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

Dylan Daniels

[No talking here]

Problem Overview

- Too many publications, not enough time
- Information overload
- Relying on shortcuts, heuristics

Image:https://img.freepik.com/premium-photo/stress-woman-scientist-with-headache-laboratory-suffering-from-burnout-migraine-pain-overworked-exhausted-frustrated-tired-worker-working-science-research-with-fatigue-tension_590464-150113.jpg

My project was motivated by the trends of increasing volume of publications and increasing specificity in the sciences.

With the huge volume of publications in this day and age, it's become increasingly difficult for scientists to stay on top of all the literature in their field. And as we all know, reading a paper and truly digesting its contents can take a long time. And when we don't have that kind of time, scientists taking shortcuts or relying on heuristics when making decisions. And all of this can lead to information fatigue which is something we'd prefer to avoid.

What would a scientist want from an ‘LLM assistant’ ?

- Given a paper...
  - Summarize key points of an article
  - Find all the bits related to their own research

What does our model need to have?

- High level of domain specificity
- Recognize and extract pertinent information
- Return a fluent, reduced representation

So the idea behind my project is that this is exactly the kind of problem that we tackle with large language models.

[Describe slide]

Why can’t we do that with Chat GPT, Gemini, etc.?

- We can, but ... there are issues with doing so
- These models tend to ‘fill in the gaps’
- We want a model that is a domain expert

Could we build a model from scratch?

- We could, but ...

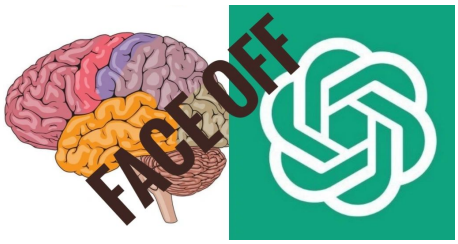


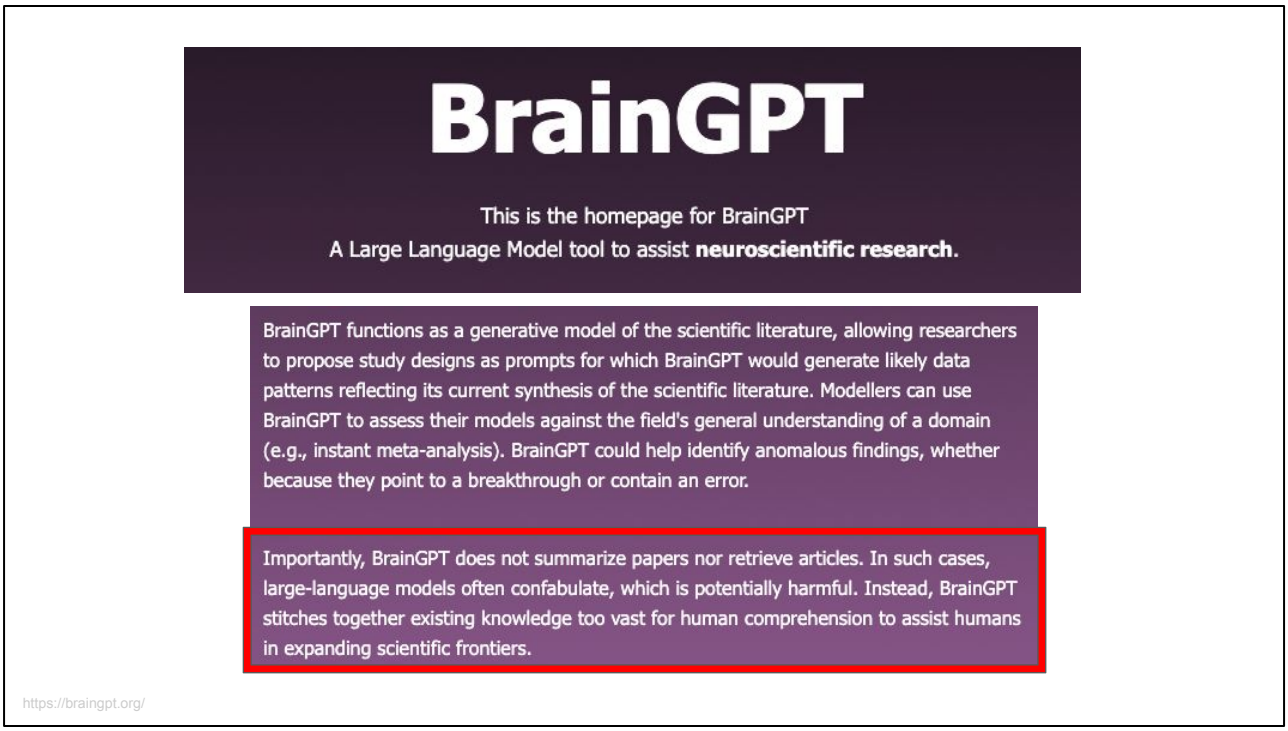
Image:https://hikmaahblog.wordpress.com/wp-content/uploads/2023/04/img_20230404_092315-01.jpeg?w=1024

And you might ask, why can’t we do this with Chat GPT, Gemini, or some other model that’s already out there?

- We can, but there are some issues with doing so
- These models tend to make things up or try to fill in the gaps with something close if they don’t know something
- And this is not good for our scientist. We want a model that is trained in our domain

And you might ask next, well wouldn’t we want to train a model from scratch to truly get that result?

- And we could do that, but it’s very computationally expensive
- And also ... [next slide]



There are people working on these kinds of things already. And it not only takes a long time, but it requires a lot of data. And getting that volume of in-domain data for building these large but narrow ‘generalist’ types of models is a really difficult challenge.

And you can see in this second paragraph that even BrainGPT doesn’t want to tackle summarization, precisely because they’re worried about that ‘filling in the gap’ problem. (Though they use the term confabulate to say it in a much smarter way)

If BrainGPT isn't willing to try, do we give up on this highly-detailed, in-domain summarization?

Image:https://nocamels.com/wp-content/uploads/2015/01/grumpycatmeme.jpg

Well maybe the BrainGPT people are right and this is irresponsible. I'm not sure. But for this project, I said no, we don't give up. We're going to try it anyway and see how far we can get.

The question: can we adapt a publicly-available model to do in-domain summarization ‘cheaply’

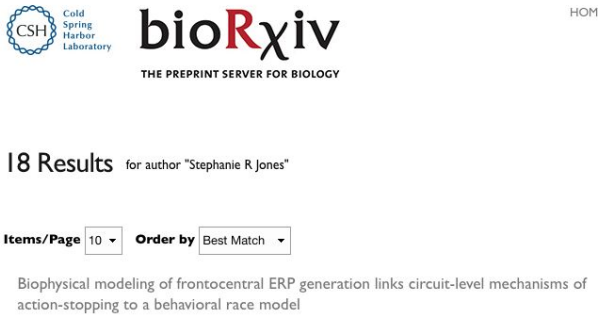
So my specifically, my question is [state question]

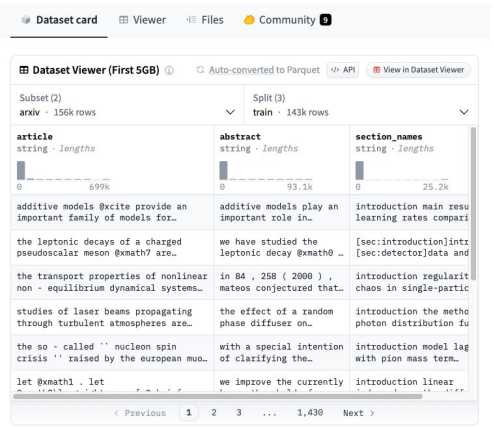
[DON’T SKIP THIS TEXT BELOW]

And as a use case for the project, I’m going to focus on summarization in the narrow domain of modeling brain rhythms in neuroscience, because that’s what I do professionally and that’s what I’m interested in.

But you could imagine adapting this approach to any narrow domain in the sciences

The data





So what data did I use for this? I'm going to describe this in reverse and start with the test data

So what I did, is I scraped all the titles and abstract from a single author, Dr. Stephanie Jones, off of bioRxiv. Dr. Jones is my PI and she does modeling of human brain rhythms. And I used her papers make up the test set that we'll evaluate again.

Next, I filtered my search on some keywords related to Dr. Jones' research, and I scraped about ~3,000 papers from that query. This makes up my dataset for fine-tuning on summarization.

Lastly, I grabbed the scientific papers dataset off of huggingface, and I similarly filtered it on some in-domain keywords. That gave me another ~2,500 abstracts, and I use these for continued pretraining with masking.

The models

BART Base

BART Fine-Tuned on In-Domain Summarization

BART With MLM Training and Fine Tuning In Domain

So for my experiments, I wanted to compare three model.

One is just the base BART model with no changes to serve as a baseline.

Second, I wanted to fine tune Bart on summarization within our domain and see how much better we can do. And to do this, I followed an approach by Callegari, Elena et al. And what they did is that they created a dataset from journal article titles and abstracts, and they used the title as a proxy for a very short summary of the abstract. And that's the approach I borrowed here.

But I want to note that this fine tuning is still pretty computationally expensive, because we have to read a long piece of text and then generate more text. And if we expand this to read entire articles and generate longer, more useful summaries, it would be even more expensive. So we want to minimize fine tuning as much as possible.

And that's where the third approach comes in. And the idea here was to do something 'cheaper'. So I continued the BART pretraining by using a

masked language model, and had BART predict masked out words. This task is computationally way easier, and so we can theoretically do a lot more of this type of training if it gives us better results.

ROUGE / BERT Score Results

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

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	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

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

So I did the training and had all the models generate summaries on the test data, and these are the results. And I'm showing ROUGE and BERT measures here, which we all now about.

And we see that both of the trained models do better than the base model, so that's great but not very surprising.

And in red i've highlighted where the MLM models do better. And we see that it does better in some domains, but those margins are really slim. So slim that I forgot to even mention the improvements in F1 in my paper writeup.

Image:https://lh5.googleusercontent.com/proxy/_6_O3nSUGrEsA_giqDBs-qGFKhtHNTvdiFZLa2IOt6SdVm9IM8_sQiYyY34x7gicPXIUTV5LjgxGQf_MzQFxo16q8Id4vP96fSIq6fU27Rh64_0v5hEpaP3toKHbBLEmDx55JkPZarpCzz-1K8cGOG8Sd6t3

And this led me to the question, are these evaluation metrics really capturing the key information we care about?
That is, are the summaries comprehensive enough, specific enough, and detailed enough to be useful?

Probably not so much. And there are a lot of reasons for that I I'll leave that to your imaginations and just tell you what I did next.

A better approach? (maybe...maybe not)

Image:https://docs.mistral.ai/img/logo.svg

Idea: use publicly available, 7b param model to judge these summaries

Used a quantized version so it would fit on our GPU

The advantage of using a model like this is that we can build a prompt, and be a lot more specific in that prompt about what we're looking for and what constitutes a 'good' summary

```
"""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:'
"""
```

And so here's the prompt that I ended up using. And I bolded some of the areas in here to show how I used this prompt to give Mistral some direction about what it should be looking for.

For example, it can be helpful to set a 'role' for the model, so I told it it's a scientist.
I told it what my preferences were
And told it that more specific and detailed summaries are better

And lastly I fed it the abstract and all of the summaries I wanted it to look at

And then I let it be the judge and give me it's result. And it also included a justification for why it chose the summary it chose, which is interesting, but I won't get into that here.

But you can see how there's a lot of room here to experiment with the prompt and kind of constrain the problem we're trying to solve

So what did Mistral think?

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

Proportion of the time Mistral chose MLM-FT for 'Best' Runs

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

Proportion of the time Mistral chose MLM-FT for All Runs

Image:https://i.kinja-img.com/image/upload/c_fill,h_675,pg_1,q_80,w_1200/f13b27b54cb5e11c7505487ddda2a1b9.png

So the left table is looking only at the 'Best Runs' based on ROUGE and BERT metrics.
And the right table is looking at all the runs, so all the summaries I generated.

And then we have four different hyperparameter sets to compare.

And the metric here is the proportion of the time that Mistral preferred the MLM over the fine-tune only model.

And on average, we see that it preferred the MLM model over 50% of the time. And this was especially true for hyperparameter sets A and B, where the range is from 50 to about 54 %.

And I would say this is a pretty meaningful result. What we've done is through this MLM approach, we've generated summaries that Mistral prefers over the fine-tune only model over 50% of the time. I would call that a meaningful improvement.

And I want to highlight that this was only with some minimal MLM

training. I definitely didn't reach the ceiling of possible improvements here.

And this isn't shown here but I think it's an important point. My fine-tune only models showed evidence of overfitting after only 3 training epochs, so I had to stop there. But my MLM model was only just starting to show convergence after 6 epochs. And I didn't train beyond that, but there was definitely still room for improvement where I stopped.

Limitations, Ideas for Improvement

Image:https://cloudarchitectmusings.com/wp-content/uploads/2016/01/625955-whats-next-meme.jpg

So I want to be mindful of some limitations to this study

- Only done on a single author
- Using abstracts instead of full articles
- Only a single pass with Mistral as its computationally expensive
- Mistral isn't specifically trained for what we want - it's a bit of a black box. But it does give you a justification for the summaries, so we do have its 'thought process' so to speak. But we're still just hoping for the best.

Areas for Improvement

- Using full articles
- Different training data (e.g., MLM on older papers by the same author, or author who cite out main author)
- Better evaluation metrics, and there are even some like SciBERT that are trained to do evaluation for these niche purposes
- HUMAN FEEDBACK for evaluating (e.g., ranking summaries generated by models, feeding those ranking back into the model so it can learn what our scientists prefer)

And that's all I have, thank you so much for your time and I hope you

enjoyed it.