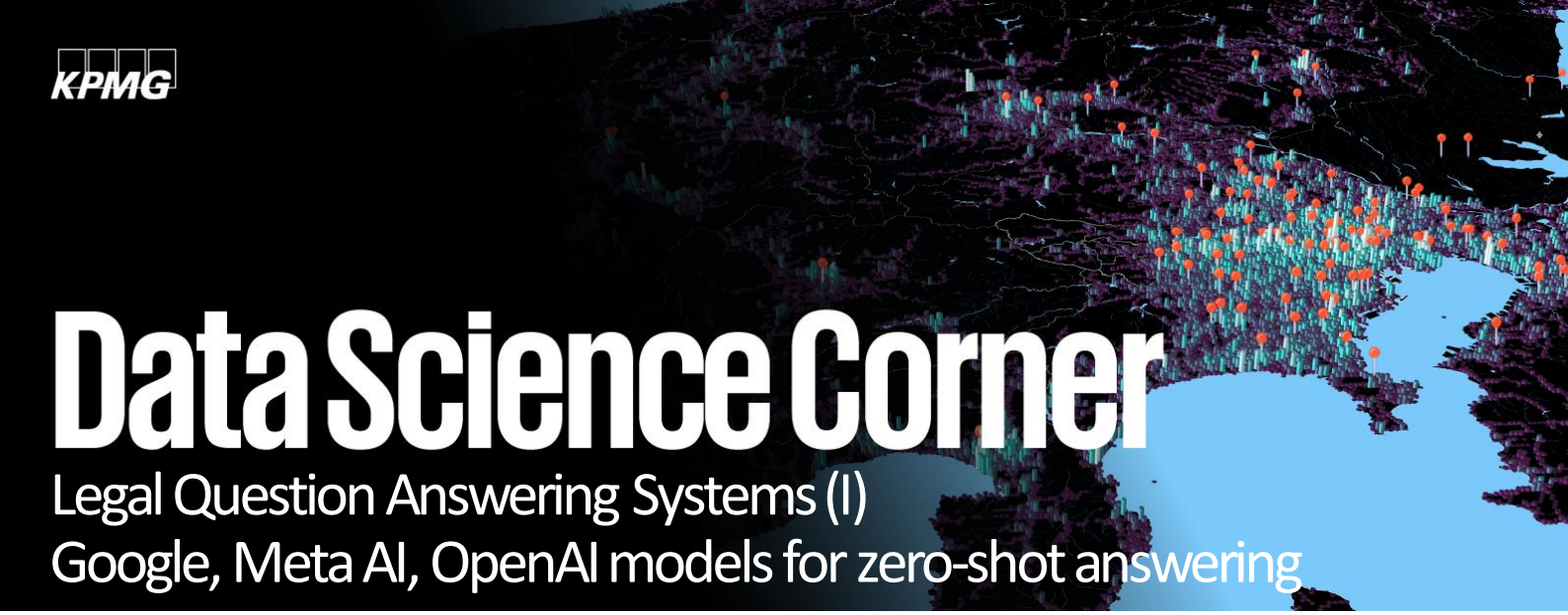

Data Science Corner

Legal Question Answering Systems (I)
Google, Meta AI, OpenAI models for zero-shot answering

Question Answering (QA) is a field of Natural Language Processing addressed to develop automatic methods for answering questions expressed in natural language. Recently, the emergence of new language models has raised the hope that QA systems are now, to some extent, capable of answering various types of questions, from simple questions whose answers can be found in a single passage to complex questions which need more complicated reasoning from several, not necessarily contiguous, passages to find the answer.

In this first of three artides focusing on **legal question answering systems**, we will document the results of an experiment to assess the accuracy of systems that are based on recent language models, namely OpenAI's ChatGPT and Davinci, Google's Flan, and Meta AI's OPT. In the second article, we will discuss the issue of retrieving relevant paragraphs from large contracts to answer a certain question. The last article will share some good engineering practices for fine-tuning and serving large question answering models.

Legal Question Answering Systems

Legal document review is the process of thoroughly reading a legal document to understand the rights and obligations of an entity signing it and assess the associated impact. There are different levels of work in legal document review[1]. The lowest level is sometimes referred as "document analysis". At this level, the reviewer needs to manually review hundreds of pages of contracts to find the relevant clauses stipulated in the document. They must identify whether relevant clauses exist and what they say if they do exist. The highest level of work is to assess risk associated with the document clauses and advise on solutions. At this level, a business client relies on highly specialized legal experts to explain not only what each clause means, but also the implications that such a clause has on client's business. This type of work is referred as "counseling".

