CYUT at SemEval-2024 Task 7: A Numerals Augmentation and Feature Enhancement Approach to Numeral Reading Comprehension

Tsz-Yeung Lau
Department of CSIE
Chaoyang University of Technology
Taichung, Taiwan
s10927116@gm.cyut.edu.tw

Shih-Hung Wu†
Department of CSIE
Chaoyang University of Technology
Taichung, Taiwan
shwu@cyut.edu.tw

Abstract

This study explores Task 2 in NumEval-2024, which is SemEval-2024(Semantic Evaluation) Task 7 , focusing on the Reading Comprehension of Numerals in Text (Chinese). The dataset utilized in this study is the Numeral-related Question Answering Dataset (NQuAD), and the model employed is BERT. The data undergoes preprocessing, incorporating Numerals Augmentation and Feature Enhancement to numerical entities before model training. Additionally, fine-tuning will also be applied. The result was an accuracy rate of 77.09%, representing a 7.14% improvement compared to the initial NQuAD processing model, referred to as the Numeracy-Enhanced Model (NEMo).

1 Introduction

Numeric information holds a crucial significance within narratives across various domains, including medicine, engineering, and finance. (Chen et al., 2021) Numerals presented in tables (Ibrahim et al., 2019) and the content (Lamm et al., 2018) of a document have garnered considerable attention from researchers. Machine-based numeral comprehension stands out as an emerging research area, still in its nascent stages. NumEval-2024 Task 2(SemEval-2024 Task 7) focus on reading comprehension of the numerals in text, models are required to identify the correct numerical value from four given options, based on a provided news article. The dataset utilized for this task is NQuAD (Chen et al., 2021), which is in Chinese.

The initial model devised for addressing this task is referred to as the Numeracy-Enhanced Model (NEMo), which achieves an accuracy of 69.95%.(Chen et al., 2021) In this study, a pre-trained BERT model will be employed to undertake the task, with the objective of enhancing accuracy through data preprocessing, Numeral Augmentation, Feature Enhancement, and fine-tuning. [1]

[1]†Contact Author

The remaining sections of this study are structured as follows: In the second section, we delve into a discussion of the data and its preprocessing. The data undergoes denoising through the removal of special symbols.

Moving on to the third section, we thoroughly examine the methodology employed, the results obtained, and the subsequent discussion. The data undergoes further denoising through stop word removal. Additionally, numeral augmentation and feature enhancement are introduced. Numeral augmentation reduces dependence on specific numerical values, while feature enhancement aims to compel the model to pay attention to numerals. As a result, our model achieves an accuracy of 77.09% on the NQuAD dataset.

The concluding section presents a summary of the insights gained, along with considerations for future research endeavors. Although the model is performing well, there are opportunities for further enhancement. We will delve into these matters in the Error Analysis and Discussion section.

2 Data Preparation

2.1 Data Source

The dataset employed in this study is the Numeral-related Question Answering Dataset (NQuAD). This dataset poses greater challenges compared to numeral-related questions in other datasets. All data were sourced from news articles spanning the period from June 22, 2013, to June 20, 2018, encompassing a total of 75,448 Chinese news articles. Notably, 59.74% of news headlines include at least one numeral, while numerals are present in 99.80% of news contents. The dataset comprises 43,787 news articles and 71,998 questions.(Chen et al., 2021)

The NQuAD dataset encompasses six columns: "news_article","question_stem","answer_options", "ans","target_num" and "sentences_containing_the

