SuryaKiran at MEDIQA-Sum 2023: Leveraging LoRA for Clinical Dialogue Summarization

Kunal Suri*, Prakhar Mishra†, Saumajit Saha† and Atul Singh

Optum, India

Abstract

Finetuning Large Language Models helps improve the results for domain-specific use cases. End-to-end finetuning of large language models is time and resource intensive and has high storage requirements to store the finetuned version of the large language model. Parameter Efficient Fine Tuning (PEFT) methods address the time and resource challenges by keeping the large language model as a fixed base and add additional layers, which the PEFT methods finetune. This paper demonstrates the evaluation results for one such PEFT method Low Rank Adaptation (LoRA), for Clinical Dialogue Summarization. The evaluation results show that LoRA works at par with end-to-end finetuning for a large language model. The paper presents the evaluations done for solving both the Subtask A and B from ImageCLEFmedical

Keywords

Dialogue Summarization, Parameter Efficient Fine Tuning, Clinical Dialogue Summarization

1. Introduction

It is important to record conversations between medical personnel and patients for compliance, training, and evaluation purposes. To that end, summaries of such conversations serve as valuable tools for medical personnel and patients to refer back to and comprehend their prior interactions. Therefore, a concise summary must be produced to facilitate the next medical consultation and provide a source for future reference. Currently, such summaries are created manually; this summarization process is costly and labour-intensive. AI-based summarization techniques can help here by reducing the time and cost associated with manual summarization and facilitating the generation of more accurate representations of doctor-patient conversations by human scribes in less time.

Sequence-to-Sequence (Seq2Seq) Architectures [1] have been at the forefront of creating summaries. Transformers [2] further improved the performance of this architecture. Over time, we have seen that the performance of these models have improved significantly [1] but it comes at the cost of increased model size which made it very difficult to fit such models on consumer grade hardware such as K80 or T4. Recently a couple of techniques such as

https://www.imageclef.org/2023/medical

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

*Corresponding author.

†These authors contributed equally.

kunal_suri@optum.com (K. Suri); prakhar_mishra29@optum.com (P. Mishra); saumajit_saha@optum.com (S. Saha); atul_singh18@optum.com (A. Singh)

[1] https://paperswithcode.com/sota/text-summarization-on-pubmed-1

LoRA [3], Prefix Tuning [4], P-Tuning [5], Prompt Tuning [6] have been introduced which are collectively referred to as *Parameter Efficient Fine Tuning* (PEFT) techniques. These techniques are used for efficiently adapting pre-trained language models (PLMs) to various downstream applications without fine-tuning all the model's parameters. PEFT methods only trains a small number of (extra) model parameters, significantly decreasing computational and storage costs because fine-tuning large-scale PLMs is prohibitively costly. For this paper, we use the PEFT implementation from Huggingface [2].

This paper presents the experimental results of our explorations with LoRA on Clinical Dialogs to accomplish both Subtask A and B of [7] Shared Tasks from [8]. The solution of SubTask B presented in this paper was ranked first among all the submissions for SubTask B. The paper uses LoRA based models for both assigning conversations to a pre-defined set of clinical notes sections and summarization of conversations. Through this work, the paper also compares the performance of fine-tuned Transformer based models with LoRA based models for classification and summarization tasks. In addition to this comparison, we also evaluate impact of ensembling outputs from multiple Seq2Seq models using [9]. Our simulations show that LoRA works as well as finetuning of Transformer-based models. This is very important because it shows that we can get the equivalent performance as we get after fine tuning Transformer models while using only a fraction of parameters which means that such models could be fine tuned on consumer grade hardware such as K80 and T4.

This paper is organized as follows. Section 3 presents a brief overview of SubTask A and B - including available labeled data and evaluation metrics. Then the paper describes current state-of-the-art for dialog classification and summarization in Section 2 that this paper builds upon. This is followed by the description of the approach used to solve SubTask A in Section 4 and SubTask B in Section 5. Then the results of our solutions for both of these subtasks are presented. Finally, the paper ends with a conclusion of the work. The paper includes an appendix containing exploratory data analysis and material that will help to better understand the solution presented in the paper.

2. Related Work

Finetuning large language models enables better performance for domain-specific use cases. In-context finetuning performs well in few-shot scenarios enabling the end users to provide examples with the prompt to enable LLMs to learn for the use case at hand. This approach does not scales as it restricts sending multiple examples with the prompt. End-to-end finetuning of LLMs is resource and time intensive and has the additional drawback of storing and managing multiple copies of large-size models.

