Abstractive Text Summarization and Improvements on Factuality

Advanced NLP and Deep Learning (Autum 2023, KSANLPD1KU)

Ernests Lavrinovics
ernl@itu.dk

Bence Zoltan Balazs
beba@itu.dk

Abstract

Text summarization proves to be an effective and efficient method for swiftly conveying crucial information. Large Language Models (LLM) hold immense potential in simplifying this task for a diverse audience. However, existing research reveals a challenge—many LLMs exhibit factual inconsistencies when generating summary texts. In this project, we embark on fine-tuning a Llama2-7bn text summarization model. Our focus is on evaluating the performance of a fine-tuned model and investigating how integration of Natural Language Inference (NLI) into the cost function can enhance factuality performance. We conduct training and evaluation on the XSum news summary dataset and employ multiple factuality consistency metrics on our final results. We find that in the current iteration of our models, there is no substantial difference between the base model and the NLI enhanced version. We propose to use better NLI model in the cost function and to train our model longer to mitigate this issue. Furthermore, we notice that the fine-tuned tend to predict parts from the source text, which is likely to be the reason that our model performs worse than the state-of-the-art models. Thus we propose an extra penalty term on taking entire sentences from the source document.

Our project source code is available on GitHub[1].

1 Introduction

Textual information contains a wealth of knowledge, part of which is in the form of news articles, novels, academic and legal documents, and others. In order to empower the information-seekers to process information faster and more efficiently, textual summarization can be used as an effective tool for this task (El-Kassas et al., 2021). Generally, literature (Widyassari et al., 2022; El-Kassas et al., 2021; Alomari et al., 2022) identifies two main approaches for summarization, *extractive* and *abstractive*.

Extractive summarization consists of contents that are purely extracted from a given document and consist of phrases directly taken for it. It is considered that extractive summarization is reaching its maturity and that research is shifting towards abstractive or real-time summarization (Widyassari et al., 2022; El-Kassas et al., 2021).

Abstractive text summarization aims at extracting all relevant information from a given source document and condenses it into a shorter, coherent version, while retaining the key points and meaning of the original text (Alomari et al., 2022; El-Kassas et al., 2021). Abstractive summarization approach essentially paraphrases specific passages, therefore delivering a more concise and fluent summarization (Koh et al., 2022).

Common issue outlined in literature for abstractive summarization are a tendency to have factual inconsistencies, also referred to as *hallucinations* (Maynez et al., 2020; Chen et al., 2022; Scialom et al., 2021; Pagnoni et al., 2021; Ji et al., 2023; Augenstein et al., 2023). Two possible causes for this may be overreliance on the underlying language model to generate fluent but inadequate words, or failure to understand the core of the input text (Chen et al., 2022). Therefore, problems with factual consistency can become a bottleneck for the technology to be deployed in production environments. Factual inconsistencies can also lead to drastic impacts if mistakes are made when summarizing news, medical records, and other sources (Adams et al., 2023; Xie et al., 2023).

Using *natural language inference* (NLI) as an aid for ensuring factual consistency can be beneficial (Falke et al., 2019; El-Kassas et al., 2021), and has already been experimented with in previous works (Zablotskaia et al., 2023; Wan and Bansal, 2022). Considering the release of a recent language model *Llama 2* (Touvron et al., 2023), we propose

[1] https://github.com/ernlavr/llamarizer

to investigate the following research questions.

1. What is the performance of a Llama 2 series model for news text summarization?
2. To what extent does Llama 2 series model suffer from *hallucinations*?
3. How will additional NLI information presented during training affect the factuality of a summary?

2 Related Work

One of the earlier works for using NLI as part of summary generation performs *re-ranking* (Falke et al., 2019). This means that a summarization model generates multiple outputs, which are then compared for entailment with respect to the source document. The authors perform two experiments: a sentence-by-sentence and summary level comparisons of entailment against the source document. The most entailing output is selected as the final system output. As for the data, the authors crowdsource labels of correct/incorrect/unclear for individual sentences, and they report summary misplacement (incorrect replaced as correct) and improvements (vice versa) in the reranking.

The NLI models were trained either on SNLI (Bowman et al., 2015) or MultiNLI (Williams et al., 2018), summaries were generated from the CNN/DM dataset (See et al., 2017).

