IUST-NLPLAB at SemEval-2024 Task 7: Numeral Prediction using gpt3.5

Sina Alinejad
Iran University of
Science and Technology
sinaalinejad4@gmail.com

Erfan Moosavi Monazzah
Iran University of
Science and Technology
moosavi_m@comp.iust.ac.ir

Sauleh Eetemadi
Iran University of
Science and Technology
sauleh@iust.ac.ir

Abstract

In this paper, we present our approach to the SemEval-2024 numeral reasoning task, which requires filling in a blank with a number based on a given sentence. We first attempted to predict the arithmetic operation needed to compute the correct answer and obtained some statistical insights from this process. We performed operation prediction in two ways: as a 9-class classification problem and as a set of binary classification problems for each operation. However, due to the low accuracy of this method, we switched to a zero-shot learning strategy that leverages natural language inference models to solve the task.

1 Introduction

Headline generation is the task of summarizing a full-length article into a brief, catchy, and informative line of text. A key challenge in this task is to preserve the numerical information from the article, as numerals often convey important facts and figures. However, existing encoder-decoder models, despite achieving high ROUGE scores, tend to generate inaccurate or unreasonable numerals in headlines. One of the main reasons for this problem is the scarcity of datasets that provide detailed annotations for numeral generation.

To address this gap, the authors of (Huang et al., 2023) introduce the NumHG dataset, which consists of more than 27,000 numeral-rich news articles with fine-grained annotations. These annotations indicate how the numerals in the headlines can be derived from the numerals in the articles, using various arithmetic operations and transformations. The NumHG dataset enables the evaluation of numeral accuracy, reasonableness, and readability in headline generation. Moreover, the dataset covers both English and Chinese languages, allowing for cross-lingual studies. By emphasizing the role of numerals, the NumHG dataset aims to advance the state-of-the-art in number-focused headline generation and foster further research in numeral-focused text generation.

In this paper, we present our system for the NumHG task, which is based on zero-shot learning using gpt3.5. We first apply some preprocessing steps to the dataset, such as tokenization, normalization, and masking. Then, we use gpt3.5 to generate headlines by reformulating the task as a natural language inference problem. We compare our system's performance on different types of operations, such as copy, trans, paraphrase, round, subtract, add, span, divide, multiply, and sround. We find that our system performs well on some operations, such as copy and trans, but poorly on others, such as round. Our system ranks 12th in the leaderboard with an accuracy of 74 percent.

Additionally, we have made our code openly accessible on GitHub[1] to facilitate reproducibility and further research endeavors.

2 Background

2.1 Dataset Description

There are 21157 samples in the training set and the validation set contains 2572 samples. each sample contains the fields "news", "masked headline", "calculation" and "ans". Table 1 demonstrates an example from the dataset. The objective is to ensure accurate numeral generation in headlines, and as such, detailed annotations on how to secure the correct numeral through specific operations are provided. The whole dataset is in the English language. In this task, we are asked to predict the correct numeral value that the masked headline must be filled with based on the news.

[1] https://github.com/sinaalinejad/SemEval2024_task7_NumEval

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1087–1091
June 20-21, 2024 ©2024 Association for Computational Linguistics

2.2 Related Work

The task of headline generation, a form of text summarization, endeavors to condense a lengthy source text into a succinct summary. Text summarization approaches typically fall into two categories: extractive and abstractive. Extractive approaches involve selecting fitting sentences from the source text to serve as the summary, while abstractive approaches strive to create new sentences to encapsulate the source text. The concept of headline generation aligns more closely with abstractive methodologies.

The emergence and development of large-scale pre-trained models like Lewis et al., Raffel et al. and Zhang et al., have notably advanced the capabilities of abstractive summarization models, to the extent that they now outperform extractive models. Some recent studies like Dou et al., Liu et al. and Wang et al., emphasize the significance of keyword sentences, asserting that these should be leveraged as guides for summary generation. GSum (Dou et al., 2021), for example, initially performs extractive summarization, then incorporates the extractive summaries into the input for abstractive summarization. Despite experimental evidence supporting GSum's effectiveness, Wang et al. argue that extractive summaries do not provide a reliable or flexible guide, potentially leading to information loss or noisy signals.

