Scientific question answering using Machine Learning techniques

Homework Documentation

Marcell Emmer
DL55DP

Máté Gedeon
HIIMNO

Csaba Kiss
B73BSA

***Abstract*—In recent years, Natural Language Processing (NLP) became a trending research field in artificial intelligence, with multiple use cases. This paper delves into employing machine learning techniques for answering scientific exam questions, with five possible answers given from the LLM Science Exam dataset available on Kaggle. Initial experiments employing BERT and SciBERT models gave promising results, but revealed limitations due to the lack of pertinent "common and scientific" knowledge. To address this, a method was devised to create an "open book" dataset by using context from Wikipedia articles to enhance the model's understanding of the questions. This contextual information was embedded into a high-dimensional vector space for similarity searches, aiding in retrieving relevant articles. Hyperparameter optimization was also performed to ensure the quality of predictions. On our test set, we reached 0.8 in MAP@3 value. With the methodology being laid down, the possibility is given to further improve the models, and continue the research.**

The research provides insight into how to develop a general solution that works beyond the predefined training and test sets. Our method is transparent and easy to understand, making it comprehensible and usable without deeper domain knowledge within the field of Natural Language Processing.

***Index Terms*—artificial intelligence, machine learning, neural networks, science**

I. Introduction

Understanding language is a fundamental human skill that we often take for granted. It may seem like a simple task, but there is a complex process that takes place within our brains to make it possible. When we hear a word, our brains do not just store it as a sequence of letters or sounds. Instead, it encodes the word's meaning based on its relationships with other words and concepts in our mental lexicon. This process involves a web of connections between different parts of the brain. As human societies evolved and developed written language, new challenges emerged in processing and analyzing large volumes of text. This need led to the creation of a new field called Natural Language Processing (NLP), which focuses on developing algorithms and techniques to enable computers to understand and process human language. Research areas in NLP include question answering, named entity recognition, semantic search, and generative models.

In this paper, we aim to explore how machine learning could be utilized to answer scientific exam questions. In the subsequent sections, we will delve into the employed methodology, the utilized dataset, the obtained results, and outline plans for future work.

II. Related Work

Nowadays, state-of-the-art models make use of the Transformer architecture, first introduced in the paper [VSP+23]. Transformers follow an encoder-decoder structure and are capable of capturing long-distance dependencies, a challenge traditional recurrent and convolutional architectures struggled with.

One of the key challenges is representing words, sentences, or even paragraphs in a manner that enables computers to comprehend their meaning. To achieve this, a technique known as embedding is employed, which involves representing text using vectors in a high-dimensional space, where each dimension signifies a different feature or attribute of the text. The objective is to create text representations that capture their relationships, allowing similar texts to be represented by similar vectors.

One of the most prominent models for embeddings is BERT (Bidirectional Encoder Representations from Transformers), which was introduced in [DCLT18]. Since its introduction, multiple models have emerged based on BERT, tailored to handle more specific tasks. Among these models is SciBERT [BLC19], which was specifically pretrained on a training corpus compiled from papers sourced from Semantic Scholar, encompassing a corpus size of 1.14 million papers and 3.1 billion tokens.

III. Dataset and evaluation metric

The dataset of our interest was provided in the context of a Kaggle competition called LLM Science Exam [WL23]. It is inspired by the OpenBookQA dataset [MCKS18] and challenges participants to answer difficult science-based questions written by a Large Language Model. It contains 200 pairs of questions, 5 possible answers (ABCDE), and a label of the correct answer. One example can be seen in Table I.

