Contextual Compression in Retrieval-Augmented Generation for Large Language Models: A Survey

Anonymous ACL submission

Abstract

Large Language Models (LLMs) showcase remarkable abilities, yet they struggle with limitations such as hallucinations, outdated knowledge, opacity, and inexplicable reasoning. To address these challenges, Retrieval-Augmented Generation (RAG) has proven to be a viable solution, leveraging external databases to improve the consistency and coherence of generated content, especially valuable for complex, knowledge-rich tasks, and facilitates continuous improvement by leveraging domain-specific insights. However, RAG is not without its limitations, including a limited context window, irrelevant information, and the high processing overhead for extensive contextual data. In this comprehensive work, we explore the evolution of Contextual Compression paradigms, providing an in-depth examination of the field. We also introduce a state-of-the-art evaluation framework and benchmark. Finally, we outline the current challenges and suggest potential research and development directions, paving the way for future advancements in this area.

1 Introduction

The pioneering accomplishments of large language models (LLMs) have galvanized research initiatives across both industrial and academic spheres. These LLMs showcase their capacity to converse with humans in a natural and articulate manner, excelling across various tasks such as document summarization, Q&A systems, conversational AI, and coding assistants. Despite their advancements, LLMs continue to struggle with tasks that require specialized knowledge or domain-specific expertise. (Kandpal et al., 2023). Notably, they may produce "hallucinations" (Zhang et al., 2023) when confronted with out-of-scope queries or requests that necessitate up-to-date knowledge. To address these challenges, Retrieval-Augmented Generation (RAG) leverages external knowledge bases to retrieve relevant document snippets, utilizing semantic similarity metrics to identify the most pertinent information. By tapping into external knowledge sources, RAG successfully alleviates the issue of generating inaccurate content, thereby increasing the reliability of LLMs and paving the way for their widespread adoption in real-world applications.

However, RAG also has its challenges. One issue is that when retrieving relevant documents, the important information may be buried in a large amount of irrelevant text, leading to inefficient and poor responses. Another challenge is that current language models have a limited input length, which causes their performance to decline when processing lengthy documents, such as academic articles, research papers, or literary works. This constraint has fueled research into developing methods to increase the input length while maintaining the model's accuracy and efficiency.

This paper aims to shed light on the latest advancements in contextual compression methods, with a focus on their application in retrieval-based systems. Our research involves a comprehensive review of methodologies, metrics, and benchmarks, which we systematically categorize into a novel taxonomy. Our taxonomy, as shown in Figure 1, presents a structured and comprehensive framework for categorizing and analyzing Contextual Compression techniques for LLMs. Our investigation involves a comprehensive analysis of established techniques, such as semantic compression, in-context auto-encoder compressors, and auto-compressors, among others. Furthermore, our research highlights the ongoing challenges in this field and provides a roadmap for future investigations. We emphasize the need for collective efforts to create a sustainable and environmentally responsible future for LLMs.

