The Review of Socionetwork Strategies (2024) 18:75–100
https://doi.org/10.1007/s12626-023-00154-y

ARTICLE

Exploring Prompting Approaches in Legal Textual Entailment

Onur Bilgin[1] · Logan Fields[1] · Antonio Laverghetta Jr.[1] · Zaid Marji[1] · Animesh Nighojkar[1] · Stephen Steinle[1] · John Licato[1]

Received: 1 September 2023 / Accepted: 14 December 2023 / Published online: 23 January 2024

Abstract

We report explorations into prompt engineering with large pre-trained language models that were not fine-tuned to solve the legal entailment task (Task 4) of the 2023 COLIEE competition. Our most successful strategy used simple text similarity measures to retrieve articles and queries from the training set. We report on our efforts to optimize performance with both OpenAI's GPT-4 and FLaN-T5. We also used an ensemble approach to find the best combination of models and prompts. Finally, we analyze our results and suggest ideas for future improvements.

Keywords AI · NLP · Reasoning · Law · Legal

1 Introduction

If we hope for AI systems to have a robust understanding of the instructions they are given or the rules they must follow, we must advance the science of how human-written rules can be automatically reasoned over and resolved. Any time a governing law, mission order, code of ethical conduct, or other verbal or written instruction is produced and given to a subordinate in a fixed, referable form (a "rule"), there is some expectation that the rule will be followed in the spirit in which it was created. Often this means there is an assumption (or hope) that the rule's intent is adequately conveyed. However, the complete conveyance of

Onur Bilgin, Logan Fields, Antonio Laverghetta Jr., Zaid Marji, Animesh Nighojkar, Stephen Steinle, and John Licato have contributed equally to this work.

✉ Onur Bilgin
onurbilgin@usf.edu

John Licato
licato@usf.edu

[1] Advancing Machine and Human Reasoning Lab, Department of Computer Science and Engineering, University of South Florida, 4202 E. Fowler Avenue, Tampa, FL 33620, USA

a rule's intent requires a multitude of background knowledge: the history behind the statement, prototypical examples of its proper and improper interpretations, the intended goals of the rule's creator, the proper scope of the rule's open-textured predicates, and so on [1–6].

Carrying out such reasoning is a challenging task, even for state-of-the-art artificially intelligent language models (LMs). A primary reason for this difficulty is the prevalence of *open-textured terms* (OTTs)—terms whose extensions are not completely and unambiguously fixed at the time of their initial use [1, 7]. For example, consider a traffic regulation stating that vehicles must "keep to the right as far as is *reasonably safe*" [3] or another example, a small recreational park and an autonomous robot guarding that park with the rule: "No motorized vehicles in the park. Violations will be rectified in an *appropriate manner*." The next day a person with a motorized wheelchair and a loud motorcycle came to that park. The robot decides that both the wheelchair and the motorcycle violated the rules. But how should it determine the "appropriate manner" to not forcefully evict the person with the wheelchair or to not only ask the motorcycle owner quietly to leave, which might be ignored [8]?

Such a regulation or rule would need to be understood and interpreted, e.g. by autonomous driving vehicles, traffic enforcement bots, or autonomous guarding robots. However, it is implausible to exhaustively list an exception-free accounting of all possible scenarios and conditions that can be considered instances of the open-textured term "reasonably safe" or "appropriate manner"—any such attempt would inevitably limit the scope of the regulation and render it fatally inflexible in the face of unpredictable conditions. Using OTTs is a necessary and unavoidable feature of regulatory and legal language [1–6, 9]. Thus ways to work with them must be addressed by any sufficiently robust account of compliance detection.

There are multiple approaches for addressing this problem in AI research. The first approach is to reduce the open-texturedness of the rules so that they can be reasoned over using transparent algorithms and formal methods. For example, our lab has recently explored the translation of rules (containing OTTs) in a collectible card game into programming language code [10], which can then allow for reasoning over the code and the game itself [11]. The second approach is to *embrace open-texturedness*; i.e., to accept that no approach will ever entirely remove the open-texturedness of languages in rules (and to acknowledge that rule systems with no open-texturedness is not desirable, either), and to instead focus on how to reason over OTTs in their natural language forms, without forcing translations into unambiguous formal languages.

Under this second approach, there are multiple alternative approaches. One of these takes the position that for artificially intelligent systems to follow human-written rules properly, they need to be able to interpret them, which requires resolving OTTs. Furthermore, the interpretation the AI chooses should be provided in a form that stakeholders can inspect, test, and use as precedent for future interpretations [8, 12, 13]. In other words, given text to be interpreted by an AI, human stakeholders need to be able to inspect: (I1) *how* the AI interpreted that text, and (I2) *why* the AI believes that interpretation is best. An emerging body of work is exploring approaches to (I2) under the topic of *interpretive argumentation* [5, 8, 14–17]. Given the difficulty of generating and evaluating interpretive argumentation, the COLIEE competition is a helpful stepping-stone towards that ultimate goal.

Task description. We focused on Task 4 of the COLIEE 2023 competition. This task was formulated as follows: We are given a set of articles A and a query q. Each article is a short text snippet which is a statute from the Japanese Civil Code translated into English, normally consisting of no more than a few sentences. The query is a short textual description of something that may or may not be true given the articles; e.g., "There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage." Our algorithms must output either 'Y' or 'N', depending on whether q follows from A. Here, the articles lead as rules to open-texturedness because of the assumptions that need to be made on the true intent of the article to answer the query. The missing history behind the article, the proper and improper interpretations, and the correct intent are missing terms that must be embraced by the reasoner to solve the task.

Overview of Results. We submitted three entries to the 2023 COLIEE competition. Our `AMHR01` submission used Flan-T5, a language model that had been instruction fine-tuned and which did well on the previous years' datasets, which we used as a validation set (in keeping with the COLIEE-2023 rules, we did not consider the test set R04 at all in selecting our models or hyperparameters). `AMHR02`, in contrast, deliberately used GPT-4—although this was a disallowed resource, we wanted to see how well it performed compared to other existing tools. Finally, our `AMHR03` submission was an ensemble approach that tried to combine the recommendations made by various models and hyperparameter settings. In our experiments, we utilized different shot selection methods and tested advanced prompting strategies. Our results showed that prompting strategies combined with intelligent shot selection methods can achieve the results of carefully designed models. We performed an in-depth analysis of those methods and strategies and the choice of temperature value in LMs.

2 Background

Methods for In-Context Learning. In recent years, advances in NLP research have been dominated by large language models (LLMs), which have tens or even hundreds of billions of parameters [18]. These models are able to solve new tasks *few-shot*, where the model is given only a small number of training examples yet can achieve strong task performance [19]. This has enabled a new paradigm of NLP engineering, where experts interact directly with LLMs and train them to solve tasks via *prompt engineering* (also called *prompt tuning*), whereby an input context is discovered, either by manual engineering or via a search algorithm, and used to prompt the model [20]. In this work, we explore two broad categories of prompts: those which focus on finding a combination of training examples (shots) to use as context (*prompt retrieval*) and those based on *chain-of-thought* prompting [21], where instructions given to the model are elaborated to induce more reliable and accurate behavior. Note that both these approaches may be used simultaneously to boost performance further [22]. Prompt retrieval aims to find the optimal strategy for selecting the training examples to use as context. Prior work has employed supervised models trained to predict the most informative shots (e.g., [23–25]). Others have used unsupervised models based on similarity metrics, such as BM25 [26] or SBERT [27]. On the other hand, chain-of-thought approaches are meant to help a model "think step-by-step" and thus embed the correct answer as well as the reasoning process [21]. Other prompting methods falling into this class include faithful chain-of-thought [28], self-taught reasoning [29], and maieutic prompting [30]. These methods have enabled high few-shot performance on challenging benchmarks of linguistic reasoning [31], and have been used in previous COLIEE competitions [32]. High performance with different prompting techniques has led to research on automatically tuning prompts with LMs. Zhou et al. introduced Automatic Prompt Engineer (APE) to generate and select instructions automatically. They report achieving human-level performance on different tasks [33].

Methods for Legal Domain. Nguyen et al. have utilized a BERT-base model [34], trained with relevant articles extracted by TF-IDF vectors from the Civil Code [35], for the entailment task. Rosa et al. have utilized DeBERTa [36] and monoT5 [37] for the legal entailment task and they have reported that monoT5 in a zero-shot setting is a robust model for the task [38]. Shao et al. have applied a semantic understanding and exact matching algorithm for legal case retrieval and entailment tasks. They have used matching-based retrieval models and a BERT-PLI model proposed by Shao et al. [39] for the legal case retrieval task. For the legal entailment task, they have applied a fine-tuned BERT. They have reported that the entailment task benefits from semantic understanding while both semantic understanding and exact matching are complementary for the retrieval task [40].

Co-occurrence-based methods such as n-grams and BM25 have also been applied to extract relevant cases [41]. Althammer et al. have successfully leveraged long documents in retrieval tasks by splitting the documents into paragraphs, applying a paragraph-level retrieval task, and summarizing the cases re-ranking with BERT; however, the latter has not improved the generalization performance [42]. Askari et al. have utilized a stepwise approach for the case law retrieval task where they first extracted meaningful sentences and n-grams. These extracted elements have been used as queries to retrieve possible relevant cases, which have later been re-ranked. They have reported issues due to long documents for both neural and statistical methods and thus have employed a cluster-driven BERT model in combination with BM25 [43].

