Questions about Contracts: Prompt Templates for Structured Answer Generation

Adam Roegiest, Radha Chitta, Jonathan Donnelly,
Maya Lash, Alexandra Vtyurina and **François Longtin**
Zuva Inc., Toronto, Ontario, Canada
{adam,radha.chitta,jonny.donnelly,maya,sasha, francois}@zuva.ai

Abstract

Finding the answers to legal questions about specific clauses in contracts is an important analysis in many legal workflows (e.g., understanding market trends, due diligence, risk mitigation) but more important is being able to do this at scale. In this paper, we present an examination of using large language models to produce (partially) structured answers to legal questions; primarily in the form of multiple choice and multiple select. We first show that traditional semantic matching is unable to perform this task at acceptable accuracy and then show how question specific prompts can achieve reasonable accuracy across a range of generative models. Finally, we show that much of this effectiveness can be maintained when generalized prompt templates are used rather than question specific ones.

1 Introduction

Contracts and other legal agreements are complex documents that specify in a myriad of ways the obligations, restrictions, and other covenants between two or more parties. Understanding such requirements for a single contract is a task that any lawyer could do if the need were to arise (e.g., whether a tenant can sublet an apartment), especially if there exist methods to automatically extract such clauses from documents (Roegiest et al., 2018; Leivaditi et al., 2020). However, such a task becomes hard to scale when reviewing hundreds to thousands of documents in an efficient manner for different needs (e.g., determining market trends, due diligence) especially when there is a desire for granular answers rather than clauses (e.g., amounts of liability). Granular answers to questions allow lawyers and other legal professionals the ability to triage the documents they need to read in order of risk, importance, or other criteria (e.g., first reviewing agreements that their client has provided unlimited liability coverage or most favoured nations guarantees). Finding a solution to allow rapid triage and review of documents for such granular answers have yet to be solved to our knowledge and we show progress towards that goal in this work.

The advent of and the increasing popularity of Generative AI and applications to "chat" with one's documents (e.g., ChatPDF[1], docGPT[2], CaseText[3]), which allows individuals to interact in a "natural" way with their documents and may facilitate a better understanding of legal obligations and restrictions, has complicated matters further due to the appeal of natural language interaction. This interaction can be convenient on a handful of documents but we believe that this approach also does not scale since "chatting" with a document means getting a human-like response that may be difficult to use in automated workflows without building complex post-processing systems (e.g., rules on how to respond to the various outputs). But added complexity means that further work is required either on the part of the end user or system builder to ensure that post processing can and does occur for outputs to be used as part of other pipelines. With these ideas in mind, this work investigates how Generative AI can help generate structured and partially structured answers to questions about clauses in legal documents in a manner that is usable in automated workflows and data pipelines.

To facilitate this process, we assume that we have a set (or sets) of clauses (potentially using ML to extract them) that a user would like to ask questions about in bulk with either predefined fixed options (i.e., multiple choice or multiple select) or text with a specific structure (e.g., lists of items, entities such as duration). Such outputs can facilitate the creation of automated workflows that can easily take action on the results of the questions without resorting to the complex understanding

[1] https://www.chatpdf.com/
[2] https://github.com/cesarhuret/docGPT
[3] https://casetext.com/

Proceedings of the Natural Legal Language Processing Workshop 2023, pages 62–72
December 7, 2023 ©2023 Association for Computational Linguistics

of a generated answer or manual human review except at predefined points in the workflow (e.g., after documents have been ranked according to the risk associated with different questions and their answers). While such outputs are perhaps less compelling than having a "conversation" with a document, we believe that this approach scales better to large document collections, workflow automation, and general analysis of trends in contracts. Moreover, we show that more natural prompts can very easily exhibit consistency and reliability issues when examining the responses generated in a "chat" interaction paradigm (Section 3).

It is tempting to *not* use LLMs to directly generate answers since there persist issues around hallucination (Ji et al., 2023) and a lack of clarity around whether such models truly reason (Valmeekam et al., 2023) and instead use more traditional unsupervised techniques (e.g., embedding similarity). We show that such techniques are not well suited to this task, especially in light of how straightforward it is to get much better results with generative techniques (Section 5.1). The downside to the generative approach is that it requires prompting engineering, a new and popular area of research (Liu et al., 2023; Reynolds and McDonell, 2021; Zhou et al., 2022; White et al., 2023) and book publication[4], but it is not a task that we expect lawyers or other legal professionals actually *want* to do on a regular basis. That is, we might expect that a lawyer would want to "chat" with a set of documents but they do not want to go to the extreme efforts to constrain outputs to a particularly useful format via prompt engineering.

