Lighter, yet More Faithful: Investigating Hallucinations in Pruned Large Language Models for Abstractive Summarization

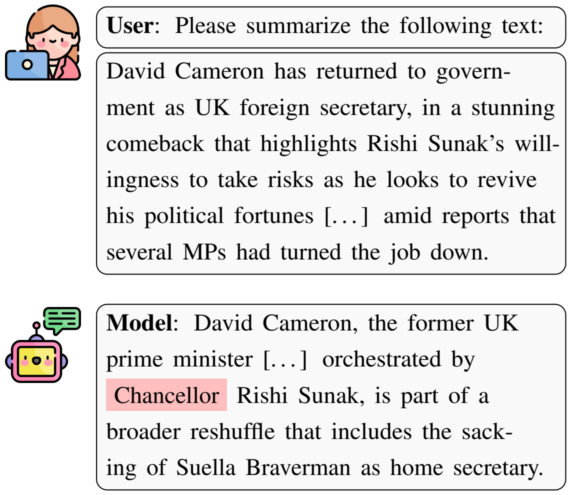
George Chrysostomou[*] **Zhixue Zhao**[*◇] **Miles Williams**[*◇] **Nikolaos Aletras**[◇]
[◇] Department of Computer Science
University of Sheffield, UK
{zhixue.zhao, mwilliams15, n.aletras}@sheffield.ac.uk

Abstract

Despite their remarkable performance on abstractive summarization, large language models (LLMs) face two significant challenges: their considerable size and tendency to hallucinate. Hallucinations are concerning because they erode the reliability of LLMs and raise safety issues. Pruning is a technique that reduces model size by removing redundant weights to create sparse models that enable more efficient inference. Pruned models yield comparable performance to their counterpart full-sized models, making them ideal alternatives when operating on a limited budget. However, the effect that pruning has upon hallucinations in abstractive summarization with LLMs has yet to be explored. In this paper, we provide an extensive empirical study on the hallucinations produced by pruned models across three standard summarization tasks, two pruning approaches, three instruction-tuned LLMs, and three hallucination evaluation metrics. Surprisingly, we find that pruned LLMs hallucinate less compared to their full-sized counterparts. Our follow-up analysis suggests that pruned models tend to depend more on the source input and less on their parametric knowledge from pre-training for generation. This greater dependency on the source input leads to a higher lexical overlap between generated content and the source input, which can be a reason for the reduction in hallucinations.[1]

1 Introduction

Abstractive summarization is the task of distilling the key information from a given document, to generate a summary consisting of text that might not appear in the original document (Cohn and Lapata, 2008; Saggion and Poibeau, 2013; Lin and Ng, 2019). Large language models (LLMs) have demonstrated strong performance on abstractive summarization (Lewis et al., 2020; Zhang et al., 2020a; Touvron et al., 2023; Almazrouei et al., 2023; Ouyang et al., 2022; OpenAI, 2023; Zhang et al., 2023). However, they face two significant challenges: their substantial size requires extensive computational resources for training and inference; and they tend to hallucinate, i.e. generate nonsensical or nonfactual contents not supported by the source document (Zhao et al., 2020; Xu et al., 2023) (see a concrete example in Figure 1).



User: Please summarize the following text:

David Cameron has returned to government as UK foreign secretary, in a stunning comeback that highlights Rishi Sunak's willingness to take risks as he looks to revive his political fortunes [. . .] amid reports that several MPs had turned the job down.

Model: David Cameron, the former UK prime minister [. . .] orchestrated by Chancellor Rishi Sunak, is part of a broader reshuffle that includes the sacking of Suella Braverman as home secretary.

Figure 1: An intuitive example of hallucinations in abstractive summarization. Highlighted text in red indicates hallucinated content (i.e. content or information not present in the source input that is not factually correct).

On the one hand, hallucinations not only undermine the performance of models but also introduce critical safety risks, ultimately eroding the trust of end users (Milintsevich and Agarwal, 2023; Tang et al., 2023; Narayan et al., 2023). For example, LLM generated summaries in the legal or health space can contain inaccurate information posing a real-life negative impact (Elaraby et al., 2023). On the other hand, large decoder-based models such as GPT-3.5 (Ouyang et al., 2022), GPT-4 (Ope-

[*] Equal contribution.

[1] We publicly release our the code: https://github.com/casszhao/PruneHall

nAI, 2023), and Llama 2 (Touvron et al., 2023) challenge the hardware capacity of many end-users. As an indication, GPT-175B requires at least five NVIDIA A100 GPUs with 80GB of memory each for half-precision (FP16) inference (Frantar and Alistarh, 2023). Pruning is a technique that enables efficient inference by removing unnecessary weights to create a sparse model (Wang et al., 2020b) with little performance degradation. Pruned models appear as attractive alternatives to full-sized models for abstractive summarization in cases of limited access to compute.

In abstractive summarization, hallucinations are a thoroughly studied subject (Cao et al., 2020; Durmus et al., 2020; Raunak et al., 2021; Narayan et al., 2023). Similarly, the effect of pruning on model performance in abstractive summarization benchmarks is also well explored (Sun et al., 2023; Xu and McAuley, 2023; Zhu et al., 2023a). However, the relationship between model pruning and hallucinations has yet to be explored. With the appeal of cost reduction and comparable downstream performance of pruned models, it is also important to establish how trustworthy their generated summaries are (i.e. if pruned models hallucinate more, the same, or less than their original counterparts).

To this end, we empirically investigate hallucinations of pruned models across three LLMs, two state-of-the-art pruning methods, three summarization datasets, and three hallucination evaluation metrics. Surprisingly, our results show that pruned models hallucinate less compared to their full-sized counterparts, which challenges "the bigger the better" stereotype (Touvron et al., 2023; OpenAI, 2023). To understand this phenomenon, we further investigate the impact of different sparsity levels on hallucination patterns. Our analysis shows that hallucinations are reduced with increasing model sparsity regardless of the pruning method. Furthermore, our results suggest that pruning encourages the model to rely more on the source input when generating text, resulting in summaries that are more lexically similar to the source input.

2 Related Work

2.1 Hallucinations in Summarization

In abstractive summarization, the model is expected to generate a concise summary of the source document input. However, prior work observed that abstractive summarization models tend to generate hallucinatory content that is not based on or cannot be entailed from the source document (Vinyals and Le, 2015; Rohrbach et al., 2018; Cao et al., 2018; Maynez et al., 2020; Raunak et al., 2021; Falke et al., 2019; Maynez et al., 2020; Chen et al., 2022). For example, Falke et al. (2019) found that 25% of the model generated summaries contain hallucinated content. On the other hand, automatic summary quality evaluation metrics such as ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020b) do not correlate with hallucinations in summaries (Zhou et al., 2021). For instance, (Zhou et al., 2021) show that even if a summary contains a large amount of hallucinatory content, it can still achieve a high ROUGE score. This has opened up new research focusing on developing approaches to detect and evaluate hallucinations (Zhou et al., 2021; Durmus et al., 2020; Guerreiro et al., 2023); and how to mitigate them (Xiao and Wang, 2021; Choubey et al., 2023; King et al., 2022).

