Saama AI Research at SemEval-2023 Task 7: Exploring the Capabilities of Flan-T5 for Multi-evidence Natural Language Inference in Clinical Trial Data

Kamal Raj Kanakarajan and **Malaikannan Sankarasubbu**
SAAMA AI Research Lab, Chennai, India
{kamal.raj, malaikannan.sankarasubbu}@saama.com

Abstract

The goal of the NLI4CT task is to build a Natural Language Inference system for Clinical Trial Reports that will be used for evidence interpretation and retrieval. Large Language models have demonstrated state-of-the-art performance in various natural language processing tasks across multiple domains. We suggest using an instruction-finetuned Large Language Models (LLMs) to take on this particular task in light of these developments. We have evaluated the publicly available LLMs under zeroshot setting, and finetuned the best performing Flan-T5 model for this task. On the leaderboard, our system ranked second, with an F1 Score of 0.834 on the official test set.

1 Introduction

Building a Natural Language Inference (NLI) system for Clinical Trial Reports (CTRs) is the objective of the Multi-evidence Natural Language Inference for Clinical Trial Data (NLI4CT) task Jullien et al. (2023), which focuses on the interpretation and retrieval of medical evidence. Clinical practitioners struggle to keep up with the large number of CTRs published yearly to provide individualised, evidence-based care DeYoung et al. (2020). The main task is separated into two subtasks. Task 1 is to determine the relationship of the claim to the information in a single CTR or to compare two CTRs to see if there is entailment or contradiction. The supporting factor required to back up the predicted label in Task 1 must be extracted from the CTRs in Task 2. English is the only language used in the task.

BERT Devlin et al. (2019), ULMFiT Howard and Ruder (2018), and GPT Radford et al. (2018) models revolutionised the field of Natural Language Processing by introducing a new technique of pretraining on Language modelling and finetuning on task-specific data using supervised data, resulting in state-of-the-art results on a variety of datasets and benchmarks Wang et al. (2018), Wang et al. (2019). When applying the same model architecture to domains such as biomedical, they also achieved state-of-the-art results on domain specific tasks; domain specific trained models like BioBERT Lee et al. (2020), PubMedBERT Gu et al. (2021), BioELECTRA raj Kanakarajan et al. (2021) further improved performance. Furthermore, the introduction of Large Language Models (LLMs) GPT-3 Brown et al. (2020), PaLM Chowdhery et al. (2022), OPT Zhang et al. (2022), GLM Zeng et al. (2022), BLOOM Scao et al. (2022) ushered in a new strategy known as Prompt-based learning Liu et al. (2023), which allows these models to be more effectively adapted to tasks. Prompt-based learning is a technique for using LLM models to predict what will happen next in a sentence. The models fill in the blanks to form the final sentence, which can then be used to determine the correct answer. With the implementation of Prompt-based learning, the model can now carry out few-shot or even zeroshot learning, enabling it to adjust to new circumstances with either a limited amount of labelled data or none at all. LLMs' potential for zeroshot success on unseen tasks was improved by supervised finetuning their instruction-following skills across multiple tasks. Instruction tuning applies to both decoder only models like GPT-3, OPT and encoder-decoder models like T5 Raffel et al. (2020).

In this paper, we investigate the effectiveness of Flan-T5 Longpre et al. (2023) models for the NLI4CT task. Flan-T5 is an instruction tuned T5 model. In the zeroshot setting, we assessed Flan-T5 models with various instructions for this task. On the official test set, our finetuned Flan-T5 model on the NLI4CT task received an F1 Score of 0.834, placing it second on the leader-board. The code to reproduce the experiments mentioned in this paper is publicly available.[1]

[1] https://github.com/kamalkraj/NLI4CT

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 995–1003

Data	No. of Samples	Type	Section					Label	
			Count	Intervention	Eligibility	Adverse Events	Results	Contradiction	Entailment
Train	1700	Single	1035	155	317	309	254	502	533
		Comparison	665	241	169	187	68	348	317
Dev	200	Single	140	26	44	32	38	70	70
		Comparison	60	10	12	20	18	30	30
Test*	500	Single	229	74	44	48	63	-	-
		Comparison	271	68	88	72	43	-	-

Table 1: The Single type is based on a single CTR, while the Comparison type is based on two CTRs. Section illustrates the count of four distinct sections from which the statements are annotated. * Test set labels are not public. More details in section 2.1.

