Domain-Specific Pre-Training for Improved Abstract Summarization in Neuroscience

Dylan Daniels
UC Berkeley, School of Information
dylansdaniels@berkeley.edu

Abstract

With the rapid pace of research and the increasing volume of publications, it has never been more challenging or more daunting for scientists to stay up to date on the latest in their respective fields. This is a domain in which we can leverage advances in NLP to build tools that address challenges in scientific research. The results of this study suggest that continued in-domain MLM pretraining and summarization fine tuning of BART can yield a model that produces improved summarization in a very narrow domain, i.e. that of a single researcher. This insight is important, as it suggests that we can considerably improve on and adapt publicly available tools to very narrow use cases without needing to pre-train large models from scratch. Pre-training from scratch, as done with BioBERT, would likely be intractable in domains as narrow as the one considered in this study due to lack of training data and computational cost, and the present study suggests that adapting publicly available tools may be a viable alternative.

1 Introduction

Staying up to date on the latest literature in a given field is a key component of a scientist's job; however, the task has become increasingly difficult. As the base of knowledge within a given field grows, so too does the corpus of information a scientist must navigate. Further, the rate of growth in publications has only increased over time, putting greater demands on scientists to parse and navigate new literature for relevant information. The problem of 'information overload' in the sciences has grown to the point that there are several publications on the topic (e.g., see Scientific literature: Information overload, Nature, 2016)[1], and the Association for Computational Linguistics now holds annual workshops on information extraction from scientific publications. With a limited capacity to consume information, scientists must increasingly rely on reduced representations of information to guide their attention.

Fortunately, recent advances in machine learning and natural language processing have led to major strides in tasks such as summarization and information extraction. Publicly-available models such as BART and T5 can do a reasonable job at text summarization out of the box. Another important challenge in the sciences, however, is the emergence of increasingly compartmentalized silos that adopt specific terms, frameworks, and even invent new terms. Capturing this domain specificity can be a challenge for models trained on a more general corpus, and thus, additional training and fine tuning is needed for these models to be useful to scientists operating in narrow domains.

To begin addressing these challenges, the present study explores whether continued in-domain pretraining and finetuning of BART, a model that is already adept at summarization, yields improvements in summarization quality for journal abstracts published by a *single, unseen* author within that same scientific domain. The domain in this study will be computational modeling of EEG brain rhythms in neuroscience. A model with this level of domain specificity would ideally perform better than out-of-the-box tools on an in-domain task, and would ideally be able to generalize to unseen authors operating in similar domains. Such a model would be highly valuable for scientists, as they need high-quality and domain-specific summaries to 1) assess the relatedness of publications to their own research, and 2) to determine if a thorough examination of a new publication will be fruitful.

2 Background

2.1 BioBERT

There have been numerous efforts in recent years to adapt popular language models to tasks in the sciences. One such example is BioBERT (Bioinformatics, 2020)[2], which is an adapted version of BERT that is pre-trained for biomedical language representation. BioBERT improved upon state-of-the-art performance in biomedical text mining tasks by doing additional pre-training and fine tuning of BERT with only minimal architectural changes, and thus it serves as an example of how additional in-domain training can lead to notable improvements in model performance. While BioBERT shows impressive results in tasks such as name-entity recognition, relation extraction, and question answering, it is not well-adapted for summarization. Since BioBERT is not adept at condensing information into a fluent summary, a significant amount of additional training and fine-tuning on labeled data would be required to adapt this model to produce high-quality summaries that are useful. BioBERT is also trained on a fairly broad domain, biomedical sciences, whereas the present study is focused on applying models to very narrow silos, i.e. those within the scope of a single researcher. As such, additional in-domain training would be needed to increase the specificity of a model such as BioBERT for use in a narrow sub domain.