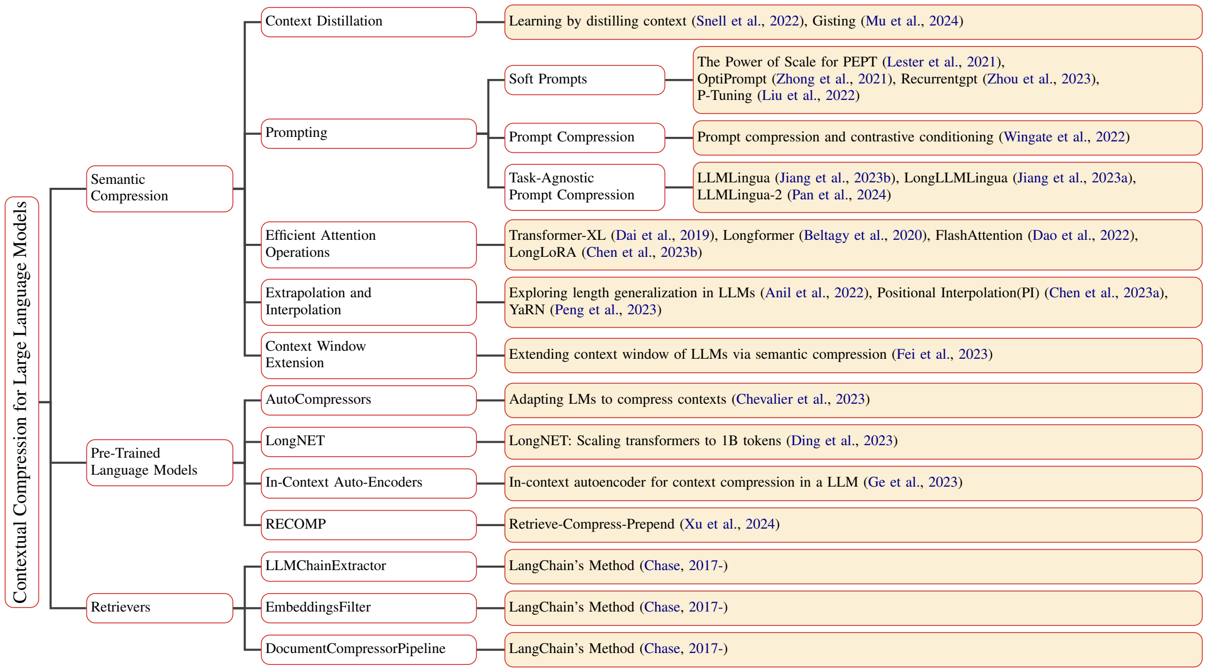

Figure 1: Taxonomy of Contextual Compression Methods for Large Language Models.

2 Methods

2.1 Semantic Compression

Semantic compression is a technique that helps identify common patterns of thought in a specific context by generalizing terms. It uses a "domain frequency dictionary" to establish the context and disambiguate multiple possible meanings of words. This approach, based on semantic networks, offers improvements over existing natural language processing techniques.

Semantic compression reduces the number of terms in a text document by replacing less frequent terms with more general terms (their hypernyms) using a semantic network and term frequency data. This compression minimizes information loss and enables efficient processing, especially in tasks involving vector space models (Baeza-Yates et al., 1999), (Erk and Padó, 2008). It also helps address linguistic (Sinha and Mihalcea, 2007) challenges like polysemy and synonymy (Krovetz and Croft, 1992) by replacing multiple rare terms with a single, more general concept. By using statistical analysis and frequency dictionaries, semantic compression can handle polysemic concepts more effectively and with lower error rates than other techniques. These efforts can be summarized into five approaches: *Context Distillation*, *Prompting*, *Efficient Attention Operations*, *Extrapolation and Interpolation*, and *Context Window Extension*.

2.1.1 Context Distillation

Recent studies have demonstrated that augmenting language models (LMs) with contextual information, such as task descriptions, illustrative examples, and explanatory notes (Chen et al., 2021), (Scheurer et al., 2022), can substantially enhance their performance capabilities. This approach can even facilitate zero-shot learning (Wei et al., 2021), (Victor et al., 2022) and enable models to tackle complex tasks by generating sequential reasoning steps (Nye et al., 2021), (Wei et al., 2022), (Zhou et al., 2022).

While LMs perform better with context tokens, this advantage disappears when the tokens are removed. Additionally, processing context tokens requires extra computation, which can be a drawback. The context tokens can also be very long, and it's unclear how to handle them when they exceed the context window size. These limitations are similar to human cognitive limitations (Wason and Evans, 1974), such as struggling with complex tasks and having limited working memory (Baddeley, 1992).

Humans overcome challenges through practice, which allows them to "distill" knowledge into habits and muscle memory. For example, learning to type a phone number becomes automatic with repetition, freeing up conscious reasoning for more complex tasks [1]. This process is essential

[1] procedural learning vs. declarative learning - https://en.wikipedia.org/wiki/Procedural_knowledge

for building skills and knowledge, enabling us to tackle increasingly intricate challenges.

Researchers in NLP (Askell et al., 2021), (Snell et al., 2022) are exploring techniques to fine-tune language models, such as context distillation and "Gisting". Context distillation involves generating "practice" questions, having the model reason step-by-step, and fine-tuning it to predict answers from simpler prompts. This helps the model internalize skills, like step-by-step addition (ref Figure 2). "Gisting" (Mu et al., 2024) compresses instructions into concise key-value attention prefixes, saving computational resources and generalizing well to new tasks. As depicted in Figure 3, the approach involves learning a gist model by incorporating gist tokens during instruction tuning, enabling the model to handle prompt compression and instruction following simultaneously.

