves fine-tuning large language models like Sentence-T5 for a candidate question retriever. In our embedding module, we also leverage Sentence-T5 to enhance the precision and effectiveness of question search in our government-related dataset. To achieve efficient similarity search of dense vectors, we utilize cosine distance specifically designed for similarity question search. By calculating similarity, we retrieve the most similar queries and obtain candidate document-level answers accordingly.

Document summarization is essential for condensing text while preserving important information. With the abundance of public text data, automatic summarization techniques are becoming increasingly important. Large language models like GPT-3 possess strong natural language understanding and generation capabilities. Comparisons with traditional fine-tuning methods show that GPT-3 exhibits excellent memory and semantic understanding. Furthermore, analysis confirms that these large language models generate answer summaries that are comparable to those produced by human experts. To improve the conciseness, readability, and logical consistency of answers derived from original documents, we have integrated these models into our question answering system.

In summary, this paper presents the following contributions:

1) Introduction of DSQA-LLM, a domain-specific question answering system designed to provide relevant and concise answers to trigger questions within a specific domain.
2) Proposal of a novel technique that combines text classification, sentence embedding, and answer generation, utilizing both traditional fine-tuning models and large language models (LLMs) to enhance accuracy and reasonableness.
3) Extensive experiments on domain-specific question answering datasets to demonstrate the effectiveness of our approach.

The subsequent chapters are structured as follows: Sect. 2 provides an overview of related work on DSQA-LLM. Section 3 presents our proposed approach and implementation details. Section 4 outlines the experiments conducted and presents the results. Finally, Subsect. 5 concludes our work.

2 Related Work

2.1 Question Classification

Question classification techniques can be categorized into Statistics-based, NN-based, Attention-based, and Transformer-based methods. Statistics-based techniques, such as Naive BayesSupport Vector Machine [1], offer accuracy and stability. Recent methods like XGBoost [2] show promise in this area. NN-based techniques, such as TextCNN [3], employ neural networks for text classification. Attention-based techniques like HAN [5] have achieved success in text classification by leveraging informative components and addressing imbalances in few-shot scenarios Transformer-based models, like ALBERT [6], and BART [7], excel at handling large-scale datasets and capturing bidirectional context, demonstrating excellent performance in text classification tasks.

2.2 Question Embedding

Deep learning techniques, such as XLNet [15], RoBERTa [17], SimCSE [8], and Sentence-T5 [9], have been effective in modeling sentence similarity. These Transformer-based models have achieved impressive performance in tasks like question answering. The Transformer model, introduced by Vaswani et al. [11], is successful in sequence-to-sequence tasks. Cer et al. [12] and Radford et al. [13] employed Transformer encoder and decoder for transfer learning and language modeling. BERT [14], with contextualized representations, is a notable advancement. XLNet improves upon BERT through the Transformer-XL architecture [16]. SimCSE and Sentence-T5 stand out by introducing the contrastive loss. Among these models, Sentence-T5 demonstrates innovative design choices and training strategies, outperforming SimCSE with superior performance.

2.3 Answer Extraction and Generation

Advancements in deep neural networks have accelerated extractive summarization. Sequential neural models like recurrent neural networks [24] and pre-trained language models are widely used [18]. However, limited exploration has been done with large language models, such as ChatGPT. Studies have explored the application of large language models in text summarization. Goyal et al. [19] compared GPT-3-generated summaries with traditional methods. Zhang et al. [20] also examined ChatGPT's performance in extractive summarization and proposed an extract-then-generate pipeline. LLMs have also been used for summarization evaluation, outperforming previous methodologies.

3 Approach

3.1 Overview

Our Question Answer system, DSQA-LLM, consists of three phases: “Question Processing”, “Similarity Searching”, and “Answer Processing”. When given a

