CSE 582: Natural Language Processing Final Project Report

Danling Jiang
dzj5189@psu.edu

Kevin Zhu
qbz5050@psu.edu

May 5, 2023

Abstract

The goal of this project is to design a novel text generation task, which involves collecting data and selecting a model to solve the task. We designed a task of generating descriptions for university-level courses base on their titles. We scraped the course information from University Bulletin of Penn State, then fine-tuned a pre-trained transformer model T5. We conducted different experiments on the fine-tuning process. The results showed that with carefully designed fine-tuning process, the "pre-train, fine-tune" paradigm can achieve good results on our newly designed text-to-text generation tasks. Our codes are available at this GitHub link

1 Introduction

Natural Language Processing (NLP) is a field of study in artificial intelligence and computational linguistics that deals with the interaction between computers and human languages. It addresses how to turn language data into computable data to serve human purposes. There are two human purposes: text classification and text generation. While both of them are important in the NLP community, text generation will be the particular focus in this project. Text generation is the process of automatically producing natural language text that is coherent and consistent with a given input. There are many applications in the field of text generation, including but not limited to machine translation, summarization, conversational chatbot, and image captioning. Text generation can also be divided into categories based on the input types, such as image-to-text, text-to-text, and audio-to-text etc. Most of the problems can be solved using the Markov processes, deep generative models like Long-Short-Term-Memory networks (LSTM), and transformers. In this project, we will examine text-to-text generation task using the transformer architecture.

2 Motivation

In this section, we will discuss the motivation behind the proposed task and selected transformer model.

2.1 Task

We came up with a novel text-to-text task of generating course descriptions for university-level courses base on their titles. While most of the researchers focused on summarization or translation tasks, there is a lack of resources on this type of "text expansion" task. Our task is also different from the language modeling that focuses on predicting the next word in a sentence. In this task, we try to find a mapping between a shorter text "title" and a longer text "description".

Course selection choices are important during students' years of study at a university. They can possibly shape the future of their career or research. Most universities provide a publicly available catalog for their offering of the courses. Each entry normally contains a course name abbreviation, a course number, a short title, and a course description. Courses are usually either prescribed or elective. For elective courses, the descriptions are important as they can provide the students with some insights and expectations. Most of the time, the schools or professors will write the descriptions. This often limits the breadth and depth of the descriptions. Our idea of incorporating artificial intelligence (AI) into this process enables the descriptions to cover the most crucial, recent and advanced topics in each

field of study. The AI model can take many data from academia into consideration when generating the course descriptions. In the era of deep learning, the more data an AI model has, the more accurate their outputs can be. Then, the school can design the courses according to the generated descriptions to attract prospective students and to benefit current students.

2.2 Model

As mentioned earlier in the introduction section, most of the text generation tasks can be solved by either a probabilistic model like Markov process, neural network based models like LSTM, and transformers. We chose to use a transformer-based model called T5. In the original paper where a group of researchers proposed the transformer architecture, the results showed that the transformers outperformed the previous state-of-the-art models such as convolutional sequence to sequence learning (ConvS2S) and LSTM in the task of machine translation [9]. Since machine translation and our task are both text-to-text generation, we will use T5 in this project. Its text-to-text framework is illustrated in the Figure 1.

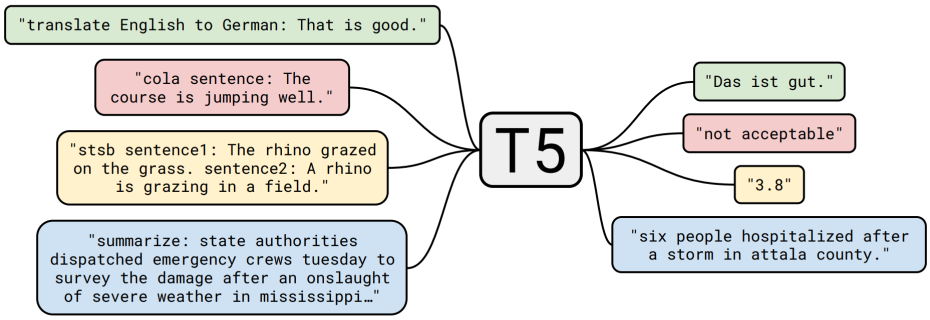

Figure 1: A diagram of T5's text-to-text framework.

It follows the standard encoder-decoder transformer architecture showed in Figure 2.

