INFO-H-512 : Current Trends of AI

ChatACE : answering questions on the rules of a student association using Large Language Models

WAFFLARD Guillaume (479740)
guillaume.wafflard@ulb.be

June 11, 2023

Abstract

We developed a web application, called chatACE. This app lets students ask questions about the official rules of the Association des Cercles Etudiants (ACE), a student association at ULB. The questions are answered by a Large Language Model. 3 models are proposed: GPT4, Vicuna, and Bloom. The first offers accurate and well formulate responses, while the two others remain more experimentals in this context. The Large Language Model is fed with chunks of the documents, that are selected according to their semantic similarities. Moreover, an option extends this functionality to let a user ask questions about any PDF file, as long as this document contains text.

Contents

1 Introduction **1**
1.1 Some context . . . 1
1.2 Inspiration from seminar . . . 1
1.3 Language Model and chatBots . . . 1
1.4 The application . . . 2

2 Large Language Models **2**
2.1 How it works . . . 2
2.1.1 Fine-tuning . . . 3
2.1.2 Inference vs following-instruction . . . 3
2.2 Which models should be selected for our project ? . . . 3
2.2.1 LLaMa . . . 3
2.2.2 Alpaca . . . 3
2.2.3 Vicuna . . . 4
2.2.4 Vigogne . . . 4
2.2.5 Bloom . . . 4
2.2.6 BloomChat . . . 4
2.2.7 InLegalBert . . . 4
2.2.8 GPT4 . . . 5
2.3 Selected models . . . 5

3 The project **5**
3.1 The main page . . . 5
3.2 Option ACE . . . 5
3.3 Parsing . . . 7
3.4 New PDF option . . . 7
3.5 Analyze the files . . . 7
3.6 User question . . . 8
3.7 Generate an answer . . . 8
3.8 Input prompt . . . 8

4 Results **9**
4.1 Example of execution . . . 9
4.1.1 GPT4 . . . 9
4.1.2 Bloom . . . 9
4.1.3 Vicuna . . . 9
4.2 Analyze of the results . . . 10
4.3 Limitation . . . 10

5 Conclusion **10**

1 Introduction

This document serves as a report for the project of the course INFO-H-512 : Current Trends of Artificial Intelligence. This section will introduce our work, its inspiration, and its purpose. Section 2 will introduce Large Language Models, explain what they are, and present the models we considered using for our project. The description of the application is presented in Section 3

1.1 Some context

The Association des Cercles Etudiants (ACE) is a collective of student fraternities, known as "Cercles étudiants," at the ULB (Université Libre de Bruxelles). It serves as a unifying body for the 31 folkloric student fraternities, regrouping them in one assembly, and fostering increased collaboration and communication. The ACE also represents these fraternities in discussions with university authorities and commercial suppliers.

As a nonprofit organization, the ACE operates under the legal status of ASBL (Association Sans But Lucratif), a status shared by each individual fraternity. The ACE is governed by official Statutes, a 37-page document submitted to Belgian authorities, and a 27-page Rules of Procedure document, known in French as "Réglement d'Ordre Intérieur" or ROI. Additionally, six Charters (e.g. *Charte Ecologie, Charte Alcool, and Chartes Egalité & Inclusivité, etc*) provide 23 pages of additional rules that all fraternity members are expected to adhere to. In total, the ACE is guided by 37 pages of Statutes and 50 pages of internal rules, all written in French.

These regulations play a crucial role in the ACE's operations. However, a current challenge is that very few individuals are fully familiar with all these rules. As a result, it's not uncommon for people to realize that certain events have violated some rules only after the event has taken place. The addition of new charters and the creation of more rules only exacerbate this issue.

1.2 Inspiration from seminar

As I became aware of this issue, my attention was drawn by the seminar *"How can AI help law? How can law help AI ?"* by Grégory Lewkowicz. This seminar discussed many topics linked to AI and law, such as Rules as Code (RaC) a new way of writing some laws that aim to change government rulemaking by creating a machine-consumable version of some types of government rules, to exist alongside the existing natural language counterpart. These official versions of machine-consumable rules could be directly consumed by third parties and citizens to get an unambiguous interpretation of the law. It also discussed how AI can find similarities between legal cases to help anticipate the outcome of the judgment, and also how laws will govern AI practices in the near future.

