Text Summarization with Contrasting Layers in the Decoder with LLaMA

Lohith Kumar Bhambore
Department of Computer Science, University of California Riverside
(Lbham003@ucr.edu)

Abstract

Contrastive learning is an approach which is gaining a lot of traction in improving the performance of large language models on different tasks. For example, it's been very good in the the field of unsupervised visual representation learning. And there have been remarkable improvements in increasing accuracy in tasks of Question Answering. The recent approach using a dynamic contrasting between layers in decoders for this task has shown remarkable improvements. In this work, we explore this method for the task of text summarization.

With the exponential growth of digital content, there's an overwhelming amount of data available. Summarization helps in digesting and extracting key insights from vast amounts of text efficiently. And it's one of the most important tasks that we would want an AI to be able to perform well. Since, it is essential across various domains where managing and understanding large volumes of textual data for efficient decision-making, information retrieval, and knowledge extraction, it is logical to have better models. We use one of the most basic large language models to test the effectiveness of this method of contrasting. Using the data set of CNN daily mail and even without a lot of pre-processing and fancy modifications we find that including this method of contrasting improves performance significantly.

The code is available at - *https://github.com/EmileS24/DoLa-ForTextSummarization*

1 Introduction

Text summarization is about extracting useful and relevant information from a large document and making them much more concise and short while also preserving the most important parts including facts, opinions, reasons and conclusion.

Since it's a big problem, of course it has been studied and explored for a long time. There have been various approaches introduced in the community so far. Some which have had huge successes and some more subtle ones. All of these works contributed to two major classification for the methods of summarization. Them being extractive summarization and abstractive summarization **(Nenkova and McKeown 2011)**

There have been many abstractive models that have shown good results with the method of having pre-trained Transformers **(Liu and Lapata 2019; Raffel et al. 2020; Dong et al. 2019; Lewis et al. 2020)**. But the main problem of them using the goal of minimizing the negative log likelihood compared to the reference summaries cannot guarantee that the summary is still carrying the factuality of the document and not introducing new information (hallucinations) in them.

In this work we are exploring the method of extractive summarization. We have seen some of the most popular and powerful enhancements in this field in the rise of models like OpenAI's ChatGPT, Google's Bard, MPT-Instruct, Flan-T5-xl and many more.

But these are huge models trained with hundreds of billions of parameters, maybe even trillions and a huge amount of data. An extractive summarization methods work better when we want to explicitly require the preservation of the orginal context of the original document.

Of course, there is also the thing that it's much easier along with the fact that the risk of the model making up things (hallucinations) are much lower.

Although it is currently unknown why hallucinations occur in Language Models, one theory is that it could be due to the goal of maximizing the likelihood, which reduces the forward KL divergences between the data distribution and the distribution from the model.

So basically it could create a situation where the model assigns probabilities greater than zero to phrases even though they don't conform to the

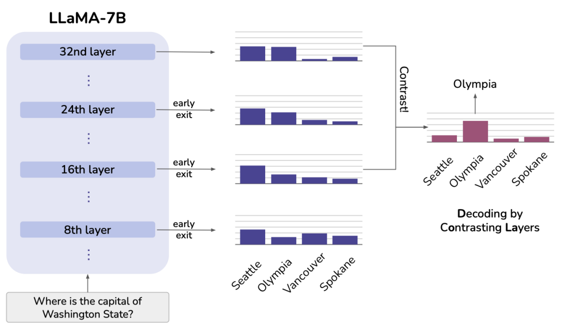

Figure 1: An Illustration of how Decoding by contrasting for Question Answering. We can see that contrasting the way the probability for Olympia increases over the layers. The whole idea is to exploit this occurrence by contrasting between these layers to increase the probabilities of the right words. **(DoLa - Yung-Sung et al. 2023)**

facts present in the source document.

Most of the time it is thought that the Large Language Models (LLMs) recognize only superficial patterns among the different examples when training without actually understanding and relating to the actual facts which are extracted from the original document.

Most LLMs use transformers as a basic block and it is believed that a transformer encodes low-level information like part-of-speech tags in the earlier layers and a lot more semantic and relational information in the layers which are close to the output layer. (**Tenney et al., 2019**).

