DUTh at SemEval 2024 Task 5: A multi-task learning approach for the Legal Argument Reasoning Task in Civil Procedure

Ioannis Maslaris **Avi Arampatzis**

Database & Information Retrieval research unit,
Department of Electrical & Computer Engineering,
Democritus University of Thrace, Greece.
{imaslari,avi}@ee.duth.gr

Abstract

Text-generative models have proven to be good reasoners. Although reasoning abilities are mostly observed in larger language models, a number of strategies try to transfer this skill to smaller language models. This paper presents our approach to SemEval 2024 Task-5: The Legal Argument Reasoning Task in Civil Procedure. This shared task aims to develop a system that efficiently handles a multiple-choice question-answering task in the context of the US civil procedure domain. The dataset provides a human-generated rationale for each answer. Given the complexity of legal issues, this task certainly challenges the reasoning abilities of LLMs and AI systems in general. Our work explores fine-tuning an LLM as a correct/incorrect answer classifier. In this context, we are making use of multi-task learning to incorporate the rationales into the fine-tuning process.

1 Introduction

In recent years, Large Language Models (LLM) development has witnessed unprecedented advancements, with Large Language Models such as GPT-3 demonstrating remarkable capabilities in understanding and generating human-like text. However, the effectiveness of these models in reasoning tasks remains an area of ongoing exploration and enhancement. While LLMs excel in linguistic fluency and context understanding, their capacity for reasoning often falls short of human-level comprehension (Huang and Chang, 2022).

In this paper, we describe the DUTh participation in *SemEval 2024 Task 5: The Legal Argument Reasoning Task in Civil Procedure* (Bongard et al., 2022)[1], on leveraging the reasoning capabilities of Large Language Models for multiple-choice question-answering in the context of US civil procedure. The task can be formulated as follows: *given an introduction to a case, a question, and a candidate answer, classify if the given answer is correct or wrong*. The dataset is based on *The Glannon Guide To Civil Procedure* by Joseph Glannons (Glannon, 2023). The multiple-choice questions come from the book's exercises, which aim to test the reader.

The training set has a size of 666 entries, which is smaller compared to other similar datasets aiming to examine the capabilities of LLMs using human-generated rationales (Hancock et al., 2019). Although the complexity of legal domain text and the number of details engulfed in real legal cases are large. The cognitive skills required to understand and handle legal cases make this task an interesting challenge for LLMs' reasoning abilities.

Our proposed system is a LegalBERT (Chalkidis et al., 2020) classifier, fine-tuned on a downstream task incorporating rationales for each answer. Our code implementation builds on the organizers' and is publicly available.[2] Additionally, we experimented with a multi-task Flan-T5 model. This strategy involves a different way to use rationales in the training process. The model is trained to predict the correct labels and, at the same time, generate relevant rationales. Its performance is evaluated through a custom loss function that accounts for the loss of the label prediction task and the loss of the rationale generation task separately. Although it did not surpass the performance of the LegalBERT classifier, it is an interesting approach that can be further examined on the current task.

2 Background

2.1 Related Work

Large Language Models have demonstrated remarkable few-shot capabilities (Smith et al., 2022; Zhang et al., 2022). This models, having more than 100 billion parameters, prove to be difficult to be

[1] https://codalab.lisn.upsaclay.fr/competitions/14817

[2] https://github.com/DataMas/SemEval2024-Task5

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1053–1057

deployed for regular real-world applications. For this reason, a lot of effort is being made toward leveraging the reasoning capabilities of smaller language models.

Knowledge distillation is a fine-tuning strategy aiming to transfer knowledge from larger and more complex models into smaller and more practical models. The larger teacher model acts on a dataset to predict its labels, and then the smaller student model is trained on these generated labels. In fact, distillation can be performed on unlabeled or limited labeled data (Abbasi et al., 2021; Fu et al., 2023).