2.2 SciBERT

A second example of a tool adapted to the sciences is SciBERT (arXiv, 2019)[3]. Also based on BERT, SciBERT employed unsupervised pretraining on a large, multi-domain corpus of scientific publications with minimal changes to the base BERT architecture. The original paper trained numerous versions of SciBERT, including a version trained from scratch using SciVocab and a version using the BaseVocab. While their results showed that SciBERT outperformed BERT-base on scientific tasks, and outperformed BioBERT on a subset of those tasks, they also found that most of the benefit was due to scientific corpus pre-training rather than generating the in-domain vocabulary from scratch. This suggests that training a vocabulary from scratch, while helpful, is not as impactful as continued pretraining. This is important for the present study as it means that establishing a highly-specific in-domain vocabulary, which is more computationally expensive than continued training, is not necessary to produce improved in-domain task performance.

2.3 Enhanced Title Generation as a Motivating Framework

The previous work most related to the present study is Enhancing Academic Title Generation Using SciBERT and Linguistic Rules (Callegari et al., 2023)[4]. Callegari et al. applied BART Large, T5 Large, and Flan T5 to the task of Title Generation from scientific abstracts, training these models on abstracts in both the huggingface scientific papers and the Kaggle ArXiv datasets. They used the publication title as the label, and treated the task as a form of ‘high-level text summarization’. This is similar to the approach used in the present study (see **Methods** below), though Callegari et al. apply certain post-processing methods to the model outputs, which is not done in the present study. The post-processing methods include the use of a fine-tuned version of SciBERT to generate a probability for each generated title compared to the true title, and also the use of a set of ‘linguistic rules’ to select the best titles.

Their first-round tests showed that T5 Large performed better on almost all ROUGE metrics compared to BART Large, and Flan T5 generally performed worse than BART Large. They also showed that the inclusion of SciBERT led to notable increases in ROUGE scores for T5 Large, with the addition of linguistic rules only yielding marginal additional improvements in ROUGE-L Precision. When comparing human rankings of titles to model rankings for a set of 40 abstracts, the T5 + SciBERT + linguistic rules model performed best, getting 10 correct responses compared to 7 for T5 + SciBERT only.

A key limitation of their study, however, is that they only applied the post-processing

methods to their best performing first-round model, T5 Large, and so they cannot assess whether BART or Flan T5 may have performed as well as T5 Large with the additional post-processing methods. A second important limitation is that they only reported ROUGE metrics for BART Large, T5 Large, and Flan T5 for titles generated *after* the additional training on the scientific abstracts datasets. Since they do not provide ROUGE measures for titles generated by the untrained base models, we do not have a 'true' baselines by which to evaluate whether the first-round of training lead to improvements in summarization as measured via the ROUGE scores.

3 Methods

3.1 Datasets

For a single model, BART, the present study builds on the method employed by Callegari et al. (i.e., using the title as the label to fine tune the model for generating summaries)[4] in several ways. First, I apply their framework to a very narrow subdomain in neuroscience, training a model that can perform well at summarizing abstracts by a single author. As such, the test dataset is composed of 16 title-abstract pairs by a single author, Dr. Stephanie R. Jones, web-scraped directly from bioRxiv. Papers by other authors sharing similar names (e.g., Stephanie Jones without the 'R.') that appeared in our query were removed. To build a training dataset of papers related to Dr. Jones' work, I searched for 'EEG Beta Event' (one of her key areas of study) in bioRxiv. The search yielded 3,318 journal articles, and I again scraped the title-abstract pairs directly from bioRxiv. After removing duplicates as well as any papers already present in the test set based on their DOI, I was left with 3,241 title-abstract pairs for fine tuning on summarization.

The present study also expands on the work of Callegari et al. by exploring whether continued MLM pretraining on in-domain data can improve performance above and beyond fine tuning, as this task is computational cheaper than generating summaries for every article. For the additional training, I used abstracts from the same Huggingface Scientific Papers dataset as Callegari et al.; however, I filtered the abstract on key words and phrases within the domain of interest. The filter words were selected by examining the word counts of the article titles (the labels) in the test dataset after removing uninformative filler words. The final words chosen from this list were based on my assessment of their domain specificity, with more specific terms being preferred over more general terms, as the goal is to target a narrower domain of journal articles. The final filter words used were: 'neural', 'beta', 'biophysical', 'cortical', and 'somatosensory', yielding 2,418 unique abstracts.