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 579–585

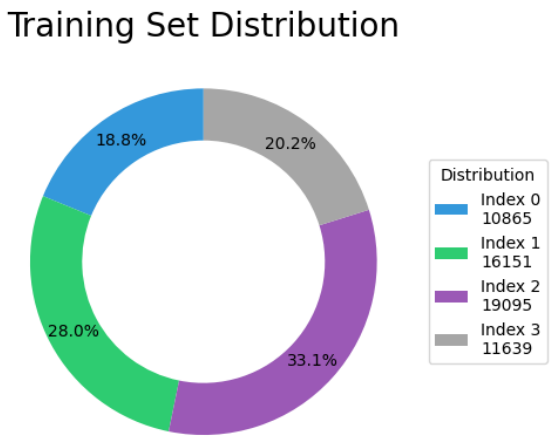

Figure 1: Training Set Distribution

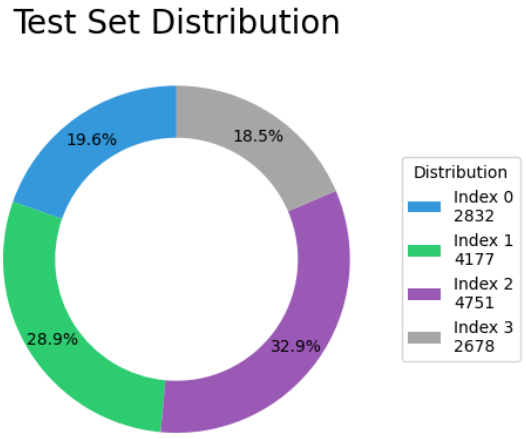

Figure 2: Test Set Distribution

_numeral_in_answer_options". The "news_article" column contains the content of the article, while the "question_stem" column represents the questions posed. The "answer_options" column consists of a list of four answer choices, with the "ans" column indicating the index of the correct answer within the "answer_options." The "target_num" column contains the content of the correct answer, and the "sentences_containing_the_numeral_in_answer_options" (scao) is a list of sentences in the article that include the correct answer.

NQuAD provides both a training set and a test set. The training set comprises 57,750 samples, while the test set includes 14,438 samples, maintaining an approximate 8:2 ratio. The distribution of the four categories varies within the training set, encompassing 10,865 instances where the correct answer index is 0, 16,151 instances for index 1, 19,095 instances for index 2, and 11,639 instances for index 3. In the test set, there are 2,832 instances with the correct answer index at 0, 4,177 instances at index 1, 4,751 instances at index 2, and 2,678 instances at index 3. The data depicted in Figure 1 and Figure 2 reveals that approximately 20% is attributed to both index 0 and index 3, while around 30% is associated with both index 1 and index 2.

2.2 Data Preprocessing

The dataset used in this study, NQuAD, provides the "scao" column, which constitutes a list of sentences within the article containing the correct answers. A new column, denoted as "article", is formed by combining the contents of the "scao", "question_stem" and "answer_options" columns. The "article" column serves as the input for the model. Additionally, the "ans" column, representing the index of the correct answer within the "answer_options" column, is employed to generate a new column named "label". The "label" column functions as the output for the model.

After creating the new input column "article", the content of the "article" column undergoes preprocessing. Special symbols such as "[# &" are removed, taking care not to eliminate symbols that may affect the semantics, such as "+ - . % $". Subsequently, HTML tags are removed, along with excess whitespaces within sentences. Finally, commas in numbers are removed, for instance, in cases like "1,000", to facilitate subsequent batch processing.

Please be aware that pre-processing will be applied to both the training set and the test set.

3 Method

In this study, the pretrained BERT model served as the baseline. To enhance model performance, two steps were implemented. First, BERT with stop word removal (BERT-SWR) was introduced. This step contributes to improved model performance through Dimensionality Reduction, Noise Reduction, and Enhanced Generalization. Building upon BERT-SWR, additional measures were taken, including numerals augmentation and feature enhancement(BERT-NAFE). Numerals in the text were replaced with a special symbol and the corresponding answer index. This substitution aimed to eliminate the model's consideration of the meaningless numerical values in the article. The use of the special symbol, represented as cashtag, compelled the model to prioritize attention to

the numerals within the text. This additional step further increased the model's accuracy. Please be aware that prior to evaluation, fine-tuning was applied to each model.

