Email Content Extraction
COS485 Final Project

Simon Park
Princeton University
juhyunp@princeton.edu

Min Lee
Princeton University
mylee@princeton.edu

Abstract

In this paper, we propose a double fine-tuning method for an email content extraction task with limited availability for annotated data. We compare the results of 1) a task-adaptive approach where a pre-trained model is first fine-tuned on a large dataset with a similar task as the target dataset before being fine-tuned again on the small target dataset and 2) a domain-adaptive approach where the dataset for the first fine-tuning stage is replaced with one with a similar data domain as the target dataset. We observe that both approaches are effective at aligning the pre-trained model for the downstream task. The code is available at: https://github.com/minniie/email_content_extraction.

1 Introduction

There are 392 undergraduate student organizations at Princeton University. On an average day, a Princeton student receives around 30-40 emails promoting events that these organizations host. The overwhelming amount of information present in the emails often discourage students from reading the emails in detail. Our work aims to extract key information from each of the emails and present it in a digest format. Based on the output of our email, users may choose to cherry pick the selected number of emails they wish to read in detail.

2 Background

2.1 Information Extraction via Question-Answering

Extracting information from a given passage is often performed in a question-answering (QA) setting — a question about the passage is appended at the end of the input, and the model is expected to output the answer to the question. When the answer exists in its exact form in the passage, we say that the task is **extractive**, and encoder-only Transformer models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) can achieve state-of-the art results by predicting the start and end indices of the answer substring. When the answer does not exist in the passage, we say that the task is **generative**, and we need the text generation abilities of decoder-based Transformer models like GPT (Radford et al., 2018) and GPT-2 (Radford et al., 2019).

2.2 Double Fine-tuning

One recent trend of NLP research has been to take a model pre-trained on a broad range of domains (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2018, 2019) and to **fine-tune** it on a specific task or domain. However, when there is limited availability of target data, fine-tuning the model on the target data can be difficult. Instead, we leverage the idea of **double fine-tuning** from Jeawak et al. (2020); Anonymous (2022); Ko and Choi (2020), where the pre-trained model is first fine-tuned on a large dataset that is similar (same task or same domain) to the desired target dataset, then fine-tuned again on the small target dataset.

2.3 Previous Works

In alignment with previous works, our model can be understood as performing a short-answer extractive [1] QA task on an email data domain. There are works that perform a similar task on a different domain, or performs a different task on a similar domain. However, to the best of our knowledge, there is no previous work that performs a short-answer QA task on an email dataset.

Similar task but different domain Stanford Question-Answering Dataset (SQuAD) (Rajpurkar et al., 2018) is an extractive question-answer task,

[1] Even though some answers are formed by concatenating substrings drawn from multiple parts of the passage, all answers are entirely contained in the input sequence. We abuse the terminology slightly and refer to this task as extractive.

where the passages are sampled from Wikipedia articles.

Similar domain but different task EmailSum (Zhang et al., 2021) is a summarization task, where the input data is drawn from existing email collections: Enron email corpus (Klimt and Yang, 2004), Avocado Research Email Collection (Oard et al., 2015), and W3C email corpus (Craswell et al., 2005).

3 Approach

For our email content extraction model, we propose double fine-tuning a decoder-based generative model. Since some of the answers span across different locations in the email, we need to leverage the text generation abilities of a decoder. We compare the performance of a task-adaptive approach, where the first fine-tuning is performed on a dataset with a similar task (extractive QA) as the target dataset (Princeton email), against a domain-adaptive approach, where the first fine-tuning is performed on a dataset with a similar domain (email). For both approaches, the second fine-tuning will be performed solely on the target dataset.

4 Experimental Setup

4.1 Data Preprocessing

4.1.1 SQuAD

SQuAD is downloaded from Huggingface. We use the full dataset with the original train and validation splits. Following conventional approach, the question is appended at the end of the context paragraph, where a special end-of-text token is inserted before and after the question. The resulting statistics for SQuAD is shown in Table 1.

# Train Samples	# Valid Samples
87,599	10,570

Table 1: Statistics for SQuAD.