A more recent work (Zablotskaia et al., 2023) leveraged more and different summarization tasks that included news, forum conversations and dialogues. The authors also use a re-ranking based approach, building upon BRIO(Liu et al., 2022) methodology, expanding it by directly incorporating NLI information as part of their fine-tuning loss function. The NLI information is generated by annotating candidate summaries using a fine-tuned T5-11B model on the Adversarial NLI dataset, and is used as part of the training loss function by scaling the entailment scores. Additionally, the authors also use a regularization term that prevents the model from over-optimizing towards high NLI scores, since the authors outline a positive correlation between length and the NLI in their dataset. The authors report Rouge, NLI as well as length and coverage scores.

FactPEGASUS incorporates factuality awareness as part of the pre-training and fine-tuning tasks. They use three separate modules for pre- and post-processing of the model's output (Wan and Bansal, 2022). The authors use contrastive learning as part of their training methodology meaning that the model is exposed to both factual and non-factual pairs.

3 Data

The core direction of the project is *news summary*. Therefore, XSum (Narayan et al., 2018) is used to fine-tune a summarization model. XSum is a popular dataset used by many other summarization works (Wan and Bansal, 2022; Liu et al., 2022) which collects 226'711 news reports from BBC, accompanied also by a single sentence professionally written summary. The dataset covers a wide range of topics such as sports, politics, business, and others. Due to computational constraints, we limit the data points to 50000 for train set and 5000 for validation split, and we skip data points that are longer than **512** tokens due to BERT-based NLI module constraint. Therefore, our final data point count is, **28301** for training and **2840** for validation.

Additionally, for fine-tuning an NLI module, we use XSum with hallucination annotations as per (Maynez et al., 2020), the factuality subsplit is taken from[2]. The dataset is relatively small, containing summaries generated by 4 different models, see Table 1. It is worth noting that most of the summaries are annotated by three separate workers indicating a *factual/notFactual* label, and each of the model is summarizing the same data point. Overall, the dataset authors note that the dataset contains 500 unique documents summarized.

BertS2S	PtGen	TConvS2S	TranS2S
1376	1375	1422	1424

Table 1: Datapoint amount for each model in XSum-Factuality

Additionally, since training is done on *causal language modeling* task, we also run experiments with preprocessed data points by nesting the input within a prompt, depicted in Listing 1. This is further elaborated in the Methodology.

4 Methodology

The popular NLP paradigm of fine-tuning to downstream tasks is the foundation of the work's methodology. Two Llama2-7bn models are trained using causal language modeling task: a baseline without

[2]https://huggingface.co/datasets/xsum_factuality

Prompt: Summarize this article '<INPUT>'; Summary: <MODEL_OUTPUT>

Listing 1: Prompt in which the <INPUT> placeholder is replaced with a source document that is followed by target summary. <MODEL_OUTPUT> is where the model starts generating its output.

any NLI information, and another using an NLI module, we refer to them as *Llamarizer-Baseline* and *Llamarizer-NLI*. The system diagram in Figure 1 depicts the *Llamarizer-NLI* training loop. The baseline version consists of removed NLI module and removed label information from the loss function. Training is performed on a single A100 GPU using the HuggingFace framework.

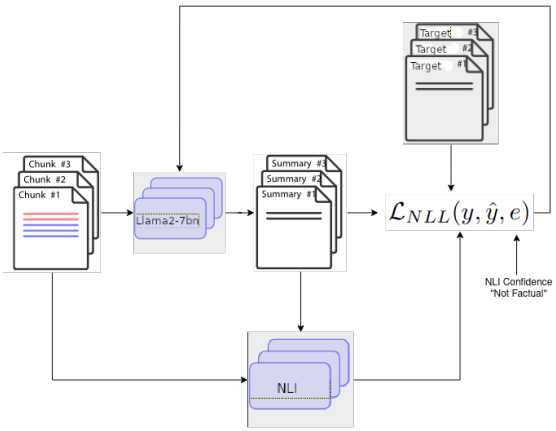

Figure 1: Summarizer with NLI enhancement

4.1 Summary Model: Llamarizer-Baseline

Due to computational and memory constraints, it is not possible to fine-tune a native Llama2-7bn model. Therefore, we quantize the model in a 4-bit float mode using BitsAndBytes[3] and train an adapter model using LoRA (Hu et al., 2021) thus also decreasing the amount of total trainable parameters from 7bn to 4'194'304.

