TextGraphs-16 Natural Language Premise Selection Task: Zero-Shot Premise Selection with Prompting Generative Language Models

Liubov Kovriguina[1], **Roman Teucher**[1],
Robert Wardenga[2]
[1]Fraunhofer IAIS Dresden, [2]InfAI Leipzig
{liubov.kovriguina,roman.teucher}@iais.fraunhofer.de
{wardenga}@infai.org

Abstract

Automated theorem proving can benefit a lot from methods employed in natural language processing, knowledge graphs and information retrieval: this non-trivial task combines formal languages understanding, reasoning, similarity search. We tackle this task by enhancing semantic similarity ranking with prompt engineering, which has become a new paradigm in natural language understanding. None of our approaches requires additional training. Despite encouraging results reported by prompt engineering approaches for a range of NLP tasks, for the premise selection task vanilla reranking by prompting GPT-3 doesn't outperform semantic similarity ranking with SBERT, but merging of the both rankings shows better results.

1 Introduction

The recently proposed task of Natural Language Premise Selection for mathematical statements (Ferreira and Freitas, 2020a) follows in line with tasks such as Mathematical Information Retrieval (Líška et al., 2011) and Mathematical Formula Understanding (e.g. (Peng et al., 2021)). Those tasks share the common objective to improve the processing and understanding of mathematical statements, which are a significant part of scientific information. On the other hand, with the advent of the attention mechanism (Vaswani et al., 2017) pretrained and fine-tuned Transformers, such as BERT (see (Devlin et al., 2018)), GPT-3 ((Brown et al., 2020b)) etc. were able to improve state of the art results for many Natural Language Task. In this short paper we investigate the use of Transformers for the Natural Language Premise Selection in the context of mathematical statements within the 1st Shared Task Natural Language Premise Selection at TextGraphs2022 Workshop(Valentino et al., 2022). We propose embedding the knowledge base with a BERT style transformer to obtain dense embedding of the statement in the knowledge base. By computing similarity of a given statement with the knowledge base we then obtain relevant candidates from the knowledge base that can be fed into a large Language model, such as GPT-3, to rank the candidates according to their importance to the given statement. We look at two structurally different transformers to compute the embeddings. 1. Sentence BERT (Reimers and Gurevych, 2019) and 2. MathBERT (Peng et al., 2021). The final ranking of the premises is done with GPT-3 using the OpenAI playground. As this approach does not require further training of the Transformer models and only uses the inherent knowledge it falls under the regime of Zero-shot Learning.

2 Related work

Transformer Models are large and deep neural network based on the attention mechanism (see (Vaswani et al., 2017)) that where pretrained originally with general Language Processing and Understanding tasks in mind (see (Vaswani et al., 2017), (Devlin et al., 2018), (Brown et al., 2020b), (Reimers and Gurevych, 2019)). Recently Transformers have been applied to tasks apart from Natural Language Processing and Generation. Models designed specifically for mathematical task can be found in (Shen et al., 2021) and (Peng et al., 2021).

Premise Selection (Ferreira and Freitas, 2020a) can be viewed as a precursor for Automated Theorem Proving (Alama et al., 2014). Automated Theorem Proving has a long history (Anderson, 1973) and is recently being tackled with approaches using Deep Neural Networks (e.g. (Ferreira and Freitas, 2020b), (Irving et al., 2016) and also (Polu and Sutskever, 2020)).

The approach in this paper is inspired by RETRO (Borgeaud et al., 2021) – a model that is able to reference a large knowledge base to solve general language tasks, by using a transformer on top of a frozen BERT retriever – and recent successes in prompt engineering for very large Language Mod-

Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 127–132
October 16, 2022.

els (Brown et al., 2020b).

3 Dataset Description

The organizers provide a dataset [1] with a tf-idf baseline (Valentino et al., 2022). Provided data consist of training, validation and test sets and a knowledge base. Each sample in the training and development sets includes *id*, *theorem text* and list of relevant *premise id's*. Texts of theorems and premises are represented in LaTeX markup. Dataset statistics are shown in Table 1. The knowledge base comprises 16205 premises.