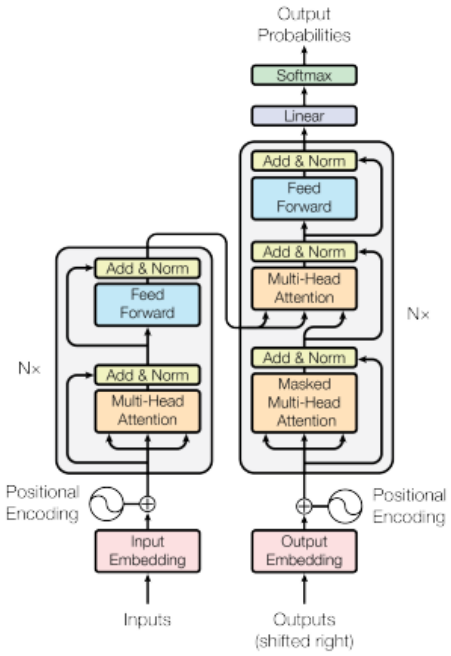

Figure 2: A standard transformer architecture proposed in the original paper.

The design of their structure is very similar to the BERT, with 12 self-attention heads in each of the 12 encoder and decoder layers [1]. Because of the limited computing resources we had, we chose to use a smaller version of T5 available on huggingface, which has 8 self-attention heads in each of the 6 decoder and encoder layers. The T5 model was pre-trained on the Colossal Clean Crawled Corpus (C4) data set, achieved state-of-the-art results on many NLP benchmarks while being flexible enough to be fine-tuned to a variety of important downstream tasks [7]. Since we are proposing a novel text generation task, we want the model to be as flexible as it can to perform well on our task. Based on their paper, the most of the tasks are working on producing a shorter or same-length text given an input text. This also showed the novelty of our "shorter-to-longer" text generation task.

3 Data set

In this section, we will discuss how we collected the data set and analyzed the data. Then we will discuss some pre-processing steps applied to data.

3.1 Web scraping

Our data set is obtained by scraping data from the University Bulletin website of Penn State [5]. A snapshot of the website is shown in Figure 3.

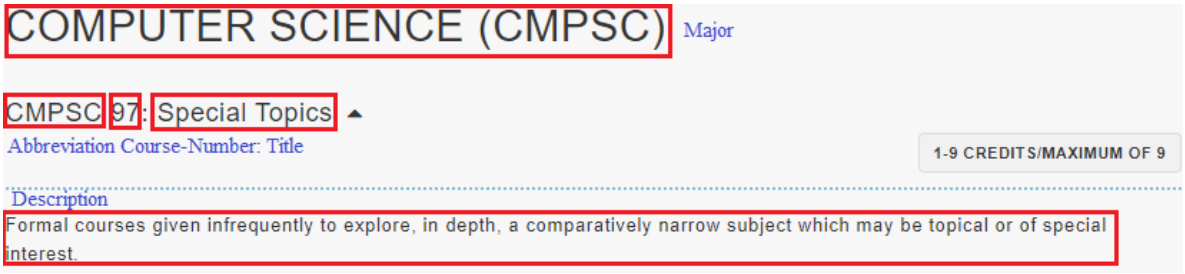

Figure 3: A snapshot of the University Bulletin website from Penn State with annotations.

The course name contains the major abbreviation, course number, and title. The course description describes the content of the course. Major abbreviation gives information about the subject in each major. Certain subjects may have their specific keywords used in the description. Course number might also provides some information on the depth and difficulty of the course. The title is the essential part that the model will be elaborating on. We grouped all of these three element from course name and named it as title, which will be the input. We collected the descriptions as the ground truth for the model, with links removed for clarity. Each pair of title and description will be one instance in our data set. Table 1 shows one example in the data set.

Title	**Description**
A-I 574: Natural Language Processing	Natural Language Processing (NLP) is a sub-field of Artificial Intelligence. This course covers basic as well as advanced concepts to gain a detailed understanding of NLP tasks such as language modeling, text to speech generation, natural language understanding, and natural language generation. Students will learn the necessary skills to design a range of applications, including sentiment analysis, translating between languages, and answering questions. Throughout the course, the practical implementation of these applications with deep neural networks is also discussed.

Table 1: An instance of the data set.