The In-Context Learning ability of LMs has been applied in the legal domain as well. Yu et al. have investigated the effectiveness of prompts utilizing zero-shot legal reasoning methods such as IRAC (Issue, Rule, Application, Conclusion) on GPT−3.5. They have reported for the legal entailment task that despite achieving high accuracy on the COLIEE 2021 dataset, their accuracy was lower on the COLIEE 2022 dataset. They have also reported the differences in model accuracy based on word selection in prompts. For example, they observed on GPT-3 that adding the word "following" to the premise increases the accuracy by nearly 2.5%, or replacing "the given premise" with "Japanese civil code statutes" decreases the accuracy by 3.7% [32]. Savelka et al. have reported similar sensitivity issues and observed a high impact on GPT-4 predictions with minor changes in prompts. They have observed a competitive performance of GPT-4 in the legal domain but chain-of-thought prompting did not noticeably improve the performance of the model [44]. Savelka et al. have reported hallucinations as one of the key limitations of LLMs. In their tests on legal corpus inserting contextual data to prompt has reduced the hallucination of GPT-4 [45].

For the COLIEE 2023 competition in Task 4, the zero-shot `flan-alpaca-xxl` model of the JNLP team achieved a test set accuracy of 0.7822. Their zero-shot `flan-ul2` and `flan-t5-xxl` models achieved an accuracy of 0.7525. The UA team reformulated the task as a Natural Language Inference task and achieved with a fine-tuned DeBERTa-large model an accuracy of 0.6634 on the test set. The KIS team worked with the Japanese dataset. Their LUKE-based ensemble models achieved an accuracy of 0.6931 and 0.6733, while their ensemble model trained for problems with and without alphabetical person names achieved an accuracy of 0.6535 [46].

Open-Texturedness in Machine Learning. Machine learning has the potential to drastically improve equity in the application of laws across racial, socioeconomic, and other categorical features. Where human judges and legal scholars may be influenced by biases [47, 48], a sufficiently trained machine learning model may be able to objectively recognize the features of a case and render an equitable decision. However, models rarely improve equity in practice because of bias preservation in the models' training [49–51] and open-texturedness in legal terminology.

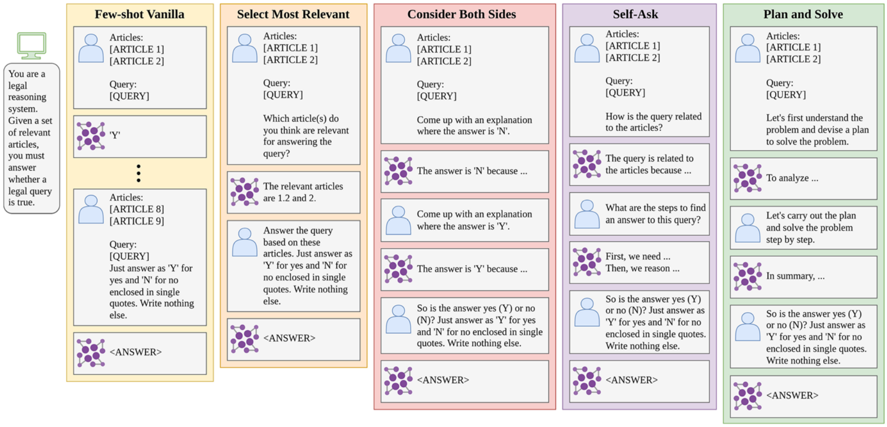
OTTs are nearly ubiquitous within legal reasoning [52, 53], where laws may have overarching downstream impacts and disagreements about their scopes are often resolved within appellate courts by expert judges. However, because the interpretation of open-texturedness relies on the discretion of "reasonable humans," which is known to suffer from biases, it presents a considerable challenge for AI models to interpret such terms in a human-like manner without perpetrating those same biases. The OTTs within legal reasoning are not only a challenge within machine learning. Legal knowledge-based systems were developed using rule-based systems to make legal decisions and their reasoning. However, these systems were limited by the open-texturedness and vagueness of the human language [54].

3 Approach 1: GPT-4

LLMs [19, 55] trained for text generation tend to outperform humans on various professional and academic benchmarks. Some of these models have been tuned to behave like "chatbots", preserving conversation history and adhering to instructions. OpenAI developed a product, GPT-4,[1] that can be used as a chatbot through their API.[2] Given a chat conversation, the API returns a chat completion response, allowing the user to set both the human's and the model's previous responses. The

[1] https://openai.com/research/gpt-4.

[2] https://platform.openai.com/docs/api-reference/chat.



You are a legal reasoning system. Given a set of relevant articles, you must answer whether a legal query is true.

Few-shot Vanilla

Articles:
[ARTICLE 1]
[ARTICLE 2]

Query:
[QUERY]

'Y'

⋮

Articles:
[ARTICLE 8]
[ARTICLE 9]

Query:
[QUERY]
Just answer as 'Y' for yes and 'N' for no enclosed in single quotes. Write nothing else.

<ANSWER>

Select Most Relevant

Articles:
[ARTICLE 1]
[ARTICLE 2]

Query:
[QUERY]

Which article(s) do you think are relevant for answering the query?

The relevant articles are 1.2 and 2.

Answer the query based on these articles. Just answer as 'Y' for yes and 'N' for no enclosed in single quotes. Write nothing else.

<ANSWER>

Consider Both Sides

Articles:
[ARTICLE 1]
[ARTICLE 2]

Query:
[QUERY]

Come up with an explanation where the answer is 'N'.

The answer is 'N' because ...

Come up with an explanation where the answer is 'Y'.

The answer is 'Y' because ...

So is the answer yes (Y) or no (N)? Just answer as 'Y' for yes and 'N' for no enclosed in single quotes. Write nothing else.

<ANSWER>

Self-Ask

Articles:
[ARTICLE 1]
[ARTICLE 2]

Query:
[QUERY]

How is the query related to the articles?

The query is related to the articles because ...

What are the steps to find an answer to this query?

First, we need ...
Then, we reason ...

So is the answer yes (Y) or no (N)? Just answer as 'Y' for yes and 'N' for no enclosed in single quotes. Write nothing else.

<ANSWER>

Plan and Solve

Articles:
[ARTICLE 1]
[ARTICLE 2]

Query:
[QUERY]

Let's first understand the problem and devise a plan to solve the problem.

To analyze ...

Let's carry out the plan and solve the problem step by step.

In summary, ...

So is the answer yes (Y) or no (N)? Just answer as 'Y' for yes and 'N' for no enclosed in single quotes. Write nothing else.

<ANSWER>

Fig. 1 Model prompting structures. The above image represents a textual conversation as seen by a language model. The green computer on the left shows the text for models that use system prompts. The various colored boxes show different prompting strategies where the conversation flows from top to bottom. The boxes with human icons show input *messages*, while the boxes with the purple graph icons show model *responses*

API also allows the user to set a “system” prompt, which persists throughout the “conversation” and helps set the model’s behavior.[3] According to OpenAI, GPT-4 scores around the top 10% of test takers on a bar exam, though they provide very few details on this exam and how the model was used to make predictions on it. GPT-4 is also instruction-tuned [56] using reinforcement learning from human feedback (RLHF) [57] to follow a variety of written instructions. This also improves zero-shot performance (especially on classification tasks [58]) of the model because examples (shots) are no longer required to *show* the expected format of responses, the user can just *tell* what the format should be (more details on this training strategy are provided in Sect. 4). We used the OpenAI API to experiment with multiple types of prompts, all of which are illustrated in Fig. 1. We incorporated the GPT-4 as a stand-alone model and in an ensemble of 5. In the ensemble, we run tests both with a fixed temperature value and each with a different temperature value. We tried zero-shot and few-shot variants of each, and the results are shown in Table 1. The first two ensemble models with an ensemble size of 5 in the GPT-4 section have a temperature value of 1 for each model in the ensemble. The remaining two ensembles have for each model the temperature values of 0, 0.25, 0.5, 0.75, and 1. The following two models with an ensemble size of 1 have a temperature value of 1. The last GPT-4 model with an ensemble size of 1 has a temperature value of 0.

[3] https://platform.openai.com/docs/guides/chat.