Hallucination evaluation metrics can broadly be split into three categories: (a) entailment-based, (b) question-answering (QA), and (c) text-generation based. Entailment-based methods (Kryscinski et al., 2020; Laban et al., 2022) typically use pre-trained neural entailment models. These models return the entailment score between the source document (premise) and the generated summary (hypothesis). The higher the entailment score, the more consistent a summary is with respect to the input. Question-answering methods (Wang et al., 2020a; Deutsch et al., 2021; Durmus et al., 2020), decompose the task of finding hallucinations to a question answering problem. For example, Faithfulness Evaluation with Question Answering (FEQA) (Durmus et al., 2020) first identifies declarative sentences in a summary, extracts entities as gold-label answers and uses a trained question generator to create questions for these answers. A pre-trained QA model then goes over the source document to extract answers for the generated questions. Extracted entities from the summary that do not match model answers are considered as hallucinations. Finally, text-generation based methods use off-the-shelf models to quantify the risk of hallucinations (Yuan et al., 2021; Son et al., 2022). For example, HaRiM (Son et al., 2022) uses the log-likelihood of a reference-free decoder model to evaluate hallucinations in a summary at the token level.

Previous work has only focused on evaluating hallucinations in summarization using full-sized models (i.e. models that have not been pruned). To the best of our knowledge, no previous work has evaluated hallucinations in the context of pruned models, which we introduce in the following section.

2.2 Pruning Large Language Models

Model compression is the task of reducing the memory footprint of a model (Ganesh et al., 2021). Pruning is a popular technique that removes redundant weights from the model (LeCun et al., 1989). Weights may be removed individually (unstructured pruning), according to defined blocks (semi-structured pruning), or in relation to model components (structured pruning) (Blalock et al., 2020; Mishra et al., 2021; Zhu et al., 2023b).

As the size of LLMs surpasses billions of parameters, pruning techniques that require re-training become impractical. Instead, post-training model compression aims to reduce model size using only a small calibration dataset (Hubara et al., 2021). In this setting, Frantar and Alistarh (2022) define the layer-wise compression problem to create a compressed version of a given layer that functions as closely as possible to the original. State-of-the-art post-training pruning techniques include SparseGPT (Frantar and Alistarh, 2023) and Wanda (Sun et al., 2023). SparseGPT introduces an efficient solution for layer-wise compression in the context of LLM pruning, relying upon an iterative weight update process using Hessian inverses, inspired by OBC (Hassibi et al., 1993). Wanda further improves upon computational efficiency, enabling pruning in a single forward pass. This is achieved using a simple pruning criterion, consisting of the element-wise product between the weight magnitude and the norm of input activations without requiring any weight updates.

Pruning methods, like Wanda and SparseGPT, have been used to prune decoder-only LLMs, showing the ability to maintain zero-shot performance in summarization tasks compared to full-sized LLMs. However, it remains unclear how pruning affects the faithfulness of LLMs (i.e. to what extent they hallucinate).

3 Methodology

Our objective is to compare hallucinations in generated summaries from a full-sized model ($\mathcal{M}$) with its pruned counterpart ($\mathcal{M}_p$). Given a source document $\mathcal{D}$, we generate summaries T and T_p using $\mathcal{M}$ and $\mathcal{M}_p$, respectively. For both type of models, we use identical prompts. We then evaluate summaries T and T_p in terms of the hallucinations that they contain. We consider whether the pruned model $\mathcal{M}_p$ performs better in comparison to the original model $\mathcal{M}$ with respect to the degree of hallucinations they produce.

3.1 Full-sized Models

We use the following decoder-only instruction-tuned models $\mathcal{M}$:[2]

Llama2 (Touvron et al., 2023) Llama 2 is a decoder-only model pre-trained on two trillion tokens. We use the 7-billion and 13-billion parameter models of the further fine-tuned version of Llama 2, i.e. Llama 2-Chat. Llama 2-Chat was instruction-tuned on 1 million human-annotated examples.

Falcon (Almazrouei et al., 2023) We use the 7-billion parameter instruction-tuned model. Falcon was pre-trained on 1.5 trillion tokens and instruction-tuned on 250 million tokens.

3.2 Pruning Methods

We obtain pruned models $\mathcal{M}_p$ using the following methods: (a) **SparseGPT** (Frantar and Alistarh, 2023); and (b) **Wanda** (Sun et al., 2023). We opt from using magnitude pruning (Hagiwara, 1994; Han et al., 2015) as it was previously shown to yield sub-par performance to SparseGPT and Wanda. Comparing the performance of pruning methods on summarization is out of the scope of our paper.

We select a 2:4 structured sparsity pattern (providing a total of 50% sparsity) following previous work (Frantar and Alistarh, 2023; Sun et al., 2023), enabling efficient hardware acceleration on GPUs (Mishra et al., 2021).[3] We use the same sparsity level in all experiments unless otherwise stated. Following Sun et al. (2023), we use the exact same set of calibration data for pruning Wanda and SparseGPT, which consists of 128 sequences sampled from C4 (Raffel et al., 2020).

3.3 Summarization Datasets

For our experiments, we include three subset datasets from the SummaC Benchmark (Laban et al., 2022): (1) **FactCC** (Kryscinski et al., 2020),

[2] We opt for experimenting with instruction-tuned models due to their wide adoption, impressive performance in abstractive summarization (Zhang et al., 2023) and large size, which makes them ideal candidates for pruning.

[3] 2:4 structured sparsity refers to zeroing two values of each four value contiguous block in the weight matrix.

(2) **Polytope** (Huang et al., 2020), and (3) **SummEval** (Fabbri et al., 2021). For each dataset, we take the source documents of the testing split and remove duplicated documents, resulting in 311, 634, and 100 source documents for FactCC, Polytope, and SummEval, respectively.

3.4 Generating Summaries with Instruction-tuned LLMs

For the summary generation, we use greedy search as our decoding strategy (i.e. selecting the token with the highest probability) for better reproducibility. We set the maximum number of new tokens to 25% of the original input tokens for each individual input. For prompting the models, we use the instruction templates of Touvron et al. (2023) for Llama 2 models and Almazrouei et al. (2023) for Falcon. As the generation performance of decoder-only LLMs can vary with differing prompts, to control for variability we run all experiments across three distinct prompts. Each prompt is hence a different way to instruct the model towards summarizing a document, under the same template. The prompts along with their templates for each model can be seen in Appendix A. As each data point corresponds to three generated summaries, we evaluate across all three generations by averaging the scores.

