YNU-HPCC at SemEval-2024 Task 7: Instruction Fine-tuning Models for Numerical Understanding and Generation

Kaiyuan Chen, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
`chenkaiyuan@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn`

Abstract

This paper presents our systems for Task 7, Numeral-Aware Language Understanding and Generation of SemEval 2024. As participants of Task 7, we engage in all subtasks and implement corresponding systems for each subtask. All subtasks cover three aspects: Quantitative understanding (English), Reading Comprehension of the Numbers in the text (Chinese), and Numeral-Aware Headline Generation (English). Our approach explores employing instruction-tuned models (Flan-T5) or text-to-text models (T5) to accomplish the respective subtasks. We implement the instruction fine-tuning with or without demonstrations and employ similarity-based retrieval or manual methods to construct demonstrations for each example in instruction fine-tuning. Moreover, we reformulate the model's output into a chain-of-thought format with calculation expressions to enhance its reasoning performance for reasoning subtasks. The competitive results in all subtasks demonstrate the effectiveness of our systems.[1]

1 Introduction

In numerous domains, precise numerical information within text is decisive in decision-making and planning. Understanding and generating text- numbers would be beneficial for improving the model's performance on specific tasks. However, it poses challenges for existing models. Also, previous research indicates that current models struggle to properly represent textual numbers (Chen et al., 2023), often leading to inaccuracies.

Therefore, Task 7 of SemEval (Chen et al., 2024) 2024 focuses on numerically-aware language comprehension and generation, which includes quantitative understanding (Chen et al., 2023), reading comprehension of the numerals in text (Chen et al., 2021), and numeral-aware headline generation (Huang et al., 2023).

We explored all the subtasks of Task 7 and designed corresponding systems for each subtask. Our work and contributions can be summarized as follows:

For Subtask 1, We adopt the paradigm of instruction tuning (Chung et al., 2022) to complete all subtasks and explore manually crafting instances. Our results demonstrate that the instruction tuning model (Flan-T5) (Chung et al., 2022) performs comparably to the BERT model (Devlin et al., 2019) on the Quantitative Understanding task.

For Subtask 2, we utilized the mT5 model (Xue et al., 2021) pre-trained on multilingual corpus and the Randeng-T5 (Wang et al., 2022) pre-trained on Chinese corpus to implement the respective systems, as this task involves Chinese. Consistent with Task 1, we designed an instruction template for inputs and employed instruction fine-tuning.

For Subtask 3, similar instances are retrieved and organized into the input-output format to further enhance model's performance in in-context learning. Specifically, we structured the model's output into the format of chain-of-thought (CoT) (Wei et al., 2022) and inserted calculation expressions to improve model's reasoning performance. Our system achieved the highest scores of ROUGE, BERTScore, and MoverScore in headline generation while ranking 3^{th} in numerical reasoning task.

The remainder of this paper is organized as follows. In Section 2, we describe the related work of our system. The system overview is presented in Section 3. The details of the experiments, main results, and a conclusion are drawn in Sections 4, 5, and 6, respectively.

2 Related Work

In-context Learning. As a novel paradigm, in-context learning (Brown et al., 2020; Chung et al., 2022) has proven to enable large language models to adapt to unseen tasks with instruction and a few