3.1 BERT

A pretrained BERT model was used as the baseline model. The complete designation for BERT is Bidirectional Encoder Representations from Transformers, a bidirectional unsupervised language representation model based on transformers, initially introduced by Google. It undergoes predominantly pre-training through the application of both the Masked Language Model (MLM) and Next Sentence Prediction (NSP) techniques. In contrast to word2vector (Pennington et al., 2014) and GloVe (Mikolov et al., 2013), which operate without considering context, BERT exhibits the capability to leverage contextual information during inference. This contextual understanding contributes to its superior performance across diverse tasks (Devlin et al., 2019).

3.2 Stop Word Removal

The preprocessing technique known as removing stop words in natural language processing (NLP) is intended to exclude commonplace yet generally uninformative words, such as "和", "你", "而是", etc., from the textual content. By eliminating these words, low-level information is excluded from the text, enabling a heightened emphasis on crucial information. This step is undertaken with the aim of enhancing model performance.

In this study, a tokenizer named jieba (Sun, 2020) was employed to segment the sentences. A stop-word list sourced from Sichuan University (goto456, 2019) was utilized to determine phrases that should be eliminated. The Python OpenCC package is employed to perform the translation from simplified Chinese to traditional Chinese for the stop-word list. For example, let's examine a phrase: "而是前一世代的iPhone 6s." Subsequent to the processing, the term "而是" is eliminated, resulting in the refined expression "前一世代的iPhone 6s."

3.3 NA and FE

3.3.1 Numerals Augmentation

Reducing the model's excessive reliance on specific numerical values and addressing numerical ambiguity are pivotal objectives in Numeral Augmentation for tasks related to Numeral Reading Comprehension.

article:
年增**22.09**% ,
精華 1565 董事昨 **23** 日通過去年盈餘分配案 ,
擬每股配發現金股利**27**元 , 營益率**34.82**% ,
精華去年EPS 35.26元寫新高 ·
擬每股配息___元 ,
22.09 , 23 , 27 , 34.82

answer_options: ['22.09', '23', '27', '34.82']

processed article:
年增**0**% ,
精華 1565 董事昨 **1** 日通過去年盈餘分配案 ,
擬每股配發現金股利**2**元 , 營益率**3**% ,
精華去年EPS 35.26元寫新高 ·
擬每股配息___元 ,
0 , 1 , 2 , 3

Figure 3: artical after NAFE pre-processing.

In this study, each answer's numeral will be replaced by its corresponding index, ranging from 0 to 3. By addressing these aspects, Numeral Augmentation contributes to enhancing the model's robustness and interpretative capabilities when confronted with diverse numerical contexts in reading comprehension tasks. For example, as illustrated in Figure 3, the numerical values in the answer options were replaced with their corresponding indices. Upon examining the processed article, the numbers 0, 1, 2, and 3 were underscored, indicating the replacements for the actual answers in the article.

3.3.2 Feature Enhancing

In this study, Feature Enhancement has been applied to numerical values. Specifically, the digits representing the answers in the text are augmented with cashtag both preceding and following them. Due to the attention mechanism of the model, there is a likelihood that attention may concentrate in the vicinity of these markers, thereby directing the model to place increased emphasis on processing the content surrounding the markers. This, in turn, enhances the model's capability for recognizing numerical. For instance, as demonstrated in Figure 3, ashtag were added both before and after the replaced answer. This step involves referencing the Tokenization Tricks presented in (Jiang et al., 2020).

3.4 Fine-tuning

Fine-tuning refers to the process in deep learning where a pre-trained model is utilized and further trained to adapt to specific tasks or domains. In this study, models based on BERT have undergone fine-tuning to enhance their ability to recognize numerical content in articles. The scrutinized parameters are delineated in Table 2. Further elaboration on these details will be provided in the Experimental Setup section.