But another interesting part of the seminar was on how AI can help the citizen to better understand the laws. Currently, legal texts are composed in a specialized legal language that can be challenging to comprehend for individuals outside the legal profession, even for those who are native speakers of the language. Artificial Intelligence can help citizens to better understand legal texts, by reformulating them in a more human-friendly formulation or by answering questions with chatBots.

1.3 Language Model and chatBots

In recent times, the advancements in chatbots, otherwise known as conversational agents, have been truly remarkable. A few years back, these agents primarily relied on pattern matching to recognize the type of question and respond with a predetermined answer. However, the progress in Natural Language Processing and Language Models has led to the rise of a new kind of chatbot. Chatbots like ChatGPT, which rapidly gained significant popularity and mainstream acceptance, are capable of conducting conversations that mimic human interactions and can answer questions across a broad spectrum of subjects. The notable enhancements in these conversational agents are primarily attributed to the extensive training they receive on comprehensive, high-quality text datasets.

Figure 1: Screenshot from the application on the page `Teste-moi`. On this page, students can ask a question, and a Language Model provides an answer, based on Internal Ruling of the ACE. In this example, the user asked *Quel est le rôle de l'Asoociation des Cercles Etudiants ?* (In English, *What is the purpose of the Association des Cercles Etudiants ?*)

1.4 The application

After considering all these aspects, we came up with the idea of developing an application that could assist students in better understanding the statutes and internal rules of the Association des Cercles Etudiants, similar to an AI that helps citizens better comprehend laws. This application, called chatACE in reference to the famous OpenAI's chatbot ChatGPT, takes the form of a chatbot that is already trained to communicate naturally in French. We have enhanced it so that it has access to all the governing documents of the ACE.

The application has been developed as a webpage, which can be accessed at the following address: `https://chatace.streamlit.app/`. It can also be run locally from the source code, which can be found in a Github repository. The steps to run it locally are explained in the Readme file. However, it is important to note that this method requires detaining some API keys to run some models.

Designed as a practical application for real-world usage, it consists of multiple pages: a main page that provides an overview of the entire project (see Section 3), an explanatory page, and a simplified page in French that allows students to ask their questions and receive answers while concealing certain process steps and technical details (Figure 1).

The website was built using the Streamlit front-end library, designed to easily turn ML script into a web app, and hosted on the Streamlit Community Cloud service.

2 Large Language Models

2.1 How it works

A large language model is a trained deep-learning model that understands and generates text in a human-like fashion. The way these models work is by using the input prompt to predict the next most probable token (a piece of a word) in a sentence. To do this, the model employs neural networks, specifically, a type of network called a transformer, which is capable of understanding the context of words in a sentence. Once the model predicts the next token, it then adds that token to the input prompt and repeats the process to predict the following token. This process continues until the model generates a complete response or reaches a specified length. The neural networks used in these models are large and complex, containing thousands of layers. Each layer has a multitude of nodes or "neurons" that are interconnected, and the strength of these connections, known as weights, is what the model learns during training. These weights, or parameters, total in the billions. For instance, Meta's LLaMA model was trained on one trillion tokens and includes models with 7 billion, 13 billion, 33 billion, and even 65 billion parameters [1]. GPT3 has 175 billion parameters and GPT4 is said to have up to hundreds of times more, but the actual numbers have not been disclosed. Training these models is a substantial task, requiring enormous computational resources. The models are trained on vast datasets composed of high-quality texts, including Wikipedia articles, scientific papers from Arxiv, public domain books, and more. By ingesting such diverse and rich data, the models learn to understand and generate a wide array of linguistic structures and concepts.

2.1.1 Fine-tuning

Once an LLM has been trained, it is great at completing sentences with words that make sense. This process is called inference. However, it is not yet made to answer questions as does a chatbot. In order to do this, it must be fine-tuned. Fine-tuning is a process of taking a pre-trained model, which has been trained on a large dataset, and then training (or "fine-tuning") it further on a smaller, specific dataset. The idea is that the pre-trained model has already learned general features from the large dataset that can be useful for many tasks, and only needs a bit of tweaking to be applied to a specific task. For instance, you might take a model trained on a large amount of general English text (like GPT-3), and then fine-tune it on a smaller dataset of legal articles to create a model for generating legal texts.