Parameter Efficient Fine Tuning (PEFT) Methods attempt to solve the problems mentioned above by finetuning a smaller number of existing or newly introduced parameters of the large language model while keeping the rest of the parameters frozen. In [10], Lilian et al. divide PEFT methods into the following four categories: additive, selective, reparameterization-based, and hybrid methods. Additive methods such as adapters [11] introduce and train only a new set of parameters or layers. Selective methods finetune only a few top layers of the network.

[2] https://huggingface.co/docs/peft/index

Reparametrization-based methods use a low-dimensional representation of the network to reduce the number of parameters to be trained during finetuning. This paper evaluates Low-Rank Adaptation (LoRA) a prominent example of this category of methods.

Parameter Efficient Fine Tuning (PEFT) methods reduce the need to host a large-sized model for each use case. They enable users to use a frozen base model with a small layer of model weights that vary with the use case. In [12], the authors compare the performance of four different PEFT techniques for scenarios where low, medium and high counts of samples are available for fine-tuning. The evaluation results show that LoRA gives near-best performance when low to medium data samples are available for summarization tasks. In another similar related study in [13], the evaluations demonstrate that the best summarization for radiology reports is achieved using a model pre-trained on the clinical text and then fine-tuned using LoRA. In this paper, the authors have used LoRA and ensembling for summarization.

3. Task Description

This Section provides a high-level overview of the MEDIQA-Sum 2023 Task (including both SubTask A and B) from ImageCLEFmed MEDIQA[8]. The Section starts with a description of different SubTask goals followed by basic counts of available labeled data. The metric used to evaluate this task is arithmetic mean of ROUGE-1 [14], Bertscore F1 [15], and BLEURT [16].

3.1. Task Definition

Given a short conversation between a Doctor and a patient or another Doctor (**Dialogue**), the goal of SubTask A is to create a system that automatically predicts the Section to which the conversation belongs to which is denoted by **Section Header**. There are twenty Sections Headers in this dataset. Some examples of Section Headers are FAM/SOCHX, GENHX, PASTMEDICALHX, CC. All of these Section Headers and their descriptions (**Section Description**) can be found in Table A2. The goal of SubTask B is to create a system that generates a summary which matches the human generated summary (**Section Text**) as closely as possible while optimizing the metric for evaluation.

3.2. Labeled Data

In this paper we have used the labeled data provided by MEDIQA-Sum 2023 organizers for training the models. A sample data point from the labeled data set for SubTask A and B can be found in Table A1. The official data consists of a training and validation split. For SubTask A and B, training data contains 1201 and validation data contains 180 <dialogue, section-text, section-header> triplets.

4. SubTask A Methodology

Given a short conversation between a doctor and a patient, the goal of SubTask A is to predict its Section Header. This Section starts with a description of the approach used to predict the Section Header.

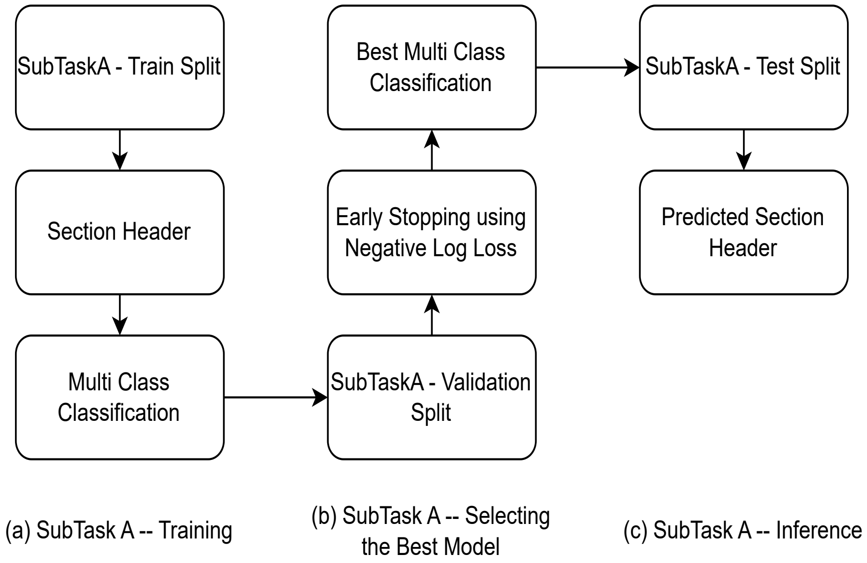
SubTaskA - Train Split
Section Header
Multi Class Classification
SubTaskA - Validation Split
Early Stopping using Negative Log Loss
Best Multi Class Classification
SubTaskA - Test Split
Predicted Section Header
(a) SubTask A -- Training
(b) SubTask A -- Selecting the Best Model
(c) SubTask A -- Inference