Table 1 Summary of prompting results

Model	Prompt type	Shots	Similarity method	Ensemble size	H30	R01	R02	R03	Train	Overall accuracy
Kano Lab 2022 [79]	n/a	n/a	n/a	n/a	**70.000%**	**63.960%**	**54.320%**	**67.890%**	n/a	**64.101%**
GPT-4	**Vanilla**	**6**	**TF-IDF**	**5**	**82.857%**	**80.180%**	**81.481%**	**88.073%**	**n/a**	**83.288%**
GPT-4	Vanilla	6	TF-IDF Balanced	5	81.429%	77.477%	80.247%	88.073%	n/a	81.941%
GPT-4	Vanilla	6	TF-IDF Balanced	5	81.429%	79.279%	80.247%	88.073%	n/a	82.480%
GPT-4	Vanilla	0	None	5	80.000%	81.081%	83.951%	86.239%	n/a	83.019%
GPT-4	Vanilla	5	TF-IDF	1	81.429%	83.784%	80.247%	88.991%	n/a	84.097%
GPT-4	Vanilla	0	None	1	80.000%	76.577%	83.951%	86.239%	n/a	81.671%
GPT-4	Vanilla	0	None	1	77.143%	80.180%	82.716%	85.321%	n/a	81.671%
T0	Vanilla	0	None	n/a	52.857%	55.856%	64.198%	65.138%	61.120%	59.839%
T0p	Vanilla	0	None	n/a	52.857%	59.460%	65.432%	66.972%	61.280%	61.725%
T0–3b	Vanilla	0	None	n/a	52.857%	55.856%	65.432%	58.716%	56.640%	58.221%
Flan-T5	Vanilla	3	TF-IDF	n/a	68.571%	72.072%	77.778%	74.312%	69.440%	73.315%
Flan-T5	Vanilla	2	TF-IDF Balanced	n/a	67.143%	71.117%	79.012%	77.817%	69.992%	74.059%
Flan-T5	**Vanilla**	**2**	**TF-IDF Balanced**	**n/a**	**70.000%**	**71.117%**	**80.247%**	**79.817%**	**71.200%**	**75.456%**
Flan-T5	Vanilla	2	TF-IDF Balanced	n/a	64.857%	65.765%	65.432%	71.559%	n/a	n/a
Flan-T5	Vanilla	4	TF-IDF	n/a	60.000%	60.360%	59.260%	67.890%	70.720%	n/a
Flan-T5	Vanilla	2	TF-IDF	n/a	60.000%	66.667%	65.432%	70.642%	69.920%	n/a
Flan-T5	Vanlilla	4	TF-IDF	n/a	58.571%	62.162%	58.024%	67.890%	69.440%	n/a
Flan-T5	Vanilla	2	TF-IDF	n/a	62.857%	63.963%	65.432%	72.477%	n/a	n/a
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	64.286%	67.568%	72.400%	75.230%	70.080%	70.255%
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	65.714%	70.270%	76.543%	76.147%	71.680%	72.507%
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	65.714%	71.117%	74.074%	75.230%	70.240%	71.952%
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	68.571%	72.973%	77.778%	76.147%	70.720%	74.124%
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	68.571%	72.973%	76.543%	77.064%	69.920%	74.124%

Table 1 (continued)

Model	Prompt type	Shots	Similarity method	Ensemble size	H30	R01	R02	R03	Train	Overall accuracy
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	64.286%	67.568%	77.778%	75.229%	70.080%	71.429%
Flan-Alpaca	Vanilla	2	TF-IDF	n/a	64.286%	63.964%	70.370%	70.642%	68.160%	67.385%
Flan-Alpaca	Vanilla	2	TF-IDF Balanced	n/a	58.571%	64.865%	69.136%	73.394%	67.840%	67.116%
Flan-Alpaca	Vanilla	4	TF-IDF	n/a	64.286%	65.766%	69.136%	66.972%	69.440%	66.577%
Flan-Alpaca	Vanilla	4	TF-IDF Balanced	n/a	57.143%	65.766%	69.136%	68.807%	67.680%	65.768%

The topmost row shows the results of the best-performing model from the 2022 Task 4 competition. The first section shows GPT-4 ablations, the second section shows results from models included in the ensemble (excluding the submitted Flan-T5-xxl model), the third section shows Flan-T5-xxl and Flan-Alpaca-xxl ablations. Underlined rows indicate the submitted Flan-T5-xxl and GPT-4 models

4 Approach 2: Instruction-Tuned Transformers

We use the Flan-T5[4] [59], T0[5] [60] and Flan-Alpaca[6] [61] checkpoints publicly available on HuggingFace and use the transformers library [62] to perform prompt-tuning using the models. Our work used `flan-t5-xxl`, `T0`, `T0p`, `T0pp`, `T0-3B` and `flan-alpaca-xxl` checkpoints. We use the text generation pipeline provided by the library and prompt-tune the model to generate the correct label, given the validation example articles and query and an optional number of training shots. We use a simple regex pattern to detect if the model generated a correct label. Given the model's raw outputs, we convert all text to lowercase, strip out leading and trailing spaces and newlines, and check if the output is any of the following strings:

1 If the correct label is "Y":
   (a) "y", "yes", "the answer is yes"
2 If the correct label is "N":
   (a) "n", "no", "the answer is no"

We found that `flan-t5-xxl` and `flan-alpaca-xxl` were quite well-behaved on this task and, in the overwhelming majority of cases, generated only the label string and thus did not require careful pattern matching to avoid false negatives. This was primarily the same with T0; however, the smaller models did not always consistently generate only the label (e.g., "the answer is yes"), and we thus added other valid patterns as possible strings to match. In cases where a valid label was not detected or the model predicted the wrong label, we marked the example as incorrect. We turn off sampling in all experiments to force determinism in all generations so that the prompt only affects the output.[7] We use a maximum sequence length of 512, the longest the models support. We experiment with the same chain-of-thought and similarity-based shot selection strategies as in our GPT-4 system (Fig. 1). However, unlike GPT-4, we found that the model's predictions for chain-of-thought were purely extractive. For example, when the prompt indicated the models should generate reasons for the answer to be either 'Y' or 'N', the model's rationales were extracted verbatim from either the articles or the query. This behavior, coupled with the much smaller maximum sequence length in these models, caused very poor performance in our chain-of-thought prompts, and we did not investigate them in detail for our Huggingface models.

[4] https://huggingface.co/google/flan-t5-xxl.

[5] https://huggingface.co/bigscience/T0pp.

[6] https://huggingface.co/declare-lab/flan-alpaca-xxl.

[7] Because sampling was disabled, the temperature does not affect these models' predictions.

5 Approach 3: Ensemble

Ensembles are a combination of multiple models used to achieve a higher prediction accuracy or better generalization because the different classifiers in the ensemble may have sensitivity for a different set of samples and have learned different subsets of features. By combining different classifiers, the goal is to reduce bias and error and to increase prediction performance [63]. Recent work has applied ensembles to the prompting of LMs, for example, by combining multiple prompts into an "ensemble of prompts" using multiple prompts with the same model and combining the predictions from each using a meta-classifier [64, 65]. We reasoned that applying this idea to our Huggingface models might boost performance even more.[8]

We tested two approaches for our ensemble model. In the first approach, we applied brute force search to select the best combination of models with the highest validation accuracy. Thus, we created an ensemble dataset from the validation set, where the features are the model predictions for each sample in the validation set. Then, for each combination of models, we calculated the model's accuracy using majority voting and selected the ensemble with the highest accuracy on the validation set. The models in our best-performing brute force ensemble are three `flan-t5-xxl` runs with balanced TF-IDF and two shots, a `flan-t5-xxl` model with TF-IDF unbalanced and three shots, and `T0`, `T0p`, and `T0 3B` each zero-shot. The resulting ensemble consists of 7 models. Further details on the shot selection strategies used by these models are found in Sect. 6.

In the second approach, we aggregated TF-IDF vectors trained on the validation set as additional features besides model predictions. Here, we reduced the vocabulary size to 10% based on TF-IDF scores to reduce the feature space. Then we applied 5-fold cross-validation for the validation set. We used support vector machines (SVMs) [66] and random forest [67] models for the training, both of which were implemented in scikit-learn [68]. The parameters are chosen based on the validation set results. For the SVM models, the radial basis function kernel is used, and the regularization parameter is 1.0. For our random forest model, the maximum depth of the tree is 5, with a total of 100 estimators. The Gini impurity is used to measure the quality of the split [69].

6 Overall Training Procedure

Per the competition specifications, we use the past four years' datasets (H30, R01, R02, R03) as validation datasets and older years' datasets as training datasets. We use only few-shot learning to tune all LMs, no additional pre-training, finetuning, or other forms of gradient updates were applied to any prompted model, and our ensembles were trained only on the outputs of the LMs and content of the articles and query in the training data. No external data was used to train any of the

[8] As we suspected our GPT-4 submission would likely be disqualified, we chose not to use this model in the ensemble.

submitted systems beyond the data used to pre-train the LMs.[9] We selected several LMs for initial testing. Specifically, we tested RoBERTa-large [70], which has shown high performance on natural language understanding (NLU) tasks [70], LegalBERT [71], which is pre-trained on legal data and has been a pivotal contribution in previous COLIEE competitions, Meta's OPT [72], Google's largest Flan-T5 [59], the T0 models from BigScience [60], EleutherAI's GPT-J [73], and OpenAI's ChatGPT [74] and GPT-4 [75]. We found that LegalBERT, RoBERTa, GPT-J, and OPT performed relatively poorly and hence chose not to run extensive ablations on them. Chat-GPT performed reasonably well, matching or exceeding the performance of both Flan-T5 and T0. However, because we believed that any OpenAI submission would likely be disqualified, we chose to use GPT-4 instead as our only submission using their API. However, we use results for Chat-GPT in some of our ablations reported below.

Across all LMs, we experimented with the following strategies for few-shot selection:

1 **Zero-Shot:** The model is given only the validation example without any further context.