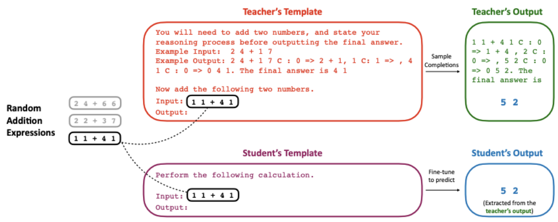

Figure 2: Internalization of step-by-step reasoning via context distillation (Snell et al., 2022)

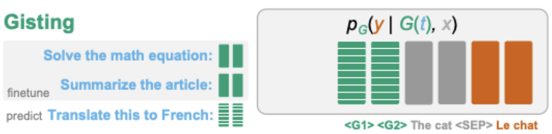

Figure 3: Gisting - Each vertical rectangle here represents a stack of Transformer activations (Mu et al., 2024)

2.1.2 Prompting

Soft Prompts - As depicted in Figure 4, soft prompt tuning enables the adaptation of pre-trained Transformers without modifying their underlying parameters, as demonstrated in recent studies (Lester et al., 2021), (Zhong et al., 2021), and (Liu et al., 2022). It entails adding novel embeddings to the input sequence and fine-tuning only these new parameters while keeping the remainder of the model's architecture frozen. This approach is categorized as a parameter-efficient fine-tuning method (PEFT) (Lialin et al., 2023), and bears resemblance to prefix tuning, which prepends task-specific vectors to the attention states instead of the input sequence (Li and Liang, 2021).

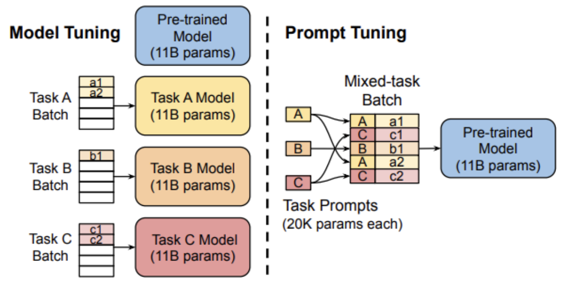

Figure 4: From 11 billion for a tuned model to just 20,480 for a tuned prompt, a reduction of over 5 orders of magnitude (Lester et al., 2021)

Prompt Compression - In their work, (Wingate et al., 2022) hypothesize using a soft prompt sp to compress information from a context ctx. They use a pre-trained LM p_{LM} to generate continuations $cty \sim p_{\text{LM}}(\cdot \mid ctx)$ based on the context, and then calibrate the model's outputs with the soft prompt sf, $p_{\text{LM}}(cty \mid sf)$ to the outputs based on the context ctx, $p_{\text{LM}}(cty \mid ctx)$. They find that soft prompts effectively preserve abstract knowledge and improve guided output. Nevertheless, this method necessitates distinct optimization for each novel context, lacking the ability to leverage knowledge across analogous contexts.

Task-Agnostic Prompt Compression - Current methods for compressing natural language prompts remove tokens or lexical units based on information entropy from a language model like LlaMa-7B. However, using information entropy as a compression metric has two limitations: 1) it only considers unidirectional context, which may miss important information, and 2) it doesn't perfectly align with the goal of prompt compression.

To address these issues, (Pan et al., 2024) propose a data distillation approach to compress prompts while retaining essential information. They introduce an extractive text compression dataset and frame prompt compression as a token classification problem (preserve or discard) (Refer to Figure 5). The key benefits are as follows:

1. *Comprehensive Information Capture:* By leveraging a Transformer encoder, the method captures essential details from the full bidirectional context.
2. *Reduced Latency:* Smaller models explicitly

learn the compression objective, leading to lower latency.

3. *Faithfulness:* The compressed prompt remains faithful to the original content.