3.2 Data analysis

The data set was separated into 2 parts: undergraduate courses only (small data set) and both undergraduate and graduate courses (large data set). The reason was to test if the scale of the data set affect the accuracy of fine-tuning. The small data set contains 9468 examples, and the large data set contains 13824 examples. About 46% increase in the size of data set should give reliable results showing if the scale can affect the accuracy.

Figure 4 shows the distribution of the lengths of the descriptions. They follow a similar distribution for both the small and large data sets. The 90th percentile values are smaller than 2000, and the averages are smaller than 1000. This information was useful for deciding how much to truncate or pad the ground truth descriptions during fine-tuning. It also puts a limit on the length of generated results. Figure 5 shows the distribution of the lengths of the titles. Titles in both data sets are shorter than 128, which is useful for truncating and padding the input titles.

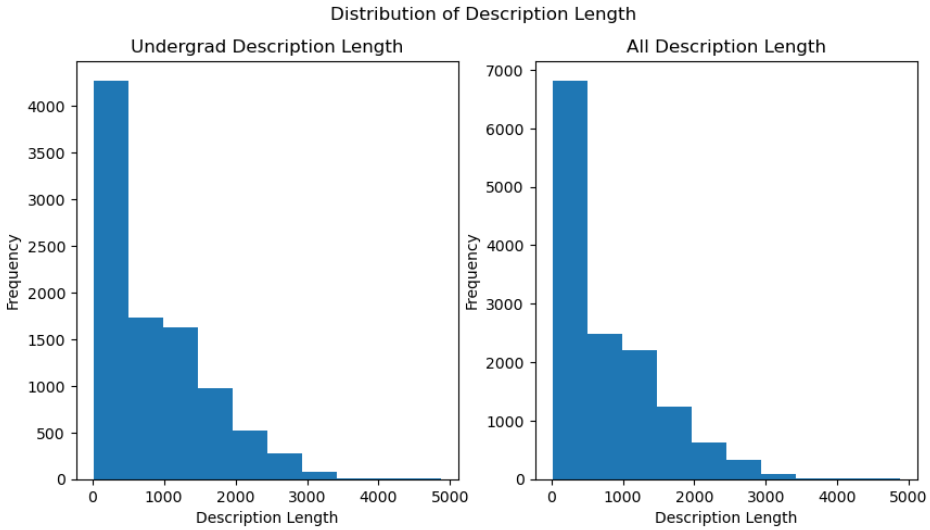

Figure 4: Distribution of length of the descriptions.

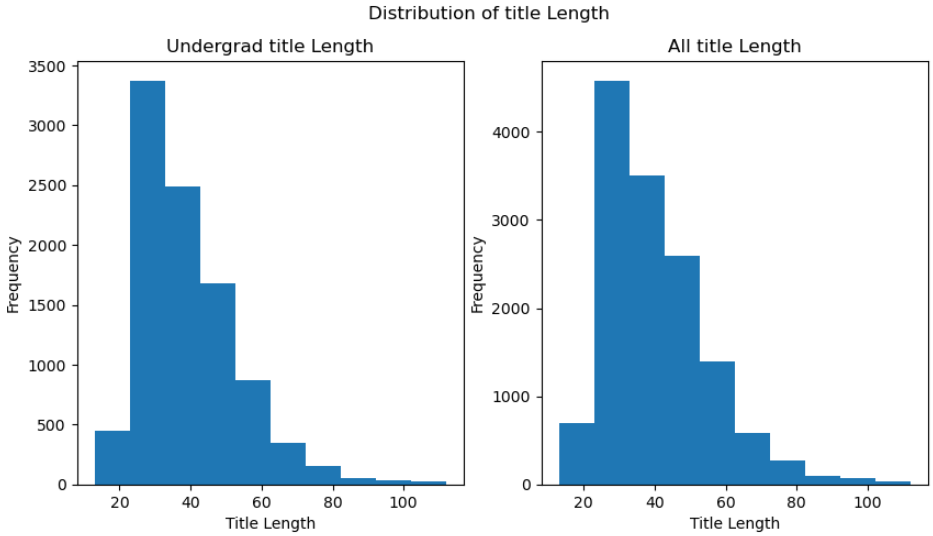

Figure 5: Distribution of length of the titles.