---

[1] Our code is available at `https://github.com/ChenKy23/semeval2024-Task7`

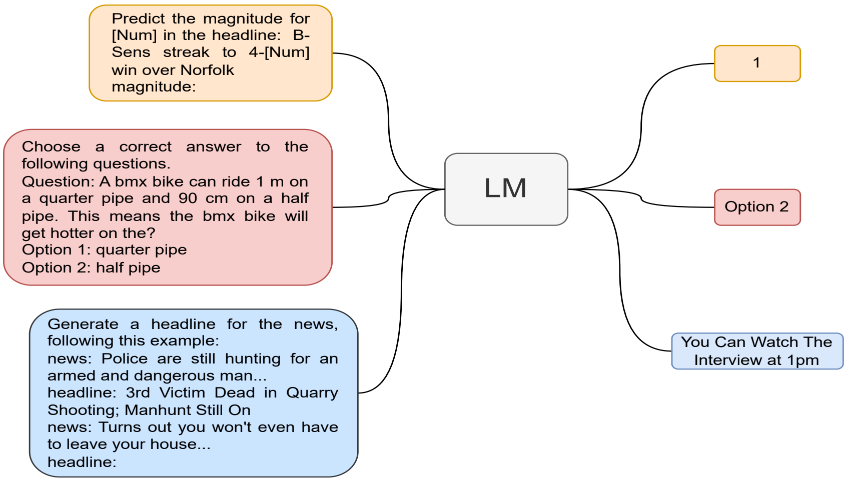
Predict the magnitude for [Num] in the headline: B-Sens streak to 4-[Num] win over Norfolk magnitude:
Choose a correct answer to the following questions.
Question: A bmx bike can ride 1 m on a quarter pipe and 90 cm on a half pipe. This means the bmx bike will get hotter on the?
Option 1: quarter pipe
Option 2: half pipe
Generate a headline for the news, following this example:
news: Police are still hunting for an armed and dangerous man...
headline: 3rd Victim Dead in Quarry Shooting; Manhunt Still On
news: Turns out you won't even have to leave your house...
headline:
LM
1
Option 2
You Can Watch The Interview at 1pm

Figure 1: The application of instruction tuning across different tasks. In our system, LM represents either the Flan-T5 or T5 model. Different tasks employ different instruction templates, with or without demonstrations.

demonstrations, and it doesn't conduct any parameter updates. Furthermore, selecting semantically similar instances can further enhance the model's performance. (Liu et al., 2022; Rubin et al., 2022). Recent work has also applied in-context learning to fine-tuning small models (Fu et al., 2023).

Chain of Thoughts Prompt. CoT prompting (Wei et al., 2022; Kojima et al., 2022) is considered as a method to guide large language models (LLMs) in multi-step reasoning. In the numerical reasoning task of Subtask 3, the model's output can be reconstructed into a CoT format to enhance its reasoning performance. It's important to note that, unlike existing distillation methods (Fu et al., 2023), we don't use a LLMs to generate CoT rationales for each example. Instead, the original labels provided can be used to generate CoT rationales which is a more efficient way.

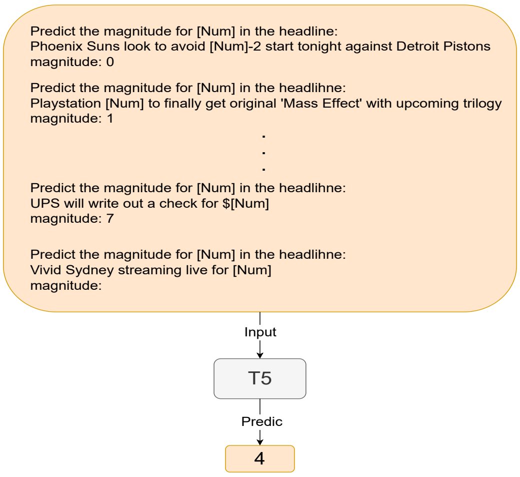
Predict the magnitude for [Num] in the headline:
Phoenix Suns look to avoid [Num]-2 start tonight against Detroit Pistons
magnitude: 0
Predict the magnitude for [Num] in the headlihne:
Playstation [Num] to finally get original 'Mass Effect' with upcoming trilogy
magnitude: 1
Predict the magnitude for [Num] in the headlihne:
UPS will write out a check for $[Num]
magnitude: 7
Predict the magnitude for [Num] in the headlihne:
Vivid Sydney streaming live for [Num]
magnitude:
Input
T5
Predic
4

Figure 2: An example of instruction tuning with demonstrations from QP. According to different magnitudes, 8 instances can be selected manually.