Figure 1: SubTask A - Overall Architecture

We have achieved success using Bio-ClinicalBERT [17] for classification in the healthcare domain. Hence we choose it as the backbone and initialize LoRA layer on top of it. We use this architecture for classification of Dialogue to a Section Header in SubTask A. We limit the number of input tokens to 300 tokens because that is the length of majority of dialogues, as shown in Figure A1. We use a 3 Fold Cross Validation approach for modeling purposes. This is to ensure that we capture all information in the data. For every fold, we split its test part into validation and test. We do this so that we can use validation split to select best model using Early Stopping and test split to calculate its performance. The hyper-parameters used for training and performance for all folds can be found in Table A3. During inference, we pass a given Dialogue through all three models, take an average of the logits for all the classes and output the class with the highest logit score.

5. SubTask B Methodology

Given a short conversation between a doctor and a patient, the goal of SubTask B is to summarize it while ensuring that the generated summary is as fluent and as close to Section Text as possible. This Section starts with a description of the methodology used to summarize the conversation. For Dialogue Summarization, we have trained a LoRA layer on top of Seq2Seq models. This Section also describes the processed labeled data used for training these models, followed by the actual training steps. Then this Section looks at the steps used to generate the summary from the decoder. Finally, we discuss the approach used for ensembling the outputs of these models.

We train LoRA based Seq2Seq models using labeled data (Dialogue + Section Header, Section Text) as (Input, Output) pair. Section Text is a part of the labeled data and is a human subject matter expert-created summary of Dialogue. As a preprocessing step, we replace all new line

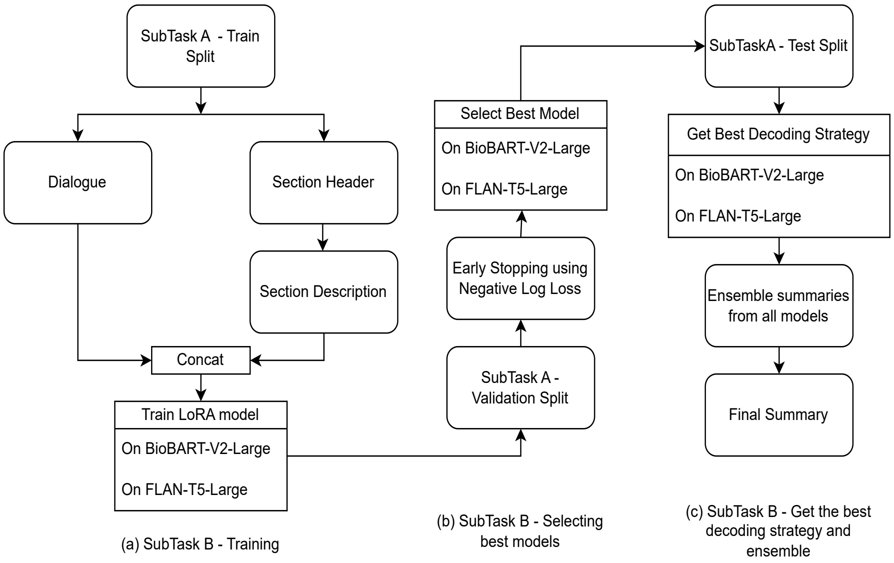

Figure 2: SubTask B - Overall Architecture

characters with whitespaces. The Dialogue is concatenated with the section description of its Section Header by the SEP token of the Seq2Seq architecture. During training and inference, we use the actual section description for the actual Section Header. No changes are made to Section Text.

We use a 3-fold cross validation scheme as described in 4 and train LoRA on two Seq2Seq architectures - BioBart-V2-Large [18] and Flan-T5-Large [19]. Here we need to select the number of input tokens for encoder and decoder. For encoder, we have selected token length of 512 tokens and for decoder, we have selected token length of 400 tokens. All the hyper-parameters used to train each of the above architecture can be found in Table A4. To select the best model, we use early-stopping [20] based on Validation Negative Log Loss. Results on the test part of each of these models can be found in Table A5. The distribution of tokens for Dialogue and Section Text can be found in Figure A2 and Figure A3 respectively.