We believe that the development of recent large language models has made possible the design and deployment of semi-automatic solutions for assisting legal practitioners in document analysis.

Legal technology, also known as Legal Tech, is a broad umbrella covering a group of technologies aimed at automating tasks such as practice management, document automation, document storage, and electronic discovery. In this context, we focus on a subset of the Question Answering (QA) challenge: Question answering with pre-defined document collections. In this case, a pre-defined set of documents is provided up front, and answers to questions are available in this set of documents. Such systems are also referred to as closed-domain systems. It's important to note that the definition of 'document' depends on the application: it can be a paragraph of a contract which represents a set of documents, or a multi-page text file such as a past legal case in a database of lawsuits. Moreover, the availability of the answer in the set of documents means that the answer may be inferred by zero or more steps in a chain of thoughts from several, not necessarily contiguous, passages of documents.

1: Hendrycks et al, CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review, NeurIPS 2021. https://arxiv.org/abs/2103.06268

In order to satisfy minimal requirements the process of automatically answering a question involves three fundamental steps (see also Figure 1):

1. **Document Retrieval** to identify documents that may contain the answer from a large source pool. This step is necessary because of the limited length of text input of language models used for answering the question and, as we will see later, because there is evidence that the shorter the input text is the more accurate the answers are;
2. **Reading Comprehension** to answer a question. Usually, the question comes in the form of a natural language interrogative sentence, although sometimes a question could also take the form of an imperative construct and starts with a verb;
3. **Evidence Extraction** to find concise spans of text that support the answer provided.

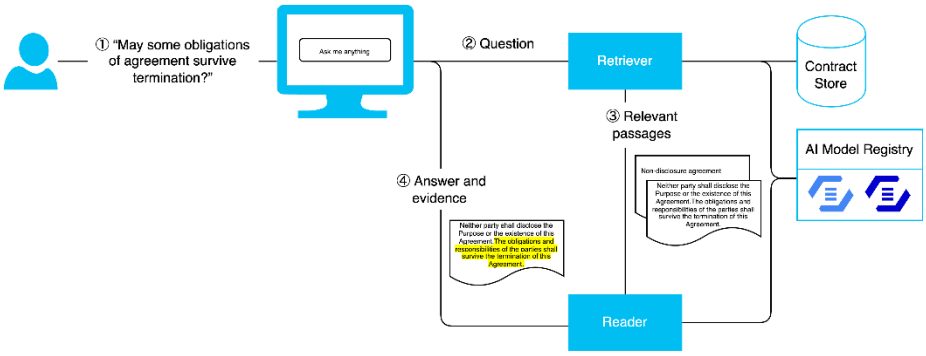

Figure 1: High level overview of a legal question answering system.

An example of a use case in KPMG for a legal question answering system is the review of engagement letters. Before an engagement team can offer professional services to a client, a local Independence Check team must verify that independence requirements are met by, among other things, reviewing the engagement letter that describes the scope of the service.

We tested the reading comprehension of some pre-trained models in a set of Yes / No questions about contract clauses without training or finetuning the models for this new task. This setting is known as zero-shot questions answering.

The Question Answering Task

The datasets (see Section The Datasets) used for the experiment are derived from ContractNLI[1], a set of 607 non-disclosure agreements annotated for a Natural Language Inference task. Given a set of 17 hypotheses (such as "Some obligations of Agreement may survive termination of Agreement.") and a contract, the task is to classify whether each hypothesis is entailed by, contradicting to or not mentioned by (neutral to) the contract as well as identifying evidence for the decision as spans in the contract. The dataset was released in October 2021.

1: https://stanfordnlp.github.io/contract-nli

...
Confidential Information: means all confidential information (however recorded, preserved or disclosed) disclosed by a Party or its Representatives to the other Party and that Party's Representatives including but not limited to:
(a) the fact that discussions and negotiations are taking place concerning the Purpose and the status of those discussions and negotiations;
(b) the existence and terms of this Agreement;
(c) any information relating to:
(i) the business, affairs, customers, clients, suppliers, plans, intentions, or market opportunities of the Disclosing Party or of the Disclosing Party's Affiliates; and
(ii) the operations, processes, product information, know-how, designs, specifications, trade secrets, computer programs or software of the Disclosing Party or of the Disclosing Party's Affiliates; and
(d) any information or analysis derived from Confidential Information.
...