4.1.2 Enron Email Corpus

Enron email corpus is downloaded from https://www.cs.cmu.edu/~enron/. We extract only the email content (body and subject line) and discard all other metadata (sender, recipient, etc.). From the email body of forwarded emails, we delete the header that contains metadata about the original email. Emails that are too short (less than 1000 characters) or too long (more than 5000 characters) are discarded. To align the task in a QA format, the input consists of the email body followed by the question "What is the subject of this email?" (with the end-of-text token before and after the question). The expected output is the subject line of the email. We perform a $4:1$ split of the filtered dataset into train and validation sets. The resulting statistics for Enron emails dataset is shown in Table 2.

# Train Samples	# Valid Samples
89,412	22,353

Table 2: Statistics for the Enron email corpus.

4.1.3 Princeton Email Dataset

We crawl promotional emails from our own Princeton Gmail inbox using Google API. Approximately 500 emails sent through https://hoagie.io/ [2] during April 10-28, 2023 are initially collected. Emails that are not promoting events are filtered out. If there are multiple emails promoting the same event, only one is chosen. From each remaining email, both authors manually label 5 different key information (title, hosting organization, location, time, and guests of the event). For each piece of information that exists in the email, 1 data point is generated, totalling a maximum of 5 data points per email. An example is shown in Table 3.

Email	Hi everyone! MASA is excited to host a Hari Raya dinner event on Sunday, April 23 7.30 - 8.30pm. Venue will be at Louis A Simpson B60. As usual there will be lots of delicious local food! Please RSVP here ASAP if you can make it. Guests are welcome. Hope to see you there! Best, Justin Ong
Title	Hari Raya dinner event
Host	MASA
Location	Louis A Simpson B60
Time	Sunday, April 23 7.30 - 8.30pm
Guest	-

Table 3: Four data points are generated from this example email from Princeton email dataset.

The input of each data point consists of the email body, followed by the natural language question corresponding to the type of content extracted from

[2]Hoagie Mail is a service to send promotional emails to all undergraduate students via residential college listservs.

the email (with the end-of-text token before and after the question). The questions are shown in Table 4.

Title	"What is the title of the event?"
Host	"What is the hosting organization of the event?"
Location	"Where does this event take place?"
Time	"When does this event take place?"
Guest	"Who are the guests of this event?"

Table 4: Questions for each content type of Princeton emails dataset.

The dataset is split into train and validation sets with ratio of $4:1$. The statistics of the final preprocessed dataset is shown in Table 5.

# Emails	# Train Samples	# Valid Samples
198	615	153

Table 5: Statistics for Princeton emails dataset.

4.2 Model

We use GPT-2 Medium (355M parameters) (Radford et al., 2019) as the pre-trained base model for our main experiment. In the first fine-tuning stage, we take two copies of the base model and fine-tune each separately on SQuAD and Enron email corpus. In the second fine-tuning stage, we take each fine-tuned copy and fine-tune it further on the Princeton email dataset. Due to physical limitations, we truncate the beginning of the input text as necessary if the number of input tokens exceeds 512 for any of the experiments.

4.3 Training Details

Each model is trained on SQuAD or Enron email corpus for 10 epochs and on Princeton email dataset for 20 epochs, where the gradient is computed from the cross entropy loss on the generated tokens. Batch size is fixed as 16 for both training and evaluation, by choosing an appropriate batch size per device and gradient accumulation step size. Learning rate is chosen as 5×10^{-5} with linear decay. AdamW (Loshchilov and Hutter, 2019) is chosen as the optimizer. On a single NVIDIA Titan RTX, the first fine-tuning takes approximately 18 hours and the second fine-tuning takes 20 minutes.

4.4 Evaluation Metric

We use the $\mathbf{F_1}$ **score** to evaluate the final model. Let $\mathbf{w}^{(o)} = w_1^{(o)} \cdots w_m^{(o)}$ be the output of the model on a particular input and let $\mathbf{w}^{(g)} = w_1^{(g)} \cdots w_n^{(g)}$ be the gold answer. Then the **precision** is defined as $p = c/n$ and the **recall** is defined as $r = c/m$ where c is the number of words that appear both in the gold answer $\mathbf{w}^{(g)}$ and the output $\mathbf{w}^{(o)}$. Then the F_1 score for the data point is computed as

$$F_1(\mathbf{w}^{(o)}, \mathbf{w}^{(g)}) := \frac{2pr}{p+r} \tag{1}$$

and the F_1 score for the dataset is computed as the average of the individual F_1 scores across all data points. During evaluation, we remove all punctuation marks, articles, and whitespace before converting the output and groundtruth text to lowercase.