3 Overview of System

3.1 Instruction Tuning

The Instruction tuning models (Flan-T5 or T5) have developed strong generalization abilities through instruction fine-tuning across various tasks. Thus, appropriate instruction can lead to better model performance. While our system is not in a zero-shot setting, introducing instructions during the fine-tuning can enhance the model's performance. Therefore, we consider the input for all subtasks as the instruction template T concatenated with the query input x, i.e., $T + x$. In different tasks, T may have different meanings. The objective function for this process is as follows:

$$L_{instr} = \frac{1}{N}\sum_{i=1}^{N} CE\left(f\left(x_i, T\right), \widehat{y}_i\right) \tag{1}$$

where assume there are a total of N examples, x_i is i-th query input from the dataset, f is the output distribution function of models, CE represents the cross-entropy loss between predicted tokens and target tokens, and $\hat{y}_i$ denotes the tokens from the i-th gold label.

It's worth noting that the input-output formats vary for each subtask. We have designed distinct instruction formats for each task and employed instruction tuning to update the models across all tasks. Figure 1 illustrates how we employ instruction tuned models across various subtasks.

Operators	Expressions
$Copy(v)$	copy v from the news
$Trans(e)$	covert e into a number which represents v
$Paraphrase(v_0, n)$	paraphrase the form of v_0 to other representations in n «v_0/n=v_1»
$Round(v_0, v_1)$	hold v_0 digits after the decimal point of c, that is v_1
$Subtract(v_0, v_1)$	subtract v_0 from v_1 «v_0-v_1=v_2»
$Add(v_0, v_1)$	add v_0 and v_1 «v_0+v_1=v_2»;
$Span(s)$	select a span s from the article which represents 1;
$Divide(v_0, v_1)$	divide v_0 by v_1 «v_0/v_1=v_2»
$Multiply(v_0, v_1)$	multiply v_0 and v_1 «v_0*v_1=v_2»

Table 1: The 9 different operators can be translated into corresponding natural language expressions. Each expression might include an additional calculated number compared to the original operator.

3.2 Instruction Tuning with Demonstrations

The form of instruction tuning can be further expanded, where instructions can be subdivided into prompt P and a list of demonstrations D. P offers explicit guidance for the current task, while D provides the model with demonstrations of the input-output format. This paradigm has been recently referred to as in-context learning in related work (Brown et al., 2020; Chung et al., 2022). Therefore, the objective function for the extended instruction tuning can be expressed as follows:

$$L_{icl} = \frac{1}{N}\sum_{i=1}^{N} CE\left(f\left(x_i, P; D\right), \widehat{y}_i\right) \quad (2)$$

Based on the ways to select demonstrations, this method can be further categorized into manual and similarity-based instruction tuning.

Manual-based Instruction Tuning. This method is employed in Subtask 1 and 2. As subtask 1 involves various aspects, including QP, QNLI, QQA, the different demonstrations can be provided for each task. An example of how instruction is used for QP is shown in Figure 2. The manual selection of demonstrations are based on covering as many different results as possible.

Similarity-based Instruction Tuning. Using similarity-based retrieval for each input is more efficient and leads to better performance (Liu et al., 2022). We employed this method in the headline generation task for Subtask 3. First, pre-trained Sentence-BERT (Reimers and Gurevych, 2019) can be utilized as an encoder to map each news article x_i to a vector v_i. Then, with cosine similarity function F, the distances between v_i and all other vectors can be computed. Finally, the news article corresponding to the vector v_j, which has the closest distance, can be selected as a similar instance. The following function can represent this process:

$$v_i = S(x_i) \quad (3)$$

$$F(v_i, v_j) = \|v_i - v_j\|_2 (or \frac{v_i \cdot v_j}{\|v_i\|_2 \|v_j\|_2}) \quad (4)$$

$${v^i}_{closest} = \underset{j \in \{1,2...N\} \cap j \neq i}{argmin} \quad F(S(x_i), S(x_j)) \quad (5)$$

where N represents the total number of instances in the training set, S is the mapping function of Sentence-BERT, F denotes the cosine similarity function, and x_i represents a news article from the dataset.