Based on this idea, rationales can also be used to supervise the fine-tuning process of a smaller model. Human-generated rationales have been used as auxiliary inputs to improve the model's performance (Fatema Rajani et al., 2019). Another approach involves using these rationales as labels in order to make a model generate similar explanations for its predictions(Eisenstein et al., 2022). Learning from LLM-generated rationales is a relatively new field of experimentation. Larger Language Models can explain their predictions by generating reasoning steps (Kojima et al., 2022). This reasoning steps can be used in the same way as human-generated rationales to improve the performance of smaller models (Pruthi et al., 2022).

Taking the previous ideas one step further, (Hsieh et al., 2023) proposed a multi-task fine-tuning framework. They essentially train a model on two separate tasks at the same time. The model is trained to not only predict the correct labels but also to generate accurate rationales explaining its predictions. They extract rationales from LLMs using Chain-of-Thought prompting. With their multi-task training, they are able to fine-tune smaller language models, which perform comparable to or better than larger models. They achieve not only to reduce the size of the final models but also the size of the needed data.

2.2 Dataset

The organizers provide a dataset from the US legal domain in English. It is essentially a multiple-choice question-answer dataset. It contains questions and possible answers regarding topics of US civil procedure. Every question concerns a legal case. Along with each question, a paragraph serving as a general introduction to the case is provided. Every possible answer is accompanied by an analysis of why its context is relevant to the case. Additionally, for every batch of possible answers corresponding to a question, a paragraph with general comments discussing all answers' rationales is given.

The training and development sets are compiled of all the features discussed above (introduction, question, answer, analysis, and explanation), while the test set excludes the features giving reasoning behind every answer. The train, development, and test sets consist of 666, 84, and 98 entries, respectively. Following, we can see the structure of the dataset clearly. The items without bold annotations are not included in the test set.

- **"introduction":** A paragraph regarding the context of the question.
- **"question":** The question regarding a legal case.
- **"answer":** A possible answer to the question.
- **"label":** A binary indicator for correct and wrong answer.
- "analysis": Reasoning on why each answer is right or wrong.
- "explanation": A paragraph discusing the reasoning of all possible answers to a question.

2.3 Evaluation Measures

Submissions are evaluated by two metrics:

- F1 score: The F1 score is defined as the harmonic mean of precision and recall, offering a single metric to asses a classifier's performance by considering both false positives and false negatives.
- Accuracy: Accuracy score is a measure used to evaluate the performance of a classification model. It is defined as the ratio of correctly predicted observations to the total observations.

Finally, participants are ranked based on the F1 score of their system.

3 System Overview

3.1 Data Pre-processing

Pre-trained Large Language Models are constrained by the maximum length of text input they can process. This limitation arises from the model's fixed-sized input layer, which can only accommodate a certain number of tokens (e.g., words or sub-words). We want to use the *introduction* and

analysis as context for the *question* and *answer* accordingly. This makes the final input exceed its token limit. The models we have used for our experiments have a limit of 512 tokens. Combining the *introduction, question, analysis* and *answer* creates input instances of length greater than 700 words on average.

In order to fit the constructed input instances, we have employed a sliding window mechanism. We use the same Sliding Window Complex (SWC) strategy proposed by the organizers (Bongard et al., 2022)[3]. This sliding window algorithm splits the inputs into chunks of specified length L, which is smaller than the limit length. In order for everything to fit, some features must be sliced. The specific details on how the features are sliced will be described later in the System architecture and Multi-task learning sub-sections. For example, one approach is to concatenate *explanation* and *answer* by keeping the whole *answer* to every chunk and pad the explanation until the limit of words is reached.

3.2 System Architecture

For our system, we utilize Legal-BERT to classify every chunk as wrong or correct. We also evaluated BERT (Devlin et al., 2018) and DistilBERT (Sanh et al., 2019), but they proved inferior. We fine-tune each model with constructed input instances. These instances are compiled by the *question*, *introduction*, *answer*, and *analysis*. The way these features are concatenated to form an input is to keep the whole *question*, *analysis*, and *answer*. The rest of the available space is filled with a part of the *introduction*.

Before the *question*, we add the distinctive feature *Q:*, while we also do the same thing before the *answer* with the distinctive feature *A:*. This way, we are making it clear to the model when a question and an answer begin. We have experimented with the learning rate and weight decay and used Optuna on the best performing model for hyperparameter optimization. Finally, training was terminated with early-stopping, with patience being set to 10.

3.3 Multi-task Learning

Based on the idea of *Distilling-step-by-step* (Hsieh et al., 2023)[4], we implement a similar system where we use the *analysis* feature provided in the dataset as rationale. During the training process, the model is trained to both predict the correct label and, at the same time, generate a comprehensive rationale for every input instance. This is done through the use of a custom loss function that accounts for the label prediction error and the rationale generation error.

$$L = (1 - w)L_{\text{label}} + wL_{\text{rationale}}$$

Adding the rationale generation loss to the training process helps the model better understand the logic behind why every answer is correct or wrong. The loss function is weighted with a factor w. Through w, we can control which task the model should focus on more during training. Choosing a $w = 0.5$ means that the model will try equally to learn both tasks. For values $w < 0.5$ the model places more importance on learning to predict labels correctly, and for values $w > 0.5$ the model is more focused on learning to generate accurate rationales.

We use the sliding window to create the input instances. These consist of an *introduction*, a *question*, and an *answer*. The distinctive features *Q:* and *A:* are also used here in the same way as described in the previous paragraph. In order to fit every instance into the limit of input tokens, every chunk has the complete *question* and *answer* and is padded with part of the *introduction*. In order to help the model distinguish between the two tasks, every instance is padded with another distinctive feature. For the label prediction task, we use the feature *Predict:* at the beginning of the instance. For the rationale generation task, we use the feature *Explain:*. This is done on our custom data collator function[5] and the result is two separate datasets.

In order to train the model in a multi-task manner, we created a custom trainer function. In this function, the model is prompted separately with the two task-specific datasets coming from the data collator. The answers of the model are evaluated, and the loss is computed through the custom function we described earlier. Finally, we define the prediction step of the model to produce answers for both tasks.

For this strategy, we utilize the small version of the Flan-T5 model (Chung et al., 2022). Because the model is trained to generate rationales, it must receive the labels as text. We transformed the labels of 0 to *Wrong* and the labels of 1 to *Correct*. Consequently, when the model is prompted to predict the

[3] github.com/trusthlt/legal-argument-reasoning-task
[4] github.com/google-research/distilling-step-by-step
[5] huggingface.co/docs/transformers/main_classes

labels of new data, it will respond with *Correct* or *Wrong*. For this reason, we have to convert the text responses to 1–0 accordingly in order to evaluate them and submit our results.

The final multi-task trained system receives a dataset and first preprocesses it. It concatenates the *introduction*, *question*, and *answer* and converts the labels to *Correct* - *Wrong* text. The model is prompted with the instances, and its responses are converted into 1–0.

Pre-trained model	F1-score	Accuracy
LegalBERT	0.5382	0.6837
BERT	0.5081	0.7245
DistilBERT	0.4269	0.7245
LegalBERT*	0.4827	0.7245
Multi-task Flan-T5	0.5324	0.6224

Table 1: Best performance of each model. *This is the score the best-performing model achieved during the evaluation phase. All the other scores have been achieved during the post-evaluation phase and are not counted for the leaderboard.

4 Experimental Setup

4.1 Chain of Thought

Before creating embeddings, we tried to fine-tune the models using a Chain of Thought strategy. During the sliding window process, we used auxiliary phrases to make the final input make more sense to the model. For example, we used the phrase *Based on the following* before adding the part of the *introduction*. After the *introduction* and before the question, we added the phrase *Answer the following question*. For the answer-analysis part, we used the phrases *The following answer* followed by the answer and *is correct/wrong because* followed by the analysis.

Although a widely used and promising technique, CoT did not prove to increase the performance of our models. At least based on the phrases and the arrangement we used. The task prefixes *Predict* and *Explain* that we used for the multi-task system can also be considered as a CoT approach. On this occasion, they were efficient in guiding the model to distinguish between the two tasks.

4.2 Experiments

Our experiments are mainly focused on fine-tuning different models under different hyperparameters. The hyperparameters we experimented on were the learning rate and the weight decay. We We came up with the best set of hyperparameters through optimization using the Optuna hyperparameter optimization framework. [6] In the first set of experiments regarding fine-tuning on a downstream classification task, we evaluated three pre-trained models: BERT, LegalBERT, and DistilBERT. The best-performing model proved to be the LegalBERT. For the second set of experiments regarding multi-task fine-tuning, we utilized the small version of the Flan-T5 model. The same hyperparameter optimization procedure was followed. We also experimented with the parameter w which controls the amount of focus on each task. A weight $w = 0.5$ proved to be slightly better.

5 Results

The comprehensive scores of our systems across the utilised models are presented on Table 1. The highest F1 score was 0.5324 achieved by LegalBERT, followed closely by the multi-task T5. According to accuracy, BERT and DistilBERT perform better with a score of 0.7245, and LegalBERT comes in second with 0.6837. LegalBERT . Although our models do not perform well, we can make some assumptions on why that is.

Firstly, regarding LegalBERT, it is possible that simply adding the rationale to the input along with the *introduction*, *question* and *answer* will not helping the model learn the logic behind justifying each answer. In fact, it makes the model perform worse compared to setups where only *introduction*, *question* and *answer* is used (Bongard et al., 2022). Additionally, our multi-task system, although incorporating a more complex training mechanism, it does not seem to be able to distinguish answers efficiently. The small version of Flan T5 is only of 80 million parameters. At this scale, it might be difficult for language models to grasp complex concepts laying on rationales. This, in fact, can be confirmed by prompting the multi-task model to generate rationales based on the input. The generated rationales barely makes any sense.

6 Conclusion

Through our experiments, we could not find a significantly performing system. Even the multi-task approach, which makes good use of the rationales to better establish a connection between input and

[6] https://github.com/optuna/optuna

label, could not perform well. But we demonstrated the possible limitations and difficulties of such tasks, where logical reasoning is needed in order for a model to perform well.

The primary benefit of multi-task learning lies in the use of rationales, enabling the model to perceive the reasons behind the correctness or incorrectness of every answer. In this work, our capabilities were constrained by hardware limitations, leading us to experiment with a smaller Language Model. However, this model's capacity to comprehend longer content is limited by its size.

Next steps could involve experimentation with bigger Language Models regarding the multi-task approach. We believe that a larger model could better grasp the context of the rationales and draw better associations between a question and possible answers. Another approach regarding the multi-task strategy is to incorporate rationales through a more efficient loss function. Another weighing strategy could be used, for example.

References

Sajjad Abbasi, Mohsen Hajabdollahi, Pejman Khadivi, Nader Karimi, Roshanak Roshandel, Shahram Shirani, and Shadrokh Samavi. 2021. Classification of diabetic retinopathy using unlabeled data and knowledge distillation. *Artificial Intelligence in Medicine*, 121:102176.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. *arXiv preprint arXiv:2211.02950*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Eisenstein, Daniel Andor, Bernd Bohnet, Michael Collins, and David Mimno. 2022. Honest students from untrusted teachers: Learning an interpretable question-answering pipeline from a pretrained language model. *arXiv preprint arXiv:2210.02498*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv e-prints*, pages arXiv–1906.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*.

Joseph W Glannon. 2023. *Glannon guide to civil procedure: learning civil procedure through multiple-choice questions and analysis*. Aspen Publishing.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. 2022. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.