The Library of Babel: Summarizing and Translating non-English Research Papers

This report provides an examination of our methods for fine-tuning models for the task of extracting contributions from non-English research papers.

Casa International
Ryotaro Takehara
Jose Torres
Naila Hajiyeva
INF 385T: Natural Language Processing/Applications (Fall 2024)

Introduction

The research landscape is predominantly dominated by English-language publications, which poses a significant barrier to accessing valuable research conducted in other languages. The project "The Library of Babel" aims to bridge this gap by extracting the "contribution" from these non-English papers, making their insights accessible to a broader audience.To be more precise, final goal of this project is to develop a method to translate and summarize non-English research paper's abstract, thus enabling English-speaking researchers to grasp the idea of contribution made by non-English research papers. This initiative not only expands the research horizon but also acknowledges the contributions of researchers working in various languages, specifically.

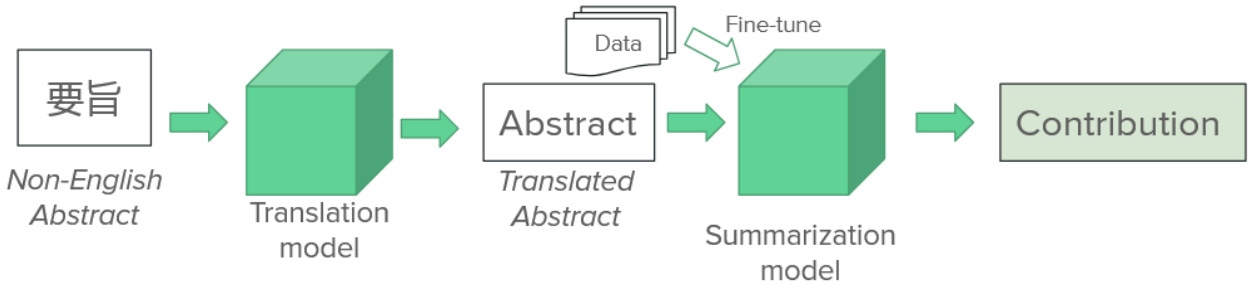
Box1: Why contribution?

This idea of extracting the main contribution derives from the paper "How to Read a Paper" written by S. Keshav. Keshav claims that, after scanning the paper once, a reader should be aware of the paper's "category", "context", "correctness", "contribution" and "clarity" (Keshav, 2007). Contribution was selected as the target for this project because it seemed to be the most important component in terms of understanding a paper, and is a suitable task for a LLM model. It might be a good future goal of this project to pursue extraction of all five Cs, especially assessing "clarity" of the paper with machine learning would be an interesting task.

Methodology

Overview

To achieve this goal, a LLM pipeline that is capable of translating and summarizing abstract was developed. In detail, the pipeline consisted of receiving a Non-English abstract as input, running it through a translation model of choice, then having the translated abstract be input into our summarization model that was fine-tuned to extract the contribution (see the diagram pictured below for the structure of the pipeline).



Main experiment occurred in this summarization part as a variety of models, fine-tuning techniques, and hyperparameters were tested to improve the appropriateness of the output summary.

Evaluation

The appropriateness of the extracted summary of contribution is then evaluated with 4 common natural language generation (NLG) metrics: **BLEU, ROUGE, METEOR, BERT Score**. Below are details of each metrics.

BLEU

We used BiLingual Evaluation Understudy (**BLEU)** to measure word overlap between the machine generated contribution and the human-generated contribution paragraph. BLEU is a precision based metric for natural language processing (NLP) which works by getting how words returned were correct over various n-gram. To get the final BLEU score we modify the precision scores p_n by taking their logarithmic and multiplying them by positive weights w_n. Then we take the geometric mean of these components, afterwards we multiply everything by a "Brevity Penalty" *BP* which is defined by 2 factors: the length of the candidate sentence *c* and *r* is the effective length of the corpus (Papineni, 2002).