2.1.2 Inference vs following-instruction

Inference in a language model context involves the generation of the most probable next word in a sequence. To transform a general language model into a following-instruction chatbot (like OpenAI did with chatGPT), it can be fine-tuned with questions-answer dataset. This enables the model to learn that its purpose is to provide answers to posed questions.

Alternatively, a well-crafted prompt can be used as input to simulate dialogue. For instance, an input might be presented as *"In a conversation between two people, Alan and Maria, the first asks [...]. The second responds: "* and the language model completes the sentence with a fitting response. This method can be effective within certain boundaries, but it carries a risk. The model can continue the sentence with words that can seem logical semantically, but for which the sense is factually incorrect. Another risk is that the model may continue the dialogue with a new question and response, reflecting the conversational exchange outlined in the prompt. This can result in the model generating a fictional dialogue between two characters rather than providing a straightforward answer.

2.2 Which models should be selected for our project ?

The choice of a specific model was complicated because multiple factors had to be taken into account. We are going to discuss here the models that we have considered, those we have chosen, and the reasons and factors that led to these choices.

2.2.1 LLaMa

In February 2023, Meta released its LLaMa model (Large Language Model Meta AI [1]) as Open-source. However, they did not release the associated weights and therefore, an intensive training of a few months should be done. This kind of training costs millions of dollars and only a small number of big companies can afford it. One week after the model release, the weights were leaked on the Internet [2]. These weights could be used for our models, however, the illegal use of an AI model for research purposes is ethically questionable. Moreover, these files weigh hundreds of Gigabytes, and the models run on expensive GPU such as the Nvidia's A100 GPU that can be rented for hundreds of dollars per hour on the cloud. This price can be affordable for a middle size company but does not fit a student project.

2.2.2 Alpaca

Standford's researcher fined-tuned LLaMa-7B (the smallest LLaMa model with 7 billion parameters) into an instruction-following and chat agent. While OpenAI fine-tuned GPT-3 into ChatGPT using expensive manual labor of human feedback, Standford's researchers used chatGPT to generate 52.000 questions-answers from chatGPT to fine-tune LLama into a model that can respond to questions and instructions.

2.2.3 Vicuna

Vicuna-13B is another model Open-source model trained by fine-tuning LLaMa. Fine-tuned with user-shared conversation from ShareGPT, a collaborative archive of chatGPT conversations, Vicuna-13B is supposed to respond as a chatbot. Its creator claim to have reached 90% of chatGPT and Bard (Google) quality, "outperforming other models like Alpaca more than 90% of cases"[3]. A Replicate server is hosting the Vicuna model and makes it callable with an API.

2.2.4 Vigogne

Vigogne[4] is a French alternative to Vicuna, that has been fine-tuned using a translated version of the fine-tuning question-responses dataset into French. We were interested by this model as we need a model that can interpret and respond in French. However, we were disappointed by the results, because it makes a lot of language mistakes and spelling errors.

Up until now, all the presented models were trained mostly in English (Vigogne is an exception, but its results were not satisfying, and LLaMa and its derived model Vicuna were trained with some documents in French, but not as a primary language). As we need to use an agent that can answer questions in French, based on documents written in French these models did not fit our requirements.

2.2.5 Bloom

BLOOM (BigScience Large Open-science Open-access Multilingual Language Model)[5] is a 176 billion parameters model developed by the BigScience Project, and was trained on the 28 petaflops Jean Zay (IDRIS) supercomputer in France. As its name indicate, this model to able to use multiple languages. It was trained on 46 different languages and 13 programming languages, and French represents a big part of the training dataset (10% of French against 30% English and the rest in other languages). This is a major advantage, considering that we require a model that can handle documents written in French.