2 Background

2.1 Task and Dataset Description

This task is based on a collection of Clinical Trial Records (CTRs) extracted from clinicaltrials.gov [2] and statements annotated by domain experts. The task uses four sections from the CTRs: Eligibility criteria, Intervention, Results, and Adverse events. The annotated statements are sentences that make some claim about the information contained in one of the sections in the CTR (premise). Task 1 is to predict the *Entailment* vs *Contradiction* between CTR-statement pairs. The statements may make claims about a single CTR or compare two CTRs. For the "Single" type, all evidence will be contained in the primary CTR, while for the "Comparison" type, evidence will have to be retrieved from both CTRs same section. Task 2 is to identify the supporting factor extracted from the CTRs (premise) to justify the prediction from task 1, given CTR, annotated statement, and prediction.

There are 999 breast cancer CTRs in the dataset. The datasets, which are divided into train, development, and test sets, contain a total of 2400 annotated statements. The distribution of labels between the train and development sets is even. Eligibility sections are used most often in Single annotated statements, while Intervention is used in Comparison statements. Detailed statistics are in table 1.

2.2 Related Work

Large Language models have shown promising results when doing tasks in the biomedical and clinical domains without domain-specific training. In Agrawal et al. (2022), evaluate InstructGPT Ouyang et al. (2022) for clinical information extractor under zeroshot and few-shot settings. In addition, they present a new dataset for benchmarking few-shot clinical information extraction. PaLM Chowdhery et al. (2022) and its instruction-tuned variant, FlanPaLM Chung et al. (2022), are evaluated on MultiMedQA Singhal et al. (2022), a collection of seven Question Answering datasets in the biomedical/clinical domain. Li'evin et al. (2022) assess InstructGPT and Codex's Chen et al. (2021) ability to answer and reason in USMLE Jin et al. (2021), MedMCQA Pal et al. (2022), and PubMedQA Jin et al. (2019) datasets. Recent additions to the LLMs family include BioGPT 1.5B Luo et al. (2022) and BioMedLM 2.7B[3]. On the MedNLI Romanov and Shivade (2018) dataset Clinical-T5-Large Lehman et al. (2023) model achieves state-of-the-art results and outperforms models double its parameters. Unlike the others mentioned above, these models are pre-trained only on domain-specific data (PubMed) using a domain-specific vocabulary.

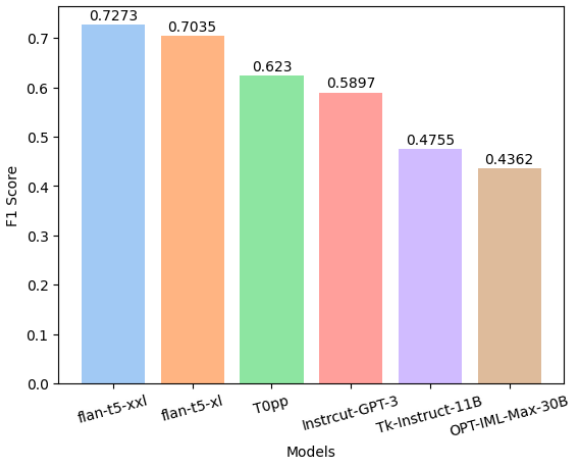

Figure 1: Zeroshot performance of various instruction-tuned models on the test set.

[2] https://clinicaltrials.gov/ct2/home

[3] https://github.com/stanford-crfm/BioMedLM

3 System overview

It has been demonstrated that finetuning the LLMs to follow natural language instructions improved the models' ability to perform better on previously unseen tasks in zeroshot and few-shot settings. For the CTRs NLI task, we will use Flan-T5, an instruction tuned T5 model. The Flan-T5 model was chosen based on its zeroshot performance in this particular task. T0pp Sanh et al. (2021), Tk-Instruct-11B Wang et al. (2022), OPT-IML-Max-30B[4] Iyer et al. (2022), Instruct-GPT-175B were the other models considered for this experiment. Figure 1 depicts the zeroshot performance of these models. For all the models, instructions are adapted to their instruction tuning style. There is no single instruction that works across all models efficiently. The expected model labels are also passed to the model via the same instruction to get the proper output from the model. Under the zeroshot settings, the Flan-T5 xxl(11B) and xl(3B) have the best F1 scores.