2 **Few-Shot no TF-IDF:** The model is randomly given k shots from the training data. The number of shots varies from two to six, depending on the model.

3 **Few-Shot with TF-IDF:** Prior work has demonstrated that the choice of shots sent to an LM significantly impacts model performance [19, 76]. Therefore, choosing the shots based on some metric is important for optimizing model performance. Following prior work [77], we use similarity-based shot selection based on the cosine similarity of TF-IDF vectors. The validation example (articles + query) is embedded using a TF-IDF vector space, and the top k most similar examples are chosen and used as context. This is done for each validation example separately, and the exact order in which shots are presented is random. We use the training data articles and queries to train our TF-IDF vector space.

4 **Few-Shot Balanced with TF-IDF:** Same as above, except the shot selection always returns a balanced number of entailment and non-entailment shots to prevent the model from overfitting on one label.

5 **Few-Shot Pruned with TF-IDF:** When the TF-IDF vectors are calculated with the complete vocabulary, they are quite sparse due to the lack of overlap among terms across documents. Therefore, we also explored applying pruning to the vectors before performing cosine similarity. After building the full TF-IDF matrix, we reduced the vocabulary to X% and rebuilt it based on this smaller size, where X% is a hyperparameter. The 6 pruned models in Table 1 have top to down the hyperparameters 15%, 30%, 45%, 60%, 75%, and 90%. Note that these approaches always used the unbalanced form of TF-IDF shot selection. As we found pruning always performed worse than the unpruned unbalanced TF-IDF, we chose not to investigate this strategy further.

[9] We refer the reader to the respective papers for details on how each model was trained. Note that, at the time of publication, OpenAI has released no details on what data GPT-4 was trained on.

Besides intelligent shot selection, prior work has also found that how the prompt is structured can significantly impact downstream performance. For example, advanced prompting strategies, including chain-of-thought [21] and maieutic prompting [30], involve asking a model a series of structured questions in order to help it arrive at a correct answer. We experimented with several such approaches with our models: (1) Select Most Relevant, (2) Consider Both Sides, and (3) Self-Ask. Furthermore, we included (4) Plan and Solve prompt type in our experiments. Figure 1 shows examples of each prompt type. Each of these approaches involves asking the model to explain why the answer should be yes (or no), either asking the model to select the most relevant information from the articles to answer the query, both of which we reasoned could aid the model in choosing the correct answer or asking to devise a plan to split the task into small subtasks and solving the task according to plan to output the answer [78].

7 Results

Table 1 shows GPT-4 and Flan-T5 systems results. We compare our systems against the results reported by the Kano Laboratory, which achieved first place in the 2022 Task 4 competition [79]. Not surprisingly, our GPT-4 based submission surpasses the prior state-of-the-art by significant margins across all validation splits and achieves an overall accuracy of 83.3%, almost 20 points higher than the prior year's results. Perhaps more surprising, however, is the strong performance achieved using `flan-t5-xxl`. This model has only 11 billion parameters, far less than GPT-3 with 175 billion parameters [80].[10] While this model is still considerably larger than the prior year's submission, we emphasize that it was not directly trained on the train set. The best Flan-T5 model uses only two shots for in-context learning, which is less than 1% of the entire train set, though new shots are sampled each time. With `flan-alpaca-xxl` models we achieved a lower accuracy than `flan-t5-xxl` models. The best Flan-Alpaca model uses two shots for in-context learning. Collectively, these results demonstrate the potential of applying prompting with generative LMs to legal reasoning tasks and show that even a relatively simple prompting strategy can outperform carefully tuned systems across multiple validation datasets. Results for each of our submitted systems on the R04 test data set are given in Table 3.

For our ensembles, the best-performing brute force approach achieved 77.0% accuracy on the validation set. With support vector machines, we achieved an ensemble accuracy of 74.9%, and for random forest, an accuracy of 74.4%, as shown in Table 4. According to the results, our brute force approach achieved better accuracy on the validation set by intensively searching for the best model combinations across the ensemble set, outperforming the ensemble's performance with support vector machines and random forest models. In our opinion, the lack of more data prevented achieving higher accuracy for those models. Thus, we selected our brute

[10] Although the exact number of parameters in GPT-4 is unknown, it is likely to be on the order of hundreds of billions, given the known size of GPT-3.

force approach as our ensemble model based on the overall validation accuracy for our submission and achieved on the test set 64.4% accuracy. We think our model combination is overfitting the validation set and therefore caused the accuracy difference between the validation and test set.

7.1 Ablations

7.1.1 Shot Selection Strategy

For both GPT-4 and Flan-T5 models, we found that TF-IDF selection balanced by label achieved the highest validation accuracy (2% increase for GPT-4, 1.3% increase for Flan-T5 compared to 0-shot prompting). Using TF-IDF with an unbalanced label selection causes a slight decrease in performance for Flan-T5. This is not surprising; prior work has found that in-context learning is highly sensitive to the prompt [81–83], and not balancing labels likely causes overfitting to the majority class in the prompt. Additionally, we found that few shot prompting with randomly chosen shots caused a substantial decrease in performance. As each validation example contains a significant amount of terminology that may not appear elsewhere, it is unclear how much information an arbitrarily chosen train example will provide for determining the label for a validation example (i.e., knowledge of entailments on contract law likely provides little information for inference on a query related to the rights of the unborn). Our results show that some sort of intelligent shot selection is necessary for few-shot learning to help.

As a selection strategy, TF-IDF is a fairly simple approach that relies on syntactic overlap among documents [84]. However, information retrieval research has developed more sophisticated methods for document similarity that employ modern contextual embeddings. Such approaches include BERTScore [85], SBERT [27], and BLEURT [86], among others. We, therefore, explored using each of these approaches as the similarity method for shot selection to see if an approach based on contextual embeddings could outperform the simpler TF-IDF vectors. We use ChatGPT for this ablation,[11]. We use standard prompting (no chain-of-thought methods), five shots, and a temperature of one for all models and compute the overall accuracy of each approach across all validation splits. Results are shown in Table 5. Although all similarity-based selection methods perform better than random selection, TF-IDF achieves the best overall accuracy. One possible explanation is that the TF-IDF vectors were the only ones trained on a legal corpus (the train set), and we relied on the pre-trained embeddings for all other methods. It is conceivable that pre-training the contextual similarity metrics on a corpus of legal documents could dramatically improve the quality of the selected shots. However, doing this is impossible with just the train set as these methods require considerably more data than TF-IDF.

Finally, we found that pruning the TF-IDF vectors to select only terms with low document frequency consistently leads to worse validation accuracy. Our goal

[11] `gpt-3.5-turbo'.

behind this method was to eliminate terms that appeared across most training examples (likely stopwords) and create better vectors for document ranking. However, it appeared to have the opposite effect. If the pruning was too aggressive, all vectors could have become orthogonal if no terms overlapped across documents. This pattern was observed regardless of the pruning factor, even with minimal pruning.

7.1.2 Advanced Prompting Strategies

We investigated various more sophisticated prompting methods (details of prompt structure discussed in Sect. 6). We focus on GPT−3.5 for our analysis because, as discussed earlier, the minimal sequence length of Flan-T5 prevented it from using any chain-of-thought approach effectively. Results are shown in Table 3. We find that no prompt outperforms the "vanilla" baseline strategy. Given that legal reasoning often involves highly open-textured phrases, the space of possible explanations may be so extensive that chain-of-thought approaches cannot effectively assist a model in arriving at a correct answer, which confirms with prior research on these prompting strategies in other specialized domains [87]. In Fig. 2, we examined the validation accuracy of GPT−3.5 models with different prompt types we used in the COLIEE competition, along with the additional "plan and solve" prompt type that we incorporated for subsequent testing, given the number of shots. Table 2 shows the results. In this table, all GPT−3.5 models have a temperature value of 1. GPT−3.5 achieves with the "plan and solve" prompt type the highest validation accuracy with five shots and with the "select most relevant" prompt type with only one shot, while the highest accuracy for the "self-ask" and "vanilla" is achieved with zero shot. Among the plots, the "consider both" shows in general lower performance while the "plan and solve" prompt type shows the best performance except for three shots. In many cases of the outputs with the "plan and solve" prompt type, it is observed that the model suggests examining the relevant articles and gives an explanation of the provided articles. After the "Let's carry out the plan and solve the problem step by step." prompt, it makes a step-by-step analysis of the articles. Nevertheless, for the best prompting method in Fig. 2, the "plan and solve" with five shots, it is also observed that the model apologizes after that prompt and outputs that the articles do not address the specific scenario, it can not carry out plans or execute actions, or it does not have access to a legal database and suggests to consult to a legal expert.

The performance of "self-ask" is low if two, three, or four shots are provided. "Self-ask" has shown to be effective at assisting GPT−3.5 with composing information it already knows in a format more suitable for reasoning. However, this relies on the model's ability to answer the sub-questions it creates accurately [88]. Given the open nature of the dataset, the model was likely unsuccessful at breaking down a legal query into more digestible parts, resulting in decreased performance. As the plots act differently for each prompt type, we believe it is hard to express the accuracy given the number of shots with one general rule. But from the plots of "select most relevant" and "self-ask" cases we can derive that between one shot and five shots, the model starts to degrade when the shot number increases. We believe that with legal context this degradation is more apparent.