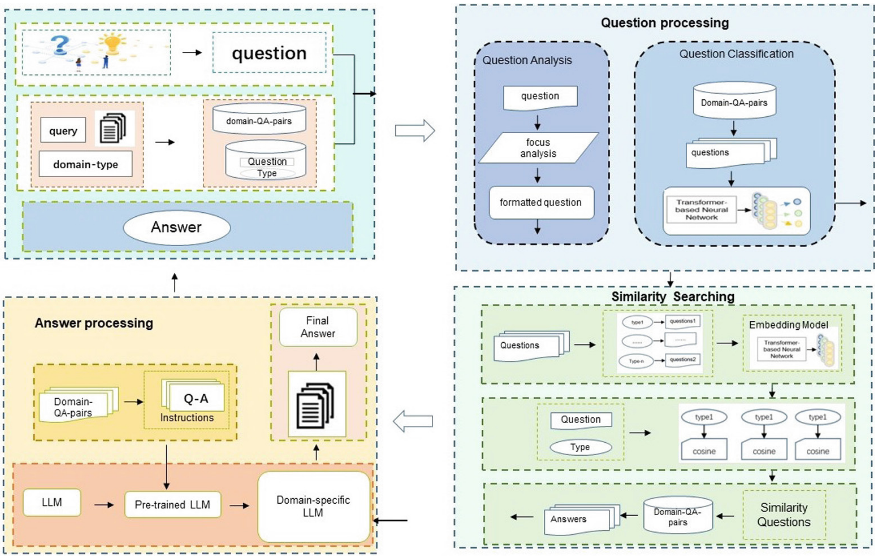

Fig. 1. Overview of our approach

question, DSQA-LLM follows these steps. Firstly, the question analyzer generates the formatted question, Qf, based on specific rules. The Question Classification Module identifies the question type T using a classification model trained on labeled questions from the government-related domain. Qf and T are then passed to the second phase. In the second phase, candidate answers are obtained using an embedding module. Similarity questions (QQP) and question-answer pairs (QAP) from the government-related domain are transformed into formatted representations using a question analyzer. The Sentence-T5 model is fine-tuned on the QQP dataset to obtain embeddings for targeted questions and questions in QAP. By matching the candidate embeddings corresponding to question type T, similarity questions are obtained, and candidate answers from the knowledge dataset QAP are acquired. A list of documents serving as candidate answers is then passed to the third phase. In the third phase, the final answer is derived using an LLM model. DSQA-LLM fine-tunes the LLM model with domain-specific prompts to address miscellaneous and redundant answers obtained in the previous phase. With the candidate answers and fine-tuned LLM model, a reasonable answer is generated, which is validated for accuracy before being presented to the user. Keep in mind that this is an overview, and certain details are omitted due to page limitations (Fig. 1).

3.2 Question Processing Phase

We introduce the popular classification model XLNet into our question process phase, and design the question process as follows.

Table 1. Example few-shot Prompts for answer extraction and generation

Prompt1 (few-shot):
Please extract Summary information for the following Content:
Content: {content1}
Summary: {summary1}
Content: {content}
Summary: {?}
Prompt2 (few-shot):
Please extract summary information from the following content, using the examples provided as guidance, note that not to make up irrelevant information yourself:
Content: {content1}
Summary: {summary1}
Content: {content}
Summary: {?}
......

Table 2. Example prompt for optimal answer selection

Please evaluate the relevance of Summary 1, Summary 2, and Summary 3 in relation to the corresponding Text. A fully relevant summary should include information that is important to the content and should not include other irrelevant information. Afterward, select one of the following options (A, B, C):
Content: {content1}
Summary 1: {summary1}
Summary 2: {summary2}
Summary 3: {summary3}
A: Summary 1 is more relevant. B: Summary 2 is more relevant. C: Summary 3 is more relevant.

Your choice (enter A, B, or C): ?

Question Analysis. The questions involved in DSQA-LLM are often colloquial and confusing, which significantly impacts the accuracy and performance of question classification and similarity search. Therefore, it is crucial to conduct specific pre-processing to generate formatted questions for subsequent use. We employ pattern matching rules to identify the main focus of the questions and remove any unnecessary information.

Question Type Classification. To understand the domain-specific information sought by the question and establish constraints on relevant data, DSQA-LLM uses the XLNet model for question type classification. The model incorporates the segment recurrence mechanism and relative encoding scheme of Transformer-XL, providing improved performance for longer text sequences. The final hidden state of [CLS] in XLNet is used as the representation for the entire sequence, and a softmax classifier predicts the probability of the label [15]. The parameters of XLNet are fine-tuned by maximizing the log-probability of the correct label. The loss function for the classification task is defined as follows,

$$\mathcal{L}_\theta = \max_\theta \mathbb{E}_{s \sim S_t} \left[\sum_{t=0}^{T} \log p_\theta (X_{s_t} | X_{s<t}) \right] \tag{1}$$

where X_t and $X_{s<t}$ represent the t_{th} element and the first $t-1$ elements of a permutation X.