The T5 is an encoder-decoder Sutskever et al. (2014) model transforms all tasks into a text-to-text format. The model has parameters ranging from 60 million to 11 billion. A denoising objective is used to pre-train the T5 model. The Flan-T5 model is trained from Lester et al. (2021) Language Modeling objective adapted T5 model. The original T5 model is trained with a maximum sequence length of 512, whereas the Flan-T5 model is trained with a maximum encoder length of 1024 and a decoder length of 256. The Flan-T5 model has been finetuned on 1836 tasks with 15M examples.

For zeroshot evaluation and finetuning experiments, we construct the input to the model using an instruction template, CTRs data, and the statement as shown in figure 2. Given the following instruction to the model, the model generates the entailment or contradiction label.

4 Experimental setup

4.1 Dataset Preprocess

The NLI task has single and comparison types, as mentioned in the section 2.1. The evidence from the primary and secondary CTRs is combined to form the premise. In the domain expert's annotated statement, they specifically use the keyword primary or secondary trial while making a claim. Even if it is a single type, they use the primary trial keyword, for Example: *"Patients eligible for the primary trial must live in the USA."*. Following that, the evidence is compiled into a single text for a single type: *"Primary trial evidence are {primary_evidence}."* and for comparison type *"Primary trial evidence are {primary_evidence} and Secondary trial evidence are {secondary_evidence}."*. The annotated statement has not been pre-processed in any way.

[4]The OPT-IML-175 is skipped as unable to get access.

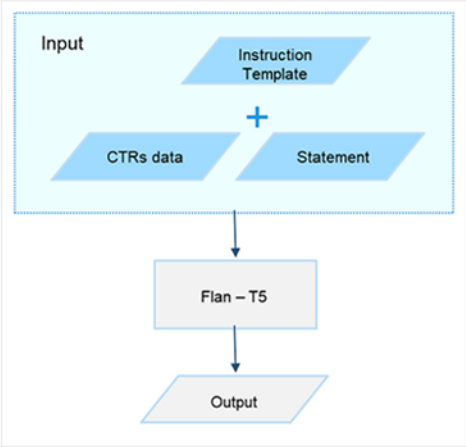

Figure 2: Flan-T5 model input and output flow.

"{premise}

Question: Does this imply that {hypothesis}?

{options}"

Figure 3: Final instruction template

4.1.1 Instruction templates

We have collected various instruction templates suitable for this task from the FlanT5 templates collection. We have used the same sentence joining method mentioned in preprocess 4.1 for all the different instruction templates. Figure 3 shows the final instruction template used for the model. The *premise* is replaced by the evidence mentioned above and *hypothesis* is replaced by the statement annotated by the domain expert. The *options* are replaced by *OPTIONS:\nEntailment\nContradiction*. Instruction templates used for zero-shot evaluation of T0pp, Instruct-GPT-3, Tk-instruct-11B, and OPT-IML-Max-30B models are available in the code open-sourced. Refer to Appendix B for various instruction templates and their corresponding F1 score on the test set.

4.2 Zeroshot

Model	Data Split	F1 Score
xxl	Train	0.701
	Dev	0.734
	Test	**0.727**
xl	Train	0.675
	Dev	0.690
	Test	0.703

Table 2: Results on Train, Dev and Test data using zeroshot Flan-T5 xl and xxl models.

Using different instructions, we evaluated the Flan-T5 model from small (60 million) to xxl (11 Billion) parameters. The initial choice of instructions comes from the open-sourced FLAN repository[5]. This experiment uses unique instruction templates from NLI task collection in Flan-T5, which includes datasets ANLI Nie et al. (2019), RTE Bentivogli et al. (2009), CB De Marneffe et al. (2019), SNLI bowman2015large, WNLI Levesque et al. (2012), QNLI Rajpurkar et al. (2018), and MNLI Williams et al. (2017). Results with best performing instruction are shown in table 2. We have also tried various NLI dataset instructions from T0pp, Tk-Instruct-11B and OPT-IML-Max-30B projects. For all the instruction templates used, the output options are mentioned using the standard FLAN format in section 4.1.1 to get the proper output from the model. Few-shot experiments are skipped as the task (NLI) is already familiar to FLAN-T5 models.