// denotes a span border

Examples of hypotheses:

Receiving Party shall not disclose the fact that Agreement was agreed or negotiated. (Evidence denoted with green highlight on upper half of text)	☑ Entailment ☐ Contradiction ☐ Not mentioned
Confidential Information shall only include technical information. (Evidence denoted with blue highlight on bottom half of text)	☐ Entailment ☑ Contradiction ☐ Not mentioned
Receiving Party shall not use any Confidential Information for any purposes other than the purpose(s) stated in Agreement. (Evidence does not exist when the hypothesis is not mentioned)	☐ Entailment ☐ Contradiction ☑ Not mentioned

Figure 2: Examples of annotations in the ContractNLI dataset. We notice that the evidence for the Contradiction example is made by sentences that are not contiguous (from arxiv.org/pdf/2110.01799.pdf).

In our experiment, we reformulated the task as a Yes / No question answering task by rewriting the hypothesis as a question (for example, "Can some obligations of Agreement survive termination of Agreement?") and assumed that we have a retrieval system capable to retrieve contracts that are not neutral to the hypothesis. Table 1 lists the 17 questions.

Question
Can Confidential Information indude verbally conveyed information?
Can the Receiving Party create a copy of some Confidential Information in some circumstances?
Can the Receiving Party acquire information similar to Confidential Information from a third party?
Can the Receiving Party disclose the fact that Agreement was agreed or negotiated?
Can the Receiving Party independently develop information similar to Confidential Information?
Can the Receiving Party retain some Confidential Information even after the return or destruction of Confidential Information?
Can the Receiving Party reverse engineer any objects which embody Disclosing Party's Confidential Information?
Can the Receiving Party share some Confidential Information with some of Receiving Party's employees?
Can the Receiving Party share some Confidential Information with some third-parties (including consultants, agents and professional advisors)?
Can the Receiving Party solicit some of Disclosing Party's representatives?
Can the Receiving Party use any Confidential Information for any purpose other than the purposes stated in Agreement?
Does the agreement grant Receiving Party any right to Confidential Information?
Can some obligations of Agreement survive termination of Agreement?
Shall Confidential Information only indude technical information?
Shall all Confidential Information be expressly identified by the Disdosing Party?
Shall the Receiving Party destroy or return some Confidential Information upon the termination of Agreement?
Shall the Receiving Party notify Disclosing Party in case Receiving Party is required by law, regulation or judicial process to disclose any Confidential Information?

Table 1. Yes / No questions derived by the hypotheses of the original natural language inference ContractNLI task.

The average length of an agreement is 1682 words[1] and the number of questions is 6173[2].

1: This is the average number of alpha and digit tokens as extracted by SpaCy en_core_web_lg model.

2: This number is different from 17 x 607 because as mentioned there are hypotheses that are neutral to some of the contracts.

Text-to-text Generation Models

We compared some of the most recent text-to-text generation models on the question answering task described in the previous section. A text-to-text generation model takes an input containing a text and some sentences that describe the task that the model is expected to accomplish on the text. The output of the model is another text for the accomplished task. Examples of common tasks addressed by text-to-text generation models are question answering, summarization, and code generation in a certain programming language. The models that we compared are:

1. **Google Flan T5 XXL**[3]. This model belongs to the family of models that were released by Google in December 2022. These models were trained with a particular focus on scaling the number of tasks and fine tuning on chain-of-thought data. The largest model has 540 billions of parameters. Flan T5 XXL has 11 billions of parameters and is the largest model of the family publicly available under the license Apache 2.0.
2. **Meta AI OPT-IML-max-30b**[4]. This is the largest model publicly available under the license Apache 2.0 among the OPT family of text-to-text generation models trained by Meta AI. Released in December 2022, the model has 30 billions of parameters and was trained on approximately 2,000 tasks.
3. **OpenAI Davinci (text-davinci-003)**[5]. This is the most powerful model in the OpenAI GPT3 family of models. According to OpenAI, Davinci is particularly suitable for applications requiring a lot of understanding of the content, like summarization, solving logic problems, and question answering for complex intent scenarios. Its latest version was released in March 2022.

3: See also Hugging Face Flan T5 XXL model card.

4: Hugging Face OPT IML model card.

5 OpenAI Davinci Documentation.

4. **ChatGPT (gpt-3.5-turbo)**[1]. This model was released in November 2022 and is quite different from the above models because of its ability to interact with users in a conversational way. The dialogue format makes it possible for ChatGPT to answer follow-up questions.