5 Results

5.1 First Fine-tuning

5.1.1 SQuAD

Figures 3 and 5 show the training and evaluation losses when first fine-tuning the base model on SQuAD. The training loss generally decreases throughout the 10 epochs, with most of the decrease coming in big steps at the end of each epoch. This suggests that some degree of memorization of training examples is happening. Also, the evaluation loss decreases only during the first epoch and increases afterwards. This also suggests that the model overfits after the first epoch. Therefore, we choose the checkpoint at 1 epoch to be the model we use for the second fine-tuning stage.

5.1.2 Enron Email Corpus

Figures 4 and 6 show the training and evaluation losses when first fine-tuning the base model on the Enron email corpus. The training loss decreases steadily over the 10 epochs, while the evaluation loss decreases sharply until 4 epochs and increases slightly for the remainder of training. We choose the checkpoint at 4 epochs to be the model we use for the second fine-tuning stage. We conjecture that Enron email corpus requires more epochs for convergence than SQuAD because the task is given as generative QA, which is more difficult for the model to perform than an extractive version.

5.2 Second Fine-tuning

5.2.1 Training and Evaluation Loss

Figures 7 and 9 show the training and evaluation losses when second fine-tuning each of the first fine-tuned models on the Princeton email dataset. For both models, the training loss decreases steadily throughout 20 epochs for both SQuAD-based and Enron-based models, and the evaluation loss decreases sharply until 1 or 2 epochs and steadily increases afterwards. Whereas the exact values of the training loss are also almost identical between the two models, the evaluation loss is consistently lower for the SQuAD-based model. Also, we observe that the evaluation loss of the Enron-based model increases faster for later epochs than the SQuAD-based model.

5.2.2 F_1 Score

Figure 1 shows the F_1 scores at each training step for both SQuAD-based and Enron-based models. Throughout training, the SQuAD-based model consistently outperforms the Enron-based model. For both models, the F_1 scores increase sharply for the first few epochs and increase gradually afterwards, even after the models start overfitting. This is most likely because the cross entropy loss compares the model output to the gold answer token-wise, and hence is highly sensitive to variations or alternate forms of the output. On the other hand, the F_1 score only takes into account the amount of overlap between the model prediction and gold answer, and hence is more resilient to alternate forms of outputs. The memorized information from the first few epochs may make it harder for the model to make the correct prediction for each token, but it allows the model to generate an output that is more accurate overall. For the second fine-tuning phase, we select the checkpoint with the highest evaluation F_1 score.

Figure 2 shows the F_1 scores of SQuAD-based and Enron-based models before any fine-tuning or after each stage of fine-tuning. Each stage of fine-tuning is shown to improve the model performance on email content extraction. Both task-adaptive and domain-adaptive approaches prove to be effective at aligning the pre-trained model for the target downstream task. Of the two approaches, the task-adaptive approach exhibits a slightly better transferability between the two fine-tuning stages. This is likely because extractive QA setting of SQuAD encourages the model to find the output from the provided input.

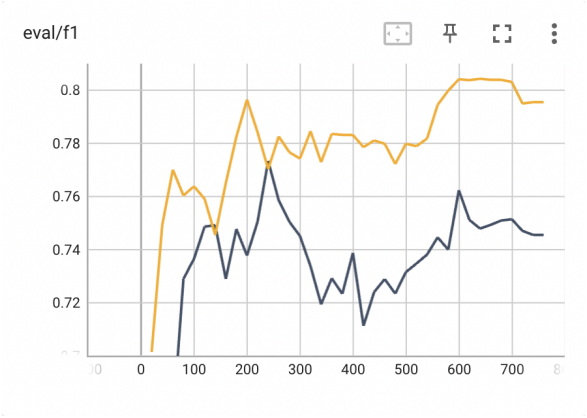

Figure 1: Evaluation F_1 score when fine-tuning SQuAD-based (yellow) and Enron-based (gray) models on Princeton email dataset.

Figure 2: Evaluation F_1 score of SQuAD-based (yellow) and Enron-based (gray) models at different stage of training.

5.2.3 Sample Generations

Table 6 shows a sample output of the SQuAD-based model before and after being fine-tuned on the Princeton email dataset. The SQuAD-based model, before the second fine-tuning, is able to understand that its task is to identify a location name inside the email, but fails to identify the correct Princeton-specific location name. Fine-tuning it on the Princeton email dataset allows it to learn the specific terminology pertinent to Princeton community.