4.3 Finetuning

Following the original research, the Flan-T5 models are finetuned to task-specific data to better adapt to the domain and task. We finetuned all five Flan-T5 models during our experiments with this task. The model was trained using both single and multiple instruction templates. For finetuning with a single instruction template, we chose the one instruction template with the highest F1 score in the zeroshot setting. We used a total of ten instruction templates across multiple instruction settings. The Flan-T5 model is based on a T5 model tailored to the language modelling task Lester et al. (2021), which is also better at following prompts. We also finetune that model directly to task data to get the baseline performance of the T5 model.

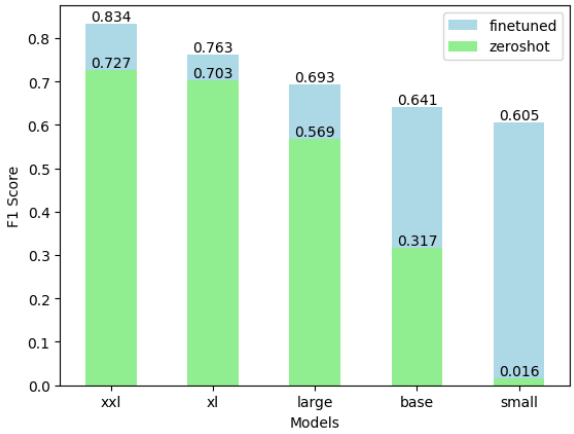

Figure 4: Zeroshot and finetuned performance of Flan-T5 model from small to xxl on the test set.

The single task finetuning in the Flan-T5 research is done with an adafactor Shazeer and Stern (2018) optimizer and a learning rate of 1e-3 with a batch size of 128 and a maximum sequence length of 512. In our experiments, we used a much lower learning rate of 7e-6 with a batch size of 8 and the Adam Kingma and Ba (2014) optimizer for two epochs. As in the original research, we use a constant learning rate. Refer to Appendix A for the full set of hyper-parameters. The Flan-T5 model is trained on samples with a maximum length of 1024 tokens. As the T5 model uses relative positional encoding, it is possible to train the model with much more tokens; however, in our experiments, training with samples with a length greater than 1024 tokens hurts model performance. As a result, any samples with more than 1024 tokens have been dropped in finetuning. There are less than 1024 tokens in 1576 of 1700 data points from the training set. We also train using development set samples for the final submission to the leaderboard. Combining the development set, we have used 1770 samples for training with less than 1024 tokens.

4.4 Software and Hardware

The entire experiment is carried out with the T5 implementation of huggingface transformers Wolf et al. (2020). We have modified the example summarization script[6] from huggingface repository for

[5] https://github.com/google-research/FLAN/tree/main/flan/v2

[6] https://github.com/huggingface/transformers/blob/v4.26.1/examples/pytorch/summarization/run_summarization.py

this task. Models with parameters up to 3B (xl) are finetuned using a 4xQuadro RTX(48GB VRAM) card, while models with parameters up to 11B (xxl) are finetuned in 8xA100(40GB VRAM) instances. DeepSpeed Rasley et al. (2020) ZeRO Rajbhandari et al. (2020) optimization is used to fit the model into the multi-gpu for training efficiently. For memory fitting of large models, we use ZeRO-3 Ren et al. (2021) specifically, which partitions and offloads parameters, optimizer states, and gradients to CPU memory. The model is trained on the A100 with BFLOAT16 (BF16) precision and on the Quadro RTX with Automatic Mixed Precision (AMP) because the original model was pretrained with BF16 precision, and model training with FLOAT16 (FP16) precision is unstable[7].

5 Results

Training Data	Test Data	Epoch 1	Epoch 2	Epoch 3
Train	Dev	0.810	0.845	0.878
	Test	0.796	0.829	0.826
Train + Dev	Dev	0.892	0.916	0.931
	Test	0.804	**0.834**	0.829

Table 3: Results on Dev, Test data using finetuned Flan-T5-xxl model trained on Train and Train + Dev data.

The finetuned Flan-T5-xxl model finished second in the leaderboard with an F1 score of 0.834. The xxl model's zeroshot evaluation on the train, dev, and test splits achieves a remarkable average score of 0.71 across the splits. Figure 4 shows that as the number of parameters in the model increases, so does the zeroshot performance. The difference in zeroshot results between xl and xxl is small, but when finetuning the model on task, the xxl results improve by +0.107, whereas the xl results improve only by +0.06. Even though the small model performs poorly in zeroshot mode, when finetuned, the F1 score improved by +0.60. For Flan-T5-xxl tasks with zeroshot settings, the model always outputs the label Entailment or Contradiction given the same label options in the model's instruction. We discovered that Flan-T5 small to xl models also output "Yes" or "No" to the instructions rather than the template's given options. We mapped the Yes and No labels from the model to Entailment and Contradiction, respectively. However, after finetuning it with task-specific data, we avoided this issue with any model. Comparison of zeroshot vs finetuned results of Flan-T5 at different scale is shown in figure 4. The zeroshot and finetined performance of model greatly varies from instruction template to template. We observed a maximum variation of $\pm$ 7 points from the best to worst instruction template.