Additionally, as shown in Listing 1 and Table 2, we perform a parameter-sweep with and without nesting the input in a prompt. The input consists of a concatenation of *source document* and *target summary*, although the *target summary* portion of the input is zero'd-out within the attention mask to prevent the model from cheating.

The model is trained using cross entropy loss,

$$\mathcal{L}(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) \quad (1)$$

[3]https://huggingface.co/blog/4bit-transformers-bitsandbytes

where y is the target summary, $\hat{y}$ is the model output, y_i and $\hat{y}_i$ are the i-th elements of y and $\hat{y}$ respectively.

To determine the optimal hyperparameters for the baseline summarizer, a parameter-sweep was performed with the following configuration as per Table 2 where the best performing configuration is highlighted in bold. Additionally, a linear learning-rate scheduler is employed with a 0.1 warmup ratio and warmup steps spanning over half an epoch and training is performed over 2 epochs. For a more detailed outlook, please refer to Appendix B.

Model	**Llama2-7bn**	Llama2-7bn-chat
Added Prompt	**Yes**	No
Batch Size	**16**	32
Learning Rate	**1e-3**	1e-4

Table 2: Grid-search training for optimum baseline summarizer. Configuration with the lowest eval loss highlighted in **bold**.

4.2 NLI Model

NLI model was seperately fine-tuned before deployment in conjunction with the summarizer. Fine-tuning was done as a sequence classification task and trained using cross-entropy loss. As noted in Section 3, the NLI XSum-Factuality dataset is heavily imbalanced. Therefore, parameter sweeps were performed with two seperate strategies for balancing the target classes, namely minority class upsampling and class weights. Table 3 summarizes all the parameters used in a parameter sweep with the most optimal configuration (lowest loss) highlighted in bold.

M.	BERT	XLM	**D.Bert**	D.Bert-MNLI
B.S.	**2**	4	8	16
W.D.	**0.001**	0.01	0.1	0.2
Cl. W.	**True**	False	-	-
Cl. Up.	True	**False**	-	-
LR	**1e-6**	1e-5	1e-4	-

Table 3: NLI model parameter sweep. DistilBERT is abbreviated as D.BERT. Row names (top to bottom): Models (M), Batch Size (B.S.), Weight Decay (W.D.), Class Weights (Cl.W), Class Upsampling (Cl.Up.), Learning Rate (LR)

NLI parameter sweeps were performed with 15 training epochs using the HuggingFace Transformers framework. For a more detailed outlook, refer to Appendix B.

4.3 Summary Model: Llamarizer-NLI

We used the best performing set of hyperparameters to repeat the training with NLI enhancements. The same training methodology as described in Section 4.1 to the NLI enhanced version, with the only modification being the use of a different loss function, and no further parameter sweeps. The loss function for this version is defined as follows:

$$\mathcal{L}(y, \hat{y}, z) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) + \frac{1}{n}\sum_{j=1}^{n} z_j \quad (2)$$

Here, z represents the NLI confidence scores computed over an entire batch for the *not-factual* label, and n denotes the batch size. The objective is to guide the summary model towards generating output that minimizes the *not-factual* label.

4.4 Evaluation Metrics

During evaluation, to extract the pure summary of the model, the corresponding *input* span is removed from the model's output. This is done by using *input_ids* array as a mask. We evaluate the final output of the models using multiple metrics to measure the factuality of the generated summaries. The *maximum new token* count for summary generation is set to be twice the size of the reference summary.

4.4.1 Rouge

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores are a set of metrics commonly used to evaluate the quality of automatic summaries by assessing the overlap between generated summaries and reference summaries. The R1, R2, and RougeL scores specifically focus on the overlap of unigram, bigram, and longest common subsequence (LCS) respectively. These metrics provide valuable insights into the informativeness and fluency of generated summaries. However, it's important to note that ROUGE scores primarily capture linguistic overlap and coherence, and they may not inherently measure the factuality or correctness of the information presented in the summaries.(Maynez et al., 2020)

4.4.2 FactCC

FactCC (Kryscinski et al., 2019) is a weakly supervised method to identify factual inconsistencies between the original text and the generated summaries. FactCC is a BERT based method trained on three tasks namely: Identifying whether sentences remain factually consistent after transformation, extracting a span in the source documents to support the consistency prediction, extracting a span in the summary sentence that is inconsistent if such inconsistency exists. The advantage of this technique compared to NLI data sets that the training data is automatically generated from the source text using a series of transformation. It made it possible to create large amount of data without human annotators. It has been shown that FactCC outperforms other actual factuality checking models trained on NLI datasets and it highly correaltes with human judgement (Pagnoni et al., 2021).