3.3 Similarity Searching Phase

As the reformulated question is submitted to the similarity searching phase, which retrieves a ranked list of relevant candidate answers for the third phase. Our Similarity Searching process consists of two modules: question embedding module and similarity searching module.

Question Embedding Module. To improve the uniformity of sentence embeddings for similarity searching, DSQA-LLM utilizes contrastive learning. This approach, known for its effectiveness in tasks like Semantic Textual Similarity (STS), involves fine-tuning Sentence-T5 representations using a contrastive loss function [8]. During training, positive examples (related sentences) are encouraged to be closer to the input sentence, while all other examples in the batch are treated as negatives. The contrastive loss is computed using in-batch sampled softmax and the similarity score calculated by the function f [26]. It utilizes paired examples $(s_i,\ s_i^+)$ where s_i represents the input sentence and s_i^+ is a related sentence, along with additional negative examples in the form of s_j^-. The loss function can be described as follows:

$$\mathcal{L} = -log\frac{exp(f(s_i, s_i^+)/\tau)}{\sum_{j \in D} exp(f(s_i, s_j^+)/\tau) + exp(f(s_i, s_j^-)/\tau)}, \tag{2}$$

Searching Module. Efficiently searching for similar questions in our knowledge base is crucial to minimize search costs in our system. To achieve this, we utilize cosine distance for type-specific similarity search, enabling the construction of a type-dominated search module. When a question is inputted, our system identifies its type using the "Question Type Classification" module. We then retrieve the corresponding embeddings using the embedding module and obtain candidate questions based on the targeted question type. This approach allows for the efficient retrieval of related questions. Furthermore, leveraging the labeled question-answer pairs in our DSQA-LLM dataset, we employ a simple matching technique to obtain candidate answers.

Answer Extraction and Generation. Abstractive summarization involves generating a summary, referred to as Y, by considering the input source document, represented as X. The source document is composed of individual sentences, creating a representation as $X = \{X_1, X_2, ..., X_T\}$. To generate the summary $Y = \{Y_1, Y_2, ..., Y_T\}$, a generative language model utilizes the following probability [26]:

$$p(Y|X, \theta) = \prod_{t}^{m} p(Y_t|Y_{<t}, X, \theta). \tag{3}$$

Large language models have demonstrated impressive task performance, even with limited training data, thanks to in-context learning (ICL). In the standard ICL approach [26], a language model M is trained on a set of example input-output pairs, represented as $\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, where x is the input text and y is the expected output. The objective is to predict the answer text $\hat{y}$ given a query text using this training. This prediction is achieved by calculating the likelihood of each candidate answer y_j using a scoring function f that incorporates the entire input sequence and the language model M.

$$\hat{y} = \underset{y_j \in Y}{argmax} \sum_{j} f_M(y_j, C, x). \tag{4}$$

Table 3. Overview Of Metrics.Qes Clas denotes classification evaluation metrics; Sim Sea denotes similarity searching evaluation metrics; Ans Gene denotes answer generation evaluation metrics

#Qes Clas	#Sim Sea	#Ans Gene
$Precision = \frac{v_j}{\sum_{i=1}^{T} v_i} \quad (5)$	$MRR = \frac{1}{\lvert Q\rvert} \sum_{i=1}^{\lvert Q\rvert} \frac{1}{rank_i} \quad (8)$	ROUGE-1
$Recall = \frac{v_i}{\sum_{j=1}^{T} v_j} \quad (6)$	$MAP = \frac{1}{\lvert Q\rvert} \sum_{i=1}^{\lvert Q\rvert} AveP(C_i, A_i) \quad (9)$	ROUGE-2
$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$	$AveP(C_i, A_i) = \frac{\sum_{k=1}^{n}(P(k) \cdot rel(k)}{min(m,n)} \quad (10)$	ROUGE-L

The set $C = \{I, s(x_1, y_1)...s(x_m, y_m)\}$ represents the collection of explanations and input-output pairs used as prompts in this formulation. Additionally, this research explores how in-context learning affects extractive summarization. In our DSQA-LLM system, we generate summaries using prompt-based approaches, including few-shot prompts following OpenAI API guidelines. In Table 1, we employ various prompts with the LLM to generate summaries. These prompts, such as "prompt1", "prompt2", and others listed, were accompanied by content and summary examples. By inputting the desired content into the LLM, we obtained three summaries: "summary1", "summary2", and "summary3". To evaluate their quality and determine the optimal summary, we utilized an evaluation LLM and collected feedback using the prompt specified in Table 2.