3.3 Pre-processing

We did some necessary pre-processing before fine-tuning the model. It involved the following steps:

1. Removing instances with missing information. There were a few instances in the data set that do not have a description or have an uninformative description "No description.". Those instances were removed from the data set.
2. Splitting the data set is into training and testing sets. 80% of the courses from every major were used as training data, while the rest of the courses were used as testing data.
3. Prepending a task-specific prefix "Generate a description of the following university course: " to every input. As detailed in the T5's paper, a prefix should be added for each task to ensure a consistent training objective during fine-tuning. Different prefixes can lead to very different results, but it is not the objective of this project to find the best prefix.
4. Truncating the longer input and pad the shorter input based on a hyper-parameter.
5. Tokenizing the input using T5's pre-trained tokenizer, which is based on the SentencePiece [3].

4 Algorithm

We followed the "pre-train, fine-tune" paradigm. The idea was introduced before the invention of the transformer model [2], but was made popular with emerging popularity of the transformer models [6]. Due to limited sources of data, fine-tuning a pre-trained transformer model can achieve relatively good performance comparing to training a model from the scratch. There were many hyper-parameters that are important for fine-tuning the model. Due to limited time, we only tuned a small subset of the hyper-parameters, while we kept rest of the hyper-parameters as default values. Some of the most important hyper-parameters will be tested in different experiments.

5 Experiments

In this section, we will explain the experiment settings of this project and results we got. The results are evaluated using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE), a metric that count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated outputs and the ground truth output [4]. It is a widely used evaluation metric for text-to-text generation tasks. We included some common variants of ROUGE, including ROUGE-1, ROUGE-2, and ROUGE-L that measures different overlaps between the predicted outputs and the ground truth. For hyper-parameters not mentioned in each experiment, they were kept same as the huggingface's default values. All experiments can be viewed at the project page on WandB.

5.1 Experiment 1 – Algorithm

In this experiment, we compared the performance between pre-trained only model and fine-tuned model to check if our fine-tuning algorithm is working correctly. Table 2 shows the results of this experiment. There are significant evidences in the results that shows our algorithm is suitable for our task.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Pre-trained only	15.5493	3.7095	11.957
Fine-tuned model	24.7184	14.3824	21.5225

Table 2: Results for experiment 1.

5.2 Experiment 2 – Scale

As we discussed in the Data set section, we prepared two different-sized data sets. There is a 46% increase in size between 2 data sets. We fine-tuned the T5 on both data sets with only 1 epoch. Max target length is set to 2048, which is larger than 90th percentile value for the target length distribution. For this experiment, time and accuracy were examined. Table 3 shows the results of this experiment. With only 1 epoch, the large data set can perform better than small data set on all metrics. Despite longer training and inference time, using more data might be preferable to achieve good accuracy when fine-tuning a transformer model. We used the large data set in all subsequent experiments.

Scale	**Training time**	**Inference time**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
9468 examples	335s	4985s	23.7331	14.2283	21.2698
13824 examples	487s	7632s	24.7184	14.3824	21.5225

Table 3: Results for experiment 2.

5.3 Experiment 3 – Max target length

Max target length is an important hyper-parameter that controls the length of the ground truth outputs during training time and generated outputs during inference time. After analyzing the data set's length distribution, we experimented 3 different sizes, 2048, 1024, and 512. As mentioned in section 3.2, 2048 is larger than 90th percentile value for the distribution, 1024 is about the average length in the distribution, and 512 is used to check if further reducing target length can affect the accuracy. Number of fine-tuning epochs was set to 1 for faster training time. In addition to time and accuracy, we also examined the average length of generated outputs. Table 4 shows the results of this experiment. With halved target length, training time stayed almost the same. However, the inference time and average length of predicted sequences roughly got halved. There is a slight increase in the ROUGE score except for ROUGE-2. One possible explanation is that the transformer model was not fine-tuned enough with small number of epochs, so it generated bad outputs. The longer the generated outputs are, the less overlap between n-grams there is. In conclusion, small number of epochs worked well with smaller max target length. When we fine-tuned the model with more epochs in the next experiment, there was no significant change in accuracy with different max target lengths.

Max target length	**Training time**	**Inference time**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**Average length**
2048	487s	7632s	24.7184	14.3824	21.5225	516
1024	497s	4418s	25.6698	15.0044	22.4257	318
512	478s	2442s	26.3792	14.8615	22.9981	197

Table 4: Results for experiment 3.

