ZXQ at SemEval-2024 Task 7: Fine-tuning GPT-3.5-Turbo for Numerical Reasoning

Zhen Qian, Xiaofei Xu, Xiuzhen Zhang
School of Computing Technologies, RMIT University, Australia
S3888611@student.rmit.edu.au
S3833028@student.rmit.edu.au
xiuzhen.zhang@rmit.edu.au

Abstract

In this paper, we present our system for the SemEval-2024 Task 7, i.e., NumEval subtask 3: Numerical Reasoning. Given a news article and its headline, the numerical reasoning task involves creating a system to compute the intentionally excluded number within the news headline. We propose a fine-tuned GPT-3.5-turbo model, specifically engineered to deduce missing numerals directly from the content of news articles. The model is trained with a human-engineered prompt that integrates the news content and the masked headline, tailoring its accuracy for the designated task. It achieves an accuracy of 0.94 on the test data and secures the second position in the official leaderboard. An examination on the system's inference results reveals its commendable accuracy in identifying correct numerals when they can be directly "copied" from the articles. However, the error rates increase when it comes to some ambiguous operations such as rounding.

1 Introduction

Huang et al. (2023) noted a deficiency in contemporary encoder-decoder models when applied to headline generation, specifically addressing imprecisions in the numerals within the generated headlines. To facilitate a thorough investigation of this issue, the authors introduced a novel dataset (i.e., NumHG dataset[1]), consisting of over 21,000 news articles rich in numerals and accompanied by detailed annotations. The dataset is linked to two sub-tasks. The first sub-task centres on numerical reasoning, requiring models to calculate the missing numerals within the headline based on the news articles. The second sub-task focuses on headline generation, requiring models to generate a headline grounded in the provided news content.

This paper focuses on the first subtask of numeral reasoning, aiming to assess the fine-tuned GPT 3.5 turbo's performance in handling numerical reasoning tasks within the context of the newly introduced dataset. Inspired by the idea of instruction tuning (Wei et al., 2022a), we carefully design the textual prompts for training GPT 3.5 turbo to calculate the missing number in the masked headline. Additionally, drawing from the concept of mapping reasoning problems to annotations alongside final answers (Amini et al., 2019; Chiang and Chen, 2019), we carry out experiments to utilize the annotations given in the NumHG dataset. The best fine-tuned model from our experiments achieves an accuracy of 0.94, securing the second position on the official leaderboard. However, we acknowledge that our best model does not rely on the annotations provided in the dataset to generate numerals. Instead, the best model calculates numerals based solely on the news content.

2 Related Work

This work draws inspiration from two key research areas: instruction tuning and leveraging intermediate reasoning steps for solving math word problems. Instruction tuning shows that incorporating prompts or instructions into training datasets, as proposed by Wei et al. (2022a), can improve the language models' performance on unseen data. Sanh et al. (2022) also suggest that using diverse prompts to augment training datasets can help language models to achieve better generalization. Regarding the use of intermediate reasoning steps, some researchers express these steps in symbolic format, such as Chiang and Chen (2019) who train language models to generate equations leading to final answers, and Amini et al. (2019) who map math word problems to predefined operations. Others opt for natural language descriptions, like Cobbe et al. (2021), who propose a system utilizing a language model to generate reasoning steps and final answers in natural language, along with a verifier

[1] https://github.com/ArrowHuang/NumHG.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 218–223

Operator	Description
Copy	Copy from the article
Trans	Convert into a number
Paraphrase	Paraphrase the form of digits to other representations
Round	Hold some digits after the decimal point of a given numeral
Subtract	Subtract a from b
Add	Add a and b
Span	Select a span from the article
Divide	Divide a by b
Multiply	Multiply a and b

Table 1: overview of the pre-defined operations given in the dataset.

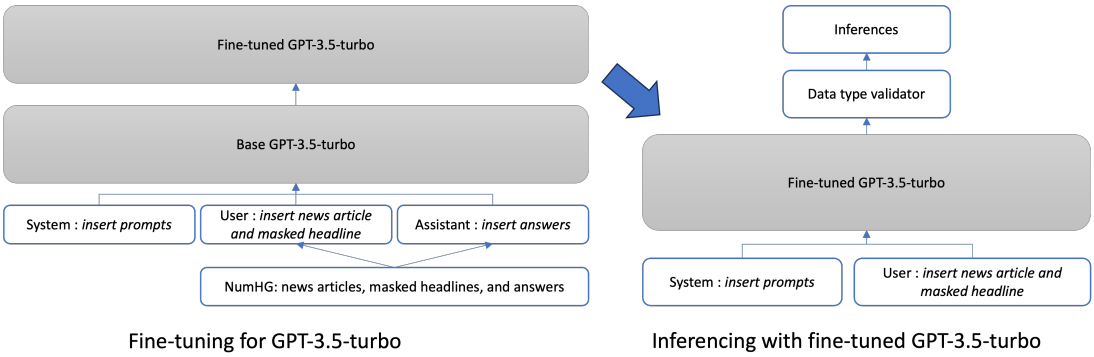

Figure 1: fine-tuning and deploying process for the GPT-3.5-turbo

model to assess the generated answers.