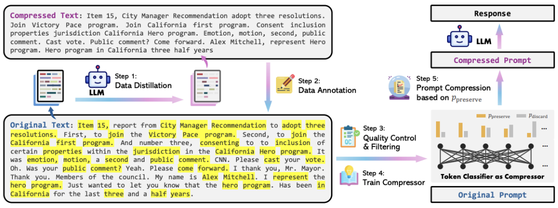

Figure 5: Overview of LLMLingua-2 (Pan et al., 2024)

2.1.3 Efficient Attention Operations

The self-attention mechanism in LLMs leads to an inference cost that scales quadratically with sequence length, prompting the development of various methods to alleviate this complexity. For example:

- *Transformer-XL (Dai et al., 2019)* - employs a recurrent architecture that operates on segments, paired with a novel positional encoding technique.
- *Longformer (Beltagy et al., 2020)* - introduces sparse attention, scaling linearly with sequence length.
- *FlashAttention (Dao et al., 2022)* - uses chunking and re-computation to avoid quadratic attention complexity.

However, these methods can be expensive to train and struggle with out-of-distribution content lengths (Ding et al., 2023). To address this, *LongLoRA (Chen et al., 2023b)* provides a computationally efficient fine-tuning method with minimal resource requirements. For further insights, refer to the study by (Huang et al., 2023).

2.1.4 Extrapolation and Interpolation

In the field of NLP, researchers are investigating methods to extend the capabilities of existing language models, initially trained on brief texts, to process longer sequences during inference (Anil et al., 2022). One approach is to alter positional embeddings, which are typically designed for shorter contexts. The Rotary Position Embeddings (RoPE) from LLaMA is a key foundation for several studies in this area. For example:

- *Position Interpolation (PI) (Chen et al., 2021)* applies a linear transformation to input positional indices.
- *YaRN (Peng et al., 2023)* leverages neural tangent kernel-inspired mechanisms to scale up the context window to 64,000 and 128,000 tokens.

2.1.5 Context Window Extension

Researchers (Fei et al., 2023) propose a semantic compression method that distills long texts into concise forms, retaining their meaning and broadening the context window (Figure 6). This method occurs before inputting tokens into pre-trained language models and is customizable and optimized for specific tasks. It outperforms existing methods in various tasks, including question answering, summarization, and few-shot learning, without requiring additional parameter updates or memory consumption, making it computationally efficient.

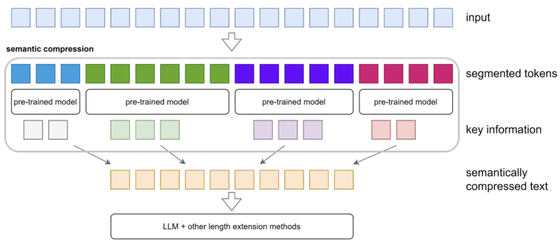

Figure 6: 1) clustering the input text into thematic groups, represented as a graph, to facilitate topic-based analysis, 2) tuning the thematic segments using pre-trained models to preserve crucial details, and 3) reassembling the refined chunks in their original order - reducing the text length by approximately 6-8 times. Additionally, other techniques like extrapolation and interpolation can be used to further extend the length (Fei et al., 2023)

2.2 Pre-Trained Language Models (PLMs)

The development of PLMs has revolutionized the field of NLP. The first generation of PLMs, such as Skip-Gram (Mikolov et al., 2013b), word2vec (Mikolov et al., 2013a), and GloVe (Pennington et al., 2014), used shallow neural networks (Qiu et al., 2020) to obtain word embeddings. The second generation, including CoVe (McCann et al., 2017), ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and GPT (Radford et al., 2018), focused on learning dynamic word embeddings using transformers. The pre-training and fine-tuning approach has achieved remarkable success in various NLP tasks. Moreover, recent breakthroughs in prompt learning (Liu et al., 2023a) have empowered PLMs to accomplish few-shot or zero-shot learning with minimal labeled data. Notable examples of successful PLMs include ChatGPT, GPT-4, Gemini, Claude, LlaMA-3, Mixtral, etc.

2.2.1 AutoCompressors

The authors of (Chevalier et al., 2023) propose teaching PLMs to compress text into summary vectors (Lester et al., 2021), which are significantly shorter than the original text (often 1-2 orders of magnitude shorter). These vectors have a two-pronged function: 1) they allow the LM to handle long documents by extending its context window with minimal computational overhead, and 2) they accelerate inference for pre-computed and cached text.