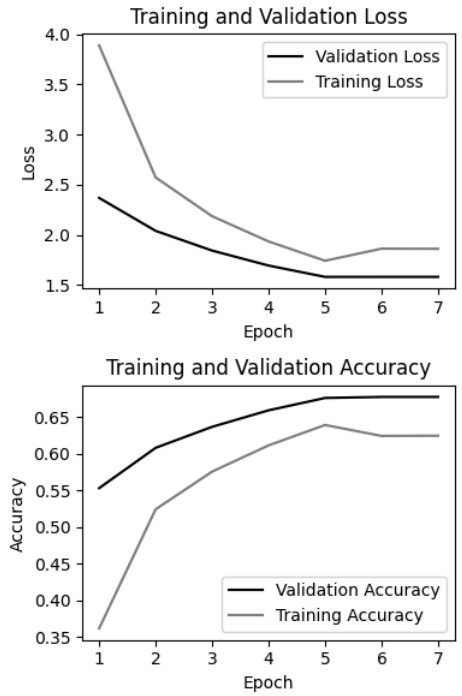

Figure 1: The training and validation losses continue to converge until after epoch 5, where performance on the training data begins to decline. The model consistently performs better on the validation set, suggesting the examples in the validation data may have been 'easier' by chance.

3.2 Continued Pretraining via MLM

As mentioned in the **3.1 Datasets**, I continued pretraining on the base BART model using a word-filtered subset of the Huggingface scientific papers dataset. The dataset was split into training and test sets using a 90-10 ratio. The data were then duplicated so that there were 10 copies of each abstract in each of the datasets. The base BART tokenizer was applied to the datasets, with a max length of 512 and padding. Words were

masked out from each abstract with a 15% probability, ensuring that neither the start-of-sentence, end-of-sentence, nor the padding tokens were part of the mask. When training, I used the Adam optimizer with a learning rate of 2e-4, and cross-entropy loss was used to compute the gradients when looping through the training data only. I ran the training loop for 10 epochs with a batch size of 4, and model checkpoints were saved after each epoch. Loss and accuracy measures for both the training and validation datasets are shown in **Figure 1**.

3.3 Summarization Fine Tuning

Fine tuning for summarization was performed on both the base BART model and the MLM trained BART model outlined above. The checkpoint after epoch 5 was used for fine-tuning, as the training and validation results began to diverge after the 5th epoch. Fine tuning was performed on the papers scraped from bioRxiv as detailed in **3.1 Datasets**. The dataset was split into training and test sets, in this case using an 80-20 ratio. The learning rate was set at 2e-5, the weight decay at 0.01, and AdamW was used as the optimizer. The gradient accumulation step was set to 2 mini batches for training and 3 for validation, and the entire training loop was run for 3 total epochs. The final training and validation losses are shown in **Figure 2** for each of the fine-tuned models. The losses for the MLM + fine tuning (MLM-FT) model begin to converge at 6 epochs (~3,900 steps). The fine-tune-only (FTO) model shows signs of overfitting to the training data after 3 epochs, when the validation loss starts to rise before leveling off at a higher value (with the exception of the final evaluation step, where the training loss abruptly increases and the validation loss concurrently drops). This suggests that fine tuning the base model without the added MLM paradigm makes the model more prone to overfitting on the training dataset for fine tuning. For summarization, I used the FTO model checkpoint at 3 epochs, and I include summaries for the MLM-FT model using both the 3-epoch and the 6-epoch checkpoints for comparison.

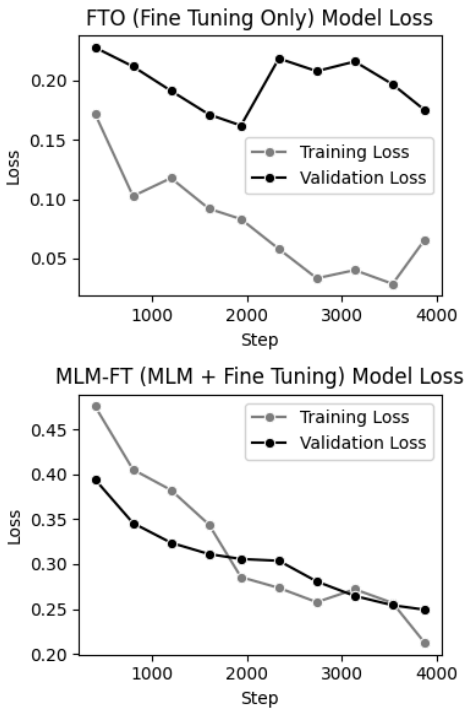

Figure 2: The FTO model shows evidence of overfitting to the training data when training for more than 3 epochs. The MLM-FT model shows convergence beginning around 3,600 steps, or just shy of 4 epochs.