3.5 Evaluating Summarization Quality

We evaluate the quality of generated summaries using a subset of the ROUGE family of metrics (Lin, 2004) and BERTScore (Zhang et al., 2020b). From ROUGE, we use two n-gram overlap metrics ROUGE-1 and ROUGE-2 and also ROUGE-L, the longest sequence overlap metric. We also report model perplexity.

3.6 Evaluating Hallucinations

To evaluate the degree of hallucinations in the generated summaries, we use three standard automatic hallucination evaluation metrics that do not require gold-reference summaries or human evaluation: HaRiM, SummaC$_{\text{Conv}}$ and SummaC$_{\text{ZS}}$.

Model	Pruning Method	Perplexity
Falcon-7B	-	8.526
	SparseGPT	11.186
	Wanda	11.583
Llama-7B	-	5.360
	SparseGPT	6.648
	Wanda	6.533
Llama-13B	-	5.283
	SparseGPT	6.410
	Wanda	6.557

Table 1: Perplexity of original and pruned models on the held-out set of WikiText (lower is better).

HaRiM (Son et al., 2022) This is a reference-free (i.e. does not rely on the source sequence) hallucination metric for abstractive summarization with encoder-decoder models. The main assumption is that in such architectures, the generation of the next token is largely dependent on the previously generated (decoder-only) text. Consequently, they re-use the same model starting with an empty encoder input, to obtain the decoder probabilities. For a single source sequence, HaRiM is then computed as follows, where L is the sequence length, p_{s2s} the predicted probability of the model given the source input and p_{lm} the generated token probability:

$$\text{HaRiM} = \frac{1}{L}\sum_{i=0}^{L}(1 - p_{s2s})(1 - (p_{s2s} - p_{lm}))$$

SummaC (Laban et al., 2022) This metric uses an off-the-shelf entailment model to assess the consistency between a source document and a generated summary. First, the document and summary are split into sentences, with the document sentences (N) being the hypothesis and the generated summary sentences (K) being the premise. The second step is to create an $K \times N$ matrix of entailment scores from the pre-trained model. A generated sentence with a low entailment score, with any of the document sentences is a potential hallucination. Finally, two approaches are used for obtaining the overall consistency score. **SummaC$_{\text{ZS}}$** obtains the row-wise maximum entailment score, which leads to a vector E of size K. In vector E each element can be interpreted as the hallucination score for each sentence in the summary. E is finally averaged to obtain a single summary hallucination score. **SummaC$_{\text{Conv}}$** obtains vector E by using a convolutional model that passes a trained kernel over each row K, to get a single score. Similarly, vector E contains the hallucination score for each sentence, which can then be averaged for the summary hallucination score.

Data	Pruning	Falcon 7B ROUGE-Av	Falcon 7B BERTScore	Llama 7B ROUGE-Av	Llama 7B BERTScore	Llama 13B ROUGE-Av	Llama 13B BERTScore
FactCC	-	0.27 (0.10)	0.88 (0.03)	0.22 (0.09)	0.85 (0.01)	0.20 (0.08)	0.84 (0.01)
	SparseGPT	0.29 (0.10)	0.87 (0.03)	0.23 (0.09)	0.85 (0.01)	0.21 (0.08)	0.84 (0.01)
	Wanda	0.29 (0.10)	0.87 (0.03)	0.23 (0.09)	0.85 (0.02)	0.22 (0.08)	0.84 (0.01)
Polytope	-	0.24 (0.09)	0.85 (0.02)	0.23 (0.08)	0.83 (0.01)	0.22 (0.08)	0.83 (0.01)
	SparseGPT	0.22 (0.10)	0.83 (0.03)	0.25 (0.08)	0.83 (0.01)	0.23 (0.08)	0.83 (0.01)
	Wanda	0.24 (0.11)	0.83 (0.03)	0.25 (0.09)	0.83 (0.01)	0.23 (0.08)	0.83 (0.01)
SummEval	-	0.27 (0.07)	0.88 (0.01)	0.25 (0.08)	0.85 (0.01)	0.23 (0.07)	0.85 (0.01)
	SparseGPT	0.29 (0.08)	0.87 (0.02)	0.26 (0.07)	0.85 (0.01)	0.24 (0.07)	0.85 (0.01)
	Wanda	0.29 (0.08)	0.86 (0.05)	0.26 (0.07)	0.85 (0.01)	0.24 (0.07)	0.85 (0.01)

Table 2: Summary generation quality measured using ROUGE-1/2/L and BERTScore, across three datasets and three original models (-) and their two pruned counterparts (SparseGPT and Wanda). For clarity, we show the average of the ROUGE scores (ROUGE-Av) with the full stack of results available in Appendix B. For all summary generation quality metrics, higher is better.

For all metrics, a higher score indicates that there is a lower prevalence of hallucinations in the generated summary.

3.7 Implementation Details

We use pre-trained models from the Hugging Face library (Wolf et al., 2020). We run all experiments on a single NVIDIA A100 GPU.

4 Results

4.1 Model Perplexity

Before comparing hallucinations between full-sized models and their pruned counterparts, we first measure model perplexity. Table 1 includes the reproduced perplexity of pruned models on the held-out dataset of WikiText (Merity et al., 2017) as per previous work (Sun et al., 2023). As expected, pruned models result in higher perplexity scores, with both pruning methods performing comparably, corroborating findings by Sun et al. (2023).

4.2 Summarization Performance

We also evaluate model performance in generating summaries. Whilst the zero-shot summarization performance of instruction tuned models is not the focus of this work, a comparable performance between pruned and non-pruned models allows a fair comparison of their hallucination quality and better justifies our study (i.e. practitioners might prefer using pruned models to full-sized if their downstream performance is similar). We therefore measure summary generation performance with ROUGE-1/2/L and BERTScore, across three datasets, three full-sized models and their two pruned counterparts (SparseGPT and Wanda). For brevity, Table 2 shows the average of the ROUGE scores (ROUGE-Av) with the full stack of results available in Appendix B. For both summarization metrics, higher is better.

We first observe that the summarization quality of pruned models does not degrade, with either of the pruning methods tested (SparseGPT and Wanda) across metrics and datasets. We see that the full-sized model slightly outperforms the pruned models in BERTScore when using Falcon 7B, whilst remaining consistent with both Llama models. For example, with Llama 7B and Polytope all models record a BERTScore of 0.83. On the contrary, when using the lexical overlap metrics (ROUGE-Av) we observe that pruned models have a slight lead over their full-sized counterparts. For example, in SummEval we observe higher ROUGE-Av scores for both pruning methods across all models tested (e.g. 0.29 ROUGE-Av with Falcon 7B pruned with Wanda or SparseGPT, versus 0.27 with the original model).

4.3 Comparing Model Hallucinations

Table 3 shows the hallucination prevalence of three full-sized models and their pruned counterparts, tested across FactCC, Polytope and SummEval. Hallucination prevalence is measured using HaRiM, SummaC$_{\text{Conv}}$ and SummaC$_{\text{ZS}}$, where a higher score indicates the model hallucinates less (i.e. lower hallucination prevalence).