AutoCompressors, proposed by (Chevalier et al., 2023), are trained To distill key information into summary vectors, generated sequentially from extended documents (Figure 7). The approach builds upon the Recurrent Memory Transformers (RMT) architecture (Bulatov et al., 2022), introducing summary accumulation and training with randomly segmented inputs. This enhances long-range information retention and facilitates reasoning across multiple passages. AutoCompressors can be seeded with PLMs and fine-tuned on long sequences. They improve perplexity for long documents and demonstrate robust compression capabilities across different domains, making them valuable for various downstream applications.

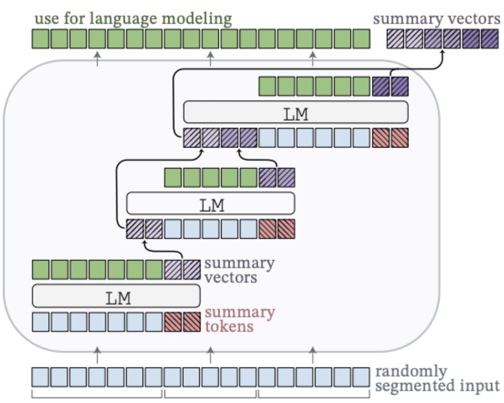

Figure 7: AutoCompressors recursively generate summary vectors from long documents, using them as soft prompts for subsequent segments (Chevalier et al., 2023)

2.2.2 LongNET

Overcoming sequence length limitations in language models has several advantages, including improved interactions with human language, better capture of complex causality and reasoning, and reduced catastrophic forgetting. However, scaling up sequence length poses a challenge in balancing computational complexity and model expressivity. RNN-style models and state space models (Gu et al., 2021), (Smith et al., 2022), (Fu et al., 2022), (Poli et al., 2023) have been proposed, but they have limitations from the perspective of parallelization and model adaptability (Fathi et al., 2023). An alternative approach is to reduce the complexity of Transformers (Vaswani et al., 2017), such as using sliding windows or convolution modules for attention, or sparse attention. LongNet (Ding et al., 2023), a novel approach, replaces the attention mechanism with "dilated attention", which achieves linear computational complexity and logarithmic dependency between tokens. This allows LongNet to efficiently scale sequence lengths to 1 billion tokens, overcoming the constraints of computation and memory.

2.2.3 In-Context Auto-Encoders

Modeling long-range dependencies is a hurdle for Transformer-based LMs (Vaswani et al., 2017) due to their self-attention mechanism. Previous research by (Beltagy et al., 2020), (Bulatov et al., 2022), and Ding (Ding et al., 2023) has attempted to cope with this issue through architectural innovations, but these approaches often struggle to maintain performance in long contexts, as underscored by (Liu et al., 2024). A novel approach, "context compression", is proposed by (Ge et al., 2023), which recognizes that an LLM can represent the same information in varying lengths. They introduce the In-context Autoencoder (ICAE), which compresses lengthy contexts into a fixed number of memory buffers using a learnable encoder and a fixed decoder (Figure 8). The ICAE is pre-trained using auto-encoding and language modeling objectives and fine-tuned using instruction data. The approach achieves 4x context compression while maintaining effective conditioning for the target LLM, enabling faster and more memory-efficient inference.

2.2.4 RECOMP

In their work, (Xu et al., 2024) introduce RECOMP, an intermediary step for Retrieval-augmented Lan-

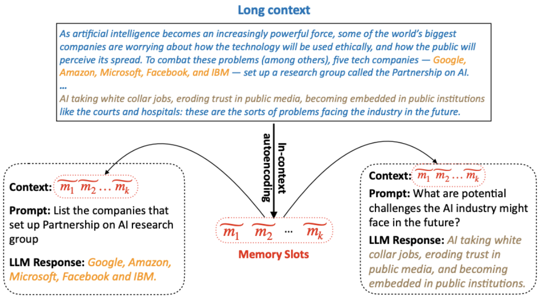

Figure 8: Condensing an extended context into a compact memory representation, which can be leveraged by the target LLM to respond to diverse prompts. (Ge et al., 2023)