5.4 Experiment 4 – Number of training epochs

In this experiment, we will test how different numbers of training epochs affect the model's accuracy. The max target length was set to 1024 for shorter inference time. Table 5 shows the results of this experiment. With increasing number of training epochs, all 3 metrics increased as expected. One interesting pattern is that the average length of generated outputs decreased, which leads to a significance decrease of the inference time. Such pattern suggested that the max target length hyper-parameter might not be too important when the model was well fine-tuned. Figure 6 shows there is a decreasing trend of loss throughout the 50 training epochs. The relatively stable decreasing loss curve indicate that the model can possibly be fine-tuned with more training epochs to further increase the accuracy.

Number of epochs	Training time	Inference time	ROUGE-1	ROUGE-2	ROUGE-L	Average length
1	497s	4418s	25.6698	15.0044	22.4257	318
10	4767s	1065s	38.3036	24.8185	34.4278	102
50	24074s	1369s	40.2179	26.2034	35.5174	104

Table 5: Results for experiment 4.

Figure 6: Loss vs. training steps.

5.5 Experiment 5 – No repeated n-gram size

Besides the training hyper-parameters, some hyper-parameters at inference time can also affect the accuracy of the output. Specifically, we tested how limiting the occurrences of unique n-grams affect the outputs using our best model fine-tuned in experiment 4. Table 6 shows that with different limits on the occurrences of n-grams, the model can generate very different outputs. The first row of the table is the trail that put no limits on n-grams. It kept generating the same sentence "The course will provide an overview of the theory and practice of business tax planning." When we apply different limits, the generated outputs are more elaborated rather than repeating the same sentence. In addition to being more coherent, ROUGE-1 score increased by 2 and ROUGE-2 score increased by 1 for putting limits on 3-gram. We did not increase the n-gram size more than 3 because the average lengths of generated outputs became smaller and smaller, which could possibly contradicts with our text generation task being "shorter-to-longer".

No repeated n-gram size	Generated output
0	This course provides an overview of the theory and practice of business tax planning. The course will provide an overview of the theory and practice of business tax planning. The course will provide an overview of the theory and practice of business tax planning. The course will also provide an overview of the theory and practice of business tax planning. The course will also provide an overview of the theory and practice of business tax planning. The course will also provide an overview of the theory and practice of business tax planning.
2	This course provides an overview of the theory and practice of business tax planning. The course will provide an introduction to the principles of taxation and the practice that govern the tax process. Students will learn the basic principles and practices of accounting and tax analysis, including the use of a tax calculator, tax code, and e-taxation.
3	This course provides an overview of the theory and practice of business tax planning. The course will provide an overview and introduction to the theory, principles, and practice that are used to prepare students for the taxation of business enterprises. The focus will be on the tax process, taxation, and taxation and the tax system that is used to manage the tax burden. The class will also examine the tax and tax system in the United States. The students will learn how to apply the tax method to the tax plan, and how to use the tax model to calculate the tax rate.

Table 6: Sample output for different n-grams occurrence limits on the course title: "ACCT 510: Business Tax Planning Theory and Practice". The ground truth output is: "Tax theory pertaining to corporations, partnerships and conduit entities, estates, trusts, ethics, and professional tax responsibilities. Business Tax Planning Theory and Practice (3) This course provides in-depth coverage of the theory and practice of tax planning for corporations, partnerships and other related pass-through entities. Topics will include tax research, corporate formation and capital structure, corporate nonliquidating distributions, corporate acquisitions and reorganizations, consolidated tax returns, partnership formation and operation, special partnership issues, S corporations, taxation of gifts, estates and trusts, and professional responsibilities and ethics."

5.6 Experiment 6 – Learning rate and optimizer

Learning rate (lr) is important in machine learning algorithms. It controls the pace of learning for an algorithm. In this experiment, we tested on the learning rate and its scheduling policy used in the T5's paper [7]. Unlike huggingface's default learning rate of 5×10^{-5} that linearly decreases after each epoch, they used a constant learning rate of 1×10^{-3} for fine-tuning. Moreover, they chose the AdaFactor as their optimizer instead of Adam that were used in our previous experiments [8]. Table 7 shows the comparison of the accuracy when the models were fine-tuned with 10 epochs. There were no significant differences in ROUGE scores.

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
Linear lr 5×10^{-5} with Adam	38.3036	24.8185	34.4278
Constant lr 1×10^{-3} with AdaFactor	38.8152	25.7961	34.3423

Table 7: Results for experiment 6.