4.4.3 ANLI

We also applied a model for factual consistency metric, that is a pre-trained for natural language inference (NLI). It is trained on a combination of four NLI datasets: SNLI, MNLI, FEVER-NLI, and ANLI (R1, R2, R3). The model is based on the RoBERTa-Large architecture and achieves state-of-the-art performance on a variety of NLI benchmarks.(Nie et al., 2019)

4.4.4 SummaC

SummaC detects inconsistencies between a source document and summaries (Laban et al., 2022). We used the convolutional version of the SummaC model, first calculating NLI score for each sentence pair between the source document and the summary text. Then, it bins all the NLI scores for each summary sentence and uses a 1-d convolutional layer to calculate a consistency score for the summary. The advantage of this technique is that it is relatively lightweight and shows state-of-the-art results on factual consistency datasets.

4.4.5 BARTScore

BARTScore is a similarity based, unsupervised evaluation metric for generated text that assesses its quality from different perspectives, including informativeness, fluency, and factuality(Yuan et al., 2021). It is defined using the weighted log probability of one text y given another text x. BARTScore is calculated as the sum of the product of weights and log probabilities over all tokens in the generated sequence

5 Results

5.1 Training of Llamarizer-Baseline

During training, we compute only a small subset of metrics due to computational restraints. Partic-

ularly the loss function and Rouge1, Rouge2 and RougeL metrics on the extracted output span as described in 4.4, as well as the same metrics on the whole span prefixed *Raw_*. Figure 2 shows the evaluation loss during training. We achieve a final loss of **2.235**. Please see our *Weights&Bias* run[4] for full details of the particular configuration.

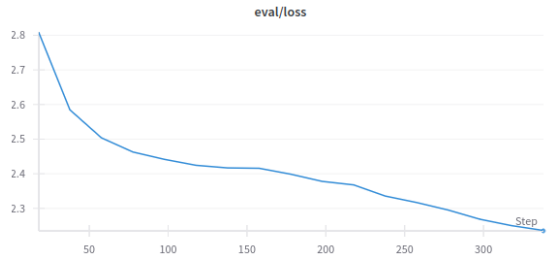

Figure 2: Llamarizer evaluation loss. Model configuration reflects Table 2. X-axis reflects evaluation step, Y axis reflects loss.

5.2 Fine-tuning NLI Module

The final NLI module was picked as per the best configuration described in the parameter sweep in Table 3. The following Table 4 describes our final NLI results on XSum-Factuality dataset.

Accuracy	F1	Loss
0.64	0.63	0.68

Table 4: Results of fine-tuning DistilBERT on XSum-Factuality dataset

Please refer to our Weights&Bias run[5] for a more detailed and full overview of results.

5.3 Training of Llamarizer-NLI

We report a combined *loss* of NLI and the summarizer, as well as the NLI *not-factual* label confidence separately in Figures 3 and 4 respectively. Note that we use the *not-factual* label confidence score as part of the Llamarizer-NLI loss function as described in Section 4.3. Our final evaluation loss is **2.679**.

Please refer to our Weights&Bias run[6] for a more detailed and full overview.

5.4 Evaluation of Llamarizer

Using the fine-tuned versions of the Llamarizer models we performed evaluations of the summaries

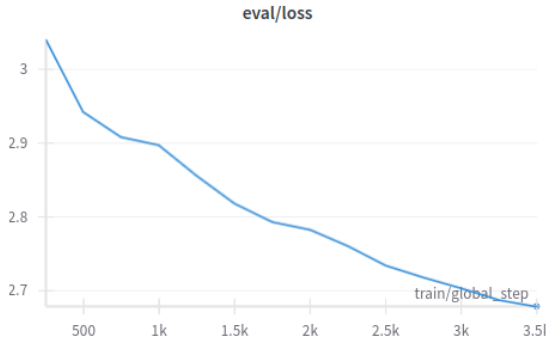

Figure 3: Evaluation loss of Llamarizer-NLI. Y-axis represents the loss and X-Axis represents evaluation step.