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \qquad \text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

ROUGE

We also used the Recall-Oriented Understudy for Gisting Evaluation (**ROUGE)** metric to measure word overlap between the machine generated contribution and the human-generated contribution. ROUGE is a recall based metric for NLP that instead counts the number of correct words that were returned over the total number of words in the reference summary. We used the ROUGE metric because of its ability to do both recall and precision which helps us avoid using too many unnecessary words and encourages using words in the reference summary. While there are a variety of the ROUGE metric, for this project we went with ROUGE-1 (Lin, 2004).

$$\frac{\text{Total \#overlapping words}}{\text{Total \#words in reference summary}}$$

METEOR

We also used the Metric for Evaluation of Translation with Explicit Ordering (**METEOR**) that works similar to BLEU but takes into account more linguistic nuances. It uses

unigram-precision and recall as well as a different measure of how well-ordered the matching words are in the machine translation to the reference. It can have multiple stages of acquiring the values for a score, the first one is matching exact words, the second one uses a stemmer, and the last is trying to find similar words using synonyms then all values are aligned between the machine text and the reference text. Finally to calculate the score we find the unigram precision (P), unigram recall (R) , finally computing F_{mean} by combining these scores as follows. It also takes into account longer matches; it does this through a penalty calculated by attempting to create the fewest chunks possible from the reference text to the machine text. The final METEOR score is calculated by $METEOR = F_{mean} * (1 - Penalty)$ (Banerjee, 2005).

$$Fmean = \frac{10PR}{R+9P} \quad Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams\_matched} \right)$$

BERTScore

For our final metric we used BERTScore which is the most different metric we are using because it is calculating a similarity score for each token in the machine text with each token in the reference text. It calculates the F1 score by first calculating the recall and precision of the tokens.

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

It also has the capability of adding weights for when more rare words using inverse document frequency (idf) scores. Using M reference sentences, the idf score of a word-piece w is

$$\text{idf}(w) = -\log \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}[w \in x^{(i)}]$$

This technique is applied to R_{BERT}, P_{BERT}, F_{BERT} for a calculated score of:

$$R_{\text{BERT}} = \frac{\sum_{x_i \in x} \text{idf}(x_i) \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{x_i \in x} \text{idf}(x_i)}$$

However because of the learned geometry of contextual embeddings the outputted scores are less readable, so to improve readability the scores are linearly scaled by computing *b* from average BERTScore computed on sentence pairs from Common Crawl monolingual datasets. Once we have b, we can rescale BERTScore linearly giving us a readable score from 0-1 (Zhang, 2019).

$$\hat{R}_{\text{BERT}} = \frac{R_{\text{BERT}} - b}{1 - b}$$

To examine the advantage of LLM, those scores from the LLM model were compared with a “base-case” model that utilizes sentence similarity, and does not leverage LLMs. In detail,

the base-case model embeds each sentence in abstract using a sentence-transformer model "all-MiniLM-L6-v2"[1]. Then similarity between each embedding and another embedding (using the same model) of a sentence "Paper's contribution, new finding, new case" was measured using cosine-similarity. Finally, two sentences that were most similar were concatenated and returned as a summary.

Models

Belows are the details of the language models tested in this project, starting from the translation model.

Only one translation model was tested as the primary focus of the project was to improve the summarization accuracy.

SeamlessM4T (translation model)

- This is a Massively Multilingual and Multimodal Machine Translation (M4T) that was picked as our pipeline's translation model because of its status as a robust state-of-art model in this task. This model has 2.3B parameters and supports over 100 languages, which includes our target language in this project: Japanese and Spanish. Although quantitative evaluation for the accuracy of the translation was not performed, from the member's subjective judgement, the quality of the translation was superb considering the fact that the abstract contained many domain-specific language. Notwithstanding that, sometimes the result of the translation included artifacts such as repeated words, and that might have interfered with the accuracy of summarization downstream (more discussions at following sections).

Following are the details of the summarization models we have tested. We have tested four different language models in total to improve the result of summarization.

Llama-2-7b

- Llama-2-7b is a member of the Llama-2 model family developed by Meta with 7 billion parameters. It is an auto-regressive language model that utilizes the optimized transformer architecture. Model was intended for commercial and research use in English and is capable of adopting to wide range of tasks[2].

Mistral-7b-v0.3

- Mistral-7b-v0.3 is a pre-trained generative text model based on transformer architecture with 7 billion parameters. Former version, Mistral-7b-v0.1 outperformed Llama-2 with 13 billion parameters with many metrics (Jiang, Albert Q., et al. 2023), and v0.3 is the version with expanded vocabulary.[3]

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[2] https://huggingface.co/meta-llama/Llama-2-7b
[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

BART large CNN

- BART is a transformer encoder-decoder model that is trained by first corrupting text then having a learning model that reconstructs the corrupted text. This is a fine-tuned version of the BART LLM that was fine-tuned using CNN daily mail with the intention of being able to summarize better. This model has 406M parameters and was chosen because it is a specialized LLM with the intention of summarizing text.[4]

T5 Large Medical Summarization

- The T5 Large model is a pre-trained model that was pre-trained on a broad range of medical literature that allows it to summarize medical text extremely well by capturing the domain specific terminology often lost in generic summarization models. This model has 60.5M parameters and was chosen because it is a specialized LLM with the intention of summarizing medical text which was perfect because our fine-tuning data came from PubMed.[5]

Special techniques

Resource constraint always is a major concern when handling a large model, and this project was not an exception. We have encountered several out-of-memory errors concerning GPU RAM. To overcome this issue, some special techniques introduced below are applied. Those techniques enabled us to train & use the model with much less memory consumption and computation time, and facilitated our research.