1: OpenAI ChatGPT website.

There is a limit to the length of the complete text, input and output, that can be handled by these models. For each model, this limit is expressed as maximal number of tokens in which the text can be split according to the corresponding tokenizer[2]. See Table 2.

2: Tokenization is a method of splitting a piece of text into smaller units called tokens. Depending on the method, tokens can be either words, characters, or sub-words. In general, for the same text, the number of tokens depends on the tokenizer used.

Model	Tokenizer	Maximal Input Length (num. of tokens)
Google Flan T5 XXL	Flan T5	512
Meta AI OPT-IML-max30b	OPT	2048
Davinci	Tiktoken p50k_base	2048
ChatGPT	Tiktoken cl100k_base	4096

Table 2. Maximal input length for the models in terms of number of tokens.

Note that the original ContractNLI dataset is not included in the list of datasets on which these 4 models were pre-trained.

Question Answering Datasets

In addition to evaluating the accuracy of the models on the question answering task, we also studied the effect of the length of the input text on accuracy. We prepared 4 datasets with increasing input length. All datasets had two fields, Input Text and Correct Answer, and each row corresponded to a question. The input text is part of a contract to which a question is attached. This always includes the sentences annotated as evidence to answer the question (see Figure 2). The correct answer to the question is either Yes or No. Table 3 shows a row of one of the datasets.

Input Text	Correct Answer
Data Use And Non-Disclosure Agreement Between The New York City Department of Health and Mental Hygiene And _____ B. Restrict Access to "Authorized Users". 1. Only the Data Recipient's employees and/or consultants required to use the Data to perform the functions of this Agreement that are set forth in Attachment B, and so designated by Data Recipient as "Authorized Users" in Attachment C to this Agreement, will be given access to the Data. 2. Question: Can the Receiving Party share some Confidential Information with some of Receiving Party's employees? A. Yes B. No Answer:	Yes

Table 3. A row of a dataset with input text and correct answer to the question attached to the input text.

In the first dataset only sentences that include pieces of the evidence are included. This dataset, therefore, can be used to estimate an ideal maximal performance of the models since its use in production would need an extremely accurate retrieval component for identifying all sentences relevant to the question. The other three datasets are made by including as many sentences from the contract as possible starting from the beginning in such way that the input text fits into the model according to the maximal length values (see Table 2). Below is a more technical description of how these four datasets are built:

1. **Dataset #1**. For each of the 6173 questions, the input text is the concatenation of the first sentence of the contract (to give the model the appropriate legal context), all sentences that include evidence for the answer, the string "¥nQuestion: ", the question, and the string "¥nA. Yes¥nB. No¥nAnswer:". Since we reserve 5 tokens for the answer, the input text is included in the dataset only if the number of tokens is not larger than 507 tokens according to Google Flan T5 XXL tokenizer[3]. Table 3 shows a row of this dataset.

3: More precisely, we included in the dataset only text inputs whose number of tokens is not larger than the maximal number of tokens minus 5 for all four models.

2. **Dataset #2**. For each of the 6173 questions, the input text is the concatenation of the first N sentences of the contract, the string "¥nQuestion: ", the question, and the string "¥nA. Yes¥nB. No¥nAnswer:". Here, N is the maximal number of sentences such that the complete text input has a number of tokens not larger than 507 tokens according to Google Flan T5 XXL tokenizer. The input text is included in the dataset only if all the evidence to answer the question is included.
3. **Dataset #3**. This dataset is built similarly to Dataset #2 with the difference that N is the maximal number of sentences such that the complete text input has a number of tokens not larger than 2043 tokens according to OPT-IML-max-30b tokenizer.
4. **Dataset #4**. This dataset is built similarly to Dataset #2 with the difference that N is the maximal number of sentences such that the complete text input has a number of tokens not larger than 4091 tokens according to the OpenAI tokenizer.

Table 4 reports the number of questions and the average length of the input text for the datasets.

Dataset	Number of Questions	Average Length of Input Text (num. of words)
#1	5633	168
#2	1047	301
#3	5605	1262
#4	6117	1713

Table 4. Number of questions and average number of words as calculated by SpaCy for each of the 4 datasets.

We used all 4 models for Datasets #1 and #2. Due to input length constraints, we could not use Google Flan XXL with Datasets #3 and #4. Similarly, we could not use Meta AI Meta AI OPT-IML-max-30b and OpenAI DaVinci with Dataset #4. Finally, to contain the costs of this experiment, we decided not to use OpenAI Davinci with Dataset #3.