3.5 BERT-NAFE

The optimal model in this study is BERT-NAFE. Details will be discussed in this section. Commencing with the data, symbols were eliminated during the data pre-processing, resulting in a refined dataset suitable for model training and evaluation. Expanding upon the foundational BERT model, the removal of stop words was implemented, resulting in a significant 3% increase in accuracy. Additionally, Feature Enhancement was employed, followed by Numerals Augmentation. During Numerals Augmentation, numerals were substituted with their index in the answer list, while being omitted from the article text. Cashtag were inserted before and after the substituted numeral as part of Feature Enhancement. This procedural refinement led to a notable 6% boost in accuracy and was shown to be the most effective in enhancing overall performance.

4 Experimental setup

4.1 Environment

The experimental setup is adaptable to both Google Colab and local environments. For local setup, a minimum of 6GB of GPU memory is necessary for model fine-tuning. The versions of each tool employed in the experiment will be detailed in the accompanying Table 1.

Parameters	Values
python	3.9.18
tensorflow	2.10
cudatoolkit	11.2
cudnn	8.1.0
ktrain	0.40.0

Table 1: Tools Versions Employed in the Experiment.

Parameters	Values
BERT Model	base
Batch size	3
Max length	250
Max learning rate	2e-5
epochs	2

Table 2: Parameters Value of Model Training.

4.2 Hyperparameter

For the fine-tuning parameters, a batch size of 3 and maximum sequence length of 250 were utilized in order to reduce computational expenses during fine-tuning. A max learning rate of 2e-5 was adopted based on hyperparameter tuning performed using the Ktrain learner. This approach was taken to identify the optimal learning rate, as visualized in Figure 4.

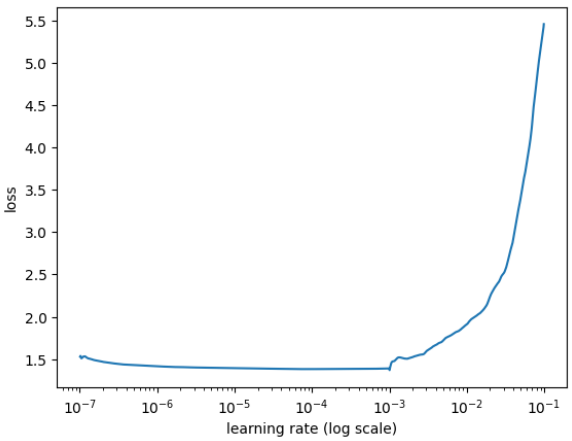

Figure 4: The loss vs. learning rate chart for finding best learning rate in our training processing experiment.

An assessment of validation performance across epochs indicated that achieving satisfactory model performance could be accomplished within just 2 epochs, as depicted in Figure 5. Additional epochs did not meaningfully improve results and instead prolonged training without benefit. Therefore, 2 epochs were deemed adequate for the model to learn from the data effectively.

Regarding the batch size, we conducted a search across values of 3, 4, 8, 16, and 32, all of which exhibited comparable performance. Consequently, we opted to utilize a batch size of 3 for our experiments.

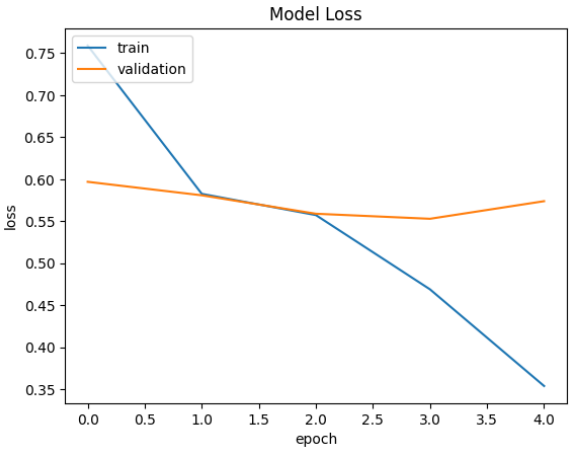

Figure 5: The loss vs. epochs chart during training.