Our end goal is to find and use prompts that provide consistent answers for the same clause and reliable answers across clauses (Section 4). In particular, we are interested in finding reusable templates that allow end users to "fill in answer options" or "specify the structure of outputs" but not need to worry about overall prompt engineering (Section 5.3). The ultimate benefit to this approach is that users of such a system would only need to worry about their area of domain expertise (i.e., determining appropriate answers) and that system builders would not need to foresee all possible different combinations of options or build bespoke prompts for every lawyer-posed question.

[4]With over 500 English language books dedicated to the topic on Amazon as of April 29, 2022.

2 Background

2.1 Large Language models

Generative models such as PaLM2[5], Llama[6], and GPT-n series[7] are self-supervised models which learn to predict the next token in a sequence of tokens (Radford et al., 2018; Touvron et al., 2023a,b). These *large* language models have billions to trillions of parameters and are pre-trained on massive corpora of texts. Unlike earlier pre-trained language models, they require little to no fine-tuning and perform well in zero and few-shot settings.

In a zero-shot generation task, the model is provided with the inputs and an instruction in natural language, and the response of the model can be parsed to obtain the answer. For open-domain QA the only input is the question but for closed-domain the input also includes the context. In a few-shot setting, the model is also provided with a few examples, which have the indirect effect of "fine-tuning" the model (Brown et al., 2020). Recent large language models (e.g., GPT-3.5-Turbo, Vicuna-13B, Alpaca) have been fine-tuned, either using supervised or reinforcement learning, with prompts for a diverse set of natural language tasks (Sanh et al., 2021; Ouyang et al., 2022; Wang et al., 2022; Taori et al., 2023; Chiang et al., 2023). They have generally outperformed earlier language models on most natural language generation benchmarks including question answering, summarization, and translation (Zhang et al., 2023). Instruction-tuned models achieve better performance than fine-tuned models but can suffer from two major issues (Ji et al., 2023): inconsistency (i.e., results generated have been found to be overly reliant on the phrasing of the prompt) and hallucination (i.e., models were found to generate false/irrelevant information in some instances).

2.2 Prompt Engineering

Prompt engineering techniques (Gao et al., 2021; Liu et al., 2023; White et al., 2023) aim to design prompts that mitigate these issues. For instance, in (Gao et al., 2021) a smaller language model, such as T5, is tuned to "auto-complete" the prompts for the large language model. Prompt templates are created in (Liu et al., 2023) and (White et al., 2023) which can be auto-filled. While these methods have

[5]https://ai.google/discover/palm2/

[6]https://ai.facebook.com/blog/large-language-model-llama-meta-ai/

[7]https://platform.openai.com/docs/models

had some success, this research is still ongoing.

2.3 LLMs in the Legal Domain

Early applications of LLMs in the legal domain involved testing how well the models perform in standard bar exams. Although early versions of ChatGPT and other LLM models did not perform very well, GPT-4 and Claude 2 were found to perform particularly well on contract-based multiple-choice questions (Choi et al., 2023; Katz et al., 2023; cla). While this was promising, most of the subsequent research into the use of LLMs in the legal domain has focused on chat-style interactions (Kuppa et al., 2023), case summarization (Nay et al., 2023), and simple yes/no questions (Trautmann et al., 2022). In our previous work (Roegiest et al., 2023), we performed a pilot exploration of prompts for generating structured outputs from contracts using LLMs. To the best of our knowledge, this work is one of the first to address the creation of prompt templates to generate structured and partially structured answers in the legal domain.

3 Open-Ended Response Generation

With the popularity of LLM powered chat bots and conversational agents, it is easy to think that "chatting" with one's documents would be advantageous for lawyers and legal professionals. Such an approach can work when dealing with one or two documents but when scaling to hundreds or thousands of documents this becomes untenable. This results in needing to triage and prioritize work to be done; that is, some documents are invariably more important than others. For example, a commercial real estate company may be concerned with environmental indemnification[8] present in their leases due to new legislation, their lawyer may seek to find which documents do not have such a clause, then documents where the tenant indemnifies the company, then where the real estate company indemnifies the tenant, and finally where this indemnification is mutual. If one can identify the presence (or absence) of such a clause then the lawyer simply needs to make a determination but, again, this does not scale effectively.

We might reasonably wonder what happens if we were to ask a LLM model about the direction of mutual indemnification in a sample clause (Figure 1). If one prompts OpenAI's GPT-3.5-Turbo model