The evaluation metric of the competition was called the Mean Average Precision @ 3 (MAP@3) [WL23].

$$MAP@3 = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{\min(n,3)} P(k) \times \text{rel}(k) \tag{1}$$

TABLE I
DATASET EXAMPLE

id	prompt	A	B	C	D	E	answer
62	What is a Schwarzschild black hole?	"A black hole that has mass but neither electric charge nor angular momentum and is not spherically symmetric according to Birkhoff's theorem."	"A black hole that has mass electric charge and angular momentum and is spherically symmetric according to Birkhoff's theorem."	"A black hole that has mass but neither electric charge nor angular momentum and is spherically symmetric according to Birkhoff's theorem."	"A black hole that has neither mass nor electric charge nor angular momentum and is not spherically symmetric according to Birkhoff's theorem."	"A black hole that has mass electric charge and angular momentum and is not spherically symmetric according to Birkhoff's theorem."	C

where U is the number of questions in the test set, $P(k)$ is the precision at cutoff k, n is the number of predictions per question, and rel(k) is an indicator function equaling 1 if the item at rank k is a relevant (correct) label, and zero otherwise.

IV. Making an open-book dataset

This part of our work is located in the context-creation.ipynb notebook that ran on Kaggle with 2xT4 GPUs.

The problem with using LLMs on this Q&A dataset is their lack of "common and scientific" knowledge. We have tried feeding the dataset into a pre-trained BERT ("bert-base-cased") and the pre-trained SciBERT ("allenai/scibert_scivocab_uncased") models. The base BERT model performed no better than random guessing with an accuracy of 20%, while the SciBERT model had an accuracy of 42.9% and $MAP@3 = 0.4233$ which is significantly better but it still left room for improvement.

Several Kaggle solutions explored the idea of providing context for a given question of the dataset so the large language model used later can learn from that before choosing the proper answer, making our dataset into an "open book science test". Our choice of contextual data was the Wikipedia Plaintext (2023-07-01) dataset on Kaggle [D$^+$23] that contains Wikipedia articles of all nature up to July 2023. We used the "sentence-transformers/all-MiniLM-L6-v2" [WWD$^+$20] model to embed both the questions and the first sentences of the articles into a 384-dimensional dense vector space where we can perform sentence similarity searches. The Similarity search was done with the IndexFlatIP index of the Faiss library [JDJ19]. We iteratively embedded the context column of the Hugging face dataset, created from the 1.76 GB "wiki_2023_index.parquet" file with a batch size of $500'000$ and added the embedding to the Faiss index in order not to run out of GPU memory. After that, we also embed the questions located in the "prompt" column of the wiki_dataset Pandas data frame and perform the similarity search with the retrieve_most_similar function. The similarity search boils down to finding the K nearest neighbors in the previously mentioned high dimensional vector space with the $K = 1$ closest neighbor being the closest article to our question. For the example provided in Table I the found context is the following: "The Schwarzschild radius or the gravitational radius is a physical parameter in the Schwarzschild solution to Einstein's field equations that corresponds to the radius defining the event horizon of a Schwarzschild black hol". The sentence cuts off after a certain value, which arguably may not provide enough information to answer the question in this case.

V. Training a model on the open-book Q&A dataset

This part of our work is located in the training-an-open-book-model.ipynb notebook that ran on Kaggle with 2xT4 GPUs. The notebook borrows ideas from a popular solution of the Kaggle - LLM Science exam problem called How To Train Open Book Model - Part 1 [Deo23] and uses the 'microsoft/deberta-v3-large' [HLGC21], [HGC21] model from Hugging Face. The model takes the openbook-qna-data.csv file created with the context-creation notebook as its input after tokenization. The data collator used in the tokenization process is not a work of our own, we used the implementation of Radek [Osm23] that became quite popular in this competition. We load the CVS file as a pandas data frame and split it into a train, validation, and test set with sizes of 100, 50, and 50 entries. The initialization of the model includes freezing the first 18 out of its 24 layers in order to speed up transfer learning. This affects the accuracy negatively. Then we use Weights & Biases Sweeps in order to perform hyperparameter optimization. The values considered during the sweeps are listed in Table II.

First we performed Bayesian hyperparameter tuning in 25 sweeps. Since the code saved too much data outside of the kaggle/working directory, we reran the code with 3 sweeps only so we had done 28 sweeps in total. The best model was generated in the sweep named 'glad-sweep-15' with a Test MAP@3 value of $MAP@3 = 0.800$. We have extracted the most important metrics of the 5 models with the highest Test MAP@3 values from our WAndB project. For more details, please visit our Weights & Biases project. The following figures illustrate the results obtained during the sweeps. Table III shows the configuration of the training arguments used during 'glad-sweep-15' and Table IV summarises the sweep results.

TABLE II
SWEEP CONFIGURATION

Method	bayes
Metric	Name: map@3 Goal: maximize
Learning Rate	Min: 1×10^{-6} Max: 1×10^{-4}
Epochs	Values: [20, 30, 50]
Weight Decay	Values: [0.0, 0.01, 0.02, 0.03, 0.04, 0.05]
Warmup Ratio	Values: [0.0, 0.05, 0.1, 0.15, 0.2]
Gradient Accumulation Steps	Values: [2, 4, 8, 16]
Early Stopping Patience	Values: [5, 10]

TABLE III
TRAINING ARGUMENTS DURING GLAD-SWEEP-15

num_train_epochs	50
learning_rate	3.868708800630949e-05
weight_decay	0.03
warmup_ratio	0
gradient_accumulation_steps	4
per_device_train_batch_size	1
per_device_eval_batch_size	2
overwrite_output_dir	True
fp16	True
logging_steps	25
evaluation_strategy	'steps'
eval_steps	25
save_strategy	'steps'
save_steps	25
load_best_model_at_end	True
metric_for_best_model	'map@3'
lr_scheduler_type	'cosine'
save_total_limit	2

TABLE IV
WEIGHTS & BIASES SUMMARY OF GLAD-SWEEP-15

train/loss	0.0426
train/learning_rate	0.00002901531600473212
train/epoch	16
train/global_step	200
_timestamp	1702060439.6846826
_runtime	695.2914175987244
_step	17
eval/loss	1.997406005859375
eval/map@3	0.7566666666666667
eval/accuracy	0.6
eval/runtime	4.6513
eval/samples_per_second	10.75
eval/steps_per_second	2.795
train/train_runtime	594.9302
train/train_samples_per_second	8.404
train/train_steps_per_second	1.009
train/total_flos	977821359296220
train/train_loss	0.624648962020874
Test MAP@3	0.8000000000000002
Test Accuracy	0.64
_wandb/runtime	694

VI. CONCLUSION AND FUTURE WORK

With the achieved 0.8 MAP@3 value, the project can be said successful. With limited computing and time resources, we still managed to accomplish the task.

Future work plans include exploring data augmentation techniques using large language models and creating a web application, which using our model can predict the answer to scientific questions, that take similar form as our dataset (one question, with five possible answers).

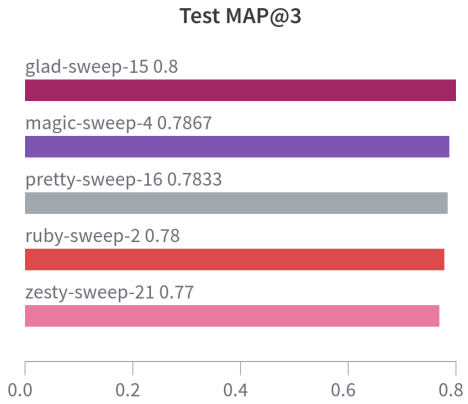

Fig. 1. Test MAP@3 values of the 5 best performing sweeps. [wan]

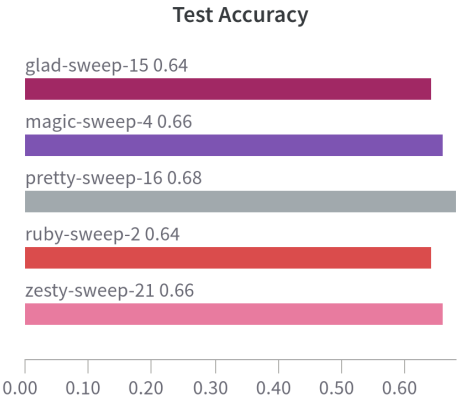

Fig. 2. Test Accuracy of the 5 best performing sweeps. [wan]

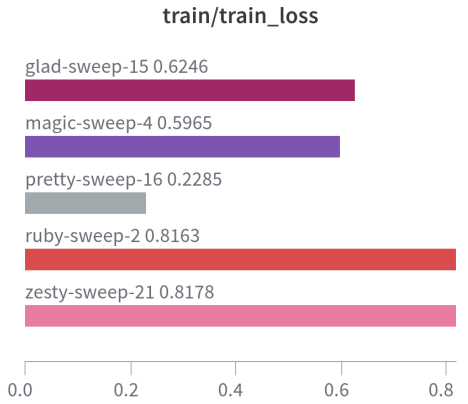

Fig. 3. Final training loss of the 5 best performing sweeps. [wan]

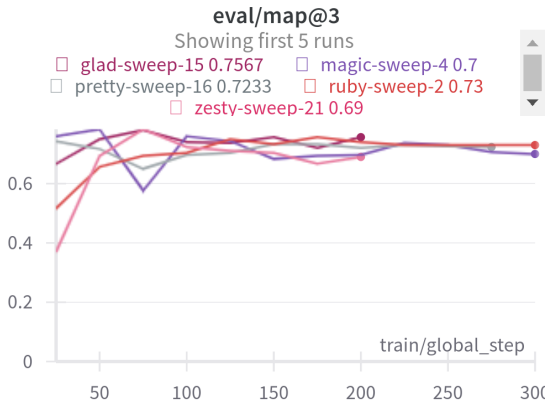

Fig. 4. MAP@3 calculated on the valuation set as a function of evaluation steps of the 5 best performing sweeps. [wan]

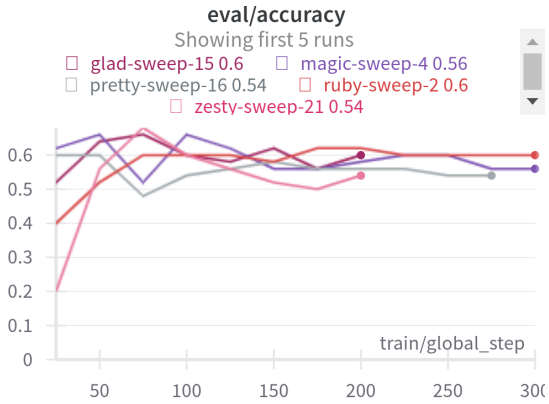

Fig. 5. Accuracy calculated on the valuation set as a function of evaluation steps of the 5 best performing sweeps. [wan]

References

[BLC19] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *EMNLP*. Association for Computational Linguistics, 2019.

[D+23] Jinho D. et al. Wikipedia dataset - july 2023. https://www.kaggle.com/datasets/jjinho/wikipedia-20230701, 2023. Accessed: 2023-12-02.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[Deo23] Chris Deotte. How to train open-book model - part 1. https://www.kaggle.com/code/cdeotte/how-to-train-open-book-model-part-1, 2023. Accessed: 2023-12-03.

[HGC21] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

[HLGC21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

[JDJ19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[MCKS18] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

[Osm23] Radek Osmulski. New dataset + deberta v3 large training! https://www.kaggle.com/code/radek1/new-dataset-deberta-v3-large-training, 2023.

[VSP+23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[wan] Weights and biases.

[WL23] Addison Howard Will Lifferth, Walter Reade. Kaggle - llm science exam, 2023.

[WWD+20] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.