5 Result

Table 3 illustrates the performance of each model developed in this study, comparing them with the initial NEMo model, which was created by the subsequent study (Chen et al., 2021), as indicated by the underlined model in Table 3. The BERT model exhibited an accuracy of 67.95%, a 2% decrease compared to the NEMo model. Upon removal of stop words, the BERT-SWR achieved an accuracy of 70.83%, presenting a marginal improvement of less than 1% compared to the NEMo model. In the case of the BERT-NAFE model, an accuracy of 77.09% was attained, reflecting a 7.14% increase compared to the NEMo model. The Jupyter Notebook employed in this investigation has been made publicly available on GitHub. For further reference, please consult the Appendix.

Model	Accuracy
Initial Model	
NEMo(Chen et al., 2021)	69.95%
Our Model	
BERT	67.95%
BERT-SWR	70.83%
BERT-NAFE	77.09%

Table 3: Comparison of model performance. (All models were fine-tuned except for NEMo.)

Therefore, it can be asserted that the removal of stop words, Numerals Augmentation, and Feature Enhancement are effective in enhancing accuracy, surpassing the performance of the existing model.

6 Discussion and Error Analysis

6.1 Discussion

The baseline model, BERT, achieved an accuracy of 67.95% after fine-tuning, closely approaching the initial model NEMo. Upon the removal of stop words from the text, there was a notable improvement of approximately 3% in accuracy. Considering dimensionality reduction, stop words, characterized as high-frequency terms, pervade most texts. Their exclusion significantly diminishes the dimensionality of the feature space, thereby reducing computational complexity. Consequently, this eases the requirements on both model training and inference processes.

The elimination of stop words contributes to noise reduction. These words often lack essential information, and retaining them may introduce interference, hindering the model's ability to concentrate on learning crucial information. Their removal enables the model to focus its attention and learning capacity on genuinely meaningful vocabulary. Additionally, excluding stop words during model training enhances the comprehension and generalization of the text. Stop words frequently serve as grammatical connectors without conveying specific semantic information. Their removal allows the model to more effectively concentrate on learning the actual relationships between words, unburdened by the intricacies of entire sentence structures.

In addition to BERT-SWR, Numerals Augmentation and Feature Enhancement were incorporated to further elevate the accuracy to 77.09%. Two reasons account for this enhancement. To address the model's inclination to overly rely on particular numerical values encountered during training, Numeral Augmentation aims to counteract this fixation. This is achieved by introducing variations in the representation of numbers, guiding the model to develop a less rigid fixation on specific instances and fostering a more generalized understanding of numerical information.

Another central focus of the augmentation process is the mitigation of numerical ambiguity. Numerals within a given context may have multiple interpretations or ambiguous meanings. To alleviate this ambiguity, the augmentation process exposes the model to diverse representations of numbers. This exposure aids the model in adapting to different contexts and facilitates the disambiguation of numeral meanings based on the surrounding textual

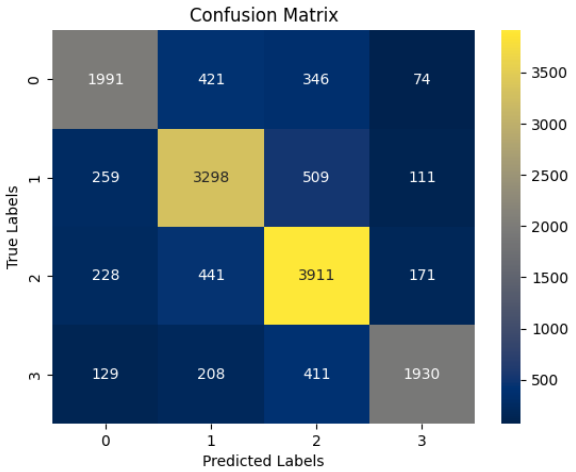

Figure 6: Confusion Matrix of model BERT-NAFE.

information.