[8]An environmental indemnity is a clause that provides a party with protection from losses as a result of environmental contamination or disaster.

```
Lessor hereby agrees to defend, indemnify and save harmless any and all Lessee
Indemnified Parties from and against any and all Environmental Liabilities
(including, without limitation, Environmental Liabilities to the extent
resulting from conditions existing at the Leased Property at the Commencement
Date or from Releases or other violations of Environmental Laws (without fault
on the part of Lessee) originating on other property but affecting the Leased
Property) other than Environmental Liabilities to the extent caused by the
grossly negligent acts or failures to act or wilful misconduct of Lessee,
Manager or subtenants of Lessee or Manager, and their respective employees,
agents or independent contractors.

Lessee hereby agrees to defend, indemnify and save harmless any and all Lessor
Indemnified Parties from and against any and all Environmental Liabilities to
the extent caused by the grossly negligent acts or failures to act or wilful
misconduct of Lessee, -47-<PAGE> 56 Manager or subtenants of Lessee or Manager,
and their respective employees, agents or independent contractors.
Lessee hereby agrees to defend, indemnify and save harmless any and all Lessor
Indemnified Parties from and against any and all Environmental Liabilities to
the extent caused by the grossly negligent acts or failures to act or wilful
misconduct of Lessee, -47-<PAGE> 56 Manager or subtenants of Lessee or Manager,
and their respective employees, agents or independent contractors.
```

Figure 1: An example of an environmental indemnity clause with mutual indemnification.

(with temperature set to 0) with the following: *In the clause below, who indemnifies whom?* and then append the example clause, we get the following responses after regenerating a handful of times:

- The Lessor indemnifies the Lessee Indemnified Parties, and the Lessee indemnifies the Lessor Indemnified Parties.
- Lessor indemnifies Lessee Indemnified Parties for any and all Environmental Liabilities, except those caused by the grossly negligent or wilful misconduct of Lessee, Manager, or subtenants of Lessee or Manager, and their respective employees, agents or independent contractors. Lessee indemnifies Lessor Indemnified Parties for Environmental Liabilities caused by the grossly negligent or wilful misconduct of Lessee, Manager, or subtenants of Lessee or Manager, and their respective employees, agents, or independent contractors.
- Lessor indemnifies Lessee Indemnified Parties. Lessee indemnifies Lessor Indemnified Parties.

While the first and third are close, they aren't identical and the second is more of an overall summary. Moreover, these still require some amount of manual work to read and potentially process the (simplified) outputs. That is, a lawyer still reads a non-trivial amount for every clause found when they might prefer to direct efforts towards leases that do not indemnify their client. While something like nugget matching (Dang et al., 2006) could help automate this process, this is also not something that we might reasonably expect a lawyer to do.

To further illustrate this problem, we present the responses from the same prompt with a different omitted environmental indemnity clause:

- Tenant indemnifies Landlord for claims arising from the introduction of hazardous material on or after the Effective Date, and Landlord indemnifies Tenant for claims arising from the existence of hazardous material or violation of Environmental Requirements prior to the Effective Date.
- In the first paragraph Tenant indemnifies and defends Landlord, and in the second paragraph Landlord indemnifies and defends Tenant.
- Tenant indemnifies Landlord. Landlord indemnifies Tenant.

We see responses that are correct but are also quite different from the responses for the previous clause. As should be apparent by now, this inconsistency and unreliability as well as response verbosity is

not ideal for simplifying work process and may just exacerbate things in the long run. The following sections detail our investigation into how well popular large language models can be prompted to consistently generate (partially) structured answers to questions based on contract clauses.

4 Models and Baseline Questions

4.1 Models

We employed OpenAI's *GPT-3.5-Turbo* (Ouyang et al., 2022) and *GPT-4* (OpenAI, 2023), Google's *PaLM2* (Anil et al., 2023) and Meta's *LlaMa2 13B* (Touvron et al., 2023a) Generative AI.

4.2 Baseline Questions

In our experimental setting, we draw upon a collection of previously annotated legal documents, annotated similarly to those described in (Roegiest et al., 2018), collected from EDGAR[9] and SEDAR[10] document repositories. These documents were originally annotated to train machine learning models to identify various legal clauses of interest to lawyers and other legal professionals (e.g., non-solicitation, environmental indemnity) and now we seek to answer high-level questions about these clauses. Our questions involve various combinations of legal reasoning, summarization, and extraction of data points in order to properly answer the question.

To test the effectiveness of our approaches, we used four legal questions, each requiring a different output format and answer options. A lawyer went through a subset of the annotated clauses for each question and provided the correct answer option or altered the set of answer options when a previously unforeseen potential answer was discovered (e.g., due to low prevalence). We provide a brief summary of the four questions used, their answer options, and the associated prevalence of each option (as appropriate) in Figure 2.

5 Structured Answer Generation

To generate structured answers, one might reasonably turn toward a more traditional solution by training a discriminative multi-class (or multi-label) classifier to produce clear answers to the questions presented in this work. Such solutions (e.g., SVMs, logistic regression) are well understood and highly optimized but come with a flaw, they require sufficient examples of each class to be effective (i.e., low prevalence classes can be hard to overcome). For example, landlord indemnification is a relatively rare occurrence in our data and one for which data augmentation is not guaranteed to work. Indeed, we have attempted to address this in prior work (Chitta and Hudek, 2019) but the solution did not scale well for extremely low prevalence or nuanced differences between options. It has been our experience that such cases arise naturally in the legal domain where rare outcomes tend to be the most problematic and equally hard to identify at scale. Accordingly, models that require little to no training data (or other labelled) examples become compelling solutions to this problem and we spend the rest of this section discussing our experiments.