3.4 Summary Generation and Evaluation

Table 1

	A	B	C	D
num_beams	4	4	4	4
no_repeat_ngram_size	3	3	3	3
length_penalty	2.0	2.0	2.0	2.0
max_length	125	50	125	50
temperature	1	1	1.1	1.1

Table 1: hyperparameter sets for summary generation

Table 1 details the hyperparameters used for generating summaries with our BART models. For the present experiment, I was most interested in comparing the differences across the different models, and so I held most parameters fixed, varying only 'Temperature' and 'Max length'. 'Min length' was set at the 25th percentile of the length of the labels in the test data. I tried two values of 'Max length', a higher value and a lower value. I opted to include the lower value for comparison given that the fine tuning was done using short titles as the label (the longest title length was 38), albeit longer summaries are better for our use case, as they would capture more of the nuance in an abstract. I also varied the

temperature of the model for each 'Max length' to introduce more variation into our summary outputs.

For each of the four models (BERT-base, FTO, MLM-FT (3), and MLM-FT (6), I performed 25 total runs (n=10 for A, and n=5 for each of B, C, and D). Each run included 16 summaries, one for each abstract, for a total of 400 summaries per model or 1,600 total summaries. I calculated ROUGE and BERT metrics (specifically, ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-LSum, and BLEU score), which include multiple different measures of semantic overlap between two bodies of text. ROUGE metrics are standard in the field for evaluating summaries and are used by Callegari, Elena et al. for this purpose. BLEU score (which assesses n-gram overlap) is most often used in translation, but it has been adapted for summary evaluation as well.[4] I also included BERT scores, which use contextual BERT embedding to compare candidate sentences (see Tianyi et al., 2019) using Precision, Recall, and F1.[5]

While ROUGE and BERT metrics are fairly standard for comparing two summaries, they are not particularly well suited to the goal of this study, which is to produce and evaluate summaries that have enough depth and specificity to be useful to a scientist operating in a narrow domain. In an attempt to address this, I also used the Mistral-7B, a highly performant model across many tasks, to select a 'best' summary for each set of summaries produced (Jiang et al., 2023)[6]. To achieve this, I prompted the Mistral model to choose the most preferred summary between FTO and the MLM-FT models (see Appendix A1 for the prompt text) for all summaries, and for a subset comprised of 15 of the 'best' runs with the highest scoring ROUGE-LSum (Longest Common Subsequences between summaries weighted by the length of the summary) and BERT Recall (which measures the proportion of relevant words in the reference summary that are captured in the generated summary). For the Mistral outputs, I counted the number of "wins" for each model (the count of the model producing the summary chosen by Mistral, aggregating across the two MLM-FT models). I used these counts to compute the proportion of summaries where Mistral selected one of the MLM-FT models as the final outcome measure.

4 Results

Table 2A below shows the average of the best-run ROUGE results for hyperparameter set A (see **Appendix A2.1** for the full set of results across all sets), and **Table 2B** shows the overall averages for all runs. **Appendix A2** shows the same measures for BERT Scores. FTO, MLM-FT (3), and MLM-FT (6) all notably outperformed the base model on both ROUGE and BERT metrics, showing that fine-tuning does indeed improve scores on these metrics. The FTO model performed better than MLM-FT on all metrics with the exception of being slightly edged out by MLM-FT (6) on ROUGE-1 and BERT Precision, though all ROUGE and BERT measures were fairly close across the three trained models.