5.3 Ablation on Model Size

We run the same set of experiments on GPT-2 Small (124M parameters) and GPT-2 Large (774M parameters) and report the F_1 scores in Table 7. We observe a similar trend as with GPT-2 Medium. Whereas the performance of GPT-2 Small underperforms the other models under every setting as

Email
Hey Princeton! The Cellists of Princeton University have been preparing all week for our Arch Play! Rain or shine, we will not be deterred from our glorious endeavor (but bring an umbrella in case we run out of room under the arch). Reminder: 5PM @ **Blair Arch**! See you there! P.S. We added Pirates of the Caribbean to our set list in case you needed another reason to come...
Gold Output
Blair Arch
Before Second Fine-tuning
Princeton University
After Second Fine-tuning
Blair Arch

Table 6: Sample generated output of the SQuAD-based model before and after being fine-tuned on the Princeton email dataset.

expected, GPT-2 Large does not always outperform GPT-2 Medium. This may be resolved by choosing a better set of hyperparameters or by choosing the model checkpoint at a later point in training.

	Small	**Medium**	**Large**
Base	0.00	0.00	0.01
S	0.40	0.49	0.45
S + P	0.77	0.80	0.79
E	0.10	0.12	0.13
E + P	0.74	0.77	0.75

Table 7: F_1 scores of models of different sizes of GPT-2, before and after each fine-tuning stage. Base refers to the pre-trained model before any fine-tuning, S, E, P refer to fine-tuning on SQuAD, Enron email corpus, and Princeton email dataset, respectively.

6 Conclusion

In this paper, we propose a double fine-tuning method for downstream tasks with limited availability for annotated data. In particular, we train a novel email content extraction model by first fine-tuning it on two different datasets: one with a similar task (extractive QA) and the other with a similar data domain (email). Both the task-adaptive and the domain-adaptive models prove to be effective at aligning the pre-trained base model for the purpose of the downstream task, but the task-adaptive approach is observed to perform slightly better.

7 Future Work

One possible future work would be to incorporate an Optical Character Recognition (OCR) process to parse the text in image attachments in the emails. During data preprocessing, we observed that a large number of the emails contained image attachments (posters, banners, etc.) that summarize the details of the event. Instead of only parsing the email body, we could consider adding an additional OCR model to parse the text from images and append to the training input.

Another direction of future research would be to modify the model to support different types of emails (e.g., asking for survey participants or receiving applications for positions). These emails are not compatible with the current setup of the model, but would be beneficial to be included in a digest. Broadening the scope of information to process, however, may make the task significantly difficult and impact the model performance.

References

Anonymous. 2022. Fine-tuning strategies for domain specific question answering under low annotation budget constraints.

Nick Craswell, Arjen de Vries, and Ian Soboroff. 2005. Overview of the trec-2005 enterprise track.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shelan Jeawak, Luis Espinosa-Anke, and Steven Schockaert. 2020. Cardiff University at SemEval-2020 task 6: Fine-tuning BERT for domain-specific definition classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 361–366, Barcelona (online). International Committee for Computational Linguistics.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, pages 217–226, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bowon Ko and Ho-Jin Choi. 2020. Twice fine-tuning deep neural networks for paraphrase identification. *Electronics Letters*, 56.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. 2015. Avocado research email collection ldc2015t03.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021. EmailSum: Abstractive email thread summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6895–6909, Online. Association for Computational Linguistics.

A Appendix

A.1 First Fine-tuning

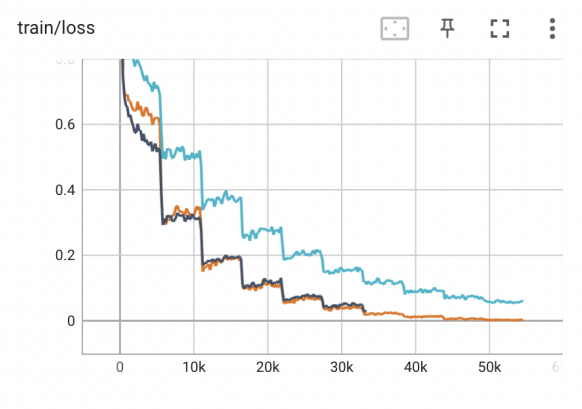

Figure 3: Training loss when fine-tuning GPT-2 Small (blue), GPT-2 Medium (gray), and GPT-2 Large (orange) on SQuAD.