3.3 Learning to Reasoning by CoT

As a part of Subtask 3, the numerical reasoning task requires deducing the numbers in masked headlines based on given news articles and approximately 20% of the questions involve reasoning and computation. Related operators can be categorized into 9 types, including *Copy*, *Add*, and others. Thus, directly predicting the numbers may be challenging. An example for NumHG (Huang et al., 2023) can be shown as follows:

Operations = Add(Subtract(5,3), Copy(3))

While this format of the execution process correctly deduces the results, it may not be very intuitive and does not provide the model with interpretable rationales. Therefore, it can be converted into a CoT format, which contains multiple immediate reasoning steps. The above example can be converted into the following CoT format:

First, subtract 3 from 5 «5-3=2»; Second, copy 3 from the news; Third, add 2 and 3 «2+3=5»;

We design corresponding natural language expressions for all 9 operators involved in the dataset and use the program to implement this process automatically. The complete correspondence between operators and natural language expressions

Notation	Model/Method	QP		QNLI					QQA	Score
		comment	headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
Original	BERT	**70.44**	57.46	64.40	**59.20**	72.29	60.42	99.91	53.20	67.17
	Link-BERT	68.81	55.70	59.94	56.85	73.43	59.01	99.91	54.14	65.97
	RoBERTa	60.46	58.03	60.15	57.64	**79.58**	58.77	98.93	51.96	65.69
	Flan-T5$_{inst}$	67.20	58.82	**77.73**	52.40	77.06	68.40	**99.94**	**59.25**	**70.10**
	Flan-T5$_{icl}$	66.68	**59.68**	74.74	52.07	76.85	**70.40**	**99.94**	56.17	69.57
Digit-based	BERT	65.38	54.74	57.86	56.46	71.36	60.11	99.11	53.75	64.85
	Link-BERT	63.76	55.41	59.54	**57.42**	73.63	60.17	99.73	53.44	65.39
	RoBERTa	**69.25**	57.65	59.40	56.69	**78.90**	62.38	99.91	54.34	67.31
	Flan-T5$_{inst}$	67.21	**58.56**	**74.70**	50.97	72.32	**68.40**	**100.00**	**58.02**	**68.77**

Table 2: The comparison between our system and previous work (Chen et al., 2023). The model used is Flan-T5-Base. $instr$ denotes fine-tuning with simple instruction prompts, while icl represents tuning with demonstrations. Refer to section 3.1 and 3.2 for more details. The *Original* refers to the inherent representation of numbers in the text, while *Digit-based* signifies the segmentation of numbers at the character level.

Model	Num Acc			ROUGE			BERTScore			MoreScore
	Overall	Copy	Reasoning	1	2	3	P	R	$F1$	
Flan-T5-Base$_{direct}$	64.247	68.828	55.904	43.64	20.21	39.16	45.56	45.08	45.33	58.84
Flan-T5-Base$_{instr}$	65.180	69.327	57.629	43.94	20.23	39.46	45.87	45.30	45.60	58.90
Flan-T5-Base$_{instr+truncate}$	65.196	69.426	57.493	44.08	20.40	39.50	46.03	45.56	45.80	58.96
Flan-T5-Base$_{icl}$	63.554	67.730	55.949	44.22	20.59	39.68	46.38	45.58	45.99	58.99
Flan-T5-XXL$_{int8\_LoRA+instr}$	**70.686**	**75.262**	**62.352**	48.57	24.40	43.66	50.86	49.62	50.25	60.32
Flan-T5-XXL$_{int8\_LoRA+icl}$	69.044	73.018	61.807	**48.90**	**24.71**	**44.22**	**51.58**	**50.10**	**50.85**	**60.55**