3 Problem Definition

The NumHG dataset comprises 21,157 pieces of news articles, each ranging between 200 to 300 words in the content. The headlines of these news articles include one or more numerals, with one numeral intentionally omitted from each headline. The novelty of the NumHG dataset lies in its provision of meticulous annotations detailing the computational processes used to arrive at the missing numerals in the headlines. This dataset articulates 12 fundamental operations for numeral computation, encompassing actions such as copying, addition, and multiplication. The pre-defined operations are listed in Table 1 (Huang et al., 2023). While in certain cases, computing the correct numerals may require a single operation, such as straightforwardly copying the numerals from the news content as the correct answers, in other instances, it may involve a sequential combination of multiple operations. For example, this could involve rounding the result after adding two numerals found in the news content. The sub-task of numerical reasoning requires us to develop a system capable of accurately calculating the omitted numerals based on the news contents and potentially utilizing the provided annotations.

4 System Overview

In this section, we provide a detailed introduction to our system. Figure 1 illustrates a general outline of our system.

4.1 Model Description

This paper aims to assess the performance of a fine-tuned GPT-3.5-turbo-0613 model in numerical reasoning, specifically its accuracy in computing the desired numerals within news headlines based on content of the news articles. Through the process of fine-tuning, we aim to harness the capability of the GPT-3.5-turbo model by training it with the NumHG dataset, tailoring its performance to our designated task. As of the period during which our experiments were conducted, GPT-3.5-turbo-0613, unveiled in June 2023, stands as the most recent iteration released by OpenAI. This version offers enhanced steerability and reduced costs associated with input tokens. The experiments conducted centre around the refinement of the base model through the fine-tuning process.

The fine-tuned model is then employed for making inferences on the test data. We also imple-

role	content
system	“you will be given a piece of news with prefix ’news:’. you will also be given an incomplete headline with the prefix ’masked headline:’. based on the news content, please output the missing number in the masked headline. please ensure the output is the number only rather than the whole sentence.”
user	“news: news content for each instance in the dataset. masked headline: masked headline for each news article in the dataset”
assistant	“the omitted number from the headline”

Table 2: prompts employed in experiment 1 – training the model to calculate the numerals directly from the news content

role	content
system	“you will be given a piece of news with prefix ’news:’. you will also be given an incomplete headline with the prefix ’masked headline:’. based on the news content, please output how the missing number in the masked headline is calculated together with the final answer. please ensure the operations follow the format below: Copy(v), Trans(e), Paraphrase(v, n), Round(v, c), Subtract(v0, v1), Add(v0, v1), Span(s), Divide(v0,v1), Multiply(v0,v1).”
user	“news: news content for each instance in the dataset. masked headline: masked headline for each news article in the dataset”
assistant	"calculations";“the omitted number”

Table 3: prompts employed in experiment 2 – training the model also to generate the operations

ment a program for verifying the data types of the model’s outputs. For the sub-task of numerical reasoning, the expected outputs should be numerical values. Our program systematically converts any non-numerical outputs to the value of 0.

4.2 Prompt Design

To fine-tune a GPT-3.5-turbo using the OpenAI API, it is necessary to convert each instance in the dataset into a format compatible with the model[2]. This involves defining the content for three distinct roles for each instance. The roles include system, user, and assistant. The content assigned to the system role consists of instructions and prompts directed towards the model. For the user role, the corresponding content involves inputs provided to the model, including questions. The content allocated to the assistant role consists of the expected outputs or answers. We mainly carried out two experiments. For the first experiment, we train the model to calculate the numerals directly from the news content. For the second, in addition to the numerals, we also instruct the model to generate the operations required to reach the numerals. The specific prompts employed in this process are presented in Table 2 and Table 3. More detailed examples are shown in Figure 2 and Figure 3.

[2] https://platform.openai.com/docs/guides/fine-tuning

5 Experiment setup

The dataset allocated for the sub-task of numerical reasoning comprises 21,157 instances, which is further split into 80% for training data and 20% for test data. Fine-tuning of the GPT-3.5-turbo-0613 model is performed via the OpenAI API, adhering to the guidelines outlined in the OpenAI API documentation. The hyper-parameters for model training, including learning rate, batch size, and epochs, are configured to auto. Throughout the training phase, evaluation metrics such as training loss, training token accuracy, validation loss, and validation token accuracy are provided by OpenAI. Following the completion of the training process, the fine-tuned model is employed to generate inferences on the test data with temperature set to default.