Table 2 Summary of prompting results

Model	Prompt type	Shots	Similarity method	Ensemble size	H30	R01	R02	R03	Overall accuracy
GPT−3.5	Vanilla	0	None	1	65.714%	63.063%	74.074%	71.560%	68.464%
GPT−3.5	Vanilla	3	TF-IDF	1	68.571%	66.667%	64.198%	69.725%	67.385%
GPT−3.5	Vanilla	5	TF-IDF	1	68.571%	62.162%	70.370%	66.055%	66.307%
GPT−3.5	Select most relevant	0	None	1	61.429%	60.360%	62.963%	67.890%	63.342%
GPT−3.5	Select most relevant	5	None	1	65.714%	63.964%	72.840%	72.477%	68.733%
GPT−3.5	Select most relevant	5	TF-IDF	1	70.000%	63.964%	70.370%	66.055%	67.116%
GPT−3.5	Consider both	0	None	1	54.286%	59.459%	55.556%	60.550%	57.951%
GPT−3.5	Consider both	5	None	1	67.143%	51.351%	48.148%	55.963%	54.987%
GPT−3.5	Consider both	5	TF-IDF	1	47.143%	54.955%	49.383%	56.881%	52.830%
GPT−3.5	Self-Ask	0	None	1	65.714%	65.766%	60.494%	69.725%	65.768%
GPT−3.5	Self-Ask	5	None	1	68.571%	60.360%	65.432%	69.725%	65.768%
GPT−3.5	Self-Ask	5	TF-IDF	1	67.143%	62.162%	58.025%	68.807%	64.151%
GPT−3.5	Self-Ask	0	None	1	70.000%	65.800%	66.700%	75.200%	69.551%
GPT−3.5	Self-Ask	3	TF-IDF	5	71.000%	61.000%	68.000%	69.000%	66.765%
GPT−3.5	Self-Ask	1	TF-IDF	5	66.000%	63.000%	63.000%	68.000%	65.035%
GPT−3.5	Plan and solve	0	None	1	64.286%	65.766%	66.667%	77.982%	69.272%
GPT−3.5	Plan and solve	1	TF-IDF	1	71.429%	67.568%	70.370%	71.560%	70.081%
GPT−3.5	Plan and solve	2	TF-IDF	1	67.143%	64.865%	77.778%	73.394%	70.620%
GPT−3.5	Plan and solve	5	TF-IDF	1	68.571%	64.865%	77.778%	74.312%	71.159%

The section shows ablations for the vanilla and advanced prompting strategies with GPT−3.5

Table 3 Test accuracy using different models

System	Test accuracy
Flan-T5	65.35%
GPT-4	**81.19%**
Ensemble	64.36%

Table 4 Overall accuracy across validation splits using different ensemble models

Ensemble model	Overall accuracy
Brute Force	**77.0%**
SVM	74.9%
Random Forest	74.4%

Table 5 Overall accuracy across validation splits using different contextual similarity metrics

Shot selection strategy	Overall accuracy
SBERT	69.542%
BERTScore	66.307%
BLEURT	68.194%
TF-IDF	**72.237%**
Random	66.038%

All results use Chat-GPT, with standard prompting, five shots, and a temperature of one

7.1.3 Choice of Temperature

LMs sample words from their vocabulary and choose which word to predict next—given a sequence—from this sample. Temperature is a hyperparameter (varying between 0 and 1) that controls the randomness of this choice. Lower values decrease randomness, and higher ones increase it. We experimented with multiple temperature values from the set 0, 0.25, 0.5, 0.75, and 1.0. However, we found that different values did not significantly affect the performance of GPT-4. Building on this, we also experimented with temperature-based ensembles where each prediction (vote) came from a GPT-4 model with a different temperature value. We used the same set of temperatures (five in total), and the majority vote was chosen as the final prediction of the ensemble. We found that this approach provided less than 0.5% improvement in accuracy and decided not to replicate it for the unbalanced GPT-4 TFIDF. Additionally, the added cost of testing GPT-4 resulted in us not performing the test for practical reasons. Results from these trials are shown in the first half of Table 1, with the rows containing multiple values indicating temperature-based ensembles.

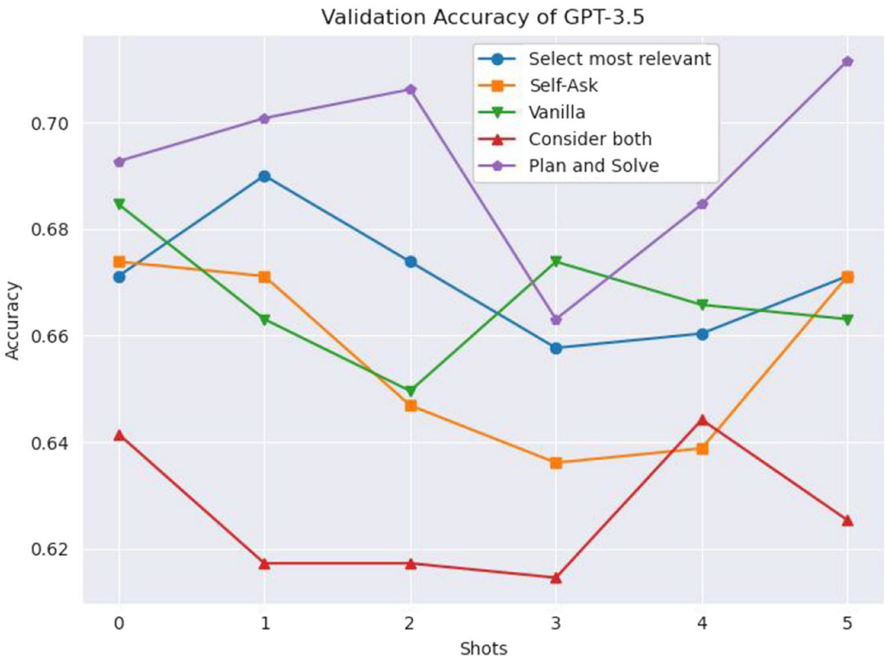


Fig. 2 Validation accuracy of GPT‒3.5 given the number of shots for different prompt types. TF-IDF is applied as the similarity metric

7.2 Error Analysis

Exploring False Predictions. We perform error analysis on our systems, focusing on GPT-4 and Flan-T5. Table 6 lists the false positive rate (FPR), false negative rate (FNR), true positive rate (TPR), and true negative rate (TNR) of the best-performing Flan-T5 and GPT-4 models on the validation sets. We find that models tend to overpredict 'Y', which leads to a much higher FPR than FNR. Scaling up models thus does not appear to eliminate this problem. The exact cause of this behavior is unclear. The validation splits are somewhat unbalanced, but not by a sufficient degree to cause such an imbalance in error rates. Our shot selection also accounted for this by ensuring the labels were balanced.

In Fig. 3, we graph a histogram of the proportion of non-ensemble models that get each validation example correct. The goal is to determine if there are examples that are consistently difficult to solve. We find that this is the case and that the distribution of accuracy scores is roughly bi-modal. Most examples appear relatively easy for our models; around 60% are correctly predicted by 80% or more of the trials. However, there is a significant fraction (roughly 80 examples) on which fewer than 30% of models get the correct answer.

Notational Conventions. Notational conventions are used in legal and contractual documents to communicate information efficiently and clearly. Those symbols and abbreviations come in queries of our validation set in the form of letters. Three

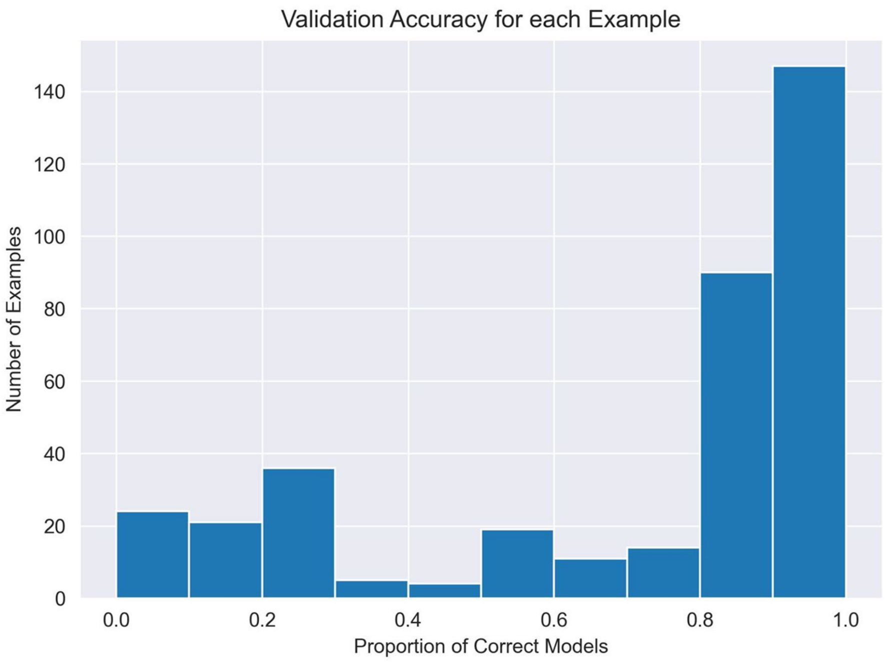


Fig. 3 Histogram of model accuracy for all examples in the validation set. For each example, we plot the proportion of models that get that example correct (x-axis). This is done across all validation examples, using only the Huggingface models