Model		Test Data
T5-xxl-lm	single + Adam	0.437
Flan-T5-xxl	single + Adam	**0.834**
	multi + Adam	0.820
	single + Adafactor	0.829
	multi + Adafactor	0.822
	single + Adam + >1024 tokens	0.793

Table 4: Results on Test data with single, multi instruction templates and Adam, Adafactor optimizer combination. >1024 indicates model trained with sequence length greater than 1024.

The xxl model performs best when finetuned on task specific data for two epochs. As we can see in table 3, the model performs best on the test set when combined both train and dev data for training. After training with the same data as Flan-T5-xxl, the T5-xxl model, adapted for language modelling, only achieves an F1 score of 0.437. The performance of the instruction tuned model is nearly twice that of the baseline language model. Furthermore, finetuning Flan-T5 with a single instruction template produced the best results, whereas using multiple instruction templates reduced the best F1 score by 0.014 points. Results in table 3 are obtained using the Adam optimizer with other hyperparameter mentioned in 4.3. Using the Adafactor optimizer, the model performance was lower by 0.005 on average across different settings compared to Adam optimizer. The model performed worse by 0.040 points when finetuned with a sequence length greater than 1024.

6 Conclusion

In our work, we have evaluated various instruction tuned Large Language models under the zeroshot setting and finetuned the best-performing instruction tuned T5, Flan-T5-xxl and achieved an F1 score of 0.834 and finished second position in the NLI4CT task. We observed that instruction tuned models are better for datasets with fewer training samples. For Flan-T5, the model performance steadily increased as the model size increased. In the zeroshot setting, we also observe that the in-

[7] https://github.com/huggingface/transformers/issues/10830

struction/prompt given to the model significantly affects its performance. We open-source instruction templates, code and pre-trained models for the reproducibility of our work.

Limitations

In our paper, we have finetuned only Flan-T5 models on the task specific data, and other models are only evaluated under zeroshot setting. This paper focuses solely on Task 1 NLI classification using various instruction template settings and hyperparameter tuning. Another experiment was to combine Tasks 1 and 2 and train the model to classify and identify the supporting factor in the CTR. However, this experiment would make the model's input and output lengths longer than Flan-T5 pretrained. Even though the Flan-T5 model can accept inputs longer than its pretrained length, we have seen a loss in model performance.

The paper doesn't discuss any results using non-generative transformer models like BioBERT. These models are typically trained with a fixed 512 positional embedding, and most input for this NLI4CT task is longer than 512. In our preliminary experiments, these models produced skewed results. Additionally, the BioMedical domain-specific GPT models BioGPT and BioMedLM are skipped because they are not instruction tuned. Their zeroshot prompt/instruction following accuracy is very low without instruction tuning.

Acknowledgements

We thank Navin Kumaran for his invaluable help setting up the AWS GPU instance used in this research project. His technical expertise and willingness to assist were instrumental in the success of our experiments. We also extend our appreciation to Samuel Gurudas for his excellent work in creating the diagrams that are included in this paper.

Furthermore, we thank the anonymous reviewers for their insightful comments and constructive criticism. Their feedback helped us improve our work's quality and strengthen our conclusions.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David A. Sontag. 2022. Large language models are few-shot clinical information extractors. In *Conference on Empirical Methods in Natural Language Processing*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*. Citeseer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

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.

Jay DeYoung, Eric P. Lehman, Benjamin E. Nye, Iain James Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models. *ArXiv*, abs/2005.04177.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donal Landers, and André Freitas. 2023. Semeval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Eric P. Lehman, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J. Smith, Zachary M. Ziegler, Daniel Nadler, Peter Szolovits, Alistair E. W. Johnson, and Emily Alsentzer. 2023. Do we still need clinical language models? *ArXiv*, abs/2302.08091.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. *KR*, 2012:13th.