LoRA

- This technique "freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the transformer architecture." This reduced the time consumption needed for fine-tuning to a great degree, and thus enabled us to test multiple models in a limited time scope. LoRA cuts down on trainable parameters and thus cuts down fine-tuning time and memory but maintains the accuracy and sometimes even improves performance (Hu, et al., 2021).

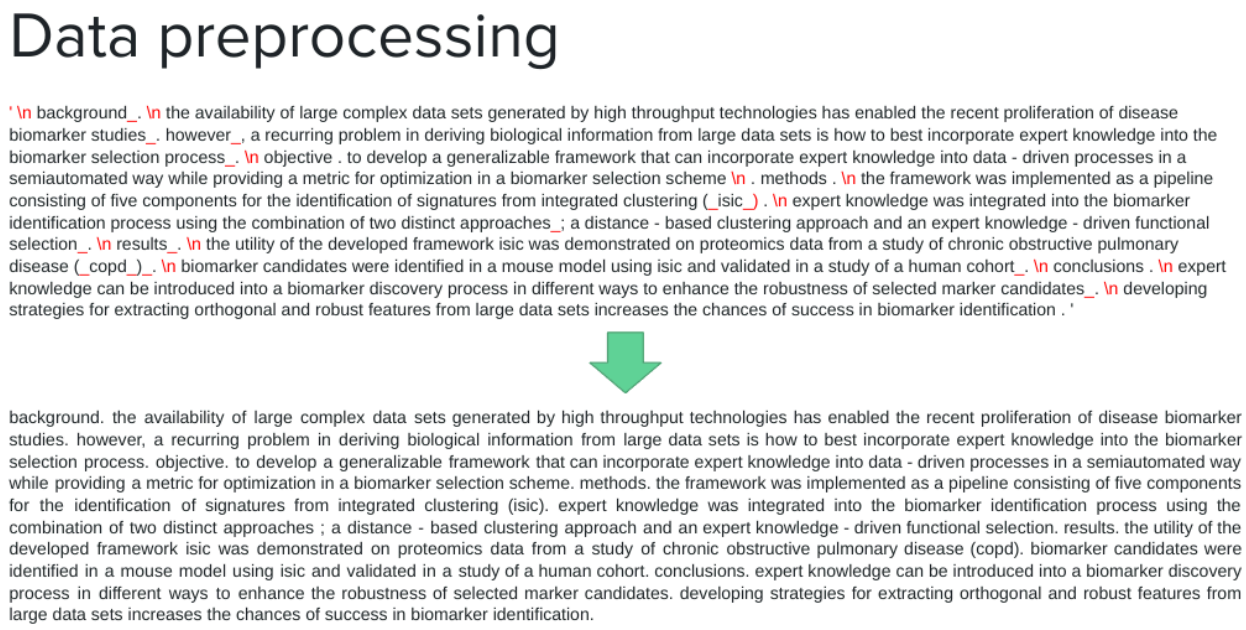
Unsloth

- Unsloth is a project to create open source versions of various LLM models that make fine-tuning faster, cheaper, and less memory intensive without harming the performance of the model. Speed-up and reduced memory consumption is achieved by manually deriving the matrix differentials themselves, and observing patterns around LoRA weights and LLama weight dimensions where correct bracket placement during chained matrix multiplication leads to huge performance gains (Han, 2023).

[4] https://huggingface.co/facebook/bart-large-cnn
[5] https://huggingface.co/Falconsai/medical_summarization

Dataset Description

The project leverages 67 scientific papers' abstracts from PubMed written in English as a training dataset for fine-tuning LLM. Papers were randomly selected from the "scientific_papers" dataset available in Hugging Face[6], which contains more than 130,000 PubMed papers (content, abstract and section names). Those abstracts were summarized to 1-2 sentences by manpower with their contribution being the main idea extracted. Then, the testing data were abstracts of scientific paper in Japanese and Spanish (Japanese dataset consists of medical paper, while Spanish dataset consists of varying disciplines in natural science), 10 in each language. Those abstracts were converted into target summaries text created by hand in abstract language. There were a lot of noisy components due to being extracted from PDF files. We used Regex to remove special characters to prepare the text for translations, escape sequences, trailing spaces, extra spaces before punctuations, page numbers, and more formatting elements. Below is the comparison of a text before and after the preprocessing.