Table 6 False positive rate (FPR), false negative rate (FNR), true positive rate (TPR), and true negative rate (TNR) for the best-performing GPT-4 and Flan-T5 runs

	Flan-T5	GPT-4
Similarity method	TF-IDF Balanced	TF-IDF
Shots	2	5
FPR	33.889%	24.444%
FNR	15.707%	9.424%
TPR	84.293%	90.576%
TNR	66.111%	75.556%

Table 7 The rate of true and false predictions for Flan-T5 given samples with or without notational conventions

Predictions	Including conventions	
	Yes	No
True	19.677%	55.795%
False	10.782%	13.747%

samples are shown in Table 8. By first converting the query text to lower case and marking the samples with the following abbreviations [“b”, “(b)”, “(a)”, “c”, “(c)”, “x”, “f”] we were able to separate all samples with notational conventions in the

Table 8 Sample queries with notational conventions in the validation set

Queries with Notational Conventions
When A was on a long-term business trip and was absent, a part of a hedge of A's house collapsed due to a strong wind. Afterward, B who has a house on a land next to A's house performed an act for A without any obligation. If B has requested a gardener to repair the hedge, B may not demand the payment of the unpaid remuneration to the gardener from A
Claim X is A's claim against B, and claim Y is B's claim against A. If F, A's creditor, attaches claim X after the due date for claim X has arrived, B may duly assert against F a set-off based on claim Y against claim X even before the due date for claim Y, which was acquired before the attachment
If beneficiary (B) sells and delivers X to a subsequent acquirer (D), who knows that the gifts of X will prejudice the obligee of A, C may claim to D to rescind the sale between B and D.

validation set from the others as samples consist of many different abbreviations. We explored the queries in the validation set to address the effects of the notational conventions on the false predictions. For `flan-t5-xxl` with balanced TF-IDF and two shots, the mean number of tokens for correct predictions was 40.71, while for false predictions the number was 47.44. The decrease in prediction accuracy with higher token numbers might be caused by the queries with notational conventions because when the queries with notational conventions are filtered out the mean number of tokens for correct predictions is 25.95 while that number is for false predictions 19.08. Table 7 shows the rate of true and false predictions given notational conventions. The queries with notational conventions in the validation set that are predicted correctly are well below the prediction accuracy of the model. Therefore, we believe that the queries with notational conventions decrease the model's generalization performance.

Perplexity Analysis. The perplexity is the inverse of the mean probability of each word in the test set. Perplexity quantifies how well LMs predict the next word and models with lower perplexity indicate a better understanding of the context and tend to have fewer errors predicting the sequence of tokens. Thus, we conducted a perplexity analysis for the model outputs given the article and query input [89]. We calculated the mean perplexity score of all samples for correct and false predictions separately. The perplexity score was for the correctly predicted queries 15.07 while for false predictions the model perplexity was 372.42. When the perplexity scores are calculated without the queries with notational conventions the perplexity score for correctly predicted queries is 4.21 while the score for falsely predicted queries is 556.87. Thus, by looking at perplexity scores for correct predictions, the model was more confident for predictions without notational conventions. This agrees with our older findings that the model struggles in queries with notational conventions.

8 Conclusion

We have demonstrated that prompting methods using LMs can achieve competitive performance on legal entailment tasks and even outperform carefully engineered systems.

Though our system performed quite well overall, several avenues remain for improvement.

We attempted to supplement the provided data using similarly structured rule sets to capture more robust open-textured terms. In the "consider both" prompt type, we attempted to create arguments for entailment and not entailment by the rule. Including this data in training decreased model performance and was not included in any of the submitted systems. However, broader rule sets may aid future work on few-shot prompting for legal entailment. Instruction tuning our Flan-T5 system on legal entailments or other legal data is also a fruitful direction for future work; we explored this option briefly but found that it required too much computing resources and training time to be viable. Nevertheless, directly training the model on this data might improve performance and generalization.

An earlier version that we experimented with used a chain-of-thought prompting approach [21], which asked the model to output explanations for why it thought the answer was 'Y' or 'N'. In our experience with this approach, open-textured terms ended up being the primary problem: without further context (which may have come from additional articles that were not included in the set of provided articles *A*), the LM did not know how to interpret certain terms of art or jargon that appeared in the articles or query. This is consistent with our view of interpretive reasoning, which suggests that properly interpreting open-textured legal terms often requires examples of how the term has been interpreted in the past. However, it should be noted that it is not clear whether explanations given through chain-of-thought prompting actually provide insight into how the language model came up with the answers, or whether it was a sort of post hoc rationalization.

Our deeper analysis of the articles and queries shows that a considerable extent of the samples consists of notational conventions. We believe that a prompting approach focused additionally on the notational conventions would increase the generalization performance. We see that LMs struggle to interpret this kind of query in a legal context. We believe that those queries might be out-of-distribution for this kind of legal reasoning task and the different prompting approaches have only limited effect to help the model generalize this task. Our analysis of different prompt types with the number of shots shows that increasing the shot number does not necessarily bring a better generalization performance for the model. This might limit the effectiveness of the prompting methods to increase the model performance. Therefore developing prompting strategies for legal tasks, that work complementary with few shots would be a good direction for future work.

Although our approach did not outperform other submissions on the test set, it was a successful endeavor overall. Our instruction-tuned transformers approach has the 7. place among all models in Task 4 of the competition and our ensemble approach achieves the 9. place [46]. As stated in this report's introduction, automated reasoning over rules is extremely important for the future of human interaction with AI, and competitions like COLIEE allow us to better understand the strengths and limitations of current natural language processing tools toward that goal. However, replicability is necessary to improve the broader impacts of the competition's efforts. Thus, in the future, we strongly recommend that certain measures be taken to ensure the integrity of the competition and to maximize its impact on the broader research community.

We recommend that all entrants require the release of full source code. The possibility of unintentionally selecting models, parameters, and hyperparameters that maximize performance on the test set is too great (even though the competition organizers explicitly disallowed the use of the test set for any of these). If code release is too limiting, a full description of methods, algorithms, parameters, and hyperparameters should be released before finalizing competition rankings in time for independent replication. This also allows for confirmation that the results listed in the final competition rankings were not simply due to luck—in our experience, many of the language models we used had non-deterministic output, and this required multiple runs in order to confirm that extremely good (or extremely bad) results were not simply flukes. With this spirit in mind, we publicly release our full source code for this competition.[12]

Declarations

Conflict of Interest The authors declare that they have no competing interests.

References

1. Hart, H. (1961). *The concept of law*. Clarendon Press.
2. Franklin, J. (2012). How much of commonsense and legal reasoning is formalizable? *A Review of Conceptual Obstacles Law, Probability and Risk, 11*(2–3), 225.
3. Prakken, H. (2017). On the problem of making autonomous vehicles conform to traffic law. *Artificial Intelligence and Law, 25*(3), 341.
4. Lawless, W. F., Mittu, R., & Sofge, D. A. (Eds.). (2020). *Human-machine shared contexts*. NY: Academic Press.
5. Licato. J., Marji, Z., & Abraham, S. (2019). Proceedings of the AAAI 2019 Fall Symposium on Human-Centered AI, Arlington, VA.
6. Licatom, J., & Marji, Z. (2018). Proceedings of the 2018 International Conference on Robot Ethics and Standards, ICRES.
7. Waismann, F. (1965). *The principles of linguistic philosophy*. St. Martins Press.
8. Licato J. (2021). How should AI interpret rules? A defense of minimally defeasible interpretive argumentation arXiv e-prints.
9. Vecht, J. J. (2020). Open texture clarified. *Inquiry*. https://doi.org/10.1080/0020174X.2020.1787222
10. Licato, J., Fields, L., & Hollis, B. (2023). Proceedings of The 36th International Florida Artificial Intelligence Research Society Conference (FLAIRS-34), AAAI Press.
11. Fields, L., & Licato, J. (2023) Proceedings of the 36th International Florida Artificial Intelligence Research Society Conference (FLAIRS-34), AAAI.
12. Licato, J. (2022). Proceedings of the AAAI 2022 Spring Workshop on "Ethical Computing: Metrics for Measuring AI's Proficiency and Competency for Ethical Reasoning".
13. Licato, J. (2022). Proceedings of the 2022 Advances on Societal Digital Transformation (DIGITAL) Special Track on Explainable AI in Societal Games (XAISG).
14. Sartor. G., Walton, D., Macagno, F., & Rotolo, A. (2014). Legal Knowledge and Information Systems. In: Proceedings of JURIX 14, pp. 21–28.

[12] https://github.com/Advancing-Machine-Human-Reasoning-Lab/COLIEE-2023-Task4.