To generate summaries that match the human generated summaries, we need a way to control the text generated by the decoder component of a Seq2Seq model. This can be done by using decoding strategies such as Beam Search [21], Top-k Sampling [22], Top-p Sampling [23], Contrastive Search [24] etc. In this module, we use Beam Search with TPESampler Algorithm from Optuna[3] to search for the optimal decoding strategy trying to maximize ROUGE-1, ROUGE-2, and BertScore rather than relying on manual tweaking of these metrics. We use TPESampler here because it supports multivariate optimization and also it handles Float, Integer, and Categorical values better than other algorithms present in Optuna[4]. We use Optuna here due to ease of implementing Hyper-parameter optimization algorithms. We did not use BLEURT during search because it is extremely time consuming. For this module, we use four hyper-parameters

[3] https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html
[4] https://optuna.readthedocs.io/en/stable/reference/samplers/index.html

for Beam Search - Early Stopping, Number of Beams, No Repeat N-gram Size, Length Penalty. The search space of each of these variables can be found in the Table 1.

The results from the different models are ensembled using **Generating Best Summary by semantic similarity** - a post-ensemble method [9] to identify the summary which is closest to all the generated summaries. The paper uses this output summary as the final summary for the given Dialogue.

Table 1
Search Space for Beam Search Decoding

Variable	Data Type	Range
Early Stopping	Categorical	[True,False]
Number of Beams	Integer	5-15
No Repeat Ngram Size	Integer	5-15
Length Penalty	Float	[-2,2]

6. SubTask A Results and Analysis

This Section presents the results for SubTask A using the approach described in Section 4. We have made only one submission for predicting Section Header whose Multi Class Accuracy was 73.5% on the test set given by the organizers, obtaining a rank of 8 among 23 submissions. In this submission, we pass Dialogues through all three LoRA based Bio-ClinicalBERT models, take an average of the logits for all the classes and output the class with the highest logit score. The table containing our team's standing can be found in the Tables A6. Standings of all the teams have been calculated using multi class accuracy. We compared performance of Bio-ClinicalBERT when it is fine-tuned end-to-end and when it is used as a backbone for LoRA. We observe that Bio-ClinicalBERT with LoRA score 73.3% on validation data whereas end-to-end fine-tuned Bio-ClinicalBERT score 72% on the same validation data.

7. SubTask B Results

This Section presents the results for SubTask B using the approach described in Section 5. We have made three submissions (mentioned as *runs* in the result tables) for generating summaries from Dialogues. For the summarization task, we have submitted results from three *runs*. In *run 1* and *run 2*, we train LoRA on BioBart-V2-Large and Flan-T5-Large respectively while *run 3* presents the results of ensembling summaries from both of these models. The details for each run are as follows:

1. Run 1 - We generate summary from BioBart-V2-Large model trained on each fold and ensemble output of all the models using 5
2. Run 2 - We generate summary from Flan-T5-Large model trained on each fold and ensemble output of all the models using *Generating Best Summary by semantic similarity*.

3. Run 3 - We generate summary from BioBart-V2-Large and Flan-T5-Large model trained on each fold and ensemble output of all the models using *Generating Best Summary by semantic similarity.*

The table containing our team's standing can be found in Table A7. Standings of all the teams have been calculated by calculating arithmetic mean of Rouge-1, Bertscore, BLEURT for the Dialogue summary.

The experiments show that Run3 performs the best scoring rank 1 out of 13 submissions. This is also intuitive since it contains summaries from 3 models of BioBART-V2-Large and 3 models of Flan-T5-Large. Run2 scored 5th rank and Run1 scored 6th rank. This is an interesting observation since Flan-T5-Large is an enhanced version of T5 that has been finetuned in a mixture of tasks whereas BioBart-V2-Large has been trained solely on medical corpus so ideally Run1 should have scored better than Run2 but it seems that bigger models work better than domain specific models although this hypothesis needs to be validated.

Table 2
Results of runs on Test Data

Run	ROUGE-1	Bertscore-F1	BLEURT	Mean Score
Run 3	0.4398	0.7231	0.5567	0.5732
Run 2	0.4209	0.7137	0.5423	0.5590
Run 1	0.4056	0.7109	0.5324	0.5496

7.1. Analysis of different Transformer Architectures on SubTask B

We compare performance of BioBart-V2-Large and Flan-T5-Large when they are fine-tuned end-to-end and they are treated as backbone for LoRA. We observe that the models trained with LoRA perform better than the models which were fine-tuned end-to-end. The performance was evaluated by calculating arithmetic mean of ROUGE-1, ROUGE-2, and BertScore-F1. We do not use BLEURT here as it is extremely time consuming and based on our observations, ROUGE-2 and BLEURT have a very strong correlation. The average score across all folds for each architecture can be found in the Table A5.