[6] https://huggingface.co/datasets/armanc/scientific_papers

Results

First attempt

In our first attempt, we tried to use the plain & fine-tuned Large Language Models (LLMs). We attempted to fine-tune these models with the simple prompt:

"*Summarize the following research abstract. Focus on main contributions. \n\n{abstract}\n\n Summary: ".*

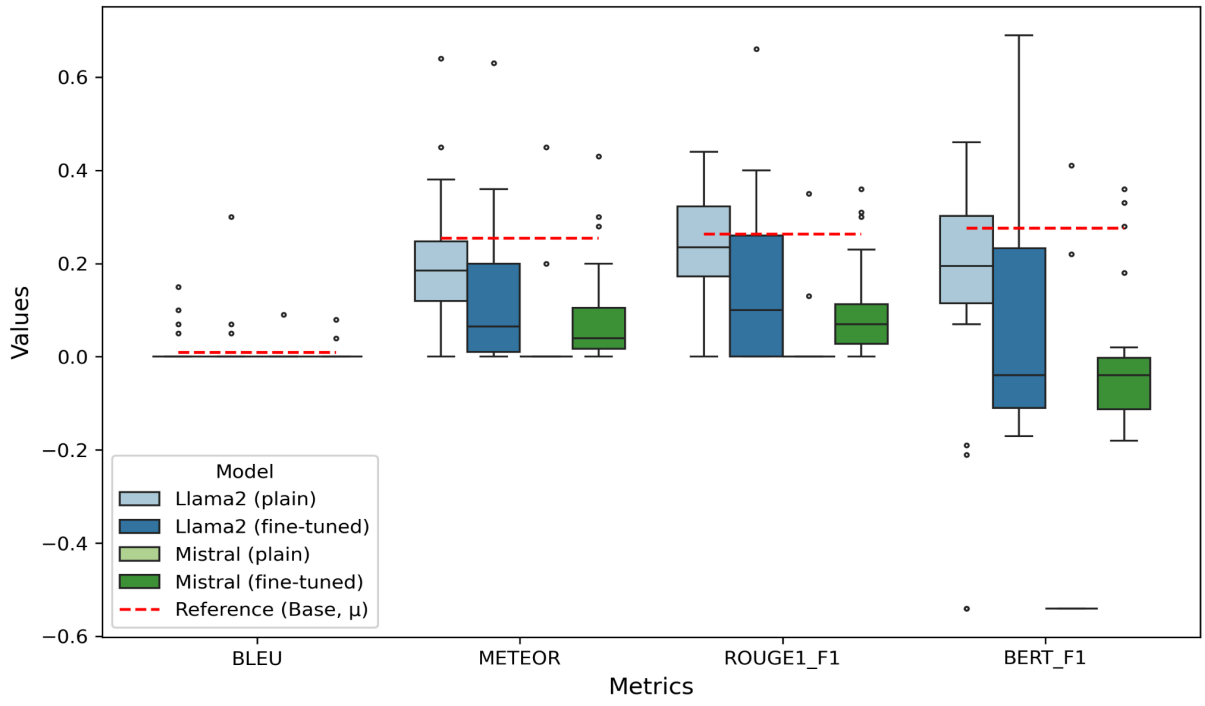
We generated the summary on test data using the same prompt on both plain and fine-tuned model to measure the effect of fine-tuning.

Below is the table that shows the parameters used in the fine-tuning. We did not perform rigorous hyper-parameter tuning, but with few tests this setting seemed to be optimal. Unfortunately, we were not able to test other batch size due to memory limitations.

Quantization	4 bits
Number of epochs	8
Batch size	4
Learning rate	1e-5

Below is the box plot that summarizes the result of the first attempt.