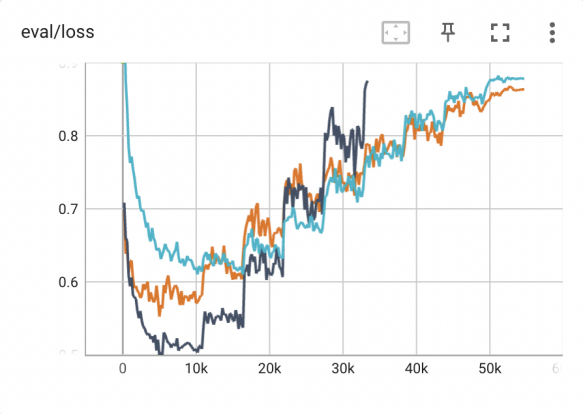

Figure 5: Evaluation loss when fine-tuning GPT-2 Small (blue), GPT-2 Medium (gray), and GPT-2 Large (orange) on SQuAD.

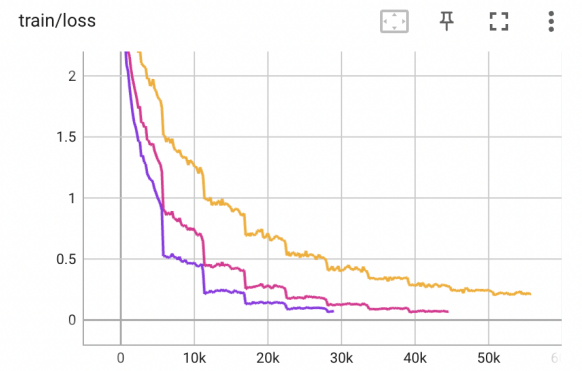

Figure 4: Training loss when fine-tuning GPT-2 Small (yellow), GPT-2 Medium (pink), and GPT-2 Large (purple) on Enron email corpus.

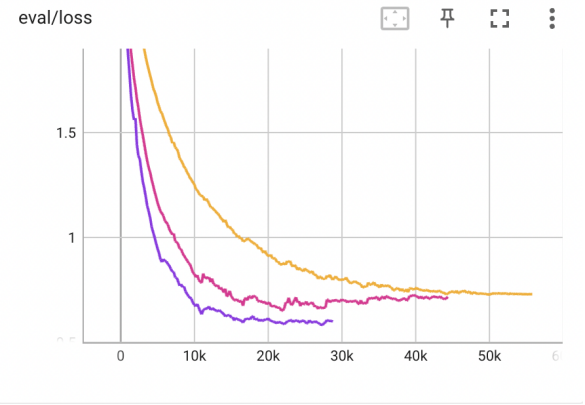

Figure 6: Evaluation loss when fine-tuning GPT-2 Small (yellow), GPT-2 Medium (pink), and GPT-2 Large (purple) on Enron email corpus.

A.2 Second Fine-tuning

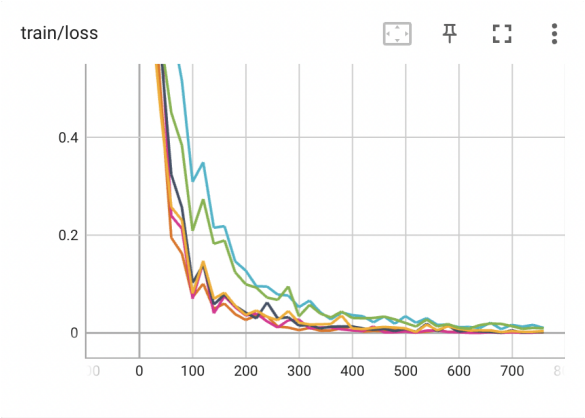

Figure 7: Training loss when fine-tuning SQuAD-based GPT-2 Small (green), GPT-2 Medium (yellow), GPT-2 Large (orange) and Enron-based GPT-2 Small (blue), GPT-2 Medium (gray), GPT-2 Large (pink) on Princeton email dataset.