Q1) **Yes/No Selection**: The question "Is there a geographical restriction on solicitation?" answers if the parties to the agreement are restricted to a specified territory while soliciting business or services. We used a set of 200 *"non-solicitation"* clauses, which describe the provisions and restrictions for solicitation. There are 15 instances where there are geographical restrictions and 185 instances where there are no geographical restrictions. Options **O1** and **O2** in Figure 2 are the tested option variants for this question.

Q2) **Single Option Selection**: We used a set of 121 *"environmental indemnity"* clauses, which describe the provisions for security or protection from losses or damage caused by environmental contamination or disasters, to answer the question "Who indemnifies whom?". Only one of the following three options applies:

a) The landlord indemnifies the tenant,
b) The tenant indemnifies the landlord, and
c) The landlord and tenant indemnify each other.

There are 6, 71, and 39 instances of the above options, respectively, and 5 instances where the clause does not contain the answer to the question. Options **O3** and **O4** in Figure 2 are the tested option variants for this question.

Q3) **Multiple Options Selection**: The question "Can the agreement be terminated for convenience?" answers the conditions under which either party to the agreement is allowed to terminate the agreement without cause. We used a set of 225 *"termination for convenience"* clauses and a set of 6 options (**O5** in Figure 2):

a) 6 instances of "agreement may be terminated without any conditions or exceptions",
b) 208 instances of "agreement may be terminated with prior notice",
c) 5 instances of "agreement may be terminated after paying a termination fee",
d) 11 instances of "agreement may be terminated after a specified time period has elapsed",
e) 7 instances of "agreement may be terminated only by mutual consent of the parties", and
f) 5 instances of "agreement may not be terminated for convenience".

Q4) **Partially Structured Response Generation**: This question "What is the timeframe of the non-solicitation clause?", also based on the non-solicitation clause, determines the duration for which the non-solicitation clause applies. As this question produces a single response extracted or summarized from the clause, it is free-form and has no fixed set of options.

Figure 2: Questions used for testing the LLM models

5.1 Zero-shot Semantic Similarity Matching

Prior to the rise of LLMs for generative tasks, a common use case was to perform some form of matching between questions and answers (Veeranna et al., 2016; Yin et al., 2019) based upon LLM embeddings of questions, potentially additional context, and the possible answers. As this is arguably a less labour intensive task than trying to find an ideal prompt for every question (or a one-size-fits-all prompt), we first turned to this approach in the hopes that it might allow end users to see value more quickly.

[9] https://www.sec.gov/edgar
[10] https://www.sedarplus.ca/

```
O1: a) Yes
    b) No

O2: a) There are geographical restrictions in the
       non-solicit clause.
    b) There are no geographical restrictions in the
       non-solicit clause.

O3: a) Landlord indemnifies Tenant.
    b) Tenant indemnifies Landlord.
    c) There is mutual indemnification.

O4: a) The landlord indemnifies the tenant.
    b) The tenant indemnifies the landlord.
    c) There is mutual indemnity between the landlord
       and the tenant.

O5: a) The agreement may be terminated without any
       conditions or exceptions.
    b) The agreement may be terminated with prior notice.
    c) The agreement may be terminated by either party but
       the terminating party has to pay a termination fee.
    d) The agreement may be terminated by either party
       after a specified time period has elapsed.
    e) The agreement may be terminated by mutual consent
       of the parties.
    f) The agreement may not be terminated for convenience.
```

Figure 3: Options used for testing the LLM models.

Options	PIL-LegalBERT	Ada
Q1: Is there a geographical restriction on solicitation?		
O1	41.2	8.0
O2	91.9	73.4
Q2: Who indemnifies whom?		
O3	32.2	5.5
O4	32.2	22.6
Q3: Can the agreement be terminated for convenience?		
O5	28.4	0.5

Table 1: Accuracy (in percentage) of the different embedding models with different answer options on the single option and multiple option questions.

Given the text of the clause T, and a set of answer options H, we use a sequence embedding model M (Selva Birunda and Kanniga Devi, 2021) to obtain the embeddings for the clause and the answer options and then predict the answer which has the highest similarity with the clause: $\tilde{h} = \arg\max_{h \in H} \cos\left(M\left(T\right), M\left(h\right)\right)$. We used two large language models to compare the performance of this method: (1) the LegalBERT model trained on the Pile-Of-Law dataset (Henderson et al., 2022)[11] and (2) the OpenAI Ada model (Neelakantan et al., 2022). We truncated the clauses to less than 512 and 8191 tokens in length, respectively, in order to adhere to model token limits.