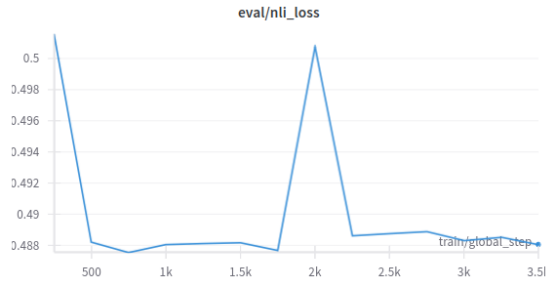

Figure 4: NLI module *not-factual* label confidence. Used as part of the Llamarizer-NLI loss function. Y-Axis represents the loss and X-Axis represents the evaluation step

using the metrics described in Section 4.4, results are shown in Table 5. Furthermore, we compare these with the results of state-of-the-art models in study (Goyal et al., 2022). We found that there is no substantial difference between the base and the NLI enhanced Llamarizer models across all the metrics. Moreover, we have found that GTP-D2 performs better for the FactCC metric. BRIO and T0 both outperform our models but with a much lower margin on the FactCC metric. There have been some good summaries predicted, such as the example in Listing 2

However, we find that our models most commonly predict the first sentence of the source an examples of this is shown in Listing 3.

6 Discussion

The training results of Llamarizer-Baseline and Llamarizer-NLI depict that the models would benefit from continued training. As per Figures 2 and 3, it is evident that the models' loss has not converged, therefore continued training could lead to improved results. Also, Figure 4 depicts that the methodology of training Llamarizer-NLI may not

[4]https://wandb.ai/ernlavr/adv_nlp2023/runs/7c8g4xcg
[5]https://wandb.ai/ernlavr/adv_nlp2023/runs/u6yzdlxz/
[6]https://wandb.ai/ernlavr/adv_nlp2023/runs/w5g89vq7

	R1	R2	RL	FactCC	ANLI	SummaC	BARTScore
Llamarizer-NLI mean	0.179	0.032	0.125	0.191	0.414	0.658	-3.699
Llamarizer-Base mean	0.181	0.034	0.127	0.179	0.417	0.656	-3.69
BRIO mean	0.497	0.260	0.410	0.203	–	–	–
T0 mean	0.442	0.207	0.358	0.222	–	–	–
GPT3-D2 mean	0.288	0.076	0.206	0.397	–	–	–
Llamarizer-NLI std	0.09	0.05	0.069	0.317	0.462	0.245	0.798
Llamarizer-Base std	0.09	0.05	0.068	0.31	0.463	0.248	0.848

Table 5: Mean and standard deviation on summarization metrics on the fine-tuned llama models. One incorporating NLI into the loss function, the other is the base model

Prediction: The Scottish Criminal Cases Review Commission has referred the case of a man jailed for 28 months after being caught with drugs at the T in the Park festival to the High Court of Justiciary.
Reference: A drug dealer caught with £870 worth of ecstasy at T in the Park will have his appeal against the length of his prison term heard at the High Court.

Listing 2: An example of a good prediction from the Llamalizer + NLI model

Source: Summarize this article: 'Nearly every bollard in Callander has been given a woolly makeover to mark the town's Winter Fest. ...';
Prediction: 'Nearly every bollard in Callander has been given a woolly makeover to mark the town's Winter Fest. ... ;
Reference: The appearance of woolly bollards in a Trossachs town has proved a major hit with visitors.

Listing 3: An example of the Llamarizer-NLI model predicting the first sentence of the source text

be sufficient for the model to optimize towards a decreased *not-factual* summary, it would be expected for the *not-factual* label confidence to gradually decrease as the training progresses.

Experimenting with different NLI weighting schemes could force the model to generate summaries with a lower *not-factual* label confidence. Although different NLI weighting schemes still may not give objectively better results, as our NLI model is comparatively weak. Table 4 shows better than *random* accuracy and F1 score, although the model itself still may not be strong enough to be reliably used within the Llamarizer-NLI loss function.

In Section 5.4 we show that our models still do not achieve as high results as the state-of-the-art models. The most likely reason is that the Llamarizers predict the first sentence of the source text as a summary. This issue could be over come by introducing a penalty term in the loss function that penalizes using entire sentences from the source document. This would increase the level of abstraction in the models.