8. Conclusion

The paper presents the solution and the results for SubTask A and B of ImageCLEFmed MEDIQA-Sum task. The solution uses LoRA to finetune Transformer based models to classify and summarise Clinical Dialogues, and our simulation results show that the performance of Transformer based models finetuned using LoRA is equivalent to the performance of Transformer based models finetuned using resource and time-intensive end-to-end finetuning. The success of Transformer based model finetunes using LoRA implies organizations can easily finetune and deploy domain-based models.

The authors observe that metrics such as ROUGE are ineffective for evaluating the performance of models like OpenAI GPT3 as they focus on syntactic similarity. Metrics such as Bertscore and BLEURT seem more suitable for such models since they focus on semantic similarity. Finally, the paper also evaluates two different ensemble techniques, and the results demonstrate that the Post Ensemble technique performs the best while giving minimum hallucinations.

A. Appendix

A.1. Data Exploration and Explanation

This section discusses data exploration and explanation so that audience can understand why we made the decisions that we made. A sample data point from dataset for SubTask A and B can be seen in Table A1.

Table A1
Sample data point for SubTask A and B

Variable	Sample Value
Section Header	FAM/SOCHX
Section Text	The patient has been a smoker since the age of 10. So, he was smoking 2-3 packs per day. Since being started on Chantix, he says he has cut it down to half a pack per day. He does not abuse alcohol
Dialogue	Doctor: Are you a smoker? Patient: Yes. I do not drink if that is any constellation. Doctor: How much do you smoke per day? Patient: I just started taking Chantix and now I am down to a half a pack a day. Doctor: How much did you smoke per day prior to starting Chantix? Patient: I was smoking about two to three packs a day. I have been smoker since I was ten years old.

The description of each of the Section Headers present in the data can be found in Table A2

The Class distribution of Section Headers for SubTask A is give by Figure A1

The Dialogue Token Distribution for SubTask A and B is give by Figure A2

The Clinical Note Token Distribution for SubTask B is give by Figure A3

The hyper-parameters and performance metrics for Predicting Section Header i.e SubTask A can be found in the Table A3.

The hyperparameters used to fine tune Seq2Seq Models and LoRA i.e. SubTask B can be found in Table A4. Each of these models were trained on 150 epochs, Gradient Accumulation of 16, Learning rate of 1e-3, AdamW optimizer, and Linear Learning Scheduler.

The performance of different Seq2Seq Models using LoRA and Fine-tuning can be found in Table A5

Table A2
Section Headers and their descriptions.

Section Header	Section Header Description
FAM/SOCHX	FAMILY HISTORY/SOCIAL HISTORY
GENHX	HISTORY OF PRESENT ILLNESS
PASTMEDICALHX	PAST MEDICAL HISTORY
CC	CHIEF COMPLAINT
PASTSURGICAL	PAST SURGICAL HISTORY
ALLERGY	ALLERGY
ROS	REVIEW OF SYSTEMS
MEDICATIONS	MEDICATIONS
ASSESSMENT	ASSESSMENT
EXAM	EXAM
DIAGNOSIS	DIAGNOSIS
DISPOSITION	DISPOSITION
PLAN	PLAN
EDCOURSE	EMERGENCY DEPARTMENT COURSE
IMMUNIZATIONS	IMMUNIZATIONS
IMAGING	IMAGING
GYNHX	GYNECOLOGIC HISTORY
PROCEDURES	PROCEDURES
OTHER_HISTORY	OTHER_HISTORY
LABS	LABS

Table A3
SubTask A - Predicting Section Header. Base Arch: Base Architecture, BS: Batch Size, LR: Learning Rate, LoRA-A: LoRA-Alpha, LoRA-D: LoRA-Dropout BVL : Best Validation Loss.