4 Experiment

4.1 Datasets

We gather a training dataset consisting of 67,840 categorized questions from various government fields for training the question classification model. An additional 14,650 questions (NT1) are utilized for testing purposes. To assess the accuracy of our approach in identifying similar questions, we compile a set of 83,790 labeled true similarity pairs and 32,174 false pairs (NT2). Additionally, we curate a collection of 127,840 question-answer pairs (NT3) to evaluate answer extraction capabilities. Further information regarding the distribution of related pairs in the training and test datasets can be found in Table 4.

4.2 Metrics and Baseline

Metrics. We evaluate our Question Type Classification with precision, recall, and F1-Score. Similarity Searching is assessed using MRR and MAP. Answer processing is evaluated using ROUGE [25] metrics, specifically ROUGE-1, ROUGE-2 and ROUGE-L. These metrics can be found in Table 3 (Table 5).

Table 4. Overview Of dataset

Dataset	#NT1	#NT2	#NT3
Training	67840	83790	127840
Test	14650	32174	/

Table 5. Result of question classification.

Metrics	Presicion (%)	Recall (%)	F1-score (%)
BLSTM-2DCNN	86.09	86.39	86.24
HAN	88.89	86.71	87.79
ALBERT	97.24	98.81	98.20
BART	97.98	98.60	98.29
RoBERTa	98.67	98.81	98.74
XLNet	**99.24**	99.06	**99.15**

Baseline. To demonstrate the effectiveness of the proposed QA system, VDQA-LLM, we compare our methods with the corresponding state-of-art efforts.

- **Question Classification.** We compare our approach, which utilizes the XLNet model, to other 5 typical text classification models, i.e. BLSTM-2DCNN [4], HAN [5], ALBERT [6], BART [7] and RoBERTa [17].
- **Similarity Searching.** We compare our Sentence-T5 based approach to other 5 typical sentence embedding models, i.e. BERT [14], SBERT [10], SRoBERTa [10], SimCSE-BERT [8], and SimCSE-RoBERTa [8].
- **Answer Extraction and Generation.** We have conducted a thorough investigation of several state-of-the-art summary generation models. These models include Seq2seq [23], Seq2seq + Att [24], BERT [14], RoBERTa [17], LLAMA [22], GLM [21], and GPT-3.5-turbo. To explore various learning strategies for Language Learning Models (LLMs), we employ few-shot learning approaches.

4.3 Results

Question Classification. Our XLNet-based approach achieves outstanding performance in question classification, boasting an impressive F1-Score of 99.15%. This remarkable accomplishment can be attributed to the XLNet-based approach's ability to overcome the limitations of previous autoregressive models by incorporating bidirectionality. Transformer-based methods consistently outperform traditional neural network-based methods due to their multi-head attention layer and self-attention module. Notably, ALBERT, BART, RoBERTa, and XLNet outshine other methods, achieving an F1-Score of over 98%. This highlights the effectiveness of BERT in text classification tasks.

Similarity Searching. The sentence embedding model's evaluation metrics include Precision, Recall, and F1-Score. Among BERT-based methods, Sentence-T5, SimCSE-BERT, and SimCSE-RoBERTa achieve impressive F1-Scores of 99.43%, 98.57%, and 97.98% respectively. The utilization of contrastive loss by these models enhances feature extraction and contributes to their superior performance. Notably, Sentence-T5 outperforms both SimCSE-BERT and

SimCSE-RoBERTa, highlighting T5's advantages in extracting sentence features. When it comes to similarity search, our Sentence-T5-based model surpasses other baselines in Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). Compared to BERT, SBERT, SRoBERTa, SimCSE-BERT, and SimCSE-RoBERTa, our model achieves significant MAP improvements of 0.097, 0.070, 0.036, 0.042, and 0.015 respectively. Similarly, the MRR improvements are 0.097, 0.086, 0.060, 0.031, and 0.022 respectively (Table 6).