Figure 1: Comparison of Llama 2 and Mistral model (plain and fine-tuned)

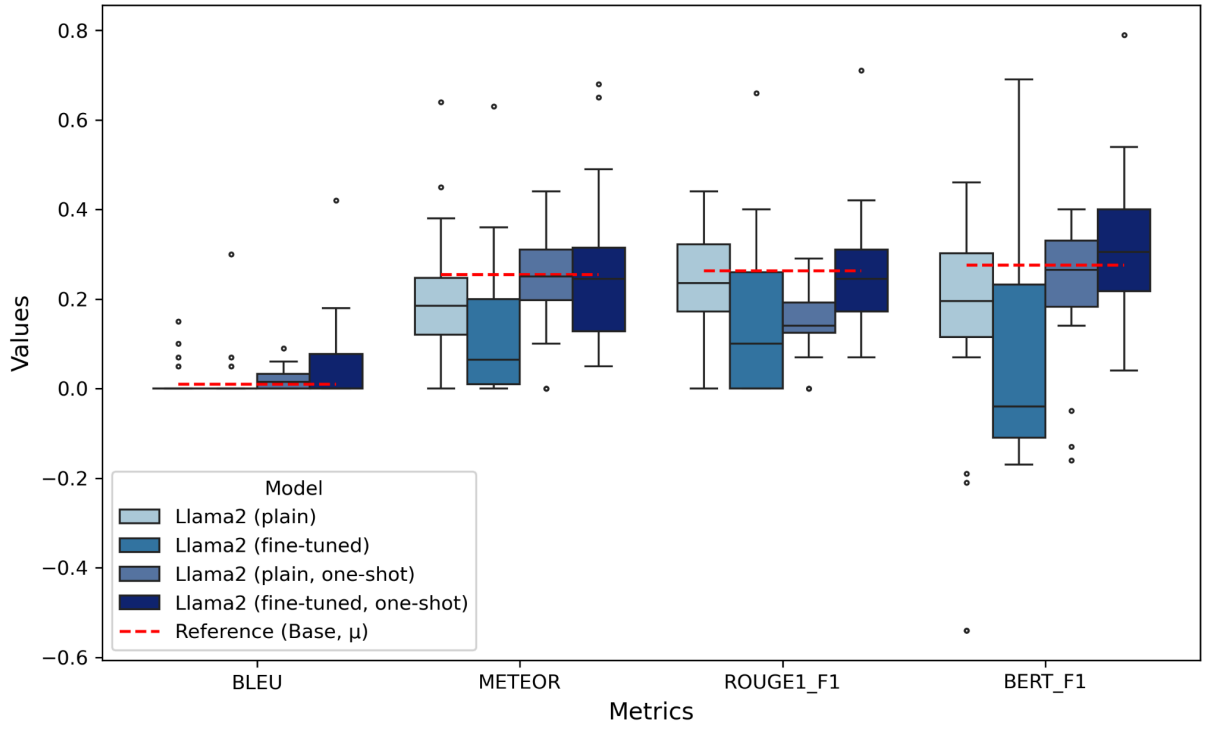


As you can see in the graph, on average the models we tested could not beat the be base case. Also, it is worth noting that in the case of Llama 2 distribution of scores deteriorated for fine-tuned model, indicating that the fine-tuning exacerbated the results. However, the fine-tuning of the Mistral model made it perform slightly better than the plain Mistral.