Variability of the Generated Output

For each input text ingested in a model, an output of maximal length of 5 tokens is produced. There is no guarantee that the model will answer according to the provided question template: "¥nA. Yes¥nB. No¥nAnswer:". In our experiment, the text generated by Flan T5 XXL was either "A." (3415 times) or "B." (3195 times). Therefore, its answers could always be mapped either to "Yes" or "No". Similarly, the text generated by Davinci was: "a. yes" (3664 times), "b. no" (3015 times), or "a" (one time). However, the text generated by Meta AI OPT-IML-max-30b and ChatGPT was more variable and not all answers could be interpreted as "Yes" or "No": For 6 text inputs Meta AI OPT-IML-max-30b generated the string "highlight the parts ("; in 641 cases, ChatGPT generated text that could not be mapped to either "Yes" or "No". Some of the most frequent cases are reported in Table 5.

ChatGPT Generated Text	Count
the answer is not	164
the agreement does not	143
a. no	55
it depends on the	54
a. no,	41
the agreement grants the	33
it is not specified	30
it depends. the	21
it depends on whether	13
b. yes,	13
the agreement states that	10
the agreement grants re	7
it is not explicitly	5

Table 5: Examples of text generated by ChatGPT that could not be mapped to either "Yes" or "No".

Results

Table 6 shows the results of our experiment.

Dataset	Model	Num. of answers not mapped	True Yes	True No	False Yes	False No	F1 Score
#1	Google Flan T5 XXL	0	2989	2252	79	313	0.94
#1	Meta AI OPT IML MAX 30b	0	2785	2094	237	517	0.88
#1	OpenAI ChatGPT	129	3024	1823	424	233	0.90
#1	OpenAI Davinci	0	2896	2081	250	406	0.90
#2	Google Flan T5 XXL	0	404	497	13	133	0.85
#2	Meta AI OPT IML MAX 30b	0	395	427	83	142	0.78
#2	OpenAI ChatGPT	26	469	422	73	57	0.88
#2	OpenAI Davinci	0	441	432	78	96	0.84
#3	Meta AI OPT IML MAX 30b	6	1954	1492	835	1318	0.64
#3	OpenAI ChatGPT	230	2816	1747	430	382	0.87
#4	OpenAI ChatGPT	256	3078	1868	494	421	0.87

Table 6: F1 Scores for the four models applied on the 4 datasets build with original ContractNLI dataset.

Observations

1. **The importance of the retrieval component:** All models performed best on Dataset #1 with the shortest input text and saw performance decline when the length of the input increases. Therefore, in production it is crucial to have good retrieval components that identify the relevant documents to make an input text as short as possible [1];
2. **Google Flan T5 XXL vs Meta AI OPT IML MAX 30b**: While one experiment is not enough to conclude that one model is better than another, among these two open source models, we were impressed by the performance of Google Flan T5 XXL, especially when considering that this model is almost 3 times smaller than that of Meta AI. The limitation of its maximal input length is another argument for a strong retrieval component.
3. **Davinci vs ChatGPT**: ChatGPT performed slightly better than Davinci in Dataset #2. The drop in performance of ChatGPT from Dataset #1 to Dataset #4 was minimal. One reason to prefer Davinci may be its ability to follow instructions and give less variable outputs. This may be an important factor if the answer needs to be processed automatically.

1: Window sliding techniques may be an alternative to a retrieval component. However, this technique may not be suitable if the evidence is split in pieces of text that are pages away from each other.

Conclusion (contribution by ChatGPT)

In this experiment, we compared four text-to-text generation models on a Yes/No question answering task using a dataset of non-disclosure agreements. We observed that the performance of the models degraded as the length of the input text increased, highlighting the importance of a good retrieval component to identify relevant documents and make the input text as short as possible.

Among the open source models, we were positively impressed by the good performance of Google Flan T5 XXL, especially considering its smaller size compared to Meta AI OPT IML MAX 30b. However, the limitation of its maximal input length is another argument for a strong retrieval component.

Overall, this experiment highlights the importance of carefully selecting and evaluating text-to-text generation models for specific tasks and considering factors such as model size, input length limitations, and output variability.

KPMG Ignition Tokyo, Inc.

This article was prepared by the following members of the Data / AI Guild:
S. Ahi, D. Aichara, R. Anand, G. De Leva, F. Khan, S. Sahoo.