5.1.1 Results

As seen in Table 1, **O1** is not a good option set for **Q1** due to the lack of context for any meaningful semantic matching but, in contrast,**O2** does much better because the options are clear fully-formed sentences. Both models suffer from the same issue by rarely predicting option (a) correctly, regardless of the option set, with the Ada model getting this right once and LegalBERT never predicting it correctly. The seemingly high accuracy is only due to the imbalanced distribution of the answers in the test dataset (185 instances of option (b) to 15 of option (a)). A similar situation holds for **Q2** with neither model being able to correctly predict option (a) and option (b) consistently.

The situation is worse for question **Q3** as without a tedious cut-off selection process (that may be untenable in the real world due to prevalence issues), similarity matching can only be used to easily predict the most similar option. With this in mind, even being lenient in our evaluation (i.e., does a model predict any correct option) neither model is able to perform well on this question. Overall, the LegalBERT model performs better than the Ada model, in all cases, because it has been trained on legal data but still performs far too poorly for end user applications. While more recent and complex systems, like Logiformer (Xu et al., 2022), may improve these results, such approaches are substantially more complicated than our prompt engineering technique and our preliminary exploration with them suggested additional fine-tuning would be needed.

5.2 Question Specific Prompts

Based upon preliminary manual investigations (Roegiest et al., 2023) into whether one could prompt LLMs to produce structured outputs, we turned to a more thorough and structured investigation of different LLMs and prompt styles. Although we tested our prompts on all four models, we focused on GPT-3.5-Turbo and optimized the prompts for it because of its reasonable costs, ease of incorporation into software (i.e., no infrastructure support), and general widespread adoption. The ideal would be to create bespoke prompts for every generative model but we did not believe that this approach would yield substantive improvements other than to indicate the delta between a tuned and untuned prompt. Essentially, if we know these prompts work consistently well on a single model and perform reasonably well on other similar models then we can have confidence that prompt templates are a viable solution irrespective of which model one chooses.

Our prompts (Figure 4) were constructed using learnings from our initial pilot exploration (Roegiest et al., 2023)[12] (across thousands of different

[11] https://huggingface.co/pile-of-law/legalbert-large-1.7M-2

[12] A prompt testing tool that can be used to run and test the prompts listed in this paper is available at https://github.com/zuvaai/gpt-tool.

```
P1: Read the following non-solicitation legal clause:
    {{Clause}}
    Pretend you are a party to the agreement in which the
    non-solicitation legal clause you have read exists in.
    You only know what you have read in this prompt.
    Is there a geographical restriction in the
    non-solicitation legal clause?
    If the clause does not specify, respond with: "Unable
    to determine". In your response, only include the
    single most correct option:
    {{Options}}
    In your response, only include the bucket names above.
    Do not provide an explanation or additional
    information.
```

```
P2: Read the following environmental indemnity legal clause:
    {{Clause}}
    Pretend you are a party to the agreement in which the
    environmental indemnity legal clause you have read exists
    in. You only know what you have read in this prompt.
    Who indemnifies whom?
    If the clause does not specify, respond with: "Unable
    to determine". In your response, only include the single
    most correct option:
    {{Options}}
    In your response, only include the bucket names above. Do
    not provide an explanation or additional information.
```

```
P3: Read the following termination for convenience legal clause:
    {{Clause}}
    Pretend you are a party to the agreement in which the termination for convenience legal clause you have read exists in.
    You only know what you have read in this prompt.
    Can the agreement be terminated for convenience according to the termination for convenience legal clause you have read?
    If the clause does not have any information about termination for convenience, respond with: "Unable to determine". In
    your response, only include the following most correct groups:
    {{Options}}
    In your response, only include the bucket names above. Do not provide an explanation or additional information.
```

```
P4: Read the following non-solicitation legal clause:
    {{Clause}}
    Pretend you are a party to the agreement in which the
    non-solicitation legal clause you have read exists in.
    You only know what you have read in this prompt.
    Extract the duration for which the non-solicitation
    clause is applicable. Do not extract any other
    information from the clause. If the clause does not
    contain this information, respond with
    "Unable to determine".
```

```
P5: Read the following  non-solicitation legal clause:

    {{Clause}}

    Pretend you are a party to the agreement in which the
    non-solicitation legal clause you have read exists in.
    You only know what you have read in this prompt.

    Extract the timeframe of the non-solicitation legal clause.
    Do not extract any other information from the clause.
    The response should be exact words from the text you have
    read. If the clause explicitly states the duration for
    which the non-solicitation clause applies, respond with
    this information extracted from the clause.
    If the clause does not explicitly state the duration for
    which the non-solicitation applies, and the
    non-solicitation applies for the whole duration of the
    agreement, respond with "term of the agreement".
    If the clause does not contain this information, respond
    with "Unable to determine".
```