Table 6. Result of question searching.

Metrics	MAP	MRR
BERT	0.870	0.892
SBERT	0.897	0.903
SRoBERTa	0.904	0.929
SimCSE-BERT	0.925	0.958
SimCSE-RoBERTa	0.952	0.971
Sentence-T5	**0.967**	**0.989**

Table 7. Result of answer generation

Metrics	ROUGE-1	ROUGE-2	ROUGE-L
Seq2seq	17.9	7.8	13.7
Seq2seq + Att	11.2	13.2	20.5
BERT	31.0	16.5	22.8
RoBERTa	38.6	17.4	25.4
Vicuna-7b	39.2	19.4	27.5
GLM-6b	40.3	20.1	28.9
GPT-3.5-turbo	**43.2**	**21.6**	**33.5**

Answer Extraction and Generation. As presented in Table 7, LLMs demonstrate superior extractive capabilities in summarization compared to traditional methods, with BERT-based models outperforming seq2seq models in terms of ROUGE scores. This is due to their extensive training on large amounts of textual data and the advantages of transformer architectures. GPT-3.5-turbo outperforms Vicuna-7b and GLM-6b in few-shot learning scenarios, indicating its strong performance in extractive summarization. Despite being designed as a generation model, GPT-3.5-turbo exhibits deep understanding of problem formulation and semantic meaning. Its decoder-only structure sets it apart from encoder-decoder models like BERT. Fine-tuning also improves the performance of GLM-6b and Vicuna-7b in few-shot learning scenarios.

5 Conclusion

In this paper, we propose DSQA-LLM, a novel technique for QA in the vertical government-related domain. DSQA-LLM combines LLM and FAISS to improve precision in searching and obtaining accurate answers. Our framework integrates popular NLP techniques such as XLNet, Sentence-T5, and GPT-3.5-turbo to further enhance the precision and optimize searching time. Through the implementation of our prototype framework, DSQA-LLM, and extensive empirical experiments, we validate the effectiveness of our approach. The results demonstrate precise and efficient Question Processing, Similarity Searching, and Answer Generation.

References

1. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0026683
2. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
3. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
4. Zhou, P., Qi, Z., Zheng, S., et al.: Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. arXiv preprint arXiv:1611.06639 (2016)
5. Yang, Z., et al.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2016)
6. Lan, Z., Chen, M., Goodman, S., et al.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
7. Lewis, M., Liu, Y., Goyal, N., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
8. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
9. Ni, J., Ábrego, G.H., Constant, N., et al.: Sentence-T5: scalable sentence encoders from pre-trained text-to-text models. arXiv preprint arXiv:2108.08877 (2021)
10. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
11. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
12. Cer, D., Yang, Y., Kong, S., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
13. Radford, A., Narasimhan, K., Salimans, T., et al.: Improving language understanding by generative pre-training (2018)
14. Devlin, J., Chang, M.W., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Yang, Z., Dai, Z., et al.: XLNet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
16. Dai, Z., Yang, Z., Yang, Y., et al.: Transformer-XL: attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
17. Liu, Y., Ott, M., Goyal, N., et al.: RoBERTa: a robustly optimized BERT pre-training approach. arXiv preprint arXiv:1907.11692 (2019)
18. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345 (2019)
19. Goyal, T., Li, J.J., Durrett, G.: News summarization and evaluation in the era of GPT-3. arXiv preprint arXiv:2209.12356 (2022)
20. Luo, Z., Xie, Q., Ananiadou, S.: ChatGPT as a factual inconsistency evaluator for abstractive text summarization. arXiv preprint arXiv:2303.15621 (2023)

21. Du, Z., Qian, Y., Liu, X., et al.: GLM: general language model pretraining with autoregressive blank infilling. arXiv preprint arXiv:2103.10360 (2021)
22. Touvron, H., Lavril, T., Izacard, G., et al.: LLaMA: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
23. Cho, K., Van Merriënboer, B., Gulcehre, C., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
24. Nallapati, R., Zhou, B., Gulcehre, C., et al.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv preprint arXiv:1602.06023 (2016)
25. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text summarization Branches Out, pp. 74–81 (2004)
26. Zhang, H., Liu, X., Zhang, J.: Extractive summarization via chatGPT for faithful summary generation. arXiv preprint arXiv:2304.04193 (2023)