Pruned models hallucinate less. The almost unanimous green cells in Table 3 indicate that, in general, pruned models hallucinate less compared

Data	Model	Pruning	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{\text{ZS}}$
FactCC	Falcon 7B	-	4.36 (0.83)	0.57 (0.21)	0.6 (0.25)
		SparseGPT	**4.77 (0.85)**	**0.67 (0.19)**	**0.73 (0.23)**
		Wanda	**4.83 (0.83)**	**0.68 (0.18)**	**0.72 (0.24)**
	Llama 7B	-	3.44 (0.66)	0.33 (0.09)	0.26 (0.16)
		SparseGPT	**3.56 (0.74)**	**0.38 (0.11)**	**0.35 (0.19)**
		Wanda	**3.56 (0.74)**	**0.38 (0.11)**	**0.36 (0.19)**
	Llama 13B	-	2.86 (0.49)	0.32 (0.08)	0.24 (0.15)
		SparseGPT	**2.96 (0.53)**	**0.37 (0.10)**	**0.34 (0.17)**
		Wanda	**2.95 (0.58)**	**0.37 (0.11)**	**0.34 (0.18)**
Polytope	Falcon 7B	-	3.67 (0.63)	0.51 (0.15)	0.63 (0.21)
		SparseGPT	**4.02 (0.71)**	**0.58 (0.17)**	**0.72 (0.21)**
		Wanda	3.70 (0.78)	0.50 (0.15)	0.63 (0.23)
	Llama 7B	-	3.32 (0.54)	0.35 (0.10)	0.41 (0.19)
		SparseGPT	**3.48 (0.55)**	**0.42 (0.13)**	**0.50 (0.18)**
		Wanda	3.36 (0.61)	**0.41 (0.13)**	**0.51 (0.19)**
	Llama 13B	-	2.82 (0.49)	0.34 (0.09)	0.39 (0.18)
		SparseGPT	**2.97 (0.58)**	**0.41 (0.12)**	**0.51 (0.19)**
		Wanda	**3.09 (0.60)**	**0.40 (0.11)**	**0.50 (0.18)**
SummEval	Falcon 7B	-	4.35 (0.52)	0.55 (0.18)	0.63 (0.20)
		SparseGPT	**4.67 (0.67)**	**0.65 (0.17)**	**0.76 (0.17)**
		Wanda	4.52 (0.73)	0.59 (0.19)	**0.70 (0.23)**
	Llama 7B	-	3.77 (0.65)	0.34 (0.11)	0.33 (0.17)
		SparseGPT	**3.99 (0.52)**	**0.39 (0.13)**	**0.43 (0.18)**
		Wanda	**3.99 (0.56)**	**0.39 (0.14)**	**0.45 (0.18)**
	Llama 13B	-	3.14 (0.49)	0.34 (0.10)	0.34 (0.17)
		SparseGPT	3.27 (0.53)	**0.39 (0.11)**	**0.43 (0.18)**
		Wanda	3.19 (0.52)	**0.40 (0.13)**	**0.45 (0.19)**

Table 3: Model hallucination comparison, averaging over the three prompts (higher is better). Green cells indicate that the pruned model hallucinates less on average compared to its non-pruned counterpart, whilst red the opposite. **Bold** values denote that the difference between the pruned model and the original are significant (paired t-test; $p < 0.05$)

to their non-pruned counterparts. The hallucination degree of pruned models is significantly lower compared to the original models in 46 out of 54 total comparisons (bold values in green cells). For example, with Llama 2 7B in SummEval, we observe significantly higher scores with all three metrics, with SummaC$_{\text{ZS}}$ recording a 10 point increase with SparseGPT (0.43 from 0.33) and 12 with Wanda (0.45 from 0.33). Additionally, in the two instances where a pruned model records lower scores (i.e. hallucinates more) compared to the non-pruned counterparts, this difference is not statistically significant (e.g. 0.51 using SummaC$_{\text{Conv}}$ with original Llama 7B compared to 0.50 with Wanda pruning in Polytope).

These findings seem counter-intuitive, considering that pruned models typically perform albeit comparably, slightly worse in perplexity and downstream tasks. We hypothesize that *by removing unused parameters, we potentially remove some of the model's parametric knowledge (i.e. knowledge obtained via pre-training and fine-tuning). This perhaps "forces" the model to rely more on the source document during the summary generation and in turn reduces hallucinations.* We examine this in more detail in Section 5.

SparseGPT is more consistent. Whilst both pruned models hallucinate less than their counterpart full-sized models, there are subtle differences between the two pruning methods tested. Results suggest that SparseGPT in particular is more consistent compared to Wanda, recording significantly better results compared to the full model in 26 out of 27 comparisons (in contrast Wanda records significantly better results in 20 out of 27 comparisons). For example, with Llama 7B in Polytope, SparseGPT records higher scores compared to Wanda in all metrics. A possible reason behind