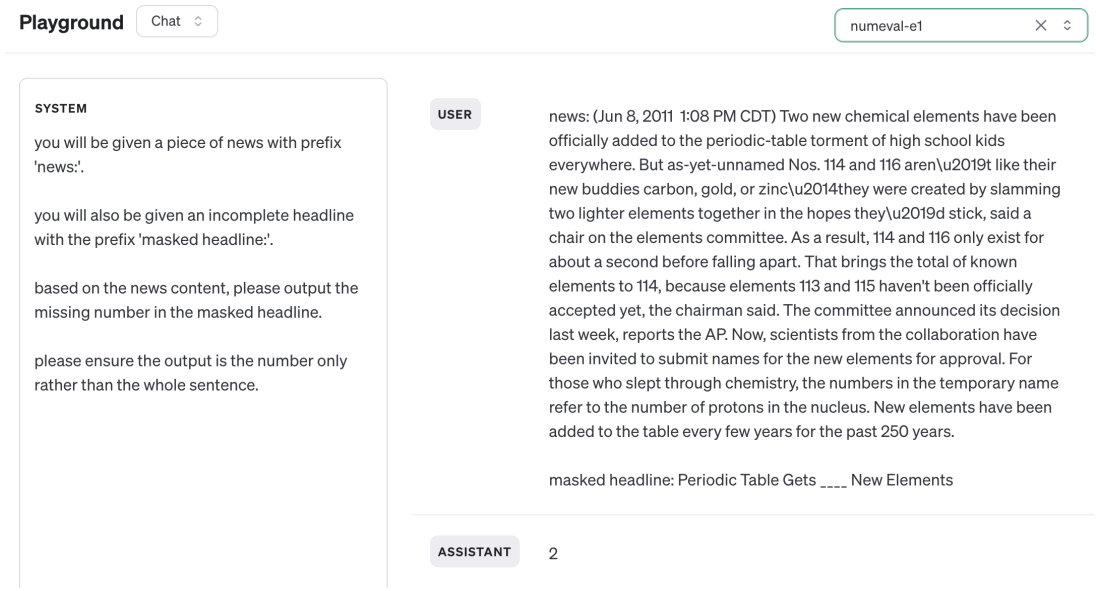

Figure 2: one example for the prompts used for instructing the model to calculate the numerals directly

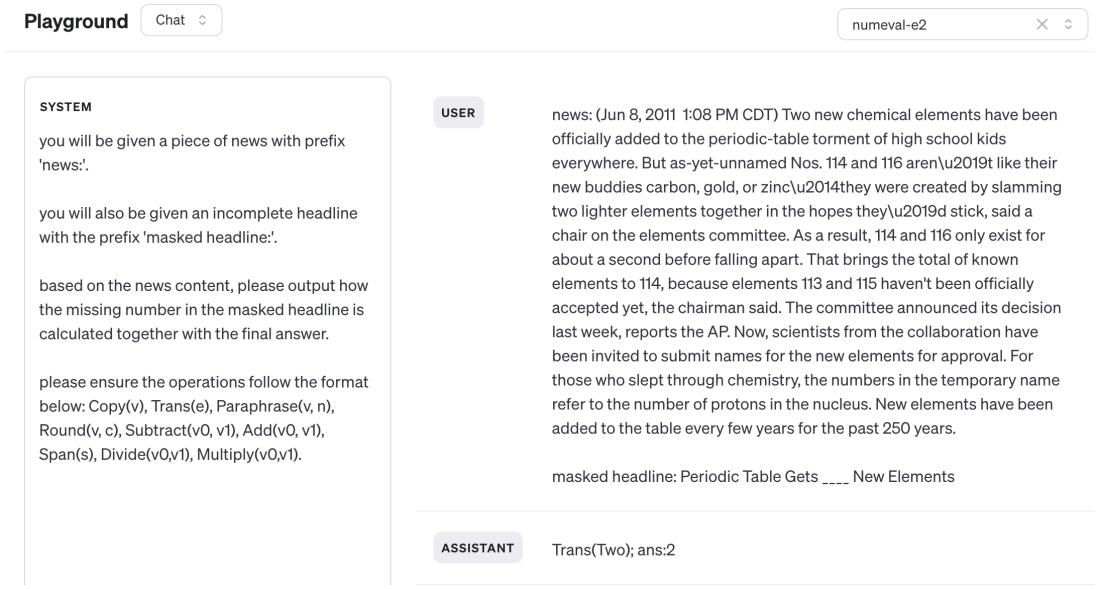

Figure 3: one example for the prompts used for instructing the model to also generate the operations

6 Results

The fine-tuned GPT-3.5-turbo-0613 model from the experiment 1, which calculates the numerals directly from news content, demonstrates a commendable accuracy of 0.94, securing the second position on the leaderboard. The model from the experiment 2, which also output the intermediate operations required to arrive at the numerals, only achieves an accuracy of 0.90. An analysis of model errors has been conducted, with detailed statistics presented in Table 4.