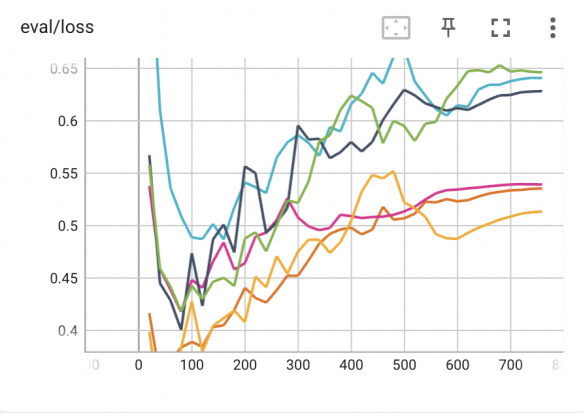

Figure 9: Evaluation loss when fine-tuning SQuAD-based GPT-2 Small (green), GPT-2 Medium (yellow), GPT-2 Large (orange) and Enron-based GPT-2 Small (blue), GPT-2 Medium (gray), GPT-2 Large (pink) on Princeton email dataset.

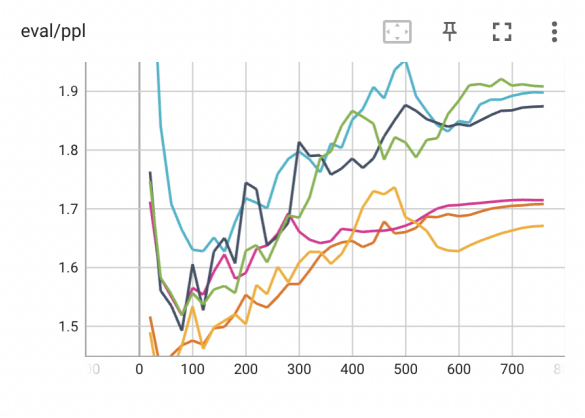

Figure 8: Perplexity (PPL) when fine-tuning SQuAD-based GPT-2 Small (green), GPT-2 Medium (yellow), GPT-2 Large (orange) and Enron-based GPT-2 Small (blue), GPT-2 Medium (gray), GPT-2 Large (pink) on Princeton email dataset.

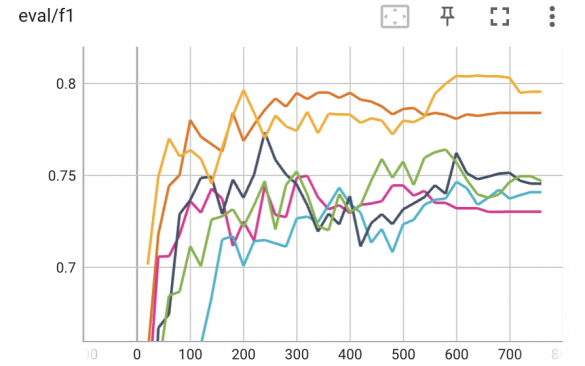

Figure 10: F_1 score when fine-tuning SQuAD-based GPT-2 Small (green), GPT-2 Medium (yellow), GPT-2 Large (orange) and Enron-based GPT-2 Small (blue), GPT-2 Medium (gray), GPT-2 Large (pink) on Princeton email dataset.

A.3 Sample Generations

Email
Do you like laughing? Then see 15 of Princeton's funniest ex-gifted children in the annual Triangle Spring Show! There will be crystal magic, foot fetishes, sports betting, How-to-tutorials, superheroes, babies, bees?? and of course sing-songy jokes that'll knock your socks off! The show is in the **Class of 1970 Theatre, Whitman College**. Tickets can be found at: tickets.princeton.edu
Gold Output
Class of 1970 Theatre, Whitman College
Before Second Fine-tuning
Triangle Spring Show
After Second Fine-tuning
Whitman College
Email
The University Press Club cordially invites you to attend the Press Club's annual Bill Rukeyser'54 lecture led by **Professor Channing Joseph** (she/he) on the theme "Queer Black Histories." The lecture will be held on April 27th, at 7:00 pm in 100 Arthur Lewis Auditorium in Robertson Hall. Professor Channing Joseph is an award-winning journalist with two decades of experience covering race, poverty, social justice and other topics in the U.S. and abroad. Thank you for your attention, and we hope to see you on April 27th.
Gold Output
Professor Channing Joseph
Before Second Fine-tuning
PRESS CLUB ANNUAL BRADSHAH
After Second Fine-tuning
Professor Channing Joseph

Table 8: Sample generated outputs before and after a SQuAD-based (above) and an Enron-based (below) GPT-2 Medium model being fine-tuned on the Princeton email dataset.