To tackle this issue, SEASON(Wang et al., 2022) adopts a dual approach, learning to predict the informativeness of each sentence and using this predicted information to guide abstractive summarization. Meanwhile, BRIO(Liu et al., 2022) employs pre-trained abstractive models to generate candidate summaries, assigning each a probability mass according to their quality and defining a contrastive loss across the candidates. By considering both token-level prediction accuracy and sequence-level coordination, BRIO combines cross-entropy loss and contrastive loss for abstractive summarization.

3 System Overview

3.1 Zero-Shot system

Our system is simply inferring the output by zero-shot learning. The input is given to gpt-3.5-turbo along with a prompt. The prompt is: "Act as a news expert. I have a text of news and its masked headline with a mask token specified as [MASK]. The mask should be filled with a numerical value. you should just give me the numerical value to put instead of the [MASK]. You should do some calculations to obtain the final number to put instead of [MASK] and these calculations are as follows: Copy(v): Copy v from the article Trans(e): Convert e into a number Paraphrase(v,n): Paraphrase the form of digits to other representations Round(v,c): Hold c digits after the decimal point of v Subtract(v0,v1): Subtract v1 from v0 Add(v0,v1): Add v0 and v1 Span(s): Select a span from the news Divide(v0,v1): Divide v0 by v1 Multiply(v0,v1) Multiply v0 and v1 the news is: <NEWS> the masked headline is: <MASKED HEADLINE>. your response should be in the format of JSON with the key of ans and value of the numerical answer, so do not include any of your calculation processes."

Table 1: An annotation example in NumHG.
News: At least 30 gunmen burst into a drug rehabilitation center in a Mexican border state capital and opened fire, killing 19 men and wounding four people, police said. Gunmen also killed 16 people in another drug-plagued northern city. The killings in Chihuahua city and in Ciudad Madero marked one of the bloodiest weeks ever in Mexico and came just weeks after authorities discovered 55 bodies in an abandoned silver mine, presumably victims of the country's drug violence. More than 60 people have died in mass shootings at rehab clinics in a little less than two years. Police have said two of Mexico's six major drug cartels are exploiting the centers to recruit hit men and drug smugglers, ...
Headline (Question): Mexico Gunmen Kill _____
Answer: 35
Annotation: Add(19,16)

Table 1: An annotation example in NumHG.

In the next step, we tried this system on a set of 100 samples from the training dataset with different prompts and the best result was an accuracy of 80 percent. Then we decided to have a set of 200 samples from the validation set but this time, the distribution of different records based on the field "calculation" was the same as the whole validation set; This time the accuracy was 77 percent.

At the end, we extract the number from the model response.

Metric	Value
Precision	0.32
Recall	0.27
F1	0.28
Accuracy	0.81

Table 2: Different metrics in operation prediction in 9-way classification using gpt2

Operation	Acc	Operation	Acc
Copy	0.9	Add	0
Trans	0.64	Span	0
Paraphrase	0.72	Divide	0
Round	0.37	Multiply	0
Subtract	0.05	Sround	0

Table 3: Accuracies for each operation in operation prediction in 9-way classification using gpt2

3.2 Operation prediction

In our investigation, we endeavored to forecast arithmetic operations using textual information extracted from news articles. To achieve this, we meticulously fine-tuned the GPT-2 language model for this specific task. The culmination of our efforts yielded the following outcomes:

Model Fine-Tuning: We conducted rigorous fine-tuning of the GPT-2 model, adapting it to the novel context of arithmetic prediction based on news content.

Performance Evaluation: Subsequently, we evaluated the model's accuracy for each arithmetic operation. The results are shown in Table 2:

The results for each operation are succinctly summarized in Table 3.

In our research endeavor, we revisited the application of the GPT-2 language model to binary classification tasks. Specifically, we aimed to predict the outcome of various arithmetic operations. Our investigation involved meticulous dataset creation, model fine-tuning, and performance evaluation. Below, we outline the key steps and findings of our study. To construct robust binary classification datasets, we adhered to a balanced approach. For each arithmetic operation (e.g., addition, subtraction, multiplication, etc.), we meticulously curated positive and negative samples.

1. Positive Samples: We collected all positive samples corresponding to each arithmetic operation.

2. Negative Samples: Achieving parity between positive and negative samples was crucial. Therefore, we ensured that the number of negative samples matched that of positive ones. However, the challenge lay in diversifying the negative samples. To address this, we introduced Distribution-Based Sampling in which each arithmetic operation in the negative sample was selected based on its distribution across all negative instances. For instance, if we were dealing with the "copy" operation and we gathered 50 positive instances from relevant data sources and the "trans" operation constituted 10% of all negative samples, we allocated 5 negative samples specifically for this operation.

$$50 * 0.1 = 5$$

The results for this method was around random classification, so we didn't continue on that.