Table 2

A. Best Run Averaged Across Abstracts

Model	rouge1	rouge2	rougeL	rougeLsum
base BART	0.2400	0.0641	0.1647	0.1647
FTO	0.4192	0.1778	0.3579	0.3579
MLM-FT (3)	0.4140	0.1755	0.3337	0.3337
MLM-FT (6)	0.4212	0.1635	0.3396	0.3396

B. Average Across All Runs and Abstracts

Model	rouge1	rouge2	rougeL	rougeLsum
base BART	0.2024	0.0438	0.1441	0.1441
FTO	0.3533	0.1346	0.2939	0.2939
MLM-FT (3)	0.3157	0.1099	0.2567	0.2567
MLM-FT (6)	0.3195	0.1103	0.2539	0.2539

Table 3 below shows the results of using Mistral to judge the model summaries, with the metric being the mean proportion of times the Mistral model chose one of the MLM-FT models over the FTO model within each group. **3.i and 3.ii** both suggest that parameter sets A and B produced summaries that were preferable over parameter sets C and D. For all answers and for the 'best' subset of answers, Mistral preferred the MLM-FT summaries at least 50% of the time for sets A and B. The summaries for parameter sets C and D were lower on average, ranging from approximately 45% to 50%. It is also worth noting

that the proportion of times the model chose MLM-FT was slightly higher when including all runs than when looking at the 'best' subset per ROUGE-LSum and BERT Recall. This suggests that ROUGE and BERT measures may not capture key components of the summaries that Mistral uses in choosing a best summary amongst the possible candidates.

Table 3

i.

Metadata Set	Runs (n)	Proportion MLM-FT
A	6	0.5417
B	3	0.5000
C	3	0.4583
D	3	0.4583

Proportion Mistral chose MLM-FT for 'Best' Runs

ii.

Metadata Set	Runs (n)	Proportion MLM-FT
A	10	0.5375
B	5	0.5250
C	5	0.4875
D	5	0.5000

Proportion Mistral chose MLM-FT for All Runs

Appendix A3 shows which summaries were preferred at the abstract level. We see that Mistral selected the MLM-FT model most often for 50% (8 out of 16) of the abstracts when pooling across all hyperparameter sets, and for the same 8 abstracts (though the percentages were higher) when looking at the higher-performing sets A and B. For the other 8 abstracts, percentages varied, with one abstract in particular always yielding 0.

5 Discussion

There are several limitations to the present study that must be considered. First, it is important to note that this study only conducted tests for a single, held out author in one domain. Our results, therefore, do not provide direct evidence that this approach will generalize to other authors or domains. Second, I did not perform an exhaustive test of all possible model hyperparameters, and so I was likely not using the most optimized configuration for either the FTO or the MLM-FT models in the end. Third, I only ran the Mistral evaluations once for the FTO and MLM-FT models. As a next step, I would want to repeat this query multiple times to quantify the variation in Mistral's selections. Furthermore, while Mistral is very performant, it is not specifically trained for what we're asking of it; namely, to say which summary is better in an extremely narrow domain. This is a nuanced task that requires some level of subjective judgment, as we do not have a true 'label' (an ideal summary) to work from for each abstract. Following Callegari, Elena et al., one way to account for this in future research would be to have our best models generate a number of different summaries for each abstract, and have domain-experts rank these summaries either against each other or on an absolute scale. We could use this human feedback as labels to train and improve our model summaries.

In spite of these limitations, the results of this study are promising. They suggest that additional MLM training, which is relatively cheap computationally, has the potential to improve summarization over models that are only fine tuned on summarization, a much more expensive task. There is also reason to believe that there is a great deal of room to improve upon the results presented in this study. For example, while the FTO model showed evidence of overfitting after ~ 3 training epochs (suggesting diminishing returns to additional training), the MLM-FT model was only starting to reach convergence after 6 epochs. It may be the case that we would see further gains with additional training epochs. There are also many opportunities to experiment with different data and different training paradigms. As an example, we could scrape papers citing our particular author of interest, and use those papers in the MLM training paradigm to have more relevant training data. Lastly, as alluded to above, our fine tuning was performed using the paper titles as a proxy for a short summary, but our use case truly calls for much more detailed and comprehensive overviews. Having better labels (or even summary rankings via human feedback as described above) for our training data would undoubtedly improve our model summaries. Regardless, being able to generate summaries for our particular use case that were preferred by Mistral over 50% of the time demonstrates the viability of the approach, even in an unoptimized setting.

References

[1] Landhuis, E. Scientific literature: Information overload. *Nature* **535**, 457–458 (2016). https://doi.org/10.1038/nj7612-457a

[2] Lee, Jinhyuk, et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." *Bioinformatics* 36.4 (2020): 1234-1240.

[3] Beltagy, Iz, Kyle Lo, and Arman Cohan. "SciBERT: A pretrained language model for scientific text." *arXiv preprint arXiv:1903.10676* (2019).

[4] Callegari, Elena, et al. "Enhancing Academic Title Generation Using SciBERT and Linguistic Rules." *Proceedings of the Second Workshop on Information Extraction from Scientific Publications*. 2023.

[5] Zhang, Tianyi, et al. "Bertscore: Evaluating text generation with bert." *arXiv preprint arXiv:1904.09675* (2019).

[6] Jiang, Albert Q., et al. "Mistral 7B." *arXiv preprint arXiv:2310.06825* (2023).

Appendix

Figure A1: Mistral 7-B Prompt Text

```
"""You are a scientist. You will be given an abstract and three
summaries of the abstract, and your job is to determine which summary
does the best job of describing the abstract. Keep in mind, you
should select the summary that will be most useful to you as a
scientist in understanding what the paper will be about based on the
abstract. More specific summaries are preferred, as you want to
understand the fine details of the science.
   Here is the abstract: {abs}
   Here are the summaries you can choose from:
       Summary 1: {sum_finetuned}
       Summary 2: {sum_mlm_ft_3e}
       Summary 3: {sum_mlm_ft_6e}
   Which summary best describes the abstract? You can only choose one
answer.

   Start your response with 'Summary', followed by the summary number
and then a colon.
   For example, if you choose summary 3, start with 'Summary 3:'

"""
```

Table A2

A2.1: Mean ROUGE Results for All Runs and Hyperparameter Sets

Metadata	Model	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
A	base_bart_model	0.202390	0.043808	0.144084	0.144084
	finetuned_model	0.353330	0.134552	0.293884	0.293884
	mlm_ft_3e_model	0.315702	0.109888	0.256652	0.256652
	mlm_ft_6e_model	0.319546	0.110330	0.253929	0.253929
B	base_bart_model	0.212027	0.045424	0.154749	0.154749
	finetuned_model	0.360410	0.132655	0.296063	0.296063
	mlm_ft_3e_model	0.310752	0.094738	0.248914	0.248914
	mlm_ft_6e_model	0.303488	0.098858	0.242314	0.242314
C	base_bart_model	0.198548	0.039134	0.140942	0.140942
	finetuned_model	0.356200	0.138483	0.295452	0.295452
	mlm_ft_3e_model	0.312613	0.098210	0.247944	0.247944
	mlm_ft_6e_model	0.319714	0.117242	0.258063	0.258063
D	base_bart_model	0.201891	0.041722	0.148115	0.148115
	finetuned_model	0.364829	0.146615	0.304234	0.304234
	mlm_ft_3e_model	0.305636	0.101482	0.248382	0.248382
	mlm_ft_6e_model	0.311853	0.096028	0.249591	0.249591

A2.2: Mean ROUGE Results for Best Runs for All Hyperparameter Sets

Metadata	Model	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
A	base_bart_model	0.240027	0.064124	0.164741	0.164741
	finetuned_model	0.419230	0.177766	0.357922	0.357922
	mlm_ft_3e_model	0.413962	0.175503	0.333729	0.333729
	mlm_ft_6e_model	0.421195	0.163484	0.339645	0.339645
B	base_bart_model	0.249821	0.056085	0.172890	0.172890
	finetuned_model	0.399638	0.170030	0.324398	0.324398
	mlm_ft_3e_model	0.370305	0.142676	0.312026	0.312026
	mlm_ft_6e_model	0.399640	0.173959	0.319162	0.319162
C	base_bart_model	0.231853	0.054537	0.161003	0.161003
	finetuned_model	0.399133	0.172921	0.330472	0.330472
	mlm_ft_3e_model	0.387233	0.148823	0.309698	0.309698
	mlm_ft_6e_model	0.374959	0.151544	0.312616	0.312616
D	base_bart_model	0.237756	0.049436	0.167190	0.167190
	finetuned_model	0.413881	0.178180	0.334865	0.334865
	mlm_ft_3e_model	0.365898	0.144858	0.302739	0.302739
	mlm_ft_6e_model	0.389095	0.137738	0.308895	0.308895

A2.3: Mean BERT Scores for Best Run and All Runs for Parameter Set A

A. Best Run Averaged Across Abstracts

Model	Precision	Recall	F1
base_bart_model	0.847459	0.876872	0.859904
finetuned_model	0.884751	0.898202	0.890938
mlm_ft_3e_model	0.889835	0.897664	0.892372
mlm_ft_6e_model	0.891861	0.897321	0.892154

B. Average Across All Runs and Abstracts

Model	Precision	Recall	F1
base_bart_model	0.839080	0.863863	0.851181
finetuned_model	0.872938	0.886233	0.879343
mlm_ft_3e_model	0.873287	0.880210	0.876595
mlm_ft_6e_model	0.874708	0.883119	0.878798

A2.4: Mean BERT Scores for All Runs and Hyperparameter Sets

Metadata	Model	Precision	Recall	F1
A	base_bart_model	0.839080	0.863863	0.851181
	finetuned_model	0.872938	0.886233	0.879343
	mlm_ft_3e_model	0.873287	0.880210	0.876595
	mlm_ft_6e_model	0.874708	0.883119	0.878798
B	base_bart_model	0.841200	0.860158	0.850480
	finetuned_model	0.873504	0.886185	0.879596
	mlm_ft_3e_model	0.873105	0.881098	0.876943
	mlm_ft_6e_model	0.871103	0.879735	0.875302
C	base_bart_model	0.838232	0.863305	0.850470
	finetuned_model	0.871779	0.886264	0.878776
	mlm_ft_3e_model	0.873117	0.881701	0.877226
	mlm_ft_6e_model	0.876263	0.883656	0.879852
D	base_bart_model	0.840562	0.858982	0.849574
	finetuned_model	0.874829	0.888817	0.881563
	mlm_ft_3e_model	0.873604	0.881765	0.877510
	mlm_ft_6e_model	0.875461	0.882799	0.879002

A2.5: Mean BERT Scores for Best Runs and All Hyperparameter Sets

Metadata	Model	Precision	Recall	F1
A	base_bart_model	0.847459	0.876872	0.859904
	finetuned_model	0.884751	0.898202	0.890938
	mlm_ft_3e_model	0.889835	0.897664	0.892372
	mlm_ft_6e_model	0.891861	0.897321	0.892154
B	base_bart_model	0.844866	0.869466	0.856836
	finetuned_model	0.884665	0.895853	0.889745
	mlm_ft_3e_model	0.885210	0.890870	0.887029
	mlm_ft_6e_model	0.884150	0.893185	0.887895
C	base_bart_model	0.844514	0.876820	0.859791
	finetuned_model	0.881416	0.896182	0.887892
	mlm_ft_3e_model	0.885638	0.893477	0.888854
	mlm_ft_6e_model	0.887162	0.894173	0.889998
D	base_bart_model	0.845931	0.865999	0.855417
	finetuned_model	0.886768	0.897405	0.891282
	mlm_ft_3e_model	0.886690	0.892227	0.888065
	mlm_ft_6e_model	0.886693	0.895016	0.889221

Table A3: Proportion of Times Mistral Selected the MLM-FT Model by Question ID for All Sets and for Sets A/B Only

Abstract ID	Proportion (All Sets)	Proportion (A, B Only)
0	0.6	0.69
1	0.2	0.19
2	1.0	0.94
3	0.76	0.75
4	0.68	0.69
5	1.0	0.94
6	0.64	0.56
7	0.72	0.62
8	0.28	0.38
9	0.04	0.06
10	0.4	0.38
11	0.0	0.0
12	0.28	0.25
13	1.0	0.94
14	0.32	0.31
15	0.36	0.31