15. Bongiovanni, G., Postema, G., Rotolo, A., Sartor, G., Valentini, C., & Walton, D. (Eds.). (2018). *Handbook of legal reasoning and argumentation* (pp. 519–560). Netherlands, Dordrecht: Springer. https://doi.org/10.1007/978-90-481-9452-0_18
16. Walton, D., Macagno, F., & Sartor, G. (2021). *Statutory interpretation: Pragmatics and argumentation*. Cambridge University Press.
17. Araszkiewicz, M. (2021). Critical questions to argumentation schemes in statutory interpretation. *Journal of Applied Logics - IfCoLog Journal of Logics and Their Applications, 8*(1), 291–320.
18. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., & Brunskill, E., et al. (2021). On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258
19. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877.
20. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train,prompt, and predict: A systematic survey of prompting methods in natural language processing arXiv:abs/2107.13586
21. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E.H., Le Q., & Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models CoRR **abs/2201.11903**. https://arxiv.org/abs/2201.11903
22. Ye, X., & Durrett, G. (2023). Explanation selection using unlabeled data for in-context learning, arXiv preprint arXiv:2302.04813
23. Rubin, O., Herzig, J., & Berant, J. (2022). Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies , pp. 2655–2671.
24. Song, C., Cai, F., Wang, M., Zheng, J., & Shao, T. (2023). TaxonPrompt: Taxonomy-aware curriculum prompt learning for few-shot event classification. *Knowledge-Based Systems, 264*, 110290. https://doi.org/10.1016/j.knosys.2023.110290
25. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W. X., Dong, D., Wu, H., & Wang, H. (2021). Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Association for Computational Linguistics, Online). pp. 5835–5847. https://doi.org/10.18653/v1/2021.naacl-main.466
26. Wang, S., Xu, Y., Fang, Y., Liu, Y., Sun, S., Xu, R., Zhu, C., & Zeng, M. (2022). Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Dublin, Ireland. pp. 3170–3179. https://doi.org/10.18653/v1/2022.acl-long.226
27. Reimers, N., & Gurevych, I. (2019). in Proceedings of the 2019 Conference on Empirical Methods. In S. Padó & R. Huang (Eds.), *Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 3982–3992). Hong Kong: Association for Computational Linguistics.
28. Lyu, Q., Havaldar, S., Stein, A., Zhang, L., Rao, D., Wong, E., Apidianaki, M., & Callison-Burch, C. (2023). Faithful chain-of-thought reasoning, arXiv preprint arXiv:2301.13379
29. Zelikman, E., Wu, Y., Mu, J., & Goodman, N. (2022). The flan collection: Designing data and methods for effective instruction tuning. *Advances in Neural Information Processing Systems, 35*, 15476.
30. Jung, J., Qin, L., Welleck, S., Brahman, F., Bhagavatula, C., Bras, R. L., & Choi, Y. (2022). Maieutic prompting: Logically consistent reasoning with recursive explanations arXiv preprint arXiv:2205.11822
31. Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M. , Abid, A., Fisch, A., & Brown, A. R. A., Santoro, A. Gupta, A. Garriga-Alonso, et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models arXiv preprint arXiv:2206.04615
32. Yu, F., Quartey, L., & Schilder, F. (2023) Findings of the Association for Computational Linguistics: ACL 2023 , pp. 13582–13596.
33. Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. J. (2022). Large language models are human-level prompt engineers, arXiv preprint arXiv:2211.01910
34. Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805
35. Nguyen, H. T., Vuong, H. Y. T., Nguyen, P. M., Dang, B. T., Bui, Q. M., Vu, S. T., & Nguyen, C. M., Tran, V., Satoh, K. Nguyen, M. L. (2020) Jnlp team: Deep learning for legal processing in coliee, arXiv preprint arXiv:2011.08071

36. He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention, arXiv preprint arXiv:2006.03654
37. Lin, J., Nogueira, R., & Yates, A. (2022). *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.
38. Rosa, G. M., Rodrigues, R. C., de Alencar Lotufo, R., & Nogueira, R. (2021). Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, pp. 295–300.
39. Shao, Y., Mao, J., Liu, Y., Ma, W. , Satoh, K., Zhang, M., & Ma, S. (2020). IJCAI, pp. 3501–3507.
40. Shao, Y., Liu, B., Mao, J., Liu, Y., Zhang, M., & Ma, S. (2020). Thuir@ coliee-2020: leveraging semantic understanding and exact matching for legal case retrieval and entailment. Corr arXiv:2012.13102
41. Rosa, G.M. , Rodrigues, R.C. , Lotufo, R., & Nogueira, R. (2021). Yes, bm25 is a strong baseline for legal case retrieval, arXiv preprint arXiv:2105.05686
42. Althammer, S., Askari, A. , Verberne, S., & Hanbury, A. (2021). Proceedings of the eighth international competition on legal information extraction/entailment (COLIEE 2021), pp. 8–14.
43. Askari, A., Peikos,G., Pasi, G., & Verberne, S. (2022). Leibi@ coliee 2022: Aggregating tuned lexical models with a cluster-driven bert-based model for case law retrieval, arXiv preprint arXiv:2205.13351
44. Savelka, J., Ashley, K. D., Gray, M. A., Westermann, H., & Xu, H. (2023). Can gpt-4 support analysis of textual data in tasks requiring highly specialized domain expertise? arXiv preprint arXiv:2306.13906
45. Savelka, J., Ashley, K. D., Gray, M. A., Westermann, H., & Xu, H. (2023). Explaining legal concepts with augmented large language models, arXiv preprint arXiv:2306.09525
46. Goebel, R., Kano, Y., Kim, M. Y., Rabelo, J., Satoh, K., & Yoshioka, M. (2023). Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, pp. 472–480.
47. Berryessa, C. M., Dror, I. E., & McCormack, C. J. B. (2023). Prosecuting from the bench? Examining sources of pro-prosecution bias in judges. *Legal and Criminal Psychology, 28*(1), 1.
48. Liu, J. Z., & Li, X. (2019). Legal techniques for rationalizing biased judicial decisions: Evidence from experiments with real judges. *Journal of Empirical Legal Studies, 16*(3), 630.
49. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2022). A survey on bias and fairness in machine learning. *ACM Computing Surveys, 54*(6), 1. https://doi.org/10.1145/3457607
50. Wachter, S., Mittelstadt, B., & Russell, C. (2020). Bias preservation in machine learning: The legality of fairness metrics under EU non-discrimination laws. *West Virginia Law Review, 123*, 735.
51. Yeung, D., Khan, I., Kalra, N., Osoba, O. A. (2021). Identifying systemic bias in the acquisition of machine learning decision aids for law enforcement applications. RAND Corporation, Santa Monica, CA. https://doi.org/10.7249/PEA862-1
52. Costantini, S., & Lanzarone, G. A. (1995). Explanation-based interpretation of open-textured concepts in logical models of legislation. *Artificial Intelligence and Law, 3*, 191. https://doi.org/10.1007/BF00872530
53. Ashley, K. D., & Walker, V. R. (2013) ICAIL ’13: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. Association for Comuting Machinery, pp. 176–180. https://doi.org/10.1145/2514601.2514622
54. Bayamlıoğlu, E., Leenes, R. E. (2018) Data-driven decision-making and the ‘rule of law’ Tilburg Law School Research Paper.
55. Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., Tow, J., Rush, A. M., Biderman, S., Webson, A., Ammanamanchi, P. S., Wang, T., Sagot, B., Muennighoff, N., del Moral, A. V., Ruwase, O., Bawden, R. , Bekman, S., McMillan-Major, A. , Beltagy, I., H. Nguyen, L. Saulnier, S. Tan, P.O. Suarez, V. Sanh, H. Laurenčon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A.F. Aji, A. Alfassy, A. Rogers, A.K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, D. van Strien, D.I. Adelani, D. Radev, E.G. Ponferrada, E. Levkovizh, E. Kim, E.B. Natan, F.D. Toni, G. Dupont, G. Kruszewski, G. Pistilli, H. Elsahar, H. Benyamina, H. Tran, I. Yu, I. Abdulmumin, I. Johnson, I. Gonzalez-Dios, J. de la Rosa, J. Chim, J. Dodge, J. Zhu, J. Chang, J. Frohberg, J. Tobing, J. Bhattacharjee, K. Almubarak, K. Chen, K. Lo, L.V. Werra, L. Weber, L. Phan, L.B. allal, L. Tanguy, M. Dey, M.R. Muñoz, M. Masoud, M. Grandury, M. Šaško, M. Huang, M. Coavoux, M. Singh, M.T.J. Jiang, M.C. Vu, M.A. Jauhar, M. Ghaleb, N. Subramani, N. Kassner, N. Khamis, O. Nguyen, O. Espejel, O. de Gibert, P. Villegas, P. Henderson, P. Colombo, P. Amuok, Q. Lhoest, R. Harliman, R. Bommasani, R.L. López, R. Ribeiro, S. Osei, S. Pyysalo, S. Nagel, S. Bose, S.H. Muhammad, S. Sharma, S. Longpre, S. Nikpoor, S. Silberberg, S. Pai, S. Zink, T.T. Torrent, T. Schick, T. Thrush,