Valentin Li'evin, Christoffer Egeberg Hother, and Ole Winther. 2022. Can large language models reason about medical questions? *ArXiv*, abs/2207.08143.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6).

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pages 248–260. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. 2021. Bioelectra: pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 143–154.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. Zero-offload: Democratizing billion-scale model training. In *USENIX Annual Technical Conference*, pages 551–564.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

K. Singhal, Shekoofeh Azizi, Tao Tu, Said Mahdavi, Jason Lee Kai Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather J. Cole-Lewis, Stephen J. Pfohl, P A Payne, Martin G. Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, P. D. Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Greg S. Corrado, Y. Matias, Katherine Hui-Ling Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joëlle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge. *ArXiv*, abs/2212.13138.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, A. Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, M. Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddharth Deepak Mishra, Sujan C. Reddy, Sumanta Patro, Tanay Dixit, Xu dong Shen, Chitta Baral, Yejin Choi, Hannaneh Hajishirzi, Noah A. Smith, and Daniel Khashabi. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Hyperparameters

Hyperparameter	Value
Learning rate	7e-5
Batch size	8
Max instruction length	1024
Max output length	8
Epochs	2
optimizer	Adam

Table 5: Hyperparameter used for the best performing model.

The hyperparameter used to achieve the highest F1 score on the task is listed in Table 5, and the total hyperparameter search space is listed in Table 6. Using the Flan-T5 original hyperparameter, 1e-3 learning rate, batch size 128, results in a skewed model with only one label for all test inputs. Max output length is set to 8 as only one label needs to be generated. The maximum output length in the original Flan-T5 implementation is 256. The optimizer-specific hyperparameters have been set

to the default except for the learning rate. The T5-xxl-lm also follows the same hyperparamters as the Flan-T5-xxl model.

Hyperparameter	Value
Learning rate	1e-3, 4e-5, 5e-5, 6e-5, 5e-6, 6e-6, 7e-6, 8e-6
Batch size	8, 16, 32, 128
Max instruction length	1024
Max output length	8
Epochs	1-5
optimizer	Adafactor, Adam

Table 6: The full hyperparameter search space.

B Instruction templates

Instruction templates 1-9 are taken from the FLAN-V2 repository, and instruction template 10 is the final template used for the submission. The F1 score for these templates using the finetuned Flan-T5-xl model is shown in table 7.

1. *"{premise} Based on the paragraph above can we conclude that {hypothesis}? {options_}"*
2. *"{premise} Based on that paragraph can we conclude that this sentence is true? {hypothesis} {options_}"*
3. *"{premise} Can we draw the following conclusion? {hypothesis} {options_}"*
4. *"{premise} Does this next sentence follow, given the preceding text? {hypothesis} {options_}"*
5. *"{premise} Can we infer the following? {hypothesis} {options_}"*
6. *"Read the following paragraph and determine if the hypothesis is true: {premise} Hypothesis: {hypothesis} {options_}"*
7. *"Read the text and determine if the sentence is true: {premise} Sentence: {hypothesis} {options_}"*
8. *"Can we draw the following hypothesis from the context? Context: {premise} Hypothesis: {hypothesis} {options_}"*
9. *"Determine if the sentence is true based on the text below: {hypothesis} {premise} {options_}"*
10. *{premise} Question: Does this imply that {hypothesis}? {options_}*

Template No.	Score
1	0.7520
2	0.7400
3	0.7280
4	0.7415
5	0.7423
6	0.7467
7	0.7228
8	0.7350
9	0.7432
10	**0.7631**

Table 7: Results on Test data using finetuned Flan-T5-xl model trained on Train + Dev data.

C Implementation details

The whole experiment is done using huggingface transformers. We have modified the example summarization script[8] from huggingface repository. The script is modified to drop examples by the Max Sequence length, the original script truncated the inputs. Deepspeed ZeRO-3 configuration is directly used from the huggingface examples[9] without any changes for sequence length upto 1024, for sequence lengths greater than 1024 parameters *stage3_max_live_parameters* and *stage3_max_reuse_distance* is updated to $1e-6$ to avoid memory overflow issue. To experiment with multiple optimizers, the optimizer section in deepspeed config has been removed.

[8] https://github.com/huggingface/transformers/blob/v4.26.1/examples/pytorch/summarization/run_summarization.py

[9] https://github.com/huggingface/transformers/blob/v4.26.1/tests/deepspeed/ds_config_zero3.json