Table 3: The performance of different methods on the headline generation task of NumHG based on ROUGE, BERTScore, and Num Acc. The $direct$ indicates directly fine-tuning the model without instruction, and $truncate$ signifies truncating the input to a length of 512.

is shown in Table 1. Furthermore, expressions can be inserted for each computational operation. An external calculator (Cobbe et al., 2021) can be used for result correction. For the mentioned example, if the model output is *5-3=1*, the external calculator will correct it to the right result, which is *5-3=2*. Subsequently, string matching replaces the incorrect numerical values in the sequence.

4 Experiment Details

Datasets. Subtask 1 utilizes Quantitative 101 (Chen et al., 2023) as the dataset, encompassing three aspects: QP (Chen et al., 2019), QQA (Mishra et al., 2022), and QNLI (Ravichander et al., 2019); Subtask 2 utilizes NQuAD (Chen et al., 2021), which is a Chinese machine reading comprehension task; NumHG (Huang et al., 2023) is used in Subtask 3, which comprises over 27K annotated numeral-rich news articles and can be further divided into headline generation and numerical reasoning.

Model Selection. For Subtasks 1 and 3, Flan-T5 (Chung et al., 2022) is utilized. For Subtask 2, we employ mT5-Small (Xue et al., 2021) and Randeng-T5-77M (Wang et al., 2022). Specifically, for Subtask 3, we experimented with Flan-T5 models ranging from Base to XXL sizes. For Flan-T5-XL and Flan-T5-XXL, we applied 8-bit quantization (Dettmersλ et al., 2022) and performed parameter-efficient tuning using LoRA (Hu et al., 2022). The *all-mpnet-base-v2* [2] can be utilized as encoder to map the text to vector.

Hyper-Parameter Selection. Adamw (Loshchilov and Hutter, 2017) is employed as the optimizer. In Subtask 1, The learning rate for all four QNLI tasks and QQA is set to 5e-7. For the QP task, the learning rate is set to 3e-5. Unless specified, the learning rates, dropout and warm-up rates for remaining tasks are set to 5e-5, 1e-2 and 0.1, respectively. We also applie the PEFT[3] library for parameter-efficient tuning.

Evaluation Metrics. Quantitative-101 Score (Huang et al., 2023) is used for ranking the overall performance in Subtask 1, while Accuracy is used to evaluate Subtask 2 and the numerical reasoning task of Subtask 3. For the numerical reasoning task, based on whether the reasoning question involves calculation, they can be further categorized into *simple* and *complex*. ROUGE (Lin, 2004), BERTScore (Zhang et al., 2020), and MoverScore (Zhao et al., 2019) are used to evaluate the result of the headline generation task of Subtask 3 and nu-

---

[2] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

[3] `https://github.com/huggingface/peft`

Method/Model	Accuracy
BERT Embedding Similarity	57.30
Vanilla BERT	66.41
BERT-BiGRU	67.15
BERT-CNN	63.92
NEMo	69.95
Randeng-T5-77M	**89.71**
mT5-Small	88.82
mT5-Base$_{LoRA}$	80.42

Table 4: The comparative results on NQuAD. Some results come from previous work (Chen et al., 2021). Evaluation is based on accuracy (%).

Model	ROUGE		
	1	2	3
Flan-T5-Base$_{instr}$	44.67	20.90	40.27
Flan-T5-Large$_{instr}$	47.07	22.58	42.04
Flan-T5-XL$_{int8\_LoRA+instr}$	48.36	23.69	43.45
Flan-T5-XXL$_{int8\_LoRA+instr}$	**49.58**	**24.98**	**44.69**
Flan-T5-Base$_{icl}$	44.88	21.02	40.57
Flan-T5-XXL$_{int8\_LoRA+icl}$	**49.60**	**25.27**	**45.01**

Table 5: The results of models at different scales on the dev set of headline generation.

merical accuracy in headlines is also considered.