In further work, we could also create better comparison of our models with other state-of-the-art models using more metrics. For factuality measures we could use question answering based evaluation metric such as QuestEval, QAFactEval or QAGs. For semantical similarity measurement more similarity based metrics a such as BertScore and MoverScore.

7 Concluding remarks

Our methodology proposes experiments for analyzing and improving factuality of a Llama2 series language model. Our results do not show significant improvements between the baseline and enhanced versions of the summarizer, although this could be attributed towards prematurely stopped training and a weak NLI model which is used as part of the training loss.

Our qualitative evaluation outlines that Llama2-7bn model suffers from generating parts of the source, which results in lower performance than the state-of-the-art models. Further work needs to review the training methodology by incorporating stronger factuality detection modules during training. Our work does not analyze the zero-shot performance of a vanilla Llama2 model, nor does it experiment with stronger NLI modules or analyze the performance transferability to other summarization datasets or domains.

References

Griffin Adams, Jason Zucker, and Noémie Elhadad. 2023. A meta-evaluation of faithfulness metrics for long-form hospital-course summarization. *arXiv preprint arXiv:2303.03948*.

Ayham Alomari, Norisma Idris, Aznul Qalid Md Sabri, and Izzat Alsmadi. 2022. Deep reinforcement and transfer learning for abstractive text summarization: A review. *Computer Speech & Language*, 71:101276.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2023. Factuality challenges in the era of large language models. *arXiv preprint arXiv:2310.05189*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

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

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

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*, Florence, Italy. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *CoRR*, abs/1910.12840.

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.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. *arXiv preprint arXiv:2203.16804*.

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.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

David Wan and Mohit Bansal. 2022. Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization. *arXiv preprint arXiv:2205.07830*.

Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, and De Rosal Ignatius Moses Setiadi. 2022. Review of automatic text summarization techniques methods. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1029–1046.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Qianqian Xie, Edward J Schenck, He S Yang, Yong Chen, Yifan Peng, and Fei Wang. 2023. Faithful ai in medicine: A systematic review with large language models and beyond. *medRxiv*.

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

Polina Zablotskaia, Misha Khalman, Rishabh Joshi, Livio Baldini Soares, Shoshana Jakobovits, Joshua Maynez, and Shashi Narayan. 2023. Calibrating likelihoods towards consistency in summarization models. *arXiv preprint arXiv:2310.08764*.

A Group Contributions

All other work outside the mentioned individual contributions has been contributed equally.

Ernests:

1. *Project management and infrastructure*: Initial project definition and background research, establishment of the main codebase and general workflow, summarization codebase full implementation, finalization of NLI module.

2. *Experimental work*: NLI + Summarization model experiments, fine-tunings, hyperparameter tunings, parameter sweeps, evaluation of summary initial establishment, finalizing evaluation metrics and running result computations.

3. *Writing*: Introduction, Related Work, Data, Methodology (sect 4, 4.1, 4.2, 4.3, partly 4.4), Results (5.1, 5.2, 5.3), parts of discussion and conclusions.

Bence:

1. *Implementation*: Creating factuality evaltion metrics and integrtating them into training set up.

2. *Research*:Research potential evaluation metrics and their strength and weaknesses. Evaluate our quantitative and qualitative results.

3. *Writing*: Abstract, Evaluation metrics, Methodolgy section 4.4 ,Results 5.4 , parts of discussion and concluding remarks .

B Weights and Biases: Model Training and Parameter Sweeps

Throughout the project we make use of Weights and Biases framework for our machine learning experiment management and tracking. For a better experience of viewing in-depth results, we make our project public[7], specifically for parameter sweeps refer to[8].

To view some explicit summarization examples from training, navigate to *Sweeps* -> *Select a Sweep* -> *Select a Run* -> *Navigate to Artifacts* -> Select an Artifacts Table -> View Files -> Open the Artifact JSON, e.g. [9]

[7] https://wandb.ai/ernlavr/adv_nlp2023

[8] https://wandb.ai/ernlavr/adv_nlp2023/sweeps

[9] https://wandb.ai/ernlavr/adv_nlp2023/artifacts/run_table/run-glwgz9uz-Training_Samples/v7/files/Training_Samples.table.json