Another work by **Dai et al., 2022** found that the "knowledge neurons" as they frame it are mostly situated at the top-layers of the BERT LLM (pre-trained).

In fact one work by **Meng et al.,** found that facts learned by an LLM can be changed by changing the some feed-forward layer in a transformer LLM which is autoregressive.

The idea proposed by **Yung-sung et al.,** is about utilizing this concept to instead increase the probabilities of real-world facts in an LLM. They started using the method of Contrasting the layers in the decoder. The key idea being that this method improves the importance of the information from the top-layers while decreasing it for the bottom layers.

Our work was to verify if this concept is limited to Question Answering where the factuality is something that is ingrained and constitutes real-world truth is still valid for the summarization task where preservation of facts is important, but it is limited to the ones presented in the original document only.

After conducting the experiment on one of the most widely accepted dataset of CNN/Daily mail 3.0 using a basic LLaMa-7b model we showed that this approach give comparatively better performance than before in all evaluation metrics used (ROUGE, BLEU and BERT scores).

Next we consider whether this method of contrasting adds a significant overhead for the LLm which basically creates a small extra latency while decoding. But that seems like a small price to pay for improved accuracy and factuality in summarizing.

2 Design

Most Large Language Models have similar skeleton for the model architecture. They consist of 3 different blocks -

- The embedder
- Stacked transformers
- The affine block

And we understand that LLMs use this to generate a probability distribution for the next word.

Let's start with a sequence of tokens $x_1, x_2, x_3, ..., x_{k-1}$

The embedder embeds these tokens into a sequence of vectors

$$H_0 = \{h_1^{(0)}, ..., h_{k-1}^{(0)}\} \tag{1}$$

This H_0 is then sent to the transformers where it is processed and gives an output. Let's use H_i to denote the output of the ith layer.

Finally, probability for the next-token x_k is given as

$$p(x_k|x_{<k}) = softmax(\phi(h_k^N))_{x_k} where x_k \epsilon X \tag{2}$$

Where X is the set of vocabulary.

Now the method proposed is to contrast the top-layers with the bottom ones to obtain the probability of the next token. The computation of probability for next tokens in lower layers is given as

$$q_j(x_k|x_{<k}) = softmax(\phi(h_k^i))_{x_k}, \tag{3}$$

$where i = 1, 2, ...N$

Here N is the number of transformer layers in total. Then we use early-exit **Teerapittayanon et al., 2016; Elbayad et al., 2020; Schuster et al., 2022),** this works without any special training. The idea behind it is that the residual connnections present in the layers help the hidden represntations evolve correspondingly to sudden variations.

Now the probability of the next word can be found using -

$$\hat{p}(x_k|x_{<k}) = softmax(F(q_N(x_k), q_M(x_k)))_{x_k} \tag{4}$$

Where $M = argmax_{i\epsilon\Upsilon} d(q_N(.), q_i(.))$

The layer M stands for the premature layer and the final layer is called the mature layer. The function F is defined as below -

$$F(q_N(x_k), q_M(x_k)) = \begin{cases} log\frac{q_N(x_k)}{q_M(x_k)}, & \text{if } x_k \epsilon \nu_{head} \\ (x_k|x_{<k}) - \infty, & \text{otherwise} \end{cases} \tag{5}$$

$$\hat{p}(x_k) = softmax(F(q_N(x_k), q_M(x_k))) \tag{6}$$

It is used to contrast between the distribution of the last two layers. This is done by contrasting their corresponding distributions in log.

The premature layer which needs to be contrasted with the final layer is selected dynamically from the candidate layers in each step using a distance measure called Jenson-Shannon Divergence. This is a symmetric measure of divergence. The Υ is the set of all candidate layers.

The function d is a distance measure which uses Jenson-Shannon Divergence to select the premature layer which can provide the largest contrast.