V. Danchev, V. Nikoulina, V. Laippala, V. Lepercq, V. Prabhu, Z. Alyafeai, Z. Talat, A. Raja, B. Heinzerling, C. Si, D.E. Taşar, E. Salesky, S.J. Mielke, W.Y. Lee, A. Sharma, A. Santilli, A. Chaffin, A. Stiegler, D. Datta, E. Szczechla, G. Chhablani, H. Wang, H. Pandey, H. Strobelt, J.A. Fries, J. Rozen, L. Gao, L. Sutawika, M.S. Bari, M.S. Al-shaibani, M. Manica, N. Nayak, R. Teehan, S. Albanie, S. Shen, S. Ben-David, S.H. Bach, T. Kim, T. Bers, T. Fevry, T. Neeraj, U. Thakker, V. Raunak, X. Tang, Z.X. Yong, Z. Sun, S. Brody, Y. Uri, H. Tojarieh, A. Roberts, H.W. Chung, J. Tae, J. Phang, O. Press, C. Li, D. Narayanan, H. Bourfoune, J. Casper, J. Rasley, M. Ryabinin, M. Mishra, M. Zhang, M. Shoeybi, M. Peyrounette, N. Patry, N. Tazi, O. Sanseviero, P. von Platen, P. Cornette, P.F. Lavallée, R. Lacroix, S. Rajbhandari, S. Gandhi, S. Smith, S. Requena, S. Patil, T. Dettmers, A. Baruwa, A. Singh, A. Cheveleva, A.L. Ligozat, A. Subramonian, A. Névéol, C. Lovering, D. Garrette, D. Tunuguntla, E. Reiter, E. Taktasheva, E. Voloshina, E. Bogdanov, G.I. Winata, H. Schoelkopf, J.C. Kalo, J. Novikova, J.Z. Forde, J. Clive, J. Kasai, K. Kawamura, L. Hazan, M. Carpuat, M. Clinciu, N. Kim, N. Cheng, O. Serikov, O. Antverg, O. van der Wal, R. Zhang, R. Zhang, S. Gehrmann, S. Mirkin, S. Pais, T. Shavrina, T. Scialom, T. Yun, T. Limisiewicz, V. Rieser, V. Protasov, V. Mikhailov, Y. Pruksachatkun, Y. Belinkov, Z. Bamberger, Z. Kasner, A. Rueda, A. Pestana, A. Feizpour, A. Khan, A. Faranak, A. Santos, A. Hevia, A. Unldreaj, A. Aghagol, A. Abdollahi, A. Tammour, A. HajiHosseini, B. Behroozi, B. Ajibade, B. Saxena, C.M. Ferrandis, D. Contractor, D. Lansky, D. David, D. Kiela, D.A. Nguyen, E. Tan, E. Baylor, E. Ozoani, F. Mirza, F. Ononiwu, H. Rezanejad, H. Jones, I. Bhattacharya, I. Solaiman, I. Sedenko, I. Nejadgholi, J. Passmore, J. Seltzer, J.B. Sanz, L. Dutra, M. Samagaio, M. Elbadri, M. Mieskes, M. Gerchick, M. Akinlolu, M. McKenna, M. Qiu, M. Ghauri, M. Burynok, N. Abrar, N. Rajani, N. Elkott, N. Fahmy, O. Samuel, R. An, R. Kromann, R. Hao, S. Alizadeh, S. Shubber, S. Wang, S. Roy, S. Viguier, T. Le, T. Oyebade, T. Le, Y. Yang, Z. Nguyen, A.R. Kashyap, A. Palasciano, A. Callahan, A. Shukla, A. Miranda-Escalada, A. Singh, B. Beilharz, B. Wang, C. Brito, C. Zhou, C. Jain, C. Xu, C. Fourrier, D.L. Periñán, D. Molano, D. Yu, E. Manjavacas, F. Barth, F. Fuhrimann, G. Altay, G. Bayrak, G. Burns, H.U. Vrabec, I. Bello, I. Dash, J. Kang, J. Giorgi, J. Golde, J.D. Posada, K.R. Sivaraman, L. Bulchandani, L. Liu, L. Shinzato, M.H. de Bykhovetz, M. Takeuchi, M. Pámies, M.A. Castillo, M. Nezhurina, M. Sänger, M. Samwald, M. Cullan, M. Weinberg, M.D. Wolf, M. Mihaljcic, M. Liu, M. Freidank, M. Kang, N. Seelam, N. Dahlberg, N.M. Broad, N. Muellner, P. Fung, P. Haller, R. Chandrasekhar, R. Eisenberg, R. Martin, R. Canalli, R. Su, R. Su, S. Cahyawijaya, S. Garda, S.S. Deshmukh, S. Mishra, S. Kiblawi, S. Ott, S. Sang-aroonsiri, S. Kumar, S. Schweter, S. Bharati, T. Laud, T. Gigant, Kainuma, T., Kusa, W., Labrak, Y., Bajaj, Y. S., Venkatraman, Y., Xu, Y., Xu, Y., Xu, Y., Tan, Z., Xie, Z., Ye, Z., Bras, M., Belkada, Y., Wolf, T. (2023). Bloom: A 176b-parameter open-access multilingual language model.

56. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama,K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L. , Simens, M., Askell, A., Welinder, P., Christiano, P. J., Leike, R. Lowe, R. (2022). Training language models to follow instructions with human feedback.
57. Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems **30**
58. Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2021). Finetuned language models are zero-shot learners, arXiv preprint arXiv:2109.01652
59. Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models, arXiv preprint arXiv:2210.11416
60. Sanh, V., Webson, A., Raffel, C., Bach, S.H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E. , Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T. J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Biderman, S., Gao, L., Bers, T., Wolf, T., Rush, A.M. (2021). Multitask prompted training enables zero-shot task generalization.
61. Chia, Y. K., Hong, P., Bing, L., Poria, S. (2023). Instructeval: Towards holistic evaluation of instruction-tuned large language models, arXiv preprint arXiv:2306.04757
62. Wolf, T., Debut, L., Sanh, V. , Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf R.,, Funtowicz, M., Davison, J., Shleifer , S., von Platen, P., Ma, C. Jernite, Y., Plu, J., Xu, C., Scao T. L, Gugger, S., Drame, M. , Lhoest, Q., Rush, A. M. (2020) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Online, pp. 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

63. Dietterich, T. G. (2000). Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, Proceedings 1. Springer. pp. 1–15.
64. Abbas, A., & Deny, S. (2022). Progress and limitations of deep networks to recognize objects in unusual poses.
65. Zhou, K., Yang, J., Loy, C. C, & Liu, Z. (2022). Learning to prompt for vision-language models.
66. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273.
67. Ho, T. K. (1995). Proceedings of 3rd international conference on document analysis and recognition, vol. 1. IEEE. pp. 278–282.
68. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825.
69. Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Chapman and Hall/CRC.
70. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M. , Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., (2019). Scikit-learn: Machine learning in Python, Journal of Machine Learning Res, CoRR **abs/1907.11692**http://arxiv.org/abs/1907.11692
71. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I. (2020). Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, pp. 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261
72. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., Zettlemoyer, L. (2022). Opt: Open pre-trained transformer language models.
73. Wang, B., Komatsuzaki, A. (2021). GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax
74. OpenAI (2022). Introducing chatgpt. https://openai.com/blog/chatgpt
75. OpenAI (2023) ArXiv, https://arxiv.org/pdf/2303.08774.pdf
76. Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.) (2022). Advances in Neural Information Processing Systems, vol. 35. Curran Associates. pp. 22199–22213. https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf
77. Lu, J., Shen, J., Xiong, B., Ma, W., Staab, S., Yang, C. (2023). Hiprompt: Few-shot biomedical knowledge fusion via hierarchy-oriented prompting, arXiv preprint arXiv:2304.05973
78. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K. W., Lim, E. P. (2023). lan-and-solve prompting: Improving zero-shot chain-ofthought reasoning by large language models, arXiv preprint arXiv:2305.04091
79. Takama, Y., Yada, K., Satoh, K., & Arai, S. (Eds.). (2023). *New frontiers in artificial intelligence* (pp. 51–67). Cham: Springer Nature Switzerland.
80. Floridi, L., & Chiriatti, M. (2020). Its nature, scope, limits, and consequences. *Minds and Machines, 30*, 681.
81. Chen, Y., Zhao, C., Yu, Z., McKeown, K., He, H. (2023). On the relation between sensitivity and accuracy in in-context learning.
82. Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J. W. (Eds.). (2021). Advances in Neural Information Processing Systems.
83. Zhao, Z., Wallace, E., Feng, S., Klein, D., Singh, S. (2021) Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 139. In: Meila, M., Zhang, T. (Eds.). Proceedings of Machine Learning Research (PMLR), pp. 12697–12706. https://proceedings.mlr.press/v139/zhao21c.html
84. Leskovec, J., Rajaraman, A., & Ullman, J. (2014). *Mining of massive datasets* (3rd ed.). Stanford University.
85. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., Artzi, Y. (2020) International Conference on Learning Representations.
86. Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.) (2020). Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics. pp. 7881–7892. https://doi.org/10.18653/v1/2020.acl-main.704
87. Liévin, V., Hother, C. E., Winther, O. (2023) Can large language models reason about medical questions?

88. Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., Lewis, M. (2022). Measuring and narrowing the compositionality gap in language models, arXiv preprint arXiv:2210.03350
89. Chen, S. F., Beeferman, D., Rosenfeld, R. (1998). Evaluation metrics for language models.