5 Main Result and Analysis

Comparison Results on Quantitative 101 As results shown in Table 2, despite the distinct ways to handling queries, the instruction tuning Flan-T5 remains comparable to BERT. Notably, our system performs superior on QNLI and QQA, which have smaller datasets. The introduction of manual demonstrations (Sec 3.2) don't lead to improvement in instruction fine-tuning. This may be associated with the manual selection of examples and hyperparameters. Furthermore, in contrast to the *Digit-based* notations, utilizing the *Original* notation for numbers performs better.

Comparison Results on NQuAD As shown in Table 4, it can be observed that the T5 tuning by instruction outperformed the BERT significantly. Both mT5 and Randeng-T5 are pre-trained on multilingual or Chinese corpus, which can enhance their capability to address Chinese-related tasks effectively. Additionally, Randeng-T5, which is based on Chinese corpus, is superior to mT5. However, enlarging the model scale seemed to lead to decreased accuracy on this task.

Comparison Results on NumHG Tables 5 and 6 show that larger models perform better on both headline generation and numerical reasoning.

Table 3 shows that instruction fine-tuning indeed leads to better performance on text generation compared to direct fine-tuning, which means instructions providing proper guidance to Flan-T5. Interestingly, the introduction of similar demonstrations further enhances the model's performance on text generation evaluation metrics but comes at the cost of lower numerical accuracy, which can be observed both in the model of Base and XXL.

Method	Num Acc		
	Total	Simple	Complex
Flan-T5-Base$_{ans\_only}$	88.691	94.205	61.125
Flan-T5-Base$_{operator}$	88.753	94.548	59.780
Flan-T5-Base$_{cot}$	88.509	94.279	59.658
Flan-T5-Base$_{cot+cal}$	88.936	94.279	62.225
Flan-T5-XXL$_{int8\_LoRA+operator}$	93.704	97.164	76.406
Flan-T5-XXL$_{int8\_LoRA+cot}$	94.010	**97.359**	77.262
Flan-T5-XXL$_{int8\_LoRA+cot+cal}$	**94.173**	**97.359**	**78.240**

Table 6: The performance of different methods on the numerical reasoning task. The *cot* is the method proposed in Sec 3.3, and *cal* denotes using an external calculator (Cobbe et al., 2021) for result correction.

As for numerical reasoning, the CoT method leads to better performance on answering *Complex* questions compared to other methods and external calculator correction further amplifies this advantage, as shown in Table 6. For both Base and XXL models, the CoT method under external calculator correction achieved the best performance. However, due to the relatively limited capabilities of smaller models, the performance boost on *Complex* tasks don't contribute significantly to the overall performance for the Base model.

6 Conclusion

During Task 7 of SemEval2024, we participated in all the subtasks and implemented the corresponding systems by instruction fine-tuning. We utilized instruction fine-tuning with demonstrations to expand its format. We also reformulated the output in the form of a chain of thought to improve the model's reasoning abilities. Our approach proved to be highly effective by outstanding performance across all the subtasks. In future work, we plan to further explore the impact of varying instance quantities, instruction templates, and model sizes on the results.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051. We would like to thank the anonymous reviewers for their constructive comments.

References

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.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Nquad: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2925–2929, New York, NY, USA. Association for Computing Machinery.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmersλ, Mike Lewis, Younes Belkada, Luke Zettlemoyer, Hugging Face, and ENS Paris-Saclay. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics.

Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, Yongfeng Huang, Xiayu Li, Yanghan Wu, Junyu Lu, Xinyu Zhu, Weifeng Chen, Ting Han, Kunhao Pan, Rui Wang, Hao Wang, Xiaojun Wu, Zhongshen Zeng, Chongpei Chen, Ruyi Gan, and Jiaxing Zhang. 2022. Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *CoRR*, abs/2209.02970.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *NAACL-HLT*, pages 483–498. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.