Second attempt

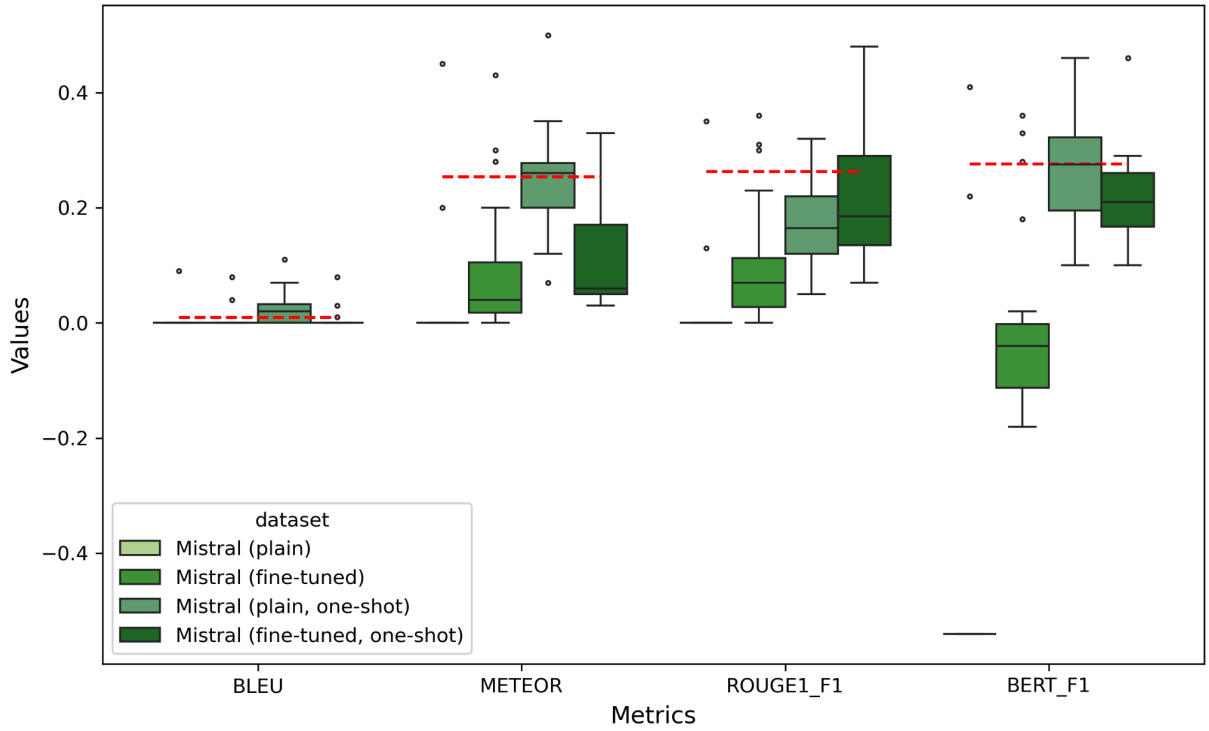
In our second attempt, we used one-shot prompting, including one summatizatoin example in the prompts to enhance the model's understanding of the task. We used the same prompt in both fine-tuning and testing process, keeping the other parameters the same across models.

Figure 2: Comparison of Llama 2 models with/without one-shot prompt (plain and fine-tuned)



As you can see in the metrics graph above, the one-shot Llama model resulted in general implrovement across metrics, doing better with the fine-tuning than without.

Figure 3: Comparison of Mistral models with/without one-shot prompt (plain and fine-tuned)



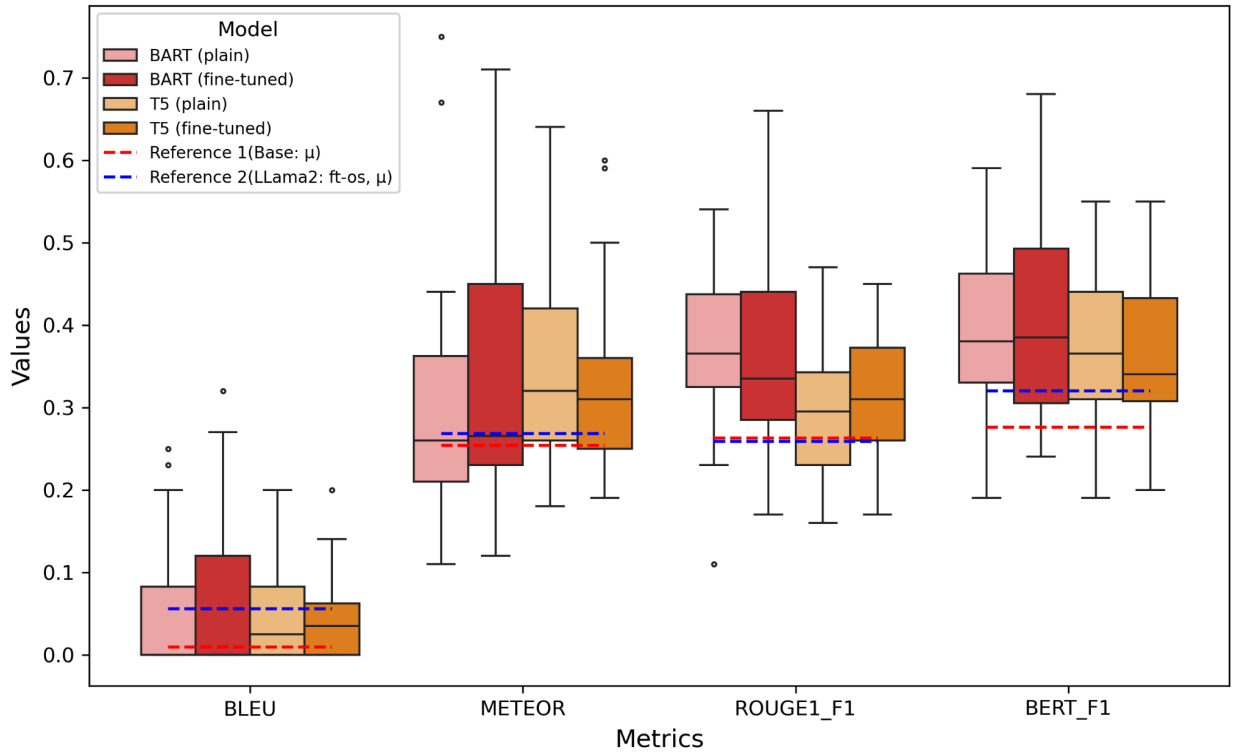
Mistral model, on the other hand, showed significant improvement, especially in the BERT score, while not showing heavy effects of the fine-tuning.