The test data comprises 4,921 instances, featuring a total of 5,237 operators in the annotations. Table 4 reveals that error rates, for both models, are notably low for operations such as Copy, Trans, and Paraphrase, while they are comparatively high for Round, Multiply, Add, and Divide. Given that 88.5% of the operations in the test data pertain to Copy, Trans, and Paraphrase, the model's commendable performance in these three operations significantly contributes to its overall accuracy.

It is important to note that our models cannot detect unanswerable questions in the test data. This anomaly arises from the fact that there are no unanswerable questions in the training data. The models do not learn to predict unanswerable questions in

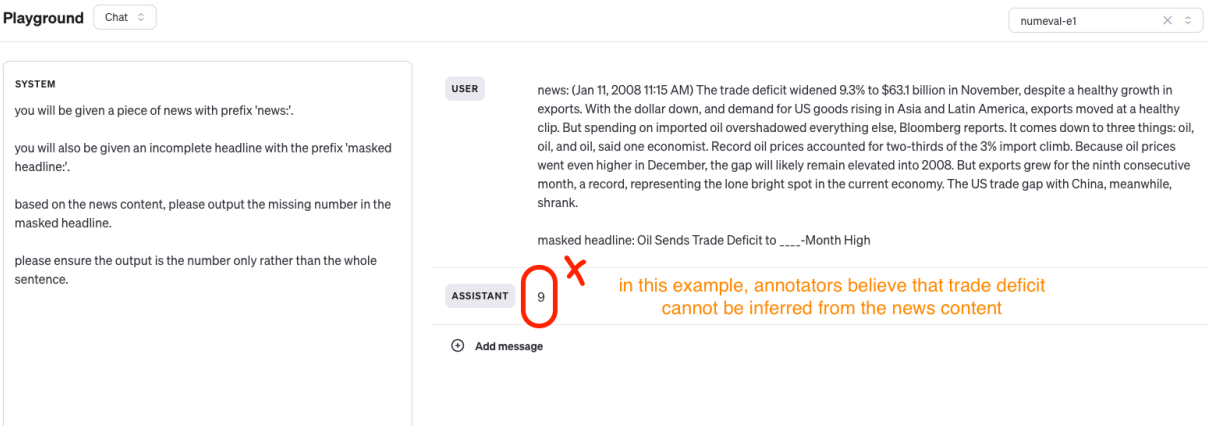

Figure 4: one example for unanswerable questions

Operator	Model 1 Error Rate	Model 2 Error Rate
Copy	4.32%	6.79%
Trans	3.13%	11.37%
Paraphrase	7.39%	15.43%
Round	41.08%	62.70%
Subtract	18.69%	43.93%
Add	24.00%	49.00%
Span	10.34%	30.17%
Divide	23.68%	55.26%
Multiply	35.71%	52.38%

Table 4: error analysis on the test data

the training phase. The unanswerable questions refer to instances identified by annotators when the numerals cannot be inferred from the news content. Figure 4 shows one example of the unanswerable questions in the test data. Another anomaly arises as the model trained for generating intermediate operations shows lower accuracy, contradicting prior works' conclusion that incorporating intermediate reasoning steps in symbolic formats should enhance language model performance by Chiang and Chen (2019) and Amini et al. (2019). To improve the model's ability to utilize annotations and boost overall accuracy, future research should explore alternative methods such as experimenting with different language models, adjusting model architecture, and employing Chain-of-Thought prompting (Wei et al., 2023; Ling et al., 2023).

7 Conclusion

In this paper, we propose to finetune the GPT-3.5-turbo specifically tailored for handling numerical reasoning in the headline generation context. Through the carefully engineered prompt that aggregates the content of news articles and the masked headlines during the fine-tuning process, we achieved an accuracy of 0.94 and ranked second in the official leaderboard. However, our model exhibits relatively high error rates particularly in operations such as rounding numbers. Additionally, further development is needed to better utilize the annotations to improve the model's accuracy.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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 2357–2367. Association for Computational Linguistics.

Ting-Rui Chiang and Yun-Nung Chen. 2019. Semantically-aligned equation generation for solving and reasoning math word problems. 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 2656–2668. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

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

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. 36:36407–36433.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.