The working theory is that the greater the contrast, the greater the accuracy for predicting the word which is more factually aligned with the original document.

$$\begin{aligned} &d(q_N.|x_{<k}), q_i(.|x_{<k})) \\ &\qquad = JSD(q_N.|x_{<k})||q_i(.|x_{<k})) \end{aligned} \tag{7}$$

Where the premature layer M satisfies the condition $0 <= M < N$. And is selected to be the premature layer with the largest divergence from the given candidate layers. This is done by -

$$\begin{aligned} M = argmax_{i\epsilon\Upsilon} \\ JSD(q_N.|x_{<k})||q_i(.|x_{<k})) \end{aligned} \tag{8}$$

3 Dataset and the LLM

3.1 LlaMA - 7B

This is a LLM which has 32-layers and the model used in this experiment continuing the line of work initially proposed in the work by **Yung-sung et al.**

As mentioned earlier, the approach uses Jenson-Shannon Divergence (JSD) to measure the distance between the premature and mature layers.

One of the patterns observed was that when checking simple facts like names or dates the JSD was very high in the later layers (closer to the mature layer). While for predicting words of less importance like $the, was, to, inetc.$ which can be copied directly from the input data, the JSD is very small around the middle layers.

Thus, the conclusion is drawn that when prediction requires very specific factual information, then it's better to contrast higher layers and contrast the lower ones for simple words.

This being the reason for the authors to use a dynamic way to select the premature layers as described above.

One of the key points to check was if the same held true in the case of summarization, but we leave that for future work.

3.2 Dataset

As is the standard, I am using the CNN/Daily Mail 3.0.0 Dataset. The CNN/Daily MailDataset, is an important asset in the field of natural language processing, consists of over 300,000 unique news articles penned by journalists associated with CNN and the Daily Mail. Initially created to aid machine understanding and comprehension, this collection of English-language data has progressed to support both extractive and abstractive summarization tasks. Each entry within the dataset includes three primary data fields: 'id,' housing the hexadecimal-formatted SHA1 hash of the article's URL source; 'article,' containing the main body of the news piece; and 'highlights,' showcasing the significant points of the article as authored by the original writer.

3.3 Task

The main task is to use the modification used by Yung-sung et al. (2023) in Question Ansering with text summarization using LLaMA-7b. Since LLaMA-7b has 32 layers, I sent the layer set $2, 4, 6, 8, 10, 12, 14$.

4 Experiments

We ran the summarization task on CNN/Daily Mail data set and used a baseline vanilla LLaMA-7b LLM and one which included the Contrasting Layers in the decoder.

4.1 Experimental setup

We utilized LLaMA-7b pretrained model from Huggingface, the CNN/Daily Mail 3.0.0 and ran them on 1 Nvidia RTX 6000 GPU with 48Gb of memory. The model was set to use a maximum of 100 tokens. The baseline utilized the new Langchain and Prompt templates, where as the modified code just runs in a loop sample at a time. The evaluation metrics were calculated for a total of 25 test samples.

4.2 Evaluation metrics

To check the performance of the modified model and the original model, we used a few widely recognized metrics.

- **BLEU scores:** BLEU serves as a measurement used to evaluate the caliber of machine translations. Its function involves assessing the resemblance between n-grams found in machine-translated sentences and those within human-translated sentences. Typically, it is observed that the BLEU score diminishes as sentence lengths increase, though deviations from this pattern might arise based on the employed translation model.

- **ROUGE Score:** The ROUGE Score evaluates the similarity of n-grams (word sequences) between the generated summary and reference summaries. It includes various metrics like ROUGE-N (unigrams, bigrams, etc.) and ROUGE-L (longest common subsequence) to gauge the level of content overlap.

- **BERT Score:** The BERT Score employs contextual embeddings derived from the BERT model to quantify the resemblance between the generated summary and reference summaries. This approach is tailored to grasp the intricacies of language and context, offering a robust evaluation metric.

4.3 Experimental results

When compared to the baseline the model which contrasted the layers in the decoder performed better in all the evaluation metrics except the BLEU scores. Which substantiates our theory that contrasting divergent layers improves factuality in our Language Model The BLEU score measures the similarity between the machine-generated summary and one or more human reference summaries. It calculates precision by comparing n-grams (sequences of words) in the generated summary with those in the reference summaries. Higher BLEU scores, generally ranging from 0 to 1, signify a closer resemblance between the machine-generated summary and the human-written summaries.