6 Result

We selected our best model based on the averages of ROUGE metrics from the experiments. The model that was fine-tuned with 50 epochs, default learning rate and scheduling, and no repeated n-gram size 3 performed the best, achieving 42.666 for ROUGE-1, 27.23 for ROUGE-2, and 35.718 for ROUGE-L. The model made predictions on some course titles that were identical to the ground truth as shown in Table 8. Some of the courses in the data set are from different majors with different course numbers, but they may have the same course names (i.e. Internship, Foreign Studies, and Special Topics). Since there is no variation in those descriptions in the ground truth, the generated outputs are perfect.

Title	**Generated description**
AFR 395: Internship	Supervised off-campus, nongroup instruction including field experiences, practica, or internships. Written and oral critique of activity required.
ANSC 499: Foreign Studies	Courses offered in foreign countries by individual or group instruction.
ACCTG 498: Special Topics	Formal courses given infrequently to explore, in depth, a comparatively narrow subject which may be topical or of special interest.

Table 8: Some examples of perfectly generated descriptions.

However, most of the courses' descriptions varied in length and topics. We picked one example where the model failed to generate a good description shown in Table 9. The result is not coherent and its ROUGE scores are very low. In the ground truth description, it starts by sharing the background and benefits of R programming language. It also mentioned many discipline-specific terms and technologies such as HTML scraping and tidyverse. But the generated description simply copied down the course name "Introduction to R" with added contents that are related to human behavior rather than statistics. Without enough knowledge in the statistical community, the model did not generate a good result.

Generated description	**Ground truth**
Introduction to R, a mathematical and mathematical approach to the study of human behavior.	R is a powerful, open-source programming language used widely for applications in statistics and data science. It is easily extendible, and thousands of user-created packages are publicly available to extend its capabilities. This course will introduce students to data computing fundamentals and a reproducible workflow using the R programming language and related tools. Students will be expected to access, join, wrangle, clean, and visualize real data from various sources (e.g. CSV, HTML scraping, web URL, R packages). The course will emphasize use of "tidyverse" R packages (e.g. dplyr, ggplot2), although students will also be exposed to Base R and other packages. In addition, students will be exposed to one or more integrated development environments (e.g. RStudio) and will be expected to write well-documented code using a reproducible workflow (e.g. RMarkdown, Git/GitHub). The course focuses on descriptive and graphical summary techniques rather than inferential statistical techniques

Table 9: An example of very low ROUGE scores. The course title is "STAT 184: Introduction to R"

7 Conclusion

After completing this project, we found out that designing a new text generation task is not easy. It requires good sources of data and novelty. We also found out that "pre-train and fine-tune" is a good strategy to handle text generation tasks. Even with little or no hyper-parameter tuning, it can produce acceptable results. It was the first time that we fine-tuned a transformer, and it is not complicated to do so. So, this strategy is user-friendly and accessible to those with little fine-tuning experience. Our project also has its limitations as the following, and it requires some future works to achieve better performance:

- Different pre-processing steps such as the prefix prepended to each input sequence might result in different accuracy. Find the best prefix for our task can be challenging since there is no deterministic algorithm currently.
- Our experiments did not cover all the hyper-parameters. It is likely that more hyper-parameter tuning can lead to better accuracy.
- We also found that more than 50 epochs will possibly improve the accuracy, but we did not have any time to train it further.
- We could not use the larger versions of T5, which will theoretically improve the results.
- Our data set can also be at larger scales. With more course information from different universities and more concepts from different disciplines in academia, the model can produce more informative descriptions.
- We need to include human evaluations in the future. ROUGE scores can be unreliable sometimes. For example, if the ground truth is "The quick brown fox jumps over the lazy dog.", the following two sentences: "The quick brown fox jumps over the dog that is lazy." and "The quick fox jumps over the dog that is brown and lazy." share the same ROUGE-1 and ROUGE-2 score, but one of them conveys totally different meanings comparing to the ground truth.

References

[1] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine 29*, 6 (2012), 82–97.

[3] Kudo, T., and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[4] Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (Barcelona, Spain, July 2004), Association for Computational Linguistics, pp. 74–81.

[5] PSU. University bulletins. https://bulletins.psu.edu/, 2023. Accessed: 2023-04-19.

[6] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training.

[7] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[8] Shazeer, N., and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost, 2018.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017.