2.2.6 BloomChat

Bloom is able to do inference, but isn't trained to answer questions or complete tasks (see Section 2.1.2). However, an instruction-following optimization for chat has been fine-tuned under the name BloomChat[6]. This model would have been ideal to use, as it is Open (open-source, open-science, open-access), multilingual (can make sentences in French) and can answer questions as a chatbot. Unfortunately, this model requires a lot of storage space to store the 176 billion weights parameters (over 360 Gb) and needs to run on GPU that we did not have at our disposal. Therefore, we will have to make do with the Bloom "normal" inference model that is usable by calling as API.

2.2.7 InLegalBert

The fine-tuning process can be applied multiple times. For example, BERT (Bidirectional Encoder Representations from Transformers), the language model introduced by Google [7], has been fine-tuned with legal texts written in English to be more familiar in this domain. The authors of this fine-tuning, I. Chalkidis et al. called this model Legal-BERT[8]. Then, researchers from the Indian Insitute of Technology used the same principle and fine-tuned Legal-Bert with Indian laws documents. This model, InLegalBert[9], is now optimized for Indian legal texts written in English. In brief, it is a general-purpose LLM in English language that has been fine-tuned for legal documents, that in turn has been optimized for Indian legal documents. This model cannot be used by our project, as it only works in English and is trained with Indian laws and not Belgian laws. However, if we had sufficient computational power at our disposal, we could have considered doing the same in our context. For instance, using a language model that is trained with French (such as Bloom, see Section 2.2.5), and then fine-tuning it with legal text written in French (e.g. text from the European Commission, or legal texts from a french-speaking country), and finally fine-tune it with Belgian legal text, such as the Belgian laws or specifically the laws concerning the ASBL. However, this option would not have been enough for our use case, for the reasons explained in Section 3.5.

2.2.8 GPT4

We cannot talk about LLM and chatbots without citing OpenAI's ChatGPT and its model GPT4. Release in March 2023, this model is still considered as of today as the more accurate Large Language Model. As a commercial product, ChatGPT is a proprietary software, and its weights are kept secret. However, OpenAI enables its use by using an API and also proposes to companies to fine-tune its model for specific needs.

2.3 Selected models

Considering the pros and cons of each model discussed in the previous section, our limited computational power and storage space, and the requirements of the project (such as the fact that the model must be able to communicate in French), we decided to select GPT4, BLOOM, and Vicuna. These three models are accessible from an API, which makes them usable from a simple laptop. Moreover, this allows us to develop the user interface as a webpage, hosted on a free cloud service, as the heavy computation is done on the API providers' servers (respectively OpenAI for GPT4, HuggingFace for Bloom and Replicate for Vicuna).
On the first hand, GPT4 is known to provide the best results but is proprietary and has a payable API. On the other hand, both Bloom and Vicuna are Open-Source and are accessible with a free (but limited API). Bloom is a multilingual model doted with 176 billion parameters but is not fine-tuned as a chatbot. On the opposite, Vicuna is supposed to act as a chatbot but was not primarily trained in French. Therefore these two models are less effective than GPT4, be we thought it was interesting to try them and to compare their results.

3 The project

As explained in Section 1.4, we developed an application that assists students in better understanding the statutes and internal rules of the Association des Cercles Etudiants de l'ULB (ACE), where they can ask questions about the rules, and receive a straightforward answer. The application is hosted on Streamlit Community Cloud and can be accessed from this page: `https://chatace.streamlit.app/`. The app can be used straightforwardly on the page *Teste-moi*. This is the page accessed by the students that are looking for an answer but who are not interested in the technical details and the underlying processes. On this page, shown in Figure 1, the user simply asks a question, and an answer is generated by the GPT4 model. The page *Explanations* provides a general explanation of how the app works (all the information is already contained in this report). Finally, the main page provides an overview of the entire project, displaying the technical steps and providing a chatbot-like interface (Figure 2). The following subsections will detail all the steps that are executed.

3.1 The main page

On the main page, the user is first invited to select an option between two choices : either *Statuts et règlements de l'ACE*, the main topic of the project, or *Upload a new document* if the user wants to query a new PDF document.

3.2 Option ACE

If the user selects the option *Statuts et règlements de l'ACE*, a drop-down list shows all the ruling documents of the Association des Cercles Etudiants, such as the Statutes, the Internal