Figure 4: Prompts used for testing the LLM models.

prompt, clause, and option combinations) which highlighted a strong need to embed the LLM role as one where they are party to the agreement (i.e., adopting a persona), provide escape hatches when no answer was appropriate (i.e., return "Unable to determine"), and limiting the potential for creative answering both in the prompt (i.e., "provide no justification") and in hyper-parameters (i.e., temperature set to 0). While these are not necessarily critical for success, we found that LLMs were generally more reliable and consistent in how they answered questions when these were factored into the prompt. Moreover, we never set a "system" message as there was no guarantee that a given model would support it or listen to it. Indeed, initial experiments using OpenAI models did not provide any strong indication that a "system" message would improve the outputs. We ensured clauses were shorter than $20,000$ characters in order to adhere to model token limits.

5.2.1 Results

For Q1, we started with the prompt **P1** shown in Figure 4. $\{\{Options\}\}$ is replaced by the options and $\{\{Clause\}\}$ is replaced by the text of the clause. The phrase "only include the single most correct option" aims to ensure that only **one** of the answer options is generated and our aforementioned guardrails (i.e., escape hatches and creative writing prevention) are used to ensure that the response from the API is able to be parsed, via regular expression, to obtain the structured answer to the question. All the models perform reasonably well with the **O1** option set (Figure 3) of just "yes" and "no," unlike the similarity matching approach. However, we find that with the more well-formed option set **O2**, both PaLM2 and Llama2 models improve substantially. The GPT-4 model, surprisingly, generates "Unable to determine" for many clauses with options **O2**, leading to a lower accuracy.

Prompt	Options	GPT-3.5 Turbo	GPT-4	PaLM2	LLaMA2-13
Q1: Is there a geographical restriction on solicitation?					
P1	O1	96.5	92.9	88.9	90.9
	O2	93.9	59.8	91.5	95.9
PT1	O2	77.4	62.2	91.4	91.9
Q2: Who indemnifies whom?					
P2	O3	66.1	94.2	79.3	47.9
	O4	67.8	95.0	84.3	44.6
PT1	O4	67.7	94.2	72.7	57.0
Q3: Can the agreement be terminated for convenience?					
P3	O5	96.0	56.4	79.6	30.2
PT2	O5	95.1	48.9	69.3	7.1

Table 2: Accuracy (in percentage) of the LLM models with the various combinations of prompts and options on the single option and multiple option questions.

Similar to **Q1**, **Q2** also needs only one option to be chosen and we use a similar prompt, **P2**, with options **O3** and **O4**. Options **O4** contain clear fully-formed sentences and are more effective than options **O3**, especially with the PaLM2 model. This is partly due to the fact that **O4** does not invite ad-

Prompt	Metric	GPT-3.5 Turbo	GPT-4	PaLM2	LLaMA2-13
Q4: What is the timeframe of the non-solicitation clause?					
P4	Rouge-l F1	0.397	0.422	0.215	0.079
	Exact match accuracy	23.6	25.1	1.5	0
	Semantic match accuracy	48.2	69.3	36.6	23.6
P5	Rouge-l F1	0.481	0.482	0.313	0.047
	Exact match accuracy	24.6	21.6	8.0	0
	Semantic match accuracy	64.8	86.9	51.3	8.0

Table 3: Performance of the LLM models with the different combinations of prompts and options on the partially structured response question.

ditional generation when compared to **O3**'s more "open-ended" phrasing as well as making it more clear what mutual indemnification means. Despite being tailored to GPT-3.5-Turbo, we see that GPT-4 and PaLM2 both answer this question much better than it. While it is not entirely clear why this happens, we suspect it may just be due to the complex nature of how indemnification, especially mutual indemnification, can be worded and that this might be captured less well by GPT-3.5-Turbo.

Q3 requires multiple option selection and so the prompt **P3** is styled to respond with the "most correct groups," and, like previous prompts, has the necessary guardrails to help constrain generation. As this is a more difficult style of question, we are more lenient and give partial credit if at least one of the correct options is present in the generated response. The results themselves are not all that revealing with GPT-3.5-Turbo being the most effective followed by the other very large models.

Finally, **Q4** requires partially structured free-text responses which can lead to increased creativity and hallucination by LLM models. Prompts for **Q4** seek to mitigate this issue by requiring models to "extract" only the relevant information, to not "extract" superfluous information in the prompt, and using our usual guardrails for generation. We evaluate the responses for this question using three metrics: (i) *Rouge-l F1-score* (Lin and Och, 2004) which measures the overlap between the longest co-occurring sequences in the generated response and the expected responses (i.e., higher is better); (ii) *exact match accuracy* which measures the percentage of clauses where the generated and expected responses match exactly; and (iii) *semantic match accuracy* which measures the percentage of clauses where a response contains all the pertinent information included in the expected response and is evaluated manually. *Semantic match accuracy* models the situation that the LLM response is correct but is not text extracted from the provided clause.