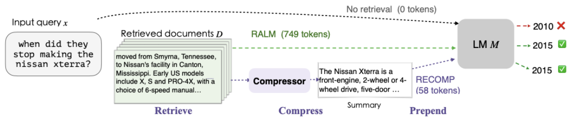

Figure 9: RECOMP's document compression technique generates a summary that serves as input to a language model, facilitating correct answer generation while minimizing encoding costs. (Xu et al., 2024)

guage Models (RALMs) (Izacard et al., 2022), (Borgeaud et al., 2022). RECOMP compresses retrieved documents into concise textual summaries before integrating them during inference, reducing computational costs and alleviating the burden on LMs to process lengthy documents. The aim is to produce summaries that balance brevity and fidelity to the original evidence documents, guiding the RALM to produce targeted outputs when the summary is used as a prefix to the input (illustrated in Figure 9). To achieve this, the authors train two types of compressors:

1. *Extractive Compressor:* This compressor filters out irrelevant sentences, retaining only the most pertinent ones from the retrieved document set.
2. *Abstractive Compressor:* This compressor produces a summary by fusing information from multiple retrieved documents.

Both compressors employ a multi-document query-based summarization approach (Xu and Lapata, 2020), summarizing evidence documents concerning the input query. The authors develop training strategies that maximize performance on the target task to guarantee accurate output. Contrastive learning is employed to train the extractive compressor enabling it to select key sentences effectively, while the abstractive compressor is distilled (West et al., 2021) from a large language model (like GPT-3 or GPT-4), achieving strong summarization performance. This approach holds promise for enhancing the efficiency and efficacy of RALMs.

2.3 Retrievers

The retriever (Chase, 2017-) is an interface that processes an unstructured query and returns a curated list of documents in response. Contextual compression aims to address the challenges of retrieval by compressing the retrieved context to only include relevant information. In this context, "compressing" encompasses both condensing the content of individual documents and eliminating irrelevant documents altogether. The Contextual Compression Retriever uses a *base retriever* and a *Document Compressor* to process queries. The base retriever retrieves the initial documents, which are then passed through the Document Compressor to shorten the list of documents by either reducing the contents of individual documents or excluding entire documents altogether.

2.3.1 LLMChainExtractor

In this approach, the base retriever is wrapped with a *ContextualCompressionRetriever*. Additionally, an *LLMChainExtractor* serves as the base compressor. The *LLMChainExtractor* iterates over the initially retrieved documents and extracts only the relevant content for the given query. It achieves this by making an additional LLM call for each retrieved document and summarizing the relevant information

2.3.2 EmbeddingsFilter

Making an additional LLM call for each retrieved document can be both costly and slow. However, the *EmbeddingsFilter* offers a more economical and faster alternative. By embedding both the documents and the query, it selectively returns only those documents that exhibit sufficiently similar embeddings to the query. This approach optimizes retrieval efficiency while maintaining relevance.

2.3.3 DocumentCompressorPipeline

The DocumentCompressorPipeline allows a seamless combination of multiple compressors in a sequence. Alongside these compressors, we can incorporate *BaseDocumentTransformers* into our pipeline. Unlike contextual compressors, these transformers don't alter the content significantly but perform specific transformations on a set of documents. For instance, *TextSplitters* can divide documents into smaller segments, while the *EmbeddingsRedundantFilter* identifies and filters out redundant documents based on embedding similarity. This modular approach enhances flexibility and adaptability in document processing. e.g.

- *Splitter:* create small chunks
- *Redundant filter:* remove similar docs — embedded
- *Relevant filter:* relevant to query

3 Metrics and Benchmarks

3.1 Metrics

Evaluating language model inference efficiency involves considering various metrics that capture different performance aspects, including accuracy, zero-shot capabilities, compression ratio, and inference time. Within the framework of RAG-based solutions, the "Triad of Metrics" [2] - Groundedness, Context Relevance, and Answer Relevance - are also employed for evaluation. Achieving satisfactory performance across these metrics helps ensure that the language model application is reliable and free from hallucinations.

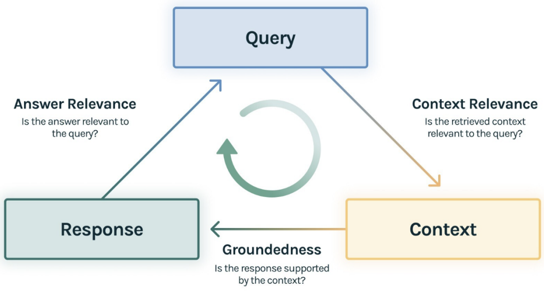

Figure 10: RAG-Triad

3.1.1 Compression Ratio

The compression ratio measures the reduction in size from the original uncompressed context to the compressed context. A higher compression ratio means that the compression is more efficient, as it achieves a greater reduction in size while preserving the context's coherence.

[2]RAG Triad (Figure 10): `https://www.trulens.org/trulens_eval/getting_started/core_concepts/rag_triad/`

3.1.2 Inference Time

Inference time, also known as latency, measures how long it takes for a Large Language Model (LLM) to process input data and generate responses. This metric is crucial for real-world applications that require quick handling of user queries or processing of large data volumes in real-time.

3.1.3 Context Relevance

In RAG applications, the first step is retrieval, and it's crucial to ensure that the retrieved context chunks are relevant to the input query. Irrelevant information in the context can lead to hallucinations in the LLM's answer. To evaluate context relevance, the structure of the serialized record can be analyzed.

3.1.4 Groundedness

After retrieving the context, an LLM transforms it into an answer. However, LLMs can sometimes stray from the facts and generate responses that are not entirely accurate. To ensure the groundedness of the application, the response can be broken down into individual claims and verified by searching for supporting evidence within the retrieved context.

3.1.5 Answer Relevance

Furthermore, our response must still effectively address the original question. We can assess this by evaluating the relevance of the final response to the user's input.

3.1.6 Others

RAG evaluation also encompasses four key abilities that reflect the model's adaptability and efficiency: noise robustness, negative rejection, information integration, and counterfactual robustness (Chen et al., 2024), (Liu et al., 2023b). The model's quality scores are heavily influenced by its ability to leverage these capabilities in diverse challenges and complex scenarios:

1. *Noise Robustness:* This metric gauges a model's capacity to distinguish between relevant and irrelevant documents, even when the latter are tangentially related to the question.
2. *Negative Rejection:* The metric measures a model's capacity to recognize when the retrieved documents are insufficient to answer a question, and to withhold a response accordingly.
3. *Information Integration:* Information integration tests a model's proficiency in combining

relevant information from multiple documents to provide well-informed answers to challenging questions.

4. *Counterfactual Robustness:* Counterfactual robustness measures a model's skill in identifying and ignoring flawed or misleading information in documents, regardless of its awareness of potential errors.

In brief, context relevance and noise robustness are crucial for evaluating the retrieval process, while answer groundedness, answer relevance, negative rejection, information integration, and counterfactual robustness are vital for assessing the quality of generated text.

3.2 Benchmarks and Datasets

The primary objective of these benchmarks and datasets is to assess the trade-offs between compressed and uncompressed contexts in terms of effectiveness, efficiency, and accuracy, covering a broad range of NLP tasks and applications.

3.2.1 Common Benchmarks and Datasets

RAG's primary function revolves around answering questions, encompassing various formats such as single-hop and multi-hop queries, multiple-choice options, and domain-specific inquiries, as well as lengthy scenarios that leverage RAG's capabilities. Moreover, RAG is constantly evolving to tackle additional tasks, including extracting relevant information, generating conversational dialogue, and searching for code snippets, documentations and even interpreting them. For more details, refer to the study by (Gao et al., 2023).

4 Challenges and Future Directions

4.1 More advanced Methods

Research on contextual compression for LLMs is still in its early stages. While previous studies have shown compressed contexts, they still lag behind uncompressed contexts in terms of performance. By exploring more advanced compression methods tailored for LLMs, we can potentially bridge this performance gap and enhance the performance of uncompressed contexts.

4.2 Performance-Size Trade-offs

Previous research highlights the importance of balancing LLM performance with context size, considering hardware limitations and practical constraints. Despite its significance, the theoretical and empirical foundations of this trade-off remain poorly understood. Future investigations should focus on conducting exhaustive examinations to drive the creation of sophisticated compression techniques that can meet the demands of increasingly complex data sets, enabling researchers to create tailored methods that effectively navigate the design space and optimize performance.

4.3 Dynamic Contextual Compression

Contemporary compression approaches still utilize manual compressors, such as retrievers, which often require an empirical methodology driven by input data or task specifications. This can be a practical hindrance to adoption, especially in scenarios like context distillation, where finding suitable student templates within computational constraints can be time-consuming and require multiple trials.

4.4 Explainability

Compressing pre-trained language models can make them hard to understand (lacking explainability). To fix this, using explainable compression methods can help make models more interpretable, easier to evaluate, and more reliable in real-life scenarios.

5 Conclusion

This in-depth analysis explores the domain of contextual compression techniques, with a focus on their application to LLMs. Our study encompasses a broad range of compression methods, evaluation metrics, and benchmark datasets, providing a comprehensive understanding of the field. By examining the complexities of contextual compression, we identify the key challenges and opportunities that arise in this area. As research in this field continues to advance, the development of specialized methodologies tailored to the needs of LLMs is crucial for unlocking their full potential across various domains. This survey aims to serve as a valuable resource, providing a detailed overview of the current landscape and encouraging further investigation into this vital topic.

Limitations

While this survey provides a comprehensive overview of contextual compression techniques for large language models, there are several limitations to acknowledge. Firstly, the field of contextual compression is rapidly evolving, and this survey may not capture the very latest advancements in the area. Additionally, the focus on large language models may not be representative of other types of language models or AI systems, which may have different compression requirements. Furthermore, the survey's reliance on existing evaluation metrics and benchmark datasets may not fully capture the complexities and nuances of contextual compression. Moreover, the need for advanced methodologies specifically designed for LLMs highlights the potential limitations of current approaches, which may not be scalable or effective for future LLM architectures. Finally, the survey's scope is limited to contextual compression, and future research may uncover new challenges and opportunities at the intersection of compression and other aspects of LLMs.

References

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Alan Baddeley. 1992. Working memory. *Science*, 255(5044):556–559.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*, volume 463. ACM press New York.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091.

Harrison Chase. 2017-. *LangChain*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 897–906.

Mahan Fathi, Jonathan Pilault, Pierre-Luc Bacon, Christopher Pal, Orhan Firat, and Ross Goroshin. 2023. Block-state transformer. *arXiv preprint arXiv:2306.09539*.

Weizhi Fei, Xueyan Niu, Pingyi Zhou, Lu Hou, Bo Bai, Lei Deng, and Wei Han. 2023. Extending context window of large language models via semantic compression. *arXiv preprint arXiv:2312.09571*.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. 2022. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Yunpeng Huang, Jingwei Xu, Zixu Jiang, Junyu Lai, Zenan Li, Yuan Yao, Taolue Chen, Lijuan Yang, Zhou Xin, and Xiaoxing Ma. 2023. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Huiqiang Jiang, Qianhui Wu, , Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. LongLLMLingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *ArXiv preprint*, abs/2310.06839.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Robert Krovetz and W Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)*, 10(2):115–141.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Yi Liu, Lianzhe Huang, Shicheng Li, Sishuo Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023b. Recall: A benchmark for llms robustness against external counterfactual knowledge. *arXiv preprint arXiv:2311.08147*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Jesse Mu, Xiang Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Ruhle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *ArXiv preprint*, abs/2403.12968.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pages 28043–28078. PMLR.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Learning from natural language feedback. In *ACL Workshop on Learning with Natural Language Supervision*.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *International conference on semantic computing (ICSC 2007)*, pages 363–369. IEEE.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. 2022. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *arXiv preprint arXiv:2209.15189*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Sanh Victor, Webson Albert, Raffel Colin, Bach Stephen, Sutawika Lintang, Alyafeai Zaid, Chaffin Antoine, Stiegler Arnaud, Raja Arun, Dey Manan, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Peter C Wason and J St BT Evans. 1974. Dual processes in reasoning? *Cognition*, 3(2):141–154.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. *arXiv preprint arXiv:2210.03162*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on empirical methods in natural language processing (EMNLP)*, pages 3632–3645.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. 2023. Recurrentgpt: Interactive generation of (arbitrarily) long text. *arXiv preprint arXiv:2305.13304*.