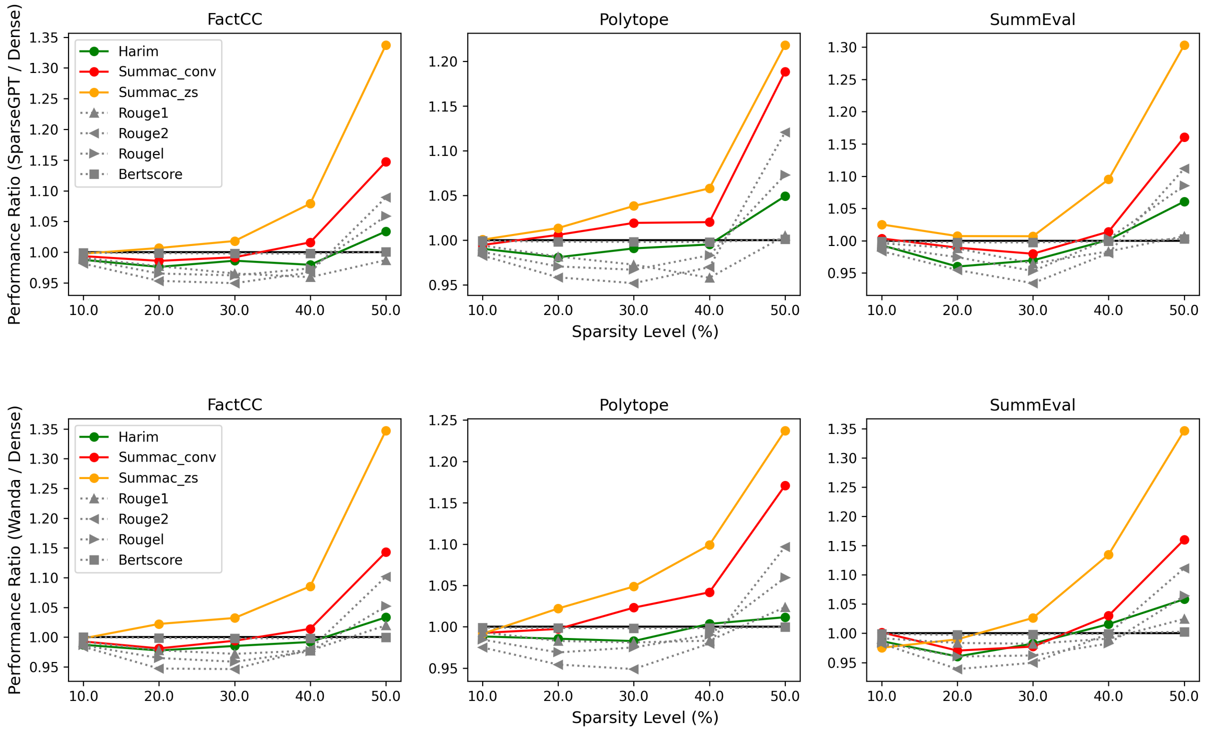


Figure 2: Performance comparison by ratio between a pruned Llama 7B model and its full-sized counterpart across 5 sparsity levels, 3 hallucination performance metrics and 4 generation performance metrics (grey dotted lines). Ratios higher than 1 indicate better performance compared to the baseline full-sized model (black-colored horizontal line).

this could be that SparseGPT updates the model weights during pruning, which might affect model behavior.

5 Impact of Sparsity on Hallucinations

We further conduct a quantitative analysis, to better understand previous results and test our working hypothesis: pruning a model removes parametric knowledge, thereby requiring the model to attend more towards the source input during generation. In turn, we assume this is the culprit behind the reduced level of hallucinations in pruned models.

For this purpose, in Figure 2 we compare the performance of Llama 7B pruned to various sparsity thresholds (using Wanda and SparseGPT) against its full-sized counterpart.[4] We increase the level of sparsity sequentially in 10% increments and compare the summarization performance and hallucination scores of the resulting model against the original. The comparison considers three hallucination performance metrics (lines with circled markers) and four generation performance metrics (dotted lines in grey). A ratio higher than one, indicates that the pruned Llama performs better in the corresponding metric compared to the original.

Hallucinations reduce as sparsity increases. Results from Figure 2 show that in most cases, hallucinations reduce with increasing sparsity.

At initial sparsity levels (20% and 30%), the hallucination performance of pruned models reduces slightly compared to the original model. However, hallucinations in pruned model summaries remain comparable with those from the full-sized model (3% difference at the lowest point with Wanda and HaRiM in SummEval with 20% sparsity). On the contrary, we observe differences between the pruned and the original model when increasing sparsity. For example in Polytope with SparseGPT, the model with 50% sparsity records an increase of 21% with SummaC$_{\text{ZS}}$, 19% with SummaC$_{\text{Conv}}$, and 5% with HaRiM compared to the full-sized model. These findings suggest that *increasing sparsity does indeed appear to reduce hallucinations in summaries generated by the model.*

Pruned models generate summaries that have greater lexical similarity to the input. Observ-

[4] We also observe similar findings for Falcon 7B and Llama 2 13B. Results are available in Appendix C.

		FactCC			Polytope			SummEval	
	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{\text{ZS}}$	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{\text{ZS}}$	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{\text{ZS}}$
SparseGPT									
ROUGE-1	0.48 (0.33)	0.17 (0.75)	0.05 (0.93)	0.55 (0.26)	0.43 (0.40)	0.30 (0.57)	0.69 (0.13)	0.64 (0.17)	0.52 (0.29)
ROUGE-2	**0.97 (0.00)**	**0.94 (0.01)**	**0.89 (0.02)**	**0.98 (0.00)**	**0.93 (0.01)**	**0.88 (0.02)**	**0.97 (0.00)**	**0.96 (0.00)**	**0.91 (0.01)**
ROUGE-L	**0.97 (0.00)**	**0.92 (0.01)**	**0.86 (0.03)**	**0.98 (0.00)**	**0.92 (0.01)**	**0.88 (0.02)**	**0.97 (0.00)**	**0.97 (0.00)**	**0.93 (0.01)**
Wanda									
ROUGE-1	**0.90 (0.02)**	0.81 (0.05)	0.70 (0.12)	0.75 (0.09)	0.79 (0.06)	0.68 (0.14)	**0.90 (0.01)**	**0.94 (0.01)**	**0.83 (0.04)**
ROUGE-2	**0.98 (0.00)**	**0.95 (0.00)**	**0.88 (0.02)**	**0.86 (0.03)**	**0.90 (0.02)**	**0.83 (0.04)**	**0.97 (0.00)**	**0.98 (0.00)**	**0.90 (0.01)**
ROUGE-L	**0.97 (0.00)**	**0.91 (0.01)**	**0.83 (0.04)**	**0.86 (0.03)**	**0.92 (0.01)**	**0.85 (0.03)**	**0.92 (0.01)**	**0.96 (0.00)**	**0.85 (0.03)**

Table 4: Pearson's correlation (p-values in the brackets) across all sparsity levels between hallucination (HaRiM, SummaC$_{\text{Conv}}$, Summac$_{\text{ZS}}$) and lexical overlap metrics (ROUGE 1/2/L) for the Llama 2 7B model. **Bold** values indicate significant correlations ($p < 0.05$).

ing lexical-based (ROUGE) and semantic-based (BERTScore) summary quality metrics across sparsity levels, the outcomes are mixed. In almost all cases for each pruning method, BERTScore scores remain comparable to the full-sized model (close to 1) up to 50% sparsity. This shows that the summaries generated by pruned models remain semantically similar to those from their original counterparts.

However, there is a stark contrast when looking at the ROUGE-based metrics. Pruned models record lower ROUGE-based scores across the 20% and 30% sparsity levels but then increase substantially beyond the original model's scores at 50% sparsity. Surprisingly, it appears that at the point where lexical-based metrics in pruned models surpass their original counterparts, we also observe a large jump in hallucination metric scores. For example with FactCC and Wanda above 40% sparsity, SummaC$_{\text{ZS}}$ jumps from 1.10 to 1.35 whilst SummaC$_{\text{Conv}}$ from 1.02 to 1.15. As summaries from pruned models remain semantically comparable to the source input with full-sized models, their *increasing lexical overlap with the source document indicates that pruned models focus more on the input document to generate a summary.*

Lexical-based metrics correlate with Hallucination metrics. To better understand the relationship between lexical overlap and hallucination metrics, we conduct a pairwise comparison across all sparsity levels between hallucination metrics (HaRiM, SummaC$_{\text{Conv}}$, Summac$_{\text{ZS}}$) and lexical overlap metrics (ROUGE 1/2/L). The rationale here is that strong correlations can reinforce our argument that: the reduced hallucinations are potentially due to the increasing lexical overlap between the source document and the generated summary by the pruned models. For this purpose in Table 4, we show the average Pearson correlation values across data points (with p-values in the brackets). **Bold** values indicate significant correlations between lexical overlap metrics and hallucination scores across sparsity levels.[5]

Our results show that there are strong correlation signals across all three datasets, all three hallucination metrics with both pruning methods in ROUGE-2 and ROUGE-L. That is, generated summaries with greater lexical overlap with their source documents, e.g. higher ROUGE scores, are less likely to contain hallucinations.[6] This corroborates findings drawn from the human annotation task by Durmus et al. (2020) which showed that summaries that are more lexically similar to the source input are less likely to contain hallucinations.