We found it more difficult to find an appropriate prompt to extract the relevant portion of the clause. Using the simple prompt **P4**, all models fail to perform well. The most common expected answer for the timeframe of non-solicitation was "during the term of the agreement" but all models were unable to produce this answer. We added additional instructions in prompt **P5**, to respond with the "term of the agreement" when applicable which resulted in substantial improvements to model efficacy.

The restriction to "not provide an explanation or additional information" is an attempt to mitigate a model's creativity but it is not consistently effective, and some amount of post-processing is required to obtain the structured response. For instance, the LLaMA2 model usually preceded the option selected with "Based on the clause you provided," and sometimes even generated garbled text. The PaLM2 model was the best at adhering to the prompt and generating only the option letter. The OpenAI models, in most cases, generated the text of the option along with the option letter.

In contrast with **Q4** despite the instructions attempting to avoid hallucinations, we found that they did occur occasionally with some of the most egregious examples coming from PaLM2 (e.g., saying a duration was 2 years when it was, in fact, 90 days). Due to the nature of how they were trained (i.e., to produce plausible English responses), all models required some amount of post-processing to remove a preamble that would make the response a valid sentence (e.g., OpenAI models had a preamble of "The duration for which the non-solicitation clause is applicable is", PaLM2 generated "The non-solicitation clause is applicable for").

We are not surprised that the Llama2 model performed particularly poorly on **Q4** based upon its prior performance, but are surprised that it generated many more unexpected responses than we would have anticipated. For example, it would sometimes respond with a question back to the user (e.g., "*Sure! I'd be happy to help you with that. Please go ahead and ask your question.*"). Some of Llama2's responses are surely attributable to it being a much smaller model relative to the others tested but may also stem from a much different set of training data. The details of its training data have not been publicly discussed in much detail (to the best of our knowledge) but it would appear that little of it resembled our prompts due to its poor performance on all but the simplest question (i.e., **Q1**). While we cannot make any claims about the training data of the other models, it appears that there may have been similar instances, either stylistically

```
PT1: Read the following {{Clause-name}} legal clause:
     {{Clause}}
     Pretend you are a party to the agreement in which the
     {{Clause-name}} legal clause you have read exists in.
     You only know what you have read in this prompt.
     Which of the following is true according to the
     {{Clause-name}} legal clause you have read?
     If the clause does not specify, respond with: "Unable
     to determine". In your response, only include the
     single most correct option:
     {{Options}}
     In your response, only include the bucket names above.
     Do not provide an explanation or additional
     information.

PT2: Read the following {{Clause-name}} legal clause:
     {{Clause}}
     Pretend you are a party to the agreement in which the
     {{Clause-name}} legal clause you have read exists in.
     You only know what you have read in this prompt.
     Which of the following is true according to the
     {{Clause-name}} legal clause you have read?
     If the clause does not specify, respond with: "Unable
     to determine". In your response, only include the
     following most correct groups:
     {{Options}}
     In your response, only include the bucket names above.
     Do not provide an explanation or additional
     information.
```

Figure 5: Prompt Templates for single option and multiple option questions.

(i.e., multiple choice) or thematically (i.e., legal), in their training data given their performance.

5.3 Reusable Prompt Templates

In the previous section, we saw promise in having specialized prompts for different questions but even then many elements were shared among them. Accordingly, we extend this approach to investigate the feasibility of "fill-in-the-blank" prompt templates that allow us to consolidate on fewer prompt possibilities. In examining our specialized prompts, we found that the key "changable" aspects of the prompts were: the name of the clause, the question being asked, and the answer options; everything else was largely equivalent or very nearly so. Based upon some analysis of our existing prompts, we created two generalized prompt templates, shown in Figure 5, that seek to satisfy single option and multiple option selection. The clause name and answer options need to be filled in by an end-user but we simplify the process by replacing bespoke questions with a generic one, "Which of the following is true according to the clause you have read?". The prompt template **PT1** asks the model to select the "single most correct option" and is meant for multiple choice questions. The prompt template **PT2**, on the other hand, asks for "the most correct groups" and is meant for questions for which more than a single option may be appropriate.

Based upon our building of these prompts and the results in Table 2, answer options can clearly play a non-trivial role when using the prompt templates as they need to contain sufficient information for the model to make a determination. Generic answer options, such as those in **O1**, obviously do not perform well but when the answer options are elaborate and contain sufficient context, the accuracy is only slightly worse than the accuracy of the question-specific prompts for most models. As before, the PaLM2 and LLaMA2 model perform worse than either OpenAI models but such an outcome is not unexpected given that these prompts were tailored to what worked for GPT-3.5-Turbo.