We provide both the 9-way classification dataset and the binary classification dataset on HuggingFace[2] for public use. Researchers and practitioners can leverage these datasets for future investigations.

Our study underscores the challenges in predicting arithmetic outcomes from news content. Future research could explore alternative models, feature engineering techniques, or domain-specific adaptations to enhance classification accuracy. Additionally, investigating the impact of dataset size and quality on model performance remains an open avenue for exploration.

In summary, while our initial results did not yield groundbreaking accuracy, the datasets we present serve as valuable resources for the scientific community. As the field of natural language processing continues to evolve, we remain optimistic about refining predictive models for diverse applications.

4 Experimental Setup

4.1 Pre-processing

The news and the masked headline are preprocessed in these manners:

1. converting new line character and tab to space

2. removing the commas from comma-separated numbers, this can help the model to better understand the numbers

[2] https://huggingface.co/Sina-Alinejad-2002

3. replacing the blank in the masked headline with a new mask

4. converting some unknown characters to the closest ASCII equivalent for example uff05 to %. This makes the context easier to understand for the model

4.2 Evaluation Metrics

As this is a prediction task, we have used an accuracy metric to evaluate our model. However, there is no training stage in our system, so this metric is not used to update any parameter and it is just for us to change some hyperparameters like the prompt.

4.3 Others

We also used the tenacity library to handle some errors that may cause the cell to stop such as TimeLimit error or RateLimit error. For this, we set a retry decorator for the main function wait for 20 seconds after an error has occurred, and retry the request to API and this is for a maximum of 3 times.

```
@retry(stop=stop\_after\_attempt(3),
wait=wait\_fixed(20))
```

5 Results

5.1 Overall Performance

The overall performance of our system on the test dataset was 74 percent. We also calculated the accuracy of each of the 10 operations and the result is shown in table 4.

5.2 Error Analysis

The system performs poorly on predicting answers that require the round operation to be applied and this is probably because the model tends just to copy the exact number in the blank or round it in different ways. On copy and trans operations, the results are the best compared to others which are around 50 percent.

6 Conclusion

In this paper, we have presented a zero-shot learning system for the NumHG task, which leverages gpt3.5 to generate headlines with accurate and reasonable numerals. Our experimental results show that our system can handle simple operations, such as copy and trans, but fails to perform complex operations, such as add, subtract, and round. This indicates that current LLMs like gpt3.5 still have limitations in capturing the numerical reasoning and arithmetic skills required for the NumHG task.

Operation	Acc	Operation	Acc
Copy	0.82	Add	0.46
Trans	0.81	Span	0.5
Paraphrase	0.54	Divide	0.54
Round	0.02	Multiply	0.4
Subtract	0.5	Sround	0

Table 4: Accuracies based on the operation used to calculate the answer

For future work, we propose to explore the possibility of using multiple agents to collaborate on the task. This could involve either having different agents specialize in different operations or having a voting mechanism to select the best answer from multiple agents. We believe that this could improve the overall performance and robustness of our system, and also provide more insights into the strengths and weaknesses of different LLMs.

We suppose that it would be also useful to extend the exploration of numeral reasoning tasks by incorporating few-shot learning techniques. This approach will allow us to delve deeper into the performance enhancements across various operations, providing a more granular understanding of the model's capabilities. Furthermore, we can transcend beyond merely predicting the final answer. Inspired by the iterative prompting methodology of Chain of Thought (Wei et al., 2023), it would be possible to endeavor to refine our model's reasoning process. This will involve guiding the model to deduce the correct set of operands and the associated operation before executing it, thereby fostering a more transparent and interpretable reasoning pathway. Such advancements will not only bolster the model's accuracy but also its ability to articulate the reasoning behind its conclusions, paving the way for more robust and reliable numeral reasoning systems. The impact of dataset size and quality is also an open avenue to explore, one such experiment has been conducted by (Jain et al., 2020).

References

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization.

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*.

Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. 2020. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3561–3562, New York, NY, USA. Association for Computing Machinery.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022. Salience allocation as guidance for abstractive summarization.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.