BBC News in a Nutshell

Abhilash Anand
Center for Data Science
New York University
aa9798@nyu.edu

Ivy Cui
Center for Data Science
New York University
mc9432@nyu.edu

Yue Lin
Center for Data Science
New York University
yl5735@nyu.edu

Abstract

Text Summarization for news not only address the data reduncy issues but also provide time effeciency. In this paper, we focus on LSTM, T5, and Llama2 models and compare their performances. As a result, Llama 2 and T5 are the best abstractive summarization models for BBC news, with Llama 2 excelling at word-level similarity and T5 at semantic similarity. (Our Code References)

1 Introduction

In the digital age, we are flooded with text from various sources. Balancing our busy lives with the desire to stay informed on a wide range of topics can be challenging. Text summarization emerges as a practical solution and helps manage it by saving time and enhancing user experience. Therefore, our project aims to provide high-quality and automated summaries by comparing the performances of different NLP algorithms and select the optimal summary models on the BBC news data.

In this project, we use LSTM, T5, and Llama2-7b models for abstractive summarization of BBC news. Specifically, LSTM is the baseline model, trained from scratch for sequential data. The second model is Google's T5 model, pre-trained on the C4 dataset and further trained on BBC news, converts NLP tasks to text generation. Our last model is Meta's Llama2-7b model, pre-trained and fine-tuned on a large text dataset, is optimized for generative tasks and fine-tuned with LoRA for multi-document summarization.

In summary, we found out that Llama 2 had the highest ROUGE scores but a low BLEU score of 0.0126. T5 had high ROUGE scores and a higher BLEU score of 0.0677. The LSTM model has the lowest performance across all metrics. Therefore, Llama 2 is the best model for capturing overlap of individual and consecutive sequences of words, while the T5 model is better at capturing the overall semantic similarity.

2 Related Work

The field of text summarization is significant in the news industry due to the ambiguity and diverse interpretations found in news articles.Tianyi Zhang et al's paper [1] benchmarks different LLMs for the task of News Summarizations, which closely align with our project goal. The paper states that LLMs performance is dependent more on instruction tuning rather than model size alone. The LLM models were evaluated by humans and each of the traditional metrics was also given a score based on the three factors: faithfulness, coherence, and relevance.This gives us a lot of insight into how we can train and evaluate our models. In addition, Bohdan M. Pavlyshenko1's paper [2] explores the application of Llama 2 GPT large language model (LLM) for the analysis of Financial news. The model, fine-tuned using the PEFT/LoRA-based approach, can highlight main points, summarize text, and extract named entities with sentiments, leading to performing a multitask financial news analysis. It provides us with insights into the potential of fine-tuning the Llama 2 model for news summarization tasks. Thus, drawing inspiration from the insights presented in the papers, we seek to incorporate these models into our project to achieve improved results and make meaningful contributions in our specific application domain.

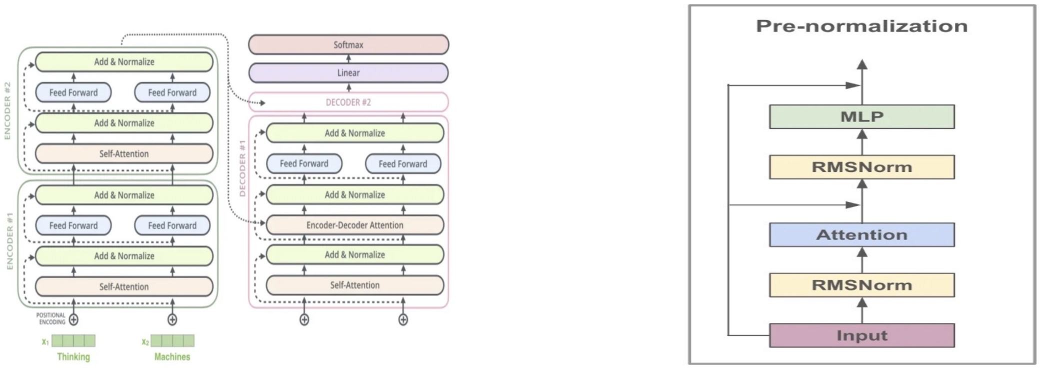
3 Approach

Given our focus on abstractive summarization, we favor generative language models like T5 or BERT over extractive methods such as graph-based or TF-IDF. Despite the imperfections of pre-trained models like T5 or GPT-3 for diverse BBC news data, we undertake the training and fine-tuning of three key models as outlined below.