It can be observed that both plain and fine-tuned LLM models are still struggling to comprehend the task and, while the on-shot strategy was effective on the smaller prompts, it was not enough as average score being tantamount to that of baseline model. Therefore, we decided to switch to a model specializing in summarization tasks.

Third attempt

In our third attempt, we turned to a specialized mode, specifically bart-large-cnn, fine-tuned on CNN daily-mail - summarizing highlights from the news source. We also tested the T5 Medical_summarization, a model fine-tuned on summarizing medical claims and procedures. Those language models were significantly smaller in parameter sizes, and their metric scores are shown in the graph below.

Figure 4: Comparison of BART and T5 model (plain and fine-tuned)

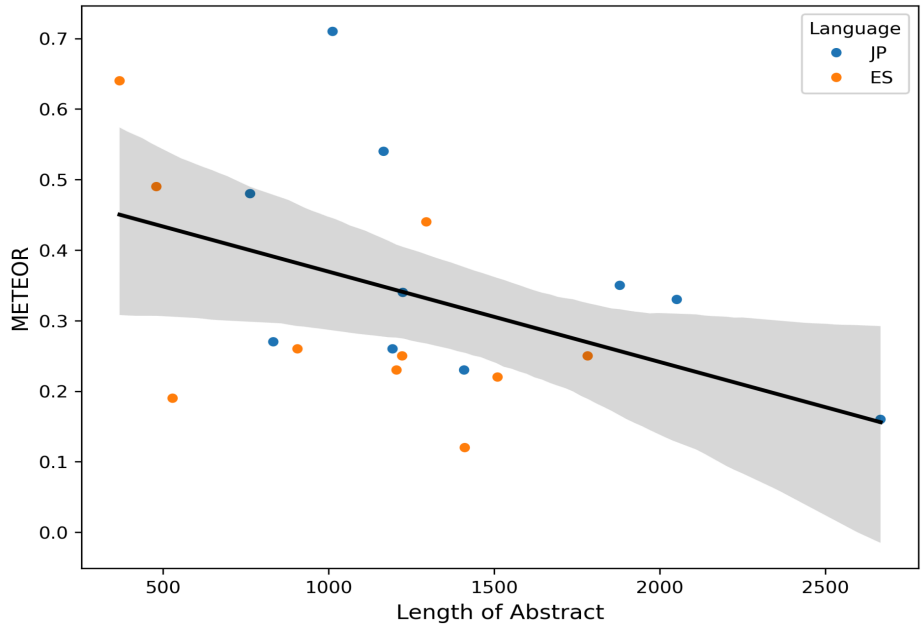


As you can see, there is a significant improvement, such as the METEOR score reaching above 70% with the BART model. Also, BART model is showing improvement in general as a result of the fine-tuning. Meanwhile, for the T5 the overall magnitutre of improvement is smaller, there is significantly less variance, stabilizing the model's performance.

Error analysis

After examining the performance of different models within our 3 attempts, we have arrived at the conclusion that the BART model turned out to be the best model. Thus, in the following session we conduct error analysis on this model in particular.

Figure 5: Correlation between length of abstract and METEOR score of summary



As you can see in the graph above, the BART fine-tuned model showed a negative correlation between the length of the input and the METEOR score. The result was slightly better for the Japanese queries (blue dots) as compared to the Spanish queries (yellow dots). This might be the result of Japanese abstracts consisting of more medical papers similar to the abstracts in training data and limited number of domain-specific terms and numbers, while some of the Spanish papers had to do with different areas of study, such as computer science or experimental physics.

Box2: Chat GPT comparison

Additionally, we compared our BART model to ChatGPT output, using the same prompt: "Summarize the following research abstract in english. Focus on main contributions. Limit the answer to 2-3 sentences."