4 Approach description

The central approach, which we have designed and evaluated for the premise selection task is **leveraging prompting methods for re-ranking**. Overall idea of it is to generate a primary ranking and further improve it by prompting generative language model with an instruction and top-k candidates from the primary ranking.

Prompt-based learning in a new paradigm in natural language processing. "Unlike traditional supervised learning, which trains a model to take in an input x and predict an output y as $P(y|x)$, prompt-based learning is based on language models that model the probability of text directly"(Liu et al., 2021). During prompt-based learning the original input x is modified using a template into a textual string prompt x' that has some unfilled slots (i.e., for model's answer), and then the language model is used for generating sequence completing the template. Due to multitasking abilities of generative language models to perform well on a wide range of tasks, there has appeared a bunch of prompt engineering approaches (prompt sharing, decomposition, noising, etc.), i.e. authors of the survey in (Liu et al., 2021) propose a typology including above 50 approaches.

For the primary ranking we have implemented two unsupervised approaches without model fine-tuning on train or validation sets: first uses sentence transformers (see Section 4.1) and second one uses MathBERT[2] (see Section 4.2) for embedding premise and theorems. Both approaches score premises by computing cosine similarity between the text of premise and text of theorem.

For re-ranking with prompt engineering we create prompts containing top-10 candidates from primary ranking and feed them to the GPT-3 model(Brown et al., 2020a) via OpenAI Playground[3], model *text-davinci-002*. Details of the approach are provided in sec. 4.3 and Appendix.

4.1 Ranking with Sentence Transformers

Sentence transformers (Sentence-BERT, SBERT) is an approach proposed in (Reimers and Gurevych, 2019) with implementation available at Gitlab[4]. It is "a modification of the pretrained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity"(Reimers and Gurevych, 2019). Authors of SBERT add a pooling operation to the output of BERT / RoBERTa to derive a fixed sized sentence embedding and experiment with three pooling strategies: using the output of the CLS-token,computing the mean of all output vectors (MEAN-strategy), and computing a max-over-time of the output vectors (MAX-strategy). In our experiments we used the default MEAN configuration.

During encoding of the texts of premises and theorems maximal length of input sequence was set to 90 tokens, that affect less than 10% of theorems in input data (see Table 1). Cosine similarity was computing using the built-in function in sentence transformers library[5]. This approach participated in evaluation phase and was ranked third among the shared task approaches (see Table 2, name **Ranking-SBERT**).

4.2 Ranking with MathBERT

There are a couple of Transformer models pretrained on Mathematical Text. Most notably MathBERT-EDU (Shen et al., 2021) and MathBERT (Peng et al., 2021). While the first is constructed for General NLP Tasks in Mathematics Education the latter focuses on Mathematical Formula Understanding. With MathBERT the authors include two pretraining tasks specifically designed to 1. relate a formula to its surrounding context (called Context Correspondence Prediction) and 2. relate parts of a formula to each other (Masked Substructure Prediction). Thus MathBERT is par-

[1] https://github.com/ai-systems/tg2022task_premise_retrieval

[2] https://huggingface.co/tbs17/MathBERT

[3] https://beta.openai.com/playground

[4] https://github.com/UKPLab/sentence-transformers

[5] https://www.sbert.net/docs/package_reference/util.html

Split	Train	Validation	Test
number of samples	8,438	2,779	2,712
average number of tokens per sample	42,65	42,81	43,01
long samples (>90 tokens) ratio	0,06	0,06	0,07

Table 1: Statistics of the training, validation, and test sets.

ticularly suitable to produce embeddings of the Knowledge base. Unfortunately The weights and source code are not available at the time of writing this article. We therefore experiment with embeddings computed using MathBERT-EDU.

Following the embedding of the knowledge base and the given statement we compute the similarity with FAISS (Johnson et al., 2019).

4.3 Re-ranking SBERT with Prompting GPT-3

This approach combines better performing **Ranking-SBERT** with **Prompting GPT-3**. The overall pipeline is shown in Fig. 1 We select top-10 candidates and design two prompt templates (see Appendix). **Prompt template a)** instructs the model to rank the premises by its relevance with the instruction *Rank premise IDs in the Knowledge by its relevance for the theorem. IDs of the most relevant premises appear first.* and **Prompt template b)** asks the model to select the most relevant premise ID: *Select most relevant premise ID for the given theorem*. None of the prompt templates includes a "helping" example. Both prompts performed reasonable in manual experiments, but **Prompt template a)** was chosen for implementation as the one with a higher possible impact on primary ranking.