After computing the current precision and recall from Figure 6 , it was observed that the system tends to produce class 2 outputs at a rate of 35.9%, surpassing the training set's rate of 33.1%. Conversely, the system's output proportion for class 3 is notably low, standing at only 15.8%. Perhaps adjusting the training set's distribution to a balanced 1:1:1:1 ratio could enhance the system's performance. It is worth noting that the numerical labels 0, 1, 2, and 3 used in the Numerals Augmentation step lack intrinsic significance and merely denote positions. For future work, adjusting the distribution ratio should be a straightforward task.

Various symbols and triple symbols for feature enhancement were also experimented with. In the case of different symbols, comparable performance was observed. However, when employing triple symbols, there was a slight decrease in accuracy, with approximately 74%, representing a 3% decline.

article:
每股虧損**0.88**元 , 至少有**2**位數的成長 ,
稅後虧損**5.06**億元 ,毛利率為**7.6%** ,
燿華：Q___獲利有很大改善，Q3至少二位數成長 ,
0.88 , **2** , **5.06** , **7.6**

question_stem:
燿華：Q___獲利有很大改善，Q3至少二位數成長
answer_options: ['0.88', '2', '5.06', '7.6']
target_num: 2

Figure 7: Example of Bad Sample

6.2 Error Analysis

In certain instances among the samples, the "scao" columns do not encompass the answers to the questions, requiring the model to predict the answer through estimation. For example, as illustrated in Figure 7, the question pertains to improving profit, yet the content in the "scao" column does not include this information.

To address this issue, the "news_article" can be employed as a substitute for the "scao" column. However, this substitution would result in an increase in token length, leading to a rise in training costs, encompassing GPU memory usage, and extending model training time.

7 Conclusions

This study has achieved significant performance improvements through fine-tuning the BERT model and optimizing text processing methods. Firstly, during the fine-tuning process, we observed a notable increase in model accuracy by removing stop words from the text, particularly high-frequency vocabulary. The exclusion of stop words not only reduced the dimensionality of the feature space, lowering computational complexity, but also contributed to noise reduction, enabling the model to better focus on meaningful vocabulary.

Secondly, the introduction of Numerals Augmentation and Feature Enhancement further elevated the model's accuracy. Numerals Augmentation, by introducing variations in the representation of numbers, assisted the model in overcoming overreliance on specific numerical values, fostering a more generalized understanding of numeric information. Additionally, Feature Enhancement strategies helped alleviate numerical ambiguity, allowing the model to adapt to different contexts and accurately interpret numeric meanings.

In summary, our research findings suggest that, building upon the BERT model, fine-tuning and text processing optimizations can effectively enhance performance in numeral reading comprehension.

Acknowledgements

This study was partially supported by the National Science and Technology Council under the grant number NSTC 112-2221-E-324-014.

References

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

goto456. 2019. scu-stopwords.

Yusra Ibrahim, Mirek Riedewald, Gerhard Weikum, and Demetrios Zeinalipour-Yazti. 2019. Bridging quantities in tables and text. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1010–1021.

Mike Tian-Jian Jiang, Yi-Kun Chen, and Shih-Hung Wu. 2020. Cyut at the ntcir-15 finnum-2 task: Tokenization and fine-tuning techniques for numeral attachment in financial tweets. In *The 15th NTCIR Conference Evaluation of Information Access Technologies*, pages 92–96. NII Testbeds and Community for Information access Research.

Matthew Lamm, Arun Chaganty, Christopher D. Manning, Dan Jurafsky, and Percy Liang. 2018. Textual analogy parsing: What's shared and what's compared among analogous facts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 82–92, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Andy Sun. 2020. jieba.

Appendix

https://github.com/anson70242/BERT-NAFE