Our overall results suggest that *higher lexical-overlaps could be responsible for reduced hallucinations, whilst increasing sparsity appears responsible for the increasing lexical-overlaps.*

6 Conclusion

In this work, we explore how hallucinations in abstractive summarization differ when LLMs are pruned. Our experimental setup consisted of two state-of-the-art pruning methods (Wanda and SparseGPT) applied to three instruction-tuned LLMs (Falcon 7B and Llama 7B & 13B). We measured hallucinations across three datasets using three established metrics. Surprisingly, our results show that as models are pruned to higher sparsity, they hallucinate less. Our analysis further shows that increased sparsity potentially encour-

[5] We also find similar outcomes with Falcon and Llama 13B, see Appendix C for all results.

[6] For ROUGE-1 we also observe positive correlations of varying degrees and strength, however, they are not significant.

ages a model to attend more to the source input for generation, offering a possible explanation for the fewer hallucinations. Our findings are supported by increasing lexical overlaps between the source input and the summary, which in turn correlate with the patterns observed in hallucination metrics across sparsity levels.

Future work includes evaluating more models and model sizes. Additionally, we plan to explore the relationship between hallucination prevalence and model pruning in other tasks such as open-book question answering (Mihaylov et al., 2018; Ciosici et al., 2021) and machine translation (Guzmán et al., 2019; Wang and Sennrich, 2020; Dale et al., 2023). Finally, whilst these metrics offer a good baseline for measuring hallucinations, we would like to expand our experiments to include human annotations.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? In *Proceedings of Machine Learning and Systems*, volume 2, pages 129–146.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Xiuying Chen, Mingzhe Li, Xin Gao, and Xiangliang Zhang. 2022. Towards improving faithfulness in abstractive summarization. In *Advances in Neural Information Processing Systems*, volume 35, pages 24516–24528. Curran Associates, Inc.

Prafulla Kumar Choubey, Alex Fabbri, Jesse Vig, Chien-Sheng Wu, Wenhao Liu, and Nazneen Rajani. 2023. CaPE: Contrastive parameter ensembling for reducing hallucination in abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10755–10773, Toronto, Canada. Association for Computational Linguistics.

Manuel Ciosici, Joe Cecil, Dong-Ho Lee, Alex Hedges, Marjorie Freedman, and Ralph Weischedel. 2021. Perhaps PTLMs should go to school – a task to assess open book and closed book QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6104–6111, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK. Coling 2008 Organizing Committee.

David Dale, Elena Voita, Loic Barrault, and Marta R. Costa-jussà. 2023. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–50, Toronto, Canada. Association for Computational Linguistics.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Mohamed Elaraby, Yang Zhong, and Diane Litman. 2023. Towards argument-aware abstractive summarization of long legal opinions with summary reranking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7601–7612, Toronto, Canada. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems*, volume 35, pages 4475–4488. Curran Associates, Inc.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. 2021. Compressing large-scale transformer-based models: A case study on BERT. *Transactions of the Association for Computational Linguistics*, 9:1061–1080.

Nuno M. Guerreiro, Pierre Colombo, Pablo Piantanida, and André Martins. 2023. Optimal transport for unsupervised hallucination detection in neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13766–13784, Toronto, Canada. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Masafumi Hagiwara. 1994. A simple and effective method for removal of hidden units and weights. *Neurocomputing*, 6(2):207–218. Backpropagation, Part IV.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

B. Hassibi, D.G. Stork, and G.J. Wolff. 1993. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. 2021. Accurate post training quantization with small calibration sets. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4466–4475. PMLR.

Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. 2022. Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 555–571, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hui Lin and Vincent Ng. 2019. Abstractive summarization: A survey of the state of the art. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9815–9822.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Kirill Milintsevich and Navneet Agarwal. 2023. Calvados at MEDIQA-chat 2023: Improving clinical note generation with multi-task instruction finetuning. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 529–535, Toronto, Canada. Association for Computational Linguistics.

Asit K. Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *CoRR*, abs/2104.08378.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional generation with a question-answering blueprint. *Transactions of the Association for Computational Linguistics*, 11:974–996.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Horacio Saggion and Thierry Poibeau. 2013. *Automatic Text Summarization: Past, Present and Future*, pages 3–21. Springer Berlin Heidelberg, Berlin, Heidelberg.

Seonil (Simon) Son, Junsoo Park, Jeong-in Hwang, Junghwa Lee, Hyungjong Noh, and Yeonsoo Lee. 2022. HaRiM$^+$: Evaluating summary quality with hallucination risk. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 895–924, Online only. Association for Computational Linguistics.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models. *CoRR*, abs/2306.11695.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR*, abs/1506.05869.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020a. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020b. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Canwen Xu and Julian McAuley. 2023. A survey on model compression and acceleration for pretrained language models. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J. Martindale, and Marine Carpuat. 2023. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11:546–564.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249, Online. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023a. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

Yunqi Zhu, Xuebing Yang, Yuanyuan Wu, and Wensheng Zhang. 2023b. Parameter-efficient fine-tuning with layer pruning on medical report summarization and medical dialogue generation. *arXiv preprint arXiv:2305.08285*.

A Prompt templates

We prompt the model to generate a summary with the following different prompts:

- "Your task is to summarize concisely and truthfully. Summarize the input below:
  Input: [document]
  Single paragraph summary: "
- "Summarize the article below in a single paragraph:
  Input: [document]
  Summary: "
- "Please write a short summary for the text below:
  Input: [document]
  Summary: "

B Full Generation Results

Dataset	Model	Pruning Method	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
FactCC	Falcon 7B	-	0.31 (0.08)	0.23 (0.10)	0.26 (0.09)	0.88 (0.03)
		SparseGPT	0.33 (0.07)	0.26 (0.11)	0.29 (0.09)	0.87 (0.03)
		Wanda	0.33 (0.08)	0.26 (0.11)	0.29 (0.09)	0.87 (0.03)
	Llama 7B	-	0.30 (0.07)	0.16 (0.06)	0.20 (0.06)	0.85 (0.01)
		SparseGPT	0.29 (0.08)	0.18 (0.07)	0.22 (0.07)	0.85 (0.01)
		Wanda	0.30 (0.07)	0.18 (0.07)	0.21 (0.07)	0.85 (0.02)
	Llama 13B	-	0.28 (0.07)	0.14 (0.05)	0.19 (0.06)	0.84 (0.01)
		SparseGPT	0.28 (0.08)	0.16 (0.06)	0.20 (0.06)	0.84 (0.01)
		Wanda	0.29 (0.07)	0.16 (0.06)	0.20 (0.06)	0.84 (0.01)
Polytope	Falcon 7B	-	0.29 (0.08)	0.21 (0.09)	0.23 (0.09)	0.85 (0.02)
		SparseGPT	0.27 (0.08)	0.17 (0.10)	0.22 (0.09)	0.83 (0.03)
		Wanda	0.30 (0.08)	0.18 (0.11)	0.23 (0.09)	0.83 (0.03)
	Llama 7B	-	0.31 (0.07)	0.18 (0.06)	0.21 (0.06)	0.83 (0.01)
		SparseGPT	0.31 (0.07)	0.20 (0.07)	0.23 (0.07)	0.83 (0.01)
		Wanda	0.32 (0.07)	0.20 (0.07)	0.23 (0.07)	0.83 (0.01)
	Llama 13B	-	0.29 (0.07)	0.16 (0.06)	0.20 (0.06)	0.83 (0.01)
		SparseGPT	0.30 (0.07)	0.18 (0.06)	0.21 (0.07)	0.83 (0.01)
		Wanda	0.30 (0.07)	0.18 (0.06)	0.22 (0.07)	0.83 (0.01)
Summeval	Falcon 7B	-	0.32 (0.06)	0.24 (0.08)	0.27 (0.07)	0.88 (0.01)
		SparseGPT	0.33 (0.06)	0.25 (0.08)	0.28 (0.07)	0.87 (0.02)
		Wanda	0.34 (0.05)	0.26 (0.09)	0.29 (0.07)	0.86 (0.05)
	Llama 7B	-	0.33 (0.05)	0.21 (0.05)	0.24 (0.05)	0.85 (0.01)
		SparseGPT	0.33 (0.05)	0.21 (0.05)	0.24 (0.05)	0.85 (0.01)
		Wanda	0.33 (0.05)	0.21 (0.06)	0.24 (0.05)	0.85 (0.01)
	Llama 13B	-	0.31 (0.04)	0.17 (0.04)	0.22 (0.04)	0.85 (0.01)
		SparseGPT	0.32 (0.04)	0.18 (0.04)	0.23 (0.04)	0.85 (0.01)
		Wanda	0.32 (0.04)	0.18 (0.05)	0.23 (0.04)	0.85 (0.01)

Table 5: Generation performance for original sized models and compressed

C Across Sparsity Additional Results

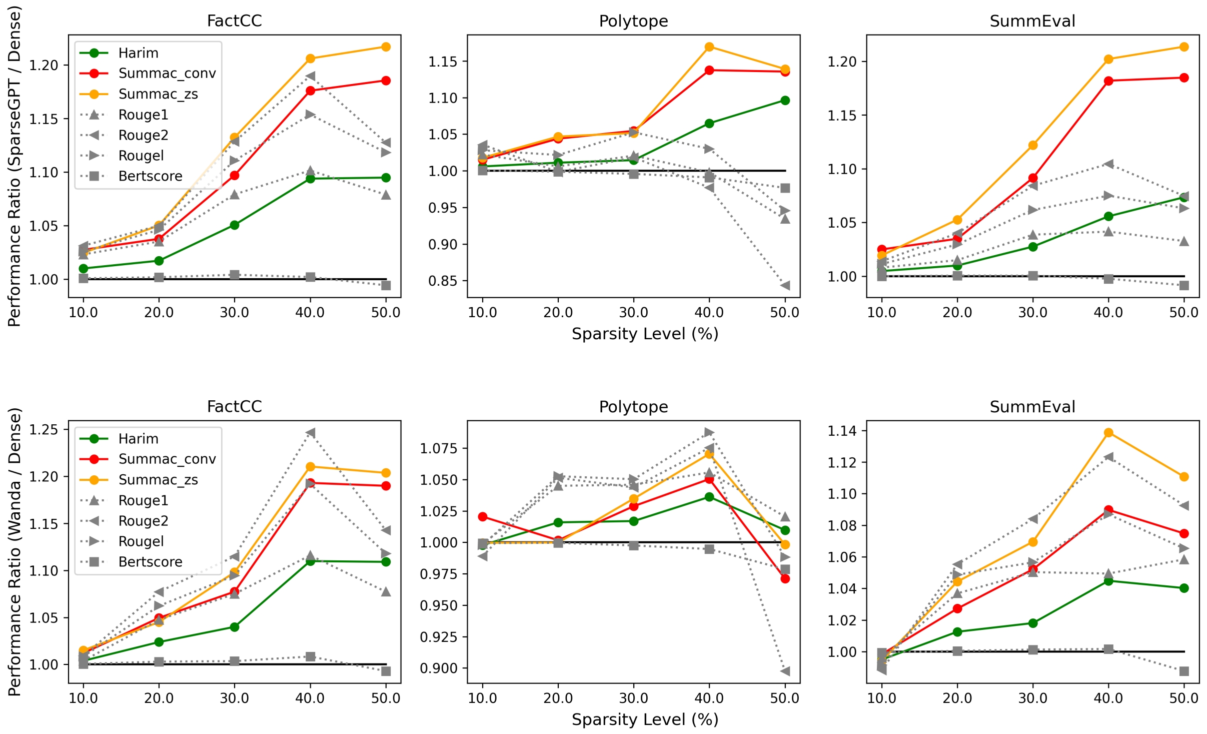


Figure 3: Performance comparison by ratio between a pruned Falcon 7B model and its full-sized counterpart across five sparsity levels, three hallucination performance metrics and four generation performance metrics (grey dotted lines). Ratios higher than 1 indicate better performance compared to the baseline full-sized model (black colored horizontal line).

	FactCC			Polytope			SummEval		
	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{ZS}$	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{ZS}$	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{ZS}$
					SparseGPT				
ROUGE-1	**0.93 (0.01)**	**0.93 (0.01)**	**0.95 (0.00)**	**-0.85 (0.03)**	-0.65 (0.16)	-0.58 (0.23)	0.80 (0.05)	**0.87 (0.02)**	**0.91 (0.01)**
ROUGE-2	**0.93 (0.01)**	**0.92 (0.01)**	**0.94 (0.01)**	**-0.89 (0.02)**	-0.72 (0.11)	-0.64 (0.17)	0.81 (0.05)	**0.89 (0.02)**	**0.91 (0.01)**
ROUGE-L	**0.95 (0.00)**	**0.94 (0.00)**	**0.96 (0.00)**	-0.64 (0.17)	-0.37 (0.47)	-0.30 (0.57)	**0.86 (0.03)**	**0.92 (0.01)**	**0.95 (0.00)**
					Wanda				
ROUGE-1	**0.89 (0.02)**	**0.90 (0.01)**	**0.91 (0.01)**	**0.92 (0.01)**	0.47 (0.35)	0.72 (0.11)	**0.88 (0.02)**	**0.91 (0.01)**	**0.90 (0.02)**
ROUGE-2	**0.91 (0.01)**	**0.92 (0.01)**	**0.93 (0.01)**	0.58 (0.23)	**0.82 (0.04)**	0.64 (0.17)	**0.95 (0.00)**	**0.98 (0.00)**	**0.97 (0.00)**
ROUGE-L	**0.92 (0.01)**	**0.93 (0.01)**	**0.94 (0.01)**	**0.90 (0.01)**	0.75 (0.08)	**0.83 (0.04)**	**0.93 (0.01)**	**0.96 (0.00)**	**0.96 (0.00)**

Table 6: Pearsons correlation statistic (with p-values in the brackets) across all sparsity levels between hallucination metrics (HaRiM, SummaC$_{\text{Conv}}$, Summac$_{ZS}$) and lexical overlap metrics (ROUGE 1/2/L) for Falcon 7B model. **Bold** values indicate significant correlations (p-value < 0.05).

	FactCC			Polytope			SummEval		
	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{ZS}$	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{ZS}$	HaRiM	SummaC$_{\text{Conv}}$	SummaC$_{ZS}$
					SparseGPT				
ROUGE-1	**0.89 (0.02)**	0.67 (0.15)	0.55 (0.26)	0.81 (0.05)	0.50 (0.31)	0.42 (0.41)	**0.89 (0.02)**	0.75 (0.09)	0.46 (0.36)
ROUGE-2	**0.93 (0.01)**	**0.82 (0.04)**	0.73 (0.10)	**0.95 (0.00)**	0.76 (0.08)	0.69 (0.13)	**0.95 (0.00)**	**0.81 (0.05)**	0.54 (0.26)
ROUGE-L	**0.95 (0.00)**	**0.85 (0.03)**	0.77 (0.08)	**0.96 (0.00)**	0.80 (0.05)	0.73 (0.10)	**0.91 (0.01)**	0.76 (0.08)	0.47 (0.34)
					Wanda				
ROUGE-1	**0.95 (0.00)**	0.79 (0.06)	0.72 (0.10)	**0.91 (0.01)**	**0.82 (0.05)**	0.70 (0.12)	**0.93 (0.01)**	**0.86 (0.03)**	0.78 (0.07)
ROUGE-2	**0.96 (0.00)**	**0.81 (0.05)**	0.75 (0.09)	**0.96 (0.00)**	**0.86 (0.03)**	0.75 (0.08)	**0.97 (0.00)**	**0.88 (0.02)**	0.77 (0.07)
ROUGE-L	**0.95 (0.00)**	**0.84 (0.03)**	0.79 (0.06)	**0.93 (0.01)**	**0.90 (0.02)**	0.80 (0.06)	**0.95 (0.00)**	**0.90 (0.01)**	0.79 (0.06)

Table 7: Pearsons correlation statistic (with p-values in the brackets) across all sparsity levels between hallucination metrics (HaRiM, SummaC$_{\text{Conv}}$, Summac$_{ZS}$) and lexical overlap metrics (ROUGE 1/2/L) for Llama 2 13B model. **Bold** values indicate significant correlations (p-value < 0.05).

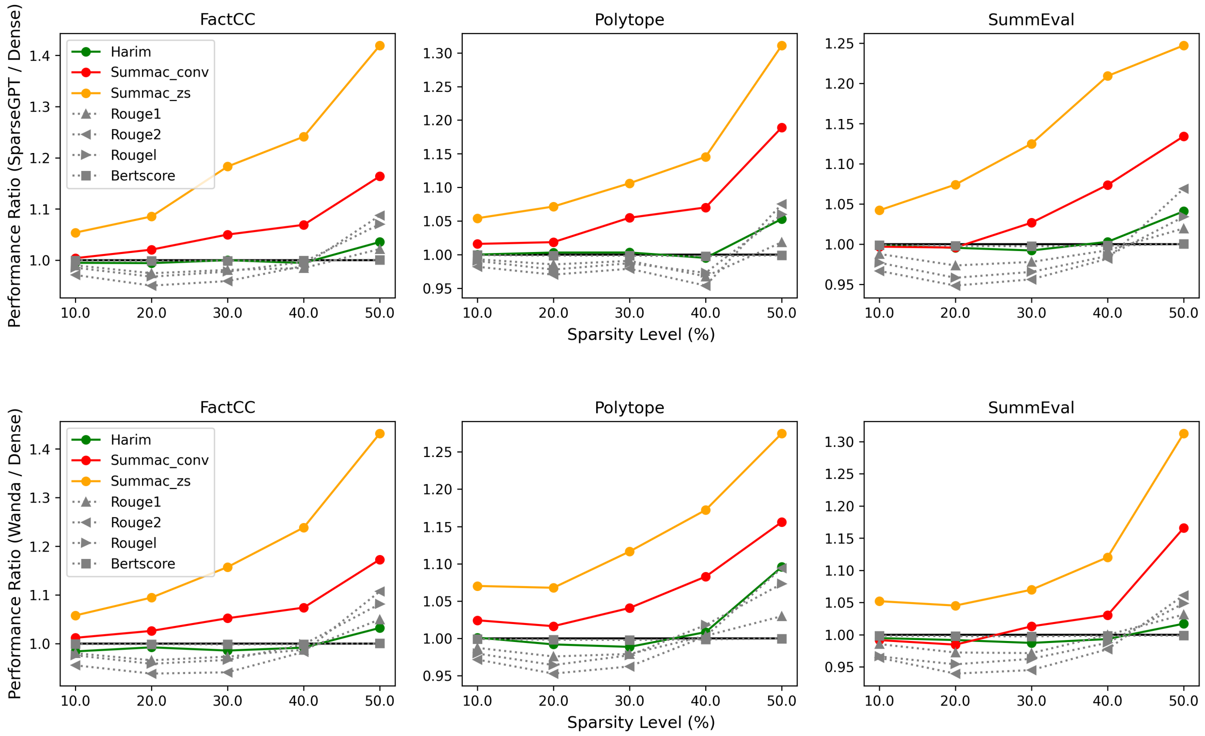


Figure 4: Performance comparison by the ratio between a pruned Llama 13B model and its full-sized counterpart across five sparsity levels, three hallucination performance metrics, and four generation performance metrics (grey dotted lines). Ratios higher than 1 indicate better performance compared to the baseline full-sized model (black-colored horizontal line).