From these results and those of the previous section, we have strong evidence that creating general prompt templates to generate structured answers for single and multiple options questions is indeed possible and that model effectiveness is not seriously impacted by doing so. However, we have not had the same success for questions requiring partially structured responses, as there is high variability in the expected responses, which need multiple escape hatches and conditions to ensure only the correct information is produced.

6 Limitations and Future Work

In this work, we focused on optimizing prompts for GPT-3.5-Turbo for the sake of simplicity and cost. Moreover, as there are increasing numbers of models being made available for commercial use (Conover et al., 2023; Biderman et al., 2023; Almazrouei et al., 2023; Touvron et al., 2023b), we felt that overall GPT-3.5-Turbo provided a reasonable baseline due to it being an obvious "initial" starting point that one might use before jumping into fine-tuning an "open" model. The benefit to this was the somewhat surprising result from Tables 2 and 3 is that prompts optimized for one LLM can produce relatively competitive results (or exceed the base LLM) with the same prompt. Further exploration remains to determine how much benefit would be gained fine-tuning a smaller model, like Llama2, given that the much larger models are "by default" apparently much better at our desired tasks. Subsequent investigation into this is planned as it is far less cost efficient to fine-tune larger LLMs and more generally reducing model size has other social benefits (e.g., less carbon footprint).

We have also largely operated under the assumption that an end user has the appropriate clause easily available about which they can ask questions. As retrieval augmented generation has become a popular topic of research (Lewis et al., 2021), combining our approach with existing approach will invariably require examining how well these prompts

and models deal with false positives. Our prompts do have escape mechanisms and guard rails to hopefully prevent obvious errors from occurring but we leave investigation of how errors propagate in end-to-end systems with our prompts to future work.

7 Conclusions

We have presented a structured answer generation task based on identifying the correct answer for a legal question given an associated clause from a document, detailed some issues with relying solely on natural language question answering, and explore solutions using unsupervised and generative methods. We have shown that the prompt-based generative methods exceed semantic similarity-based approaches and that despite question specific prompts performing better, that generalized prompts can be used with little decrease in overall effectiveness. Finally, partially structured answer generation is shown to be possible with bespoke prompts but generalized approaches do fare well due to the large amount of variance in the output formats that are required to accommodate the types of information desired.

References

Anthropic claude-2.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality.

Radha Chitta and Alexander K. Hudek. 2019. A reliable and accurate multiple choice question answering system for due diligence. In *Proceedings of the International Conference on Artificial Intelligence and Law*, page 184–188.

Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. 2023. ChatGPT goes to law school. *Social Science Research Network*.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Ali Ghodsi, Patrick Wendell, and Matei Zaharia. 2023. Hello dolly: Democratizing the magic of chatgpt with open models. https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html.

Hoa Trang Dang, Jimmy Lin, and Daine Kelly. 2006. Overview of the trec 2006 question answering track. In *Proceedings of the Text Retrieval Conference*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 3816–3830.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2023. Gpt-4 passes the bar exam. *Social Science Research Network*.

Aditya Kuppa, Nikon Rasumov-Rahe, and Marc Voses. 2023. Chain of reference prompting helps llm to think like a lawyer. In *Generative AI+ Law Workshop*.

Spyretta Leivaditi, Julien Rossi, and Evangelos Kanoulas. 2020. A benchmark for lease contract review. *CoRR*, abs/2010.10386.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the Association for Computational Linguistics*, pages 605–612.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9).

John J Nay, David Karamardian, Sarah B Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H Choi, and Jungo Kasai. 2023. Large language models as tax attorneys: A case study in legal capabilities emergence. *arXiv preprint arXiv:2306.07075*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training.

OpenAI. 2023. GPT-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm.

Adam Roegiest, Radha Chitta, Jonathan Donnelly, Maya Lash, Alexandra Vtyurina, and François Longtin. 2023. A search for prompts: Generating structured answers from contracts.

Adam Roegiest, Alexander K Hudek, and Anne McNulty. 2018. A dataset and an examination of identifying passages for due diligence. In *Proceedings of the international ACM SIGIR conference on research & development in information retrieval*, pages 465–474.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization.

S Selva Birunda and R Kanniga Devi. 2021. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020*, pages 267–281.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. LLaMA 2: Open foundation and fine-tuned chat models.

Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models – a critical investigation.

Sappadla Prateek Veeranna, Jinseok Nam, Eneldo Loza Mencıa, and Johannes Fürnkranz. 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proceeding of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 423–428.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt.

Fangzhi Xu, Jun Liu, Qika Lin, Yudai Pan, and Lingling Zhang. 2022. Logiformer: A two-branch graph transformer network for interpretable logical reasoning. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1055–1065.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers.