An Investigation into Discourse Probing of Pretrained Language Models

Jacob Hotz
University of British Columbia
jhotz@student.ubc.ca

Abstract

Language models (LMs) have knowledge of discourse, and (Koto et al., 2021) examines what types of LMs fare better or worse when tested with tasks related to discourse. I examine a surprising result—T5 (Raffel et al., 2020) vastly underperforming—and show that it is due to incorrect model parameter sizes. I conduct experiments that show that results are actually in-line with what the language models' pretraining would suggest and that T5 does not underperform.

1 Introduction

Discourse in the context of NLP revolves around understanding of coherence, relationships between entities in a text, and an understanding of order. Language models have some knowledge of discourse, and (Koto et al., 2021) conducts a study determine how much knowledge current relevant models have in an aim to understand different architecture choices and if any model is best at capturing discourse.

Discourse can be of interest as a useful tool in downstream applications such as summarization (Xu et al., 2020; Dong et al., 2020). It can also serve as a guide to designing strong benchmarks for LMs, either to test them or to understand them (Huber and Carenini, 2022; Pandia et al., 2021).

(Koto et al., 2021) reveal interesting insights in terms of the results they report. They find that the final layers of encoders tend to capture the most discourse information in general. This makes sense, as models trained on masked language modelling or span denoising are incentivized to best capture information in the final stage of the encoder. If the model has a decoder, the decoder is incentivized to form representations of what the output vocab token will be and eliminated certain words, so we can posit that there may be information gradually lost as the selection process advances.

They find that BART (Lewis et al., 2019) performs overall the best. However, I believe that ELECTRA's performance is also at least equally noteworthy given it consumed a much smaller training corpus—16GB as opposed to 160GB—and a slightly smaller parameter count, as well as having a unique training regimen as a language discriminator rather than generator (Clark et al., 2020).

One surprising result was T5's poor performance (Raffel et al., 2020). Both T5 and BART have a similar pre-training objective in that they denoise masked spans of text. T5 also has a larger training corpus. The models that the paper uses are `bart-base` and `t5-small`, which are listed in the paper as having similar parameter sizes.

2 Initial investigation into models

I first examined the models the paper uses, and saw they they are using `bart-base` and `t5-small` from the HuggingFace Transformers library.[1]

I then printed the model details [2], and found that `t5-small` has a dimensionality of 512 for embeddings, whereas `bart-base` has a dimensionality of 768. Furthermore, the feed-forward layers of `t5-small` have 2048, whereas BART's are 3072. These details are captured in are captured in Figure 1 and in Figure 2.

After confirming this by looking at the PyTorch object, I looked for further divergence in parameter size by searching for how many attention heads each has. This can be found now in Hugging Face documentation, and `t5-small` has 12 whereas `bart-base` has 16.

In total, these differences amount to `bart-base` having 139 million parameters and `t5-small` having just about 60 million.

[1] https://huggingface.co/transformers/v3.3.1/pretrained_models.html

[2] Code for my project can be found at https://github.com/hotzjacobb/cpsc532g and https://github.com/hotzjacobb/discourse_probing/tree/compute_can_python37

Model	Type	#Param	#Data	Objective
BERT	Enc	110M	16GB	MLM+NSP
RoBERTa	Enc	125M	160GB	MLM
ALBERT	Enc	11M	16GB	MLM+SOP
ELECTRA	Enc	110M	16GB	MLM+DISC
GPT-2	Dec	117M	16GB	LM
BART	Enc+Dec	139M	160GB	DAE
T5-Small	Enc+Dec	60M	866GB	DAE
T5-Base	Enc+Dec	220M	866GB	DAE

Table 1: Revised summary table of English LM's used in (Koto et al., 2021) with the addition of T5-Base. The parameter counts are fixed as compared to the original paper. "MLM" = masked language model, "NSP" = next sentence prediction, "SOP" = sentence order prediction, "LM" = language model, "DISC" = discriminator, and "DAE" = denoising autoencoder

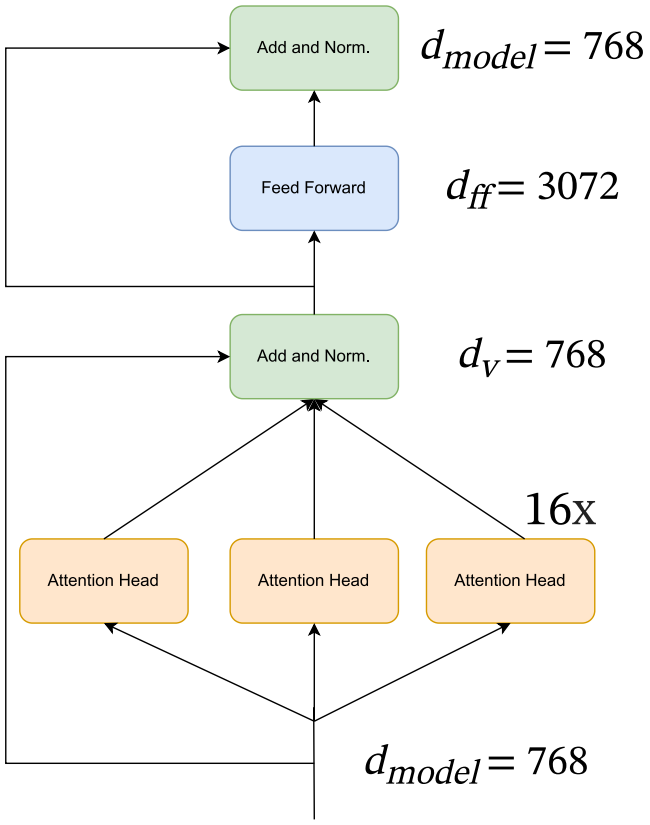

Figure 1: Size of bart-base

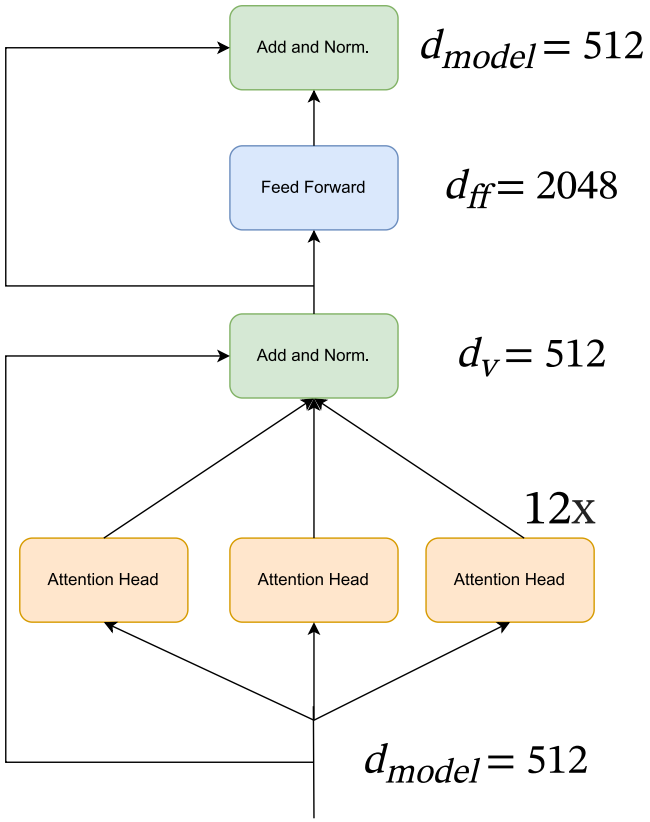

Figure 2: Size of t5-small

These differences are captured in Table 1, as well as updated counts for other models used in (Koto et al., 2021).

2.1 T5: more training data not a strict advantage

A likely explanation for `t5-small`'s poor performance on discourse tasks is it's smaller size when compared to other models. Even though it saw much more training data, it likely was unable to store more information in it's parameters.

Raffel et al., 2020 does not comment on how much of the proprietary corpus `t5-base` uses, but likely training after a point returns no further reduction in perplexity or model performance.

Thus, I hypothesize that parameter count is the limiting factor.

3 Experiments

I initially wanted to run all experiments to probe different layers of the model for latent discourse knowledge as done in (Koto et al., 2021). However, I had two main issues:

- **Unable to fully reconstruct steps**: Of the original paper's seven tasks, three use the Penn-Treebank (Carlson et al., 2001). I was able to obtain this through my institution, and thus was planning to run the tasks. Koto et al., 2021 provide a source code repository[3], but the authors mention that there are no instructions or code to convert the .rst files in the original dataset to .rs3 files that their code operates on. Thus, I decided to just run on the four tasks that do not require the Penn-Treebank.

- **Lack of compute**: For early stages of my project I had little or no GPU accessibility. I eventually was able to run my code on 4 x NVIDIA P100 Pascal GPU's more reliably, after fixing packages to work for that environment. Still, this was not enough compute to run all of the orignal paper's tasks. Particularly of note, I was not able to run the same experiments with three different seeds due to my compute, instead having to settle for running each experiment with only one seed.

[3] https://github.com/fajri91/discourse_probing

Probing Task	Dataset
4-way NSP Sentence Ordering	XSUM Articles (Narayan et al., 2018) Split: 8k/1k/1k
Discourse Connective	Sampled DisSent Dataset (Nie et al., 2019) #Labels: 15 Split: 8k/1k/1k
Cloze Story Test	(Mostafazadeh et al., 2016) Split:1683/188/1871

Table 2: Subset of tasks (originally in Koto et al., 2021, see that paper for more details on setup) used for testing T5-Base. "Split" indicates the number of train/development/test instances.

3.1 Choice of T5 model

To be able to have a fairer comparison with the models used in the *Discourse Probing*, I ideally wanted to use a model with the same amount of parameters. However, there is no publicly available pre-trained model with such characteristics, so if I wanted to do an exact comparison, I would have to pre-train a model.

I don't have the requisite compute needed for that, so instead decided to opt for `t5-base`. `t5-base` does have about 220 million parameters, which would ostensibly give it an advantage over `bart-base`.

3.2 Method for a fair comparison

However, as a key insight, we can actually just look at the first six layers of `t5-base` when running the tasks. This is a more than fair comparison, as it actually gives an advantage to `bart-base` in terms of parameters. Why?

`t5-base` has more parameters solely because of that it has twice the amount of layers, with 24 instead of 12. It has the same dimensionality for token representations as `bart-base`, and less attention heads: 12 versus 16.

Thus, by limiting ourselves to the first 6 layers, `bart-base` actually has more parameters and thus an advantage. `bart-base` has a further advantage; Koto et al., 2021 finds that in encoder-decoder models (BART and T5 are both encoder-decoder models), the layers that tend to perform best are those that are the end of the encoder.

This can be explained by the fact that, as they state, the decoder is more focused on output to the fixed vocabulary than a semantic representation. Furthermore, the last encoder layer must represent tokens semantically as it is a bottleneck for the decoder. However, previous layers can be more flexible in trying to compute information and do not necessarily provide information as a general representation of the tokens.

3.3 T5 still at a slight disadvantage

Given this, combined with `t5-base`'s disadvantage in parameter size due to less attention heads, I believe that improvement of `t5-base` over `t5-small` should be notable, and that `t5-base` returning results on par with `bart-base` should be interpreted as it doing better given its disadvantage, nor can it be dismissed if it reduces the margin to `bart-base`.

4 Results and Analysis

The results are presented in Figure 3.

We note that `t5-base` significantly improves upon `t5-small`. This was to be expected, as any model with an identical architecture, but scaled up in terms of size should perform as good or better. So the increase in model dimensionality does help T5, but overall performance is still not as high as `bart-base`.

The fact that `t5-base` is at a disadvantage to `bart-base` even in this setup was noted earlier, thus any definitive analysis here is impossible when comparing T5 and BART. However, it it notable that `t5-small` had some of it's worst results on the tasks not examined in this study (RST tasks). Therefore, its performance here in narrowing the gap is noteworthy as those tasks where it did not have such a large deficit were not present. This can be seen in the significant gain of average of normalized scores.