Base Arch	Fold	Epochs	BS	LR	LoRA-R	LoRA-A	LoRA-D	BVL
Bio-ClinicalBERT	0	150	16	1e-3	8	32	0.01	1.193
Bio-ClinicalBERT	1	150	16	1e-3	8	32	0.01	1.429
Bio-ClinicalBERT	2	150	16	1e-3	8	32	0.01	0.4961

Table A4
SubTask B - Hyperparameter Tuning for Different Architectures. Base Arch: Base Architecture, BS: Batch Size, LR : Learning Rate, LoRA-A: LoRA-Alpha, LoRA-D: LoRA-Dropout, MaxSL : Maximum Source Length, MaxTL : Maximum Target Length, MinTL : Minimum Target Length

Base Arch	BS	LR	LoRA-R	LoRA-A	LoRA-D	MaxSL	MaxTL	MinTL
Flan-T5-Large	1	1e-3	8	32	1e-3	512	400	8
Biobart-V2-Large	1	1e-3	8	32	1e-3	512	400	8

Table A5
SubTask B - Section Text Summarization Comparison

Base Architecture	LoRA-Score	Fine Tuning-Score
BioBart-V2-Large	0.4310	0.2877
FLAN-T5-Large	0.4276	0.1083

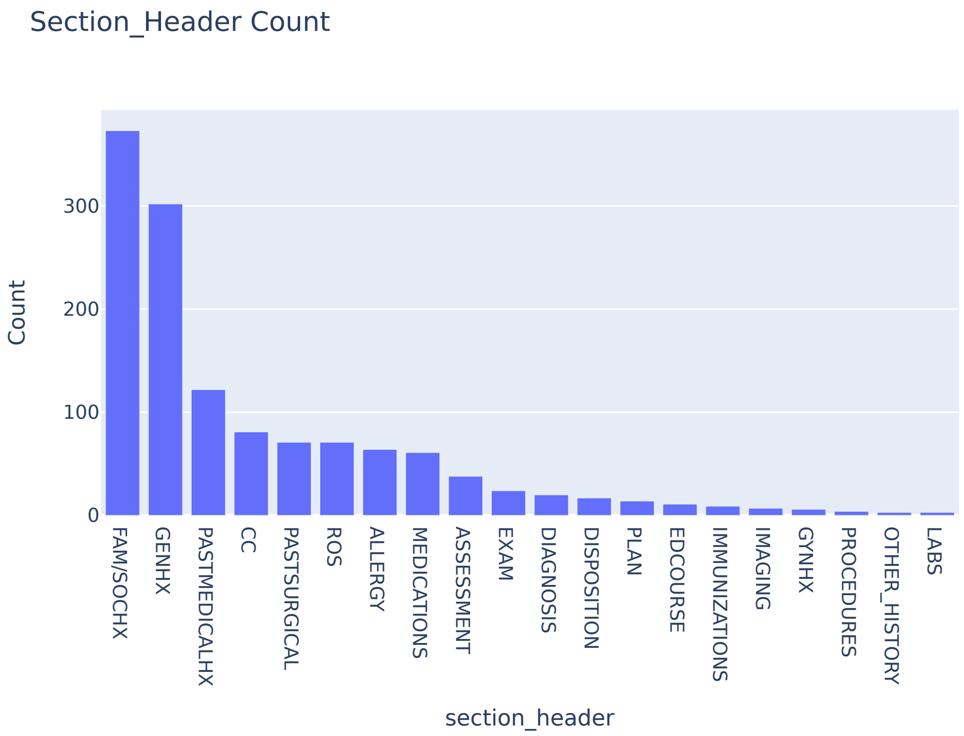

Figure A1: Class distribution of Section Headers

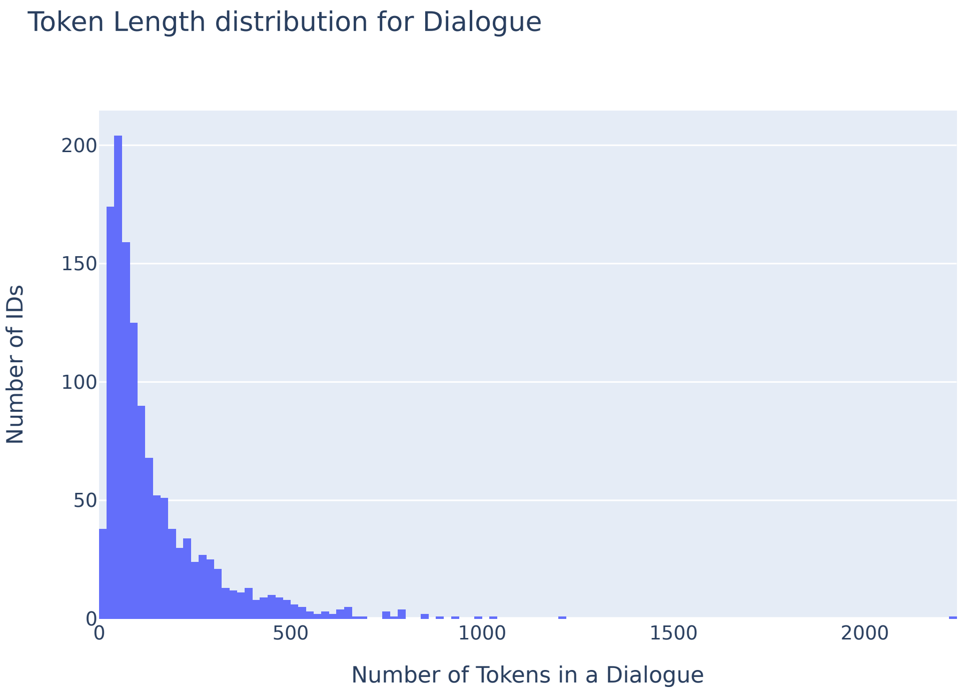

Figure A2: Dialogue Token Distribution

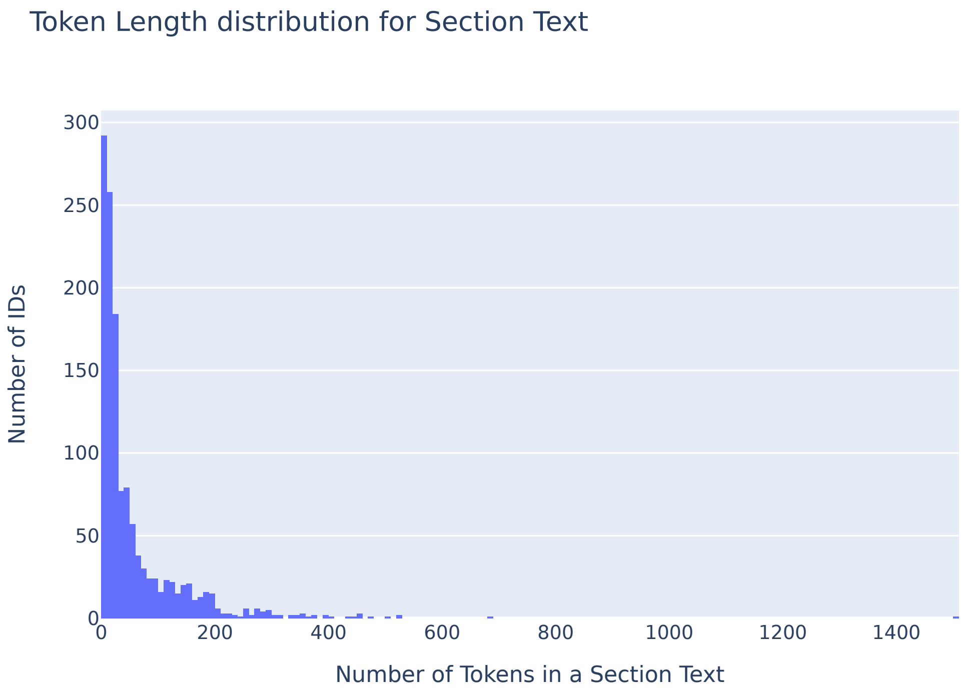

Figure A3: Clinical Note Token Distribution

A.2. Standing of our team

Our standings (in bold) for SubTask A - Section Header Classification is in Table A6. We omitted several teams from these standings and represent them by Ellipsis (**...**). This is done only to conserve space.

Table A6
SubTask A - Section Header Classification Standings

Team	Run	Accuracy	Rank
Cadence	run1	0.82	1
...			
SuryaKiran	**run1**	**0.735**	**8**
...			
SSNSheerinKavitha	run1	0.14	23

Our standings (in bold) for SubTask B - Summarization is in Table A7

Table A7
SubTask B - Section Text Summarization Standings

Team	Run	Rouge1	Bertscore_F1	Bleurt	Aggregate_score	Rank
SuryaKiran	**run3**	**0.4398**	**0.7231**	**0.5567**	**0.5732**	**1**
...						
SuryaKiran	**run2**	**0.4209**	**0.7137**	**0.5423**	**0.5590**	**5**
SuryaKiran	**run1**	**0.4056**	**0.7109**	**0.5324**	**0.5496**	**6**
...						
SKKU-DSAIL	run1	0.2603	0.5929	0.5305	0.4612	13

References

[1] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 27, Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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

[4] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, 2021. arXiv:2101.00190.

[5] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, Gpt understands, too, 2021. arXiv:2103.10385.

[6] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059. URL: https://aclanthology.org/2021.emnlp-main.243. doi:10.18653/v1/2021.emnlp-main.243.

[7] W. Yim, A. Ben Abacha, N. Snider, G. Adams, M. Yetisgen, Overview of the mediqa-sum task at imageclef 2023: Summarization and classification of doctor-patient conversations, in: CLEF 2023 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Thessaloniki, Greece, 2023.

[8] B. Ionescu, H. Müller, A.-M. Drăgulinescu, W. Yim, A. Ben Abacha, N. Snider, G. Adams, M. Yetisgen, J. Rückert, A. G. S. de Herrera, C. M. Friedrich, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, S. A. Hicks, M. A. Riegler, V. Thambawita, A. Storås, P. Halvorsen, N. Papachrysos, J. Schöler, D. Jha, A.-G. Andrei, I. Filipovich, I. Coman, V. Kovalev, A. Stan, G. Ioannidis, H. Manguinhas, L.-D. Ştefan, M. G. Constantin, M. Dogariu, J. Deshayes, A. Popescu, Overview of ImageCLEF 2023: Multimedia retrieval in medical, social media and recommender systems applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 14th International Conference of the CLEF Association (CLEF 2023), Springer Lecture Notes in Computer Science LNCS, Thessaloniki, Greece, 2023.

[9] H. Kobayashi, Frustratingly easy model ensemble for abstractive summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4165–4176. URL: https://aclanthology.org/D18-1449. doi:10.18653/v1/D18-1449.

[10] V. Lialin, V. Deshpande, A. Rumshisky, Scaling down to scale up: A guide to parameter-efficient fine-tuning, 2023. arXiv:2303.15647.

[11] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo,

M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, CoRR abs/1902.00751 (2019). URL: http://arxiv.org/abs/1902.00751. arXiv:1902.00751.

[12] G. Pu, A. Jain, J. Yin, R. Kaplan, Empirical analysis of the strengths and weaknesses of peft techniques for llms, 2023. arXiv:2304.14999.

[13] D. V. Veen, C. V. Uden, M. Attias, A. Pareek, C. Bluethgen, M. Polacin, W. Chiu, J.-B. Delbrouck, J. M. Z. Chaves, C. P. Langlotz, A. S. Chaudhari, J. Pauly, Radadapt: Radiology report summarization via lightweight domain adaptation of large language models, 2023. arXiv:2305.01146.

[14] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with BERT, CoRR abs/1904.09675 (2019). URL: http://arxiv.org/abs/1904.09675. arXiv:1904.09675.

[16] T. Sellam, D. Das, A. P. Parikh, Bleurt: Learning robust metrics for text generation, 2020. arXiv:2004.04696.

[17] E. Alsentzer, J. R. Murphy, W. Boag, W. Weng, D. Jin, T. Naumann, M. B. A. McDermott, Publicly available clinical BERT embeddings, CoRR abs/1904.03323 (2019). URL: http://arxiv.org/abs/1904.03323. arXiv:1904.03323.

[18] H. Yuan, Z. Yuan, R. Gan, J. Zhang, Y. Xie, S. Yu, Biobart: Pretraining and evaluation of a biomedical generative language model, 2022. arXiv:2204.03905.

[19] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling instruction-finetuned language models, 2022. arXiv:2210.11416.

[20] Y. Yao, L. Rosasco, A. Caponnetto, On early stopping in gradient descent learning, Constructive Approximation 26 (2007) 289–315.

[21] A. Graves, Sequence transduction with recurrent neural networks, CoRR abs/1211.3711 (2012). URL: http://arxiv.org/abs/1211.3711. arXiv:1211.3711.

[22] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 889–898. URL: https://aclanthology.org/P18-1082. doi:10.18653/v1/P18-1082.

[23] A. Holtzman, J. Buys, M. Forbes, Y. Choi, The curious case of neural text degeneration, CoRR abs/1904.09751 (2019). URL: http://arxiv.org/abs/1904.09751. arXiv:1904.09751.

[24] Y. Su, N. Collier, Contrastive search is what you need for neural text generation, 2023. arXiv:2210.14140.