In our model it seems that this happens because of insufficient formatting and not being able to use an exhaustive set of stop words. It makes our model produce some tokens in between which would cause a low BLEU score. With the baseline, since we are able to use Langchain with HuggingFacePipeline along with a prompt template it is expected to have a closer resemblance to the original summary provided, the highlights.

5 Conclusion

As expected we can see that the modified model using the Contrasting Layers in the Decoder outperforms the baseline vanilla LLaMA-7b model in the task of text summarization. This was expected and more of a verification as to whether contrasting the probabilities betweeen layers which are the most divergent really enhances the factuality of a Large Language Model. Of course this was a very small scale and limited study where in the only model used was LLaMA-7b which is pretty small compared to the really big models out there and may not be representative of all models, but it brings to light the situation that it is possible to improve the accuracy and factuality of an LLM without just throwing more data or making it deeper or even scaling it up to a lot more parameters.

Acknowledgement

The author would like to thank and acknowledge his immense gratitude to professor Yue Dong, for her guidance which proved to be invaluable in this work, the research community at University of California Riverside and finally my dear friend Dhananjay Gavade who was able to help and support me through this journey.

References

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. Improving language models by retrieving from

Model	ROUGE-1(%)↑	ROUGE-2↑	ROUGE-L↑	BERT Score (P/R/F1)↑	BLEU Score↑
LLaMA-7b(baseline)	0.129	0.033	0.119	0.75/0.82/0.78	**9.553479375665518e-232**
CLD-LLaMA-7b	**0.221**	**0.082**	**0.201**	**0.79/0.87/0.83**	7.649308833340803e-232

Table 1: Automatic Evaluation Results on ESConv.

trillions of tokens. In International conference on machine learning, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Lan- guage models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper files/paper/2020/file/ 1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pre- trained transformers. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8493–8502, 2022.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325, 2023. Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In ICLR 2020- Eighth International Conference on Learning Representations, pp. 1–14, 2020.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. Not all models localize linguistic knowledge in the same place: A layer-wise probing on bertoids' repre- sentations. In Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pp. 375–388, 2021. Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. Transactions of the Associa- tion for Computational Linguistics, 9:346–361, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

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

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

OpenAI GPT-3.5, "text-davinci-003,". [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5. [Accessed: 2023- 10-14]

MosaicML NLP Team, "Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs," 2023. [Online]. Available: www.mosaicml.com/blog/mpt-7b. [Accessed: 2023-10-14]

"MPT-7B-Instruct, Hugging Face Models." [Online]. Available: https://huggingface.co/mosaicml/mpt-7b-instruct. [Accessed: 2023-10- 14]

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Sid- dhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. ”Scaling Instruction-Finetuned Language Models,” 2022. [Online]. Available: https://arxiv.org/abs/2210.11416. [Accessed: 2023-10-14]

”Falcon-7B-Instruct, Hugging Face Models.” 2023. [Online]. Available: https://huggingface.co/tiiuae/falcon-7b-instruct. [Accessed: 2023-10-14]

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

”CNN/DailyMail Dataset, Hugging Face Datasets.” [Online]. Available: https://huggingface.co/datasets/cnn dailymail. [Accessed: 2023-10-14]

”XSum Dataset, Hugging Face Datasets.” [Online]. Available: https://huggingface.co/datasets/xsum [Accessed: 2023-10-14]

”Metric Card for BLEU, Hugging Face Metrics.” [Online]. Available: https://huggingface.co/spaces/evaluate-metric/bleu. [Accessed: 2023-10- 14]

”Metric Card for ROUGE, Hugging Face Metrics.” [Online]. Available: https://huggingface.co/spaces/evaluate- metric/rouge . [Accessed: 2023- 10-14].

”Metric Card for BERT Score, Hugging Face Metrics.” [Online]. Available: https://huggingface.co/spaces/evaluate-metric/bertscore. [Ac- cessed: 2023-10-14].

Yan, Ziyou, "Evaluation Hallucination Detection for Abstractive Sum- maries" [Online]. Available: https://eugeneyan.com/writing/abstractive/. [Accessed: 2023-10-14]

L. Basyal, ”LLMs-Text-Summarization,” GitHub. [Online]. Avail- able: https://github.com/lbasyal/LLMs-Text-Summarization. [Accessed: 2023-10-14]