Furthermore, it's interesting that on one of the the tasks that it did worst on in Koto et al., 2021,

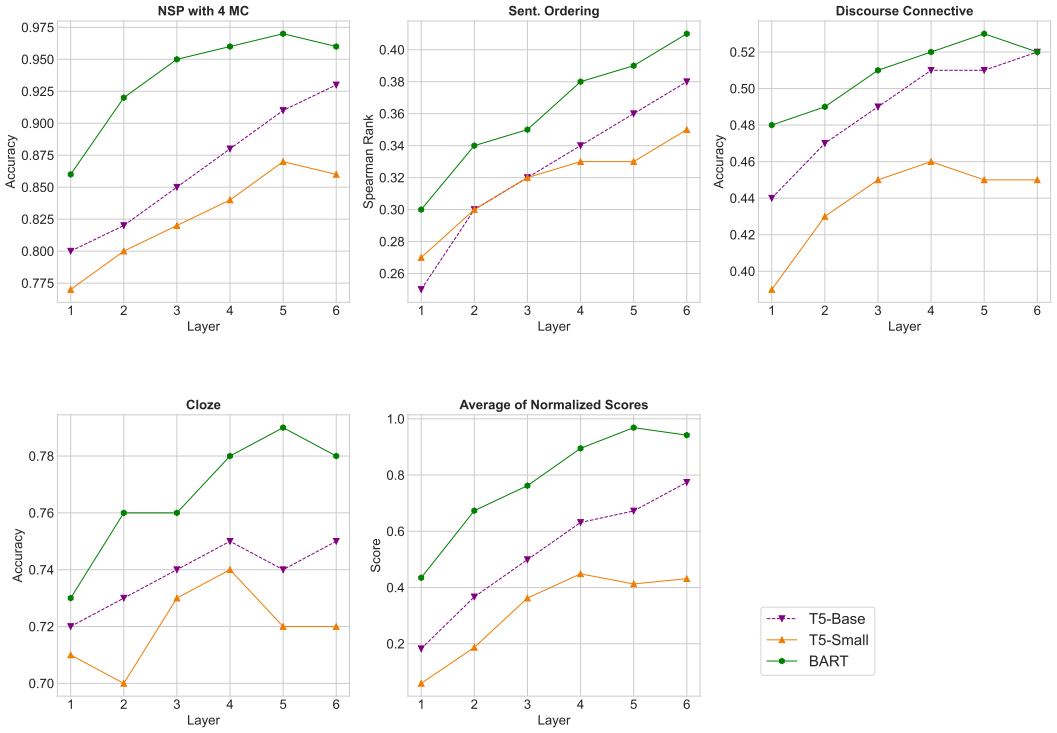

Figure 3: Results from experiments. T5-Small and BART were done in (Koto et al., 2021), whereas T5-Small are new experiments

Discourse Connectives, it greatly improved and closed the performance gap. This can be interpreted as it having an better result than BART, given its disadvantages.

The task that we see the least improvement on is the Cloze Story Test. This is understandable considering that here, we are only looking at models' encoders, and in *Discourse Parsing of Pretrained Language Models* this task was an anomaly as most models performed just as well or even better in their decoder as in their encoder. Thinking of reasons for this, it could potentially be due to the fact that the encoder produces an aggregate representation of a sequence (The Cloze Stories). Potentially, different methods such as SBERT (Reimers and Gurevych, 2019) capture the overall meaning of the sequence better.

5 Meta-analysis of the author

5.1 Lessons Learned

I will analyze my own work here with the project. If by any chance reading the paper for the results, feel free to skip this.

On this project, I decided to base my work off an existing paper. I was genuinely curious about its results and seeing if it could be explained, as sometimes in ML, papers are published with no apparent explanation for results witnessed (barring divine benevolence, see Shazeer, 2020 if curious), which I understand, but I though this result made a lot of sense to analyze since I had intuition that I could find why.

In the end, I spend a ton of time working on dependencies and being able to run on compute clusters so that I could train models with GPUs. Given that Koto et al., 2021 is now two years old, it was inevitable that a number of packages would not work, and this took a lot of engineering effort. In future, I think that I would like to do my own original projects. I did make changes to the code for this in allowing `t5-base` to run (changing encoder-decoder logic, adding the model), but got to this phase late due to my inability to run the code for a while.

I would definitely work in a team if I were to do a course project again, as here I worked alone and working in a team would likely be more productive and more fun.

One last thing is that I vastly underestimated how much time it would take me to write this report. Part of it was me being stubborn trying to get a benchmark on ELECTRA vs. other encoder models that I didn't get finished on time to include in this. But another part was just underestimating

the time it takes.

5.2 Reflections

I am happy that I was able to run non-trivial experiments by solving some of the software engineering problems required, and that I found the cause of this surprising result and could explain it. I also think that I picked an interesting question, so I'm glad about that, even if the result turned out to be not as complex as one might have thought.

In terms of project outline/planning, I think that is an area where I can criticize myself. I think of the original three objectives; T5 results explanation, ELECTRA use downstream, and ELECTRA pretraining alterations. ELECTRA pretraining was unrealistic, and would merit a whole project itself. Using ELECTRA downstream in a task was viable, and I spent time adapting code from this paper and writing my own for that purpose (available in my repository), and even finished the code for it, but I couldn't debug it or run it. I eventually ran into a dependency issue (due to adapting the older code). I spent a lot of time on this project, but a lot of it was in trying to get my code running and not working on it or exploring further ideas once I had a baseline

I probably should emphasize that I don't plan to work with a pre-existing paper if I were to do a course project unless I believe the code to be recent and robust. Also I want to explore more unique ideas if possible.

6 Future Work

The best possible comparison would be to pre-train models here with the same sizes and then compare. It doesn't seem to me like BART or T5 are opinionated about optimal attention head to feedforward dimensionality ratio, so thus a head-to-head comparison pre-training on the same corpus would seem to be definitive.

Finishing the last steps for benchmarking ELECTRA compared to a similarly sized encoder models remains to be done.

Another idea that came up in the course of this work, due to thinking about encoders as related to decoders, would be pre-training encoder models with different classifiers for different masked languages. Perhaps the encoder would learn a better semantic representation of the input sequence not constrained by one language. On a different note but slightly related, it would be interesting to see if works like SBERT (Reimers and Gurevych, 2019) are do better on the Cloze Story Task given my hypothesis about being able represent an input sequence that might have surprising continuations.

References

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.

Patrick Huber and Giuseppe Carenini. 2022. Towards understanding large-scale discourse structures in pre-trained and fine-tuned language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2376–2394, Seattle, United States. Association for Computational Linguistics.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Discourse probing of pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3849–3864, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Allen Nie, Erin Bennett, and Noah Goodman. 2019. DisSent: Learning sentence representations from explicit discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4497–4510, Florence, Italy. Association for Computational Linguistics.

Lalchand Pandia, Yan Cong, and Allyson Ettinger. 2021. Pragmatic competence of pre-trained language models through the lens of discourse connectives. In *CONLL*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

A Source Code

Source code can be found in two places for this project:

- For changes made to (Koto et al., 2021) adding `bart-base` and necessary logic, as well as changing requirements package files and scripts to run, please see `https://github.com/hotzjacobb/discourse_probing/tree/compute_can_python37`
- For a standalone notebook comparing model sizes as well as code adapted for the ELECTRA sentiment analysis benchmarks that I did not finish, please see `https://github.com/hotzjacobb/discourse_probing`

B Full Experiment Results

Layer	NSP	Sent. Ord.	Discourse Con.	Cloze St.
BART-Base				
1	.86	.30	.48	.73
2	.92	.34	.49	.76
3	.95	.35	.51	.76
4	.96	.38	.52	.78
5	.97	.39	.53	.79
6	.96	.41	.52	.78
T5-Base				
1	.80	.25	.44	.72
2	.82	.30	.47	.73
3	.85	.32	.49	.74
4	.88	.34	.51	.75
5	.91	.36	.51	.74
6	.93	.38	.52	.74
T5-Small				
1	.77	.27	.39	.71
2	.80	.30	.43	.70
3	.82	.32	.45	.73
4	.84	.33	.46	.74
5	.87	.33	.45	.72
6	.86	.35	.45	.72

Table 3: Full results for all three models. BART-Base and T5-Small taken as reported from (Koto et al., 2021)