5 Experiments

GPT-like models, despite impressive performance on many NLP tasks under the zero-shot and few-shot setup, are not capable of long-term memory. During re-ranking, the model may favor last seen premises and "forget" the relevant ones, that were presented (ranked) first in the original ranking.

We have implemented three simple experiments to estimate how the order of the premises in the prompt influences the GPT-3 generated ranking. Results are provided in Table2.

Experiment 1. Favoring the "last seen premise". In this experiment, we checked, whether GPT-3 favours "last seen" (and, probably, irrelevant) premise in the end of the prompt to more relevant ones in the middle of the prompt. For this reason, premises with ranks 1 and 2 were swapped in the ranking obtained from GPT-3. Since the Mean Average Precision (MAP) decreased, it is possible to say, that GPT-3 at least relies on the meaning of premises while re-ranking.

Experiment 2A. "Forgetting" a relevant premise. In this experiment, the premise id with rank 1 in Ranking-SBERT was moved closer to the ranking head in GPT-3 ranking (to rank 3). It slightly improved the GPT-3 ranking, but hasn't outperformed Ranking-SBERT approach.

Experiment 2B. "Forgetting" a relevant premise. In this experiment, a merged ranking was created by inserting the premise with rank 1 from Ranking-SBERT, to the re-ranked results from GPT-3. This has resulted in major improvement and has shown, that GPT-3 struggles with memorizing relevant information and tends to increase the rank of relevant premises, if they appear at the beginning of a long sequence.

6 Results and Error Analysis

Results of approaches, described in Section 4, are summarized in Table 2. For single rankings, best results were shown by **Ranking-SBERT**. **Re-ranking SBERT with Prompting GPT-3** approach performed slightly worse. However, merging these two rankings has led to the improved result (see Experiment 2B).

The results in the table steer towards the actively discussed question, have large language models (not only GPT-like) actually learned to do reasoning, or have they only memorized training examples (Li et al., 2021; Si et al., 2020), see also [6], [7]. Despite its game-changing performance for many NLP tasks, GPT-3 doesn't outperform SBERT for the natural language premise selection task, where reasoning based on a large knowledge base is required.

[6] https://jens-lehmann.org/blog/neural-language-models/

[7] https://lambdalabs.com/blog/demystifying-gpt-3/

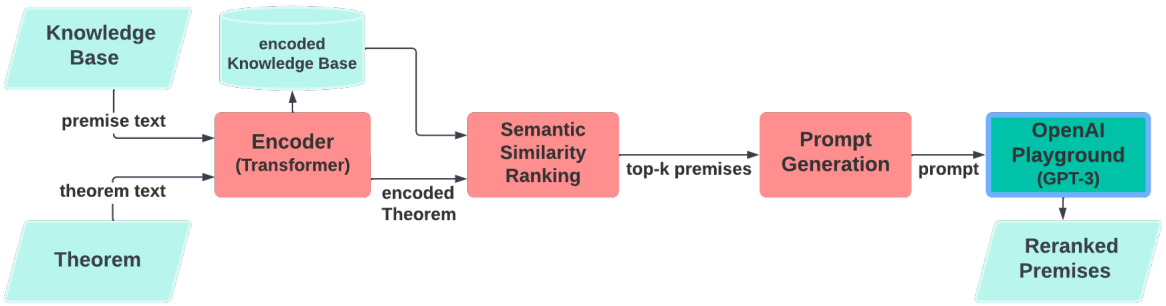

Figure 1: Premise Re-ranking with Prompts Template Design

Approach /Accuracy	Train	Validation	Test	Phase
Ranking-SBERT	n/a	n/a	**0,1460**	evaluation
Ranking-MathBERT-EDU	n/a	n/a	0.0609	post-competition
Re-ranking SBERT with Prompting GPT-3	n/a	n/a	0,1423	evaluation
Experiment 1. Favouring last seen item	n/a	n/a	0,1262	post-competition
Experiment 2A. Merged ranking	n/a	n/a	0,1450	post-competition
Experiment 2B. Merged ranking	n/a	n/a	**0,1497**	post-competition

Table 2: Approaches performance and experiments

Moreover, analysis of the GPT-3 generation output shows that the model occasionally repeated premise ids, omitted premise ids or "hallucinated" ids with comparable length during generation (total 16,5% of all premises). This erroneous output was not taken into account during re-ranking: it means, that for each sample there is a different portion of premises, re-ranked by GPT-3.

Prompt design should be implemented carefully, because GPT-3 tends to rely on the order of premises in the prompt, as well as on its meaning. Although the model doesn't really favor the last seen information in the prompt, it suffers from forgetting relevant information, if it was presented at the beginning of the prompt. This can be handled, for example by randomly shuffling elements subjected to re-ranking by GPT-3.

Overall, re-ranking by prompting generative language models, in a vanilla setup, does not improve similarity-based ranking, although merging these two rankings brings a better result.

7 Limitations and Future Work

While the approach presented here requires conceptually low resources compared to fine-tuning to the given training data, the use of GPT-3 comes with a significant cost (with the most capable model costing up to $0.02 for 1000 tokens). Furthermore, mathematical formulas are non-typical input for training general language models and hence tokenization might be less accurate thus also reducing the capability of the transformer models used for pre-ranking as they come with a maximum sequence length (512 for SBERT, MathBERT-EDU and MathBERT).

Performance of the proposed similarity ranking and prompt engineering approach, and available results from the shared task leaderboard show, that automated theorem proving is a hard task for NLU methods. LaTeXmarkup remains a hard type of input for encoders, that could possibly be overcome by using language models that have been pre-trained on LaTeX(e.g. MathBERT) or input-agnostic models, such as Perceiver. Furthermore, transformation of formulas into typesetting invariant representations should be investigated. Especially the representation of formulas as Operator Trees or translation to natural language might be beneficial in combination with general Language Models.

References

Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, Josef Urban, J Alama, T Heskes, · D Kühlwein, · E Tsivtsivadze, and · J Urban. 2014. Premise selection for mathematics by corpus analysis and kernel methods. *J Autom Reasoning*, 52:191–213.

Robert B. Anderson. 1973. Symbolic logic and mechanical theorem proving.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2021. Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

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

Deborah Ferreira and André Freitas. 2020a. Natural language premise selection: Finding supporting statements for mathematical text. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2175–2182, Marseille, France. European Language Resources Association.

Deborah Ferreira and André Freitas. 2020b. Premise selection in natural language mathematical texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7365–7374, Online. Association for Computational Linguistics.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Een, Francois Chollet, and Josef Urban. 2016. Deepmath - deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

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

Xiang Lorraine Li, Adhiguna Kuncoro, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2021. Do language models learn commonsense knowledge?

Martin Líška, Petr Sojka, Michal Ržička, and Petr Mravec. 2011. Web interface and collection for mathematical retrieval: Webmias and mrec.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. Mathbert: A pre-trained model for mathematical formula understanding. *CoRR*, abs/2105.00377.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *CoRR*, abs/2009.03393.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil T. Heffernan, Xintao Wu, and Dongwon Lee. 2021. Mathbert: A pre-trained language model for general NLP tasks in mathematics education. *CoRR*, abs/2106.07340.

Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. 2020. Benchmarking robustness of machine reading comprehension models.

Marco Valentino, Deborah Ferreira, Mokanarangan Thayaparan, André Freitas, and Dmitry Ustalov. 2022. Textgraphs 2022 shared task on natural language premise selection. In *Proceedings of the Sixteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-16)*. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Parameter name	Value
max_new_tokens	300
temperature	0.7
top_p	1
openai_frequency_penalty	0.0
openai_ presence_penalty	0.0
openai_stop_sequences	[ ]
n_responses	1

Table 3: GPT-3 Model parameters.

A Appendix

A.1 Experiment details and Parameters

The parameters for OpenAIs GPT-3 model in the OpenAI API have been chosen according to Table 3.