The baseline model is the LSTM, a type of RNN encoder-decoder that solves the vanishing gradient problem of RNNs. RNNs were popular before Large Language Models and can handle long-term dependencies and various NLP tasks. We use an LSTM encoder-decoder for training, where the encoder processes the input sequences to comprehend the context, and the decoder, using the encoder's final states, generates the summary in a sequence-to-sequence manner.For inference, the decoder uses the encoder's states to generate the output tokens step by step. We trained the LSTM from scratch on Google Colab A100 GPU (40 GB).

The second model used was Google's T5 model. The T5 model comes in multiple variations related to both size and pre-training methods. We used the T5-base (0.2B) model, pre-trained on C4, for text-to-text NLP tasks. It differs from traditional transformers by using layer norm at both ends of each layer and positional embeddings instead of absolute ones. It uses teacher forcing for training, where the decoder takes the target word as the next input. We trained it on BBC news data using Google Colab's Tesla T4 GPU (16 GB) with a maximum article length of 1800 and a maximum summary length of 600. We then switched to HPC with RTX8000 (48 GB) to increase the lengths to 3000 and 1600 respectively. We added the token 'summarize:' to the start of each input to trigger the summarization task.

Our final model incorporates the pre-trained Llama 7b model. This model has a causal decoder-only architecture which is different from the prior two models. Traditional fine-tuning approaches for pre-trained language models (PLMs) involve updating all model parameters, demanding substantial computational resources and extensive data. To overcome this limitation, Parameter-Efficient Fine-Tuning (PEFT) was employed. PEFT selectively updates a small subset of the model's parameters, resulting in a more computationally efficient process. In this case, we use Low-rank adaptation (Lora) to fine-tune our Llama model on the BBC news data.



(a) T5 model architecture

(b) Llama model architecture

Figure 1: Comparison of T5 and Llama model architectures

4 Experiments

4.1 Data

Our project will utilize the "BBC News Summary" dataset sourced from Kaggle, specifically designed for extractive text summarization tasks. This dataset consists of 2,225 news articles, each with an associated summarized paragraph that serves as the ground truth, from the BBC for the years 2004 to 2005 and is categorically distributed as follows: Business (510 files), Entertainment (386 files), Politics (417 files), Sport (511 files), Tech (401 files). Before subjecting this data to our models, we applied a set of preprocessing techniques, such as text cleaning, text tokenization, input truncation in web links, numerals and punctuation marks, and padding, to ensure it is in an optimal format for model training and evaluation. After this, the Spacy library tokenizes the purified text, segmenting it into distinct English tokens. After the data cleaning, we split the data into training, validation, and testing sets before implementing our model.

4.2 Evaluation method

We assessed the model using evaluation metrics of BLEU and ROUGE. The BLEU metric, ranges from 0 to 1, measures n-gram similarity between machine and reference texts. The higher BLEU score values indicate a better precision against the references. The ROUGE metric that ranges from 0 to 1 compares the automatically produced summary against a set of reference summaries, focusing on recall. The higher ROUGE score indicates a higher similarity between the machine-generated summary and the reference. In the following result, we had ROUGE-1, ROUGE-2, and ROUGE-L, which measure unigram, bigram, and word sequence overlaps, respectively, to understand and quantify the model performance.

4.3 Experimental details

Initially, we built the LSTM model from scratch as our base model. After we cleaned the text by removing punctuation and stopwords, we trained Word2Vec embeddings on the cleaned articles and summaries, and used them to convert the text into numerical sequences, with articles at 128 tokens and summaries at 65 tokens. For optimization, we used Adam optimizer with a learning rate of 0.001. The model trained for 50 epochs with a batch size of 64, using ModelCheckpoint to save improvements, and EarlyStopping to halt training upon stagnation. We also used the Word2Vec embeddings matrix as pre-trained weights for both the encoder and decoder, and fine-tuned them during training, and set the LSTM units at 256 to balance long-term dependencies and computational efficiency.

Our second model involved the fine-tuning of the Google T5 base model on BBC news data involving several key parameters. The training process utilized a batch size of 2, with 2 validation batches. Training epochs were set to 10, while validation epochs were limited to 1. The learning rate was established at 1×10^{-4}, and the entire process spanned 10 epochs. Article lengths were capped at 1800 during training and 3000 during validation, with summary lengths set to 600 and 1500, respectively. Each training epoch took approximately 55 minutes. Data was split into training, testing, and validation sets at a ratio of 0.6, 0.2, and 0.2, respectively. The model was run on a GPU, specifically a Tesla T4 with 16 GB for training and an RTX 8000 with 48 GB for processing.

Finally, we fine-tune the Llama2 model with LoRA, adjusting attention mechanisms using low-rank adaptations for summarization tasks. We set the LoRA parameters: attention dimension `lora_r` to 64, the scaling factor `lora_alpha` to 16, and a dropout rate `lora_dropout` of 0.1 to prevent overfitting. These optimize the attention mechanism with fewer parameters. The model trains for five epochs, with batch size of 4 per device, both for training and evaluation. We accumulate gradients for every update step, and cap gradient norm at 0.3. We use the Llama-2 tokenizer, setting padding token to end-of-sequence token, and aligning padding to the right for overflow issue with fp16 precision.

4.4 Results

The scores of the LSTM are lowest among the three models, as it is less capable of handling complex patterns than transformer-based models like T5 and Llama 2. T5 performs much better than LSTM, as it is more recent and advanced, benefiting from a transformer-based architecture, and is more effective in handling sequence-to-sequence tasks. Llama 2 scores slightly better than T5 on ROUGE-1 and ROUGE-L and slightly lower on ROUGE-2 and BLEU, suggesting its effectiveness in capturing bigram-based similarities. However, the scores are not reliable for text summarization, and human evaluation is needed on criteria such as relevance, coherence, readability, and succinctness. We found that LSTM summaries are short and incomplete, while T5 and Llama 2 summaries are readable and comprehensive.

Model	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
LSTM	0.1384	0.0615	0.1249	0.0451
T5	0.5168	0.4183	0.5147	0.0677
Llama 2	0.5984	0.4785	0.5887	0.0126

Table 1: Results

The training loss of LSTM decreases steadily, which is a good sign of learning. However, the validation loss does not decrease as much, indicating the model might not generalize well to unseen data, it's a sign of overfitting. The T5 and llama2 model shows a rapid decrease in training loss within the first few epochs, which then stabilizes. As the validation loss is decreasing alongside the training loss, implying that the model is generalizing well which indicates the model training progresses are outperformed.

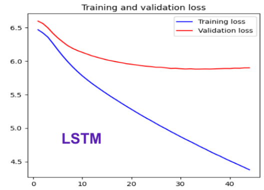


Figure 2: Lstm train/val loss

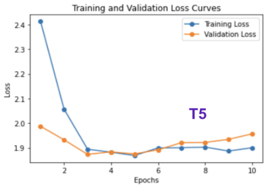


Figure 3: T5 train/val loss

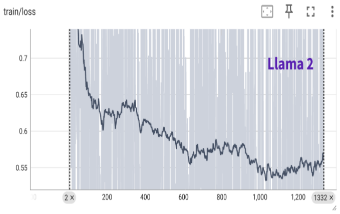


Figure 4: Llama2 training loss

5 Analysis

Upon reviewing the outputs of the LSTM model, it is able to predict short sentences with gibberish. But it faces challenges in grasping context with longer sentences and it is also constrained by length limitations. Augmenting LSTM units fails to enhance its performance, as evidenced by ROUGE and BLEU scores, and it is constrained by length limitations. Thus, increasing LSTM units does not improve its performance (ROUGE and BLEU scores). Besides, The Llama 2 model aligns moderately with the ground truth, captures information with clarity, but deviates occasionally. Its summaries maintain a high degree of clarity, ensuring comprehensibility. It also incorporates specific details moderately, avoids repetition, and maintains consistency. However, the model's proficiency in handling ambiguity varies across categories, indicating room for improvement in navigating nuanced language. Overall, the model showcases a mix of strengths and areas for improvement across diverse evaluation criteria. Lastly, the T5 base model exhibits performance comparable to the Llama 2 model, albeit slightly less effective than the Llama 2 model, particularly in the areas of Incorporation of Specific Details and Sensitivity.

6 Conclusion

The above models showed how different models' sizes and architectures can have an impact on the summaries predicted. Even though the T5 base is a smaller model than the Llama2 because of its pre-training and architecture it performs almost as well as the Llama2 model. It would be a good experiment to compare the performance T5 3 billion model with the Llama2.

We will also explore different fine-tuning methods and try larger models that may help in capturing more nuanced patterns in the data for LLama2 to see if we can improve the ROUGE-2 and BLEU scores. For the T5 model, we will experiment with a larger parameter set and versions trained on different datasets or with improved architectures that could provide further enhancements in summarization quality. In addition, we may try prompt engineering based on the fine-tuned model in depth.

7 Student contributions

Workloads are evenly distributed among all the team members including data preparation, model, analysis, presentation, and wrote report.

Abhilash Anand: Data preparation + Model + Analysis + presentation + wrote report

Ivy Cui: Data preparation + Model + Analysis + presentation + wrote report

Yue Lin: Data preparation + Model + Analysis + presentation + wrote report

References

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

[2] Bohdan M Pavlyshenko. Financial news analytics using fine-tuned llama 2 gpt model. *arXiv preprint arXiv:2308.13032*, 2023.