Japanese

	BLEU	METEOR	ROUGE-1 (F1)	BERT (F1)
0	0.03	0.18	0.42	0.52
1	0.01	0.15	0.31	0.45
2	0.01	0.13	0.28	0.48

Spanish

	BLEU	METEOR	ROUGE-1 (F1)	BERT (F1)
0	0.14	0.35	0.56	0.57
1	0.05	0.23	0.47	0.56
2	0.07	0.28	0.55	0.53

ChatGPT summaries achieved BERT scores close to 0.5, indicating average semantic similarity.

ROUGE-1 (F1) suggested decent overlap with human-written summaries, but not strong enough to make it efficient. This is likely due to words having multiple meanings, and some models' inability to differentiate based on context. Additionally, the comparison of scores in the BLEU and METEOR calculations indicate weak lexical similarity.

Conclusion

This project aimed at translating, and summarizing the abstract of research papers to enable English-speaking researchers accessing contributions done by non-English research papers. At the beginning, it was naively thought that use of fine-tuned LLM would solve this task with ease. However, it turned out that the quality of the output returned by fine-tuned LLM didn't even reach that of the base-case model. This taught us the importance of understanding the problem and selecting the correct specialized model that was trained for the summarization task, this also highlighted that the number of parameters is not the only factor for performance. Important to note is that Seq2Seq tasks on these predominantly medical documents was a tough task even for our specialized LLMs.

The random errors produced in the summaries also taught us that the need for high quality data is not something that LLMs can escape from either despite their incredible complexity. Text pre-processing part of the pipeline reduced most of the unwanted components such as extra spaces and special characters, but it could have been not thorough enough and could have affected the quality of the output summary. Our translation model was not perfect either and occasionally suffered from translating domain-specific language and existence of unwanted components, introducing some noticeable artifacts such as repeated words. This is an error that could be fixed with more cleaning of the input data and further fine-tuning the translation model.

Overall this project has taught CasaInternational a lot about the process and limitations of fine-tuning a general model for the task of summarizing domain-specific documents. Our work had many limitations regarding the quantity and quality of our training data used for fine-tuning. Future work for this project would most likely require the creation of more high quality data as well as more hyperparameter tuning.

Appendix

After presenting our work, a small Q&A followed where some very interesting and insightful questions were asked. Below is the summarization of Q&A.

Q1: How does the system handle domain-specific terminology across diverse disciplines?

The summarization model seems to be learning topics related to medicine through fine-tuning, as the BART model showed a better score after fine-tuning. Also, this is largely helped by the accuracy of translation made by the seamless model. Expansion of the domain can be done through further fine-tuning of both translation and summarization models using papers from various disciplines.

Q2: Are there plans to improve translation fidelity to reduce artifacts before summarization?

Yes. One way would be fine-tuning through domain specific papers. Another, and more important way is to further enhance the text-cleaning pipeline to get rid of all the unwanted components and unique characters. It was observed that models tend to generate unexpected results such as repeated words and empty sentences when there are multiple unwanted components (extra white spaces & special characters). It might even be effective to get rid of all numbers and units as those are hard to process as a part of a sentence, and summary doesn't usually include those detailed numbers.

Q3: How do you intend to scale the dataset for more robust fine-tuning?

Two fold. 1: Increase amount of dataset. 67 seems to be too less. 2: Widen the discipline. Also, as suggested above, clean the text thoroughly to avoid artifacts.

Q4: Why did fine-tuning degrade performance for some models, and what steps can be taken to address this?

Many times, degrades of performance caused by introduction of unwanted components such as repeat of instruction or unwanted components such as html tags in the output. It is highly possible that some training data was not cleaned enough. Increasing the amount of data should also be a key for complicated seq2seq tags. Final option is to combine reinforcement learning to penalize those unwanted components.

References

Banerjee, Satanjeev, and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Cohan, Arman, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. NAACL-HLT 2018.

Han, Daniel. 2023. Introducing Unsloth: 30x faster LLM training. https://unsloth.ai/introducing.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. D. L., ... & Sayed, W. E. (2023). Mistral 7B. *arXiv preprint arXiv:2310.06825*

Keshav, S. How to Read a Paper. https://web.stanford.edu/class/ee384m/Handouts/HowtoReadPaper.pdf.

Lewis, M. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Patwardhan, N., Marrone, S., & Sansone, C. 2023. Transformers in the Real World: A Survey on NLP Applications. Information, 14(4), 242.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

Code

We have all of our code in this Github repository. Everything needed to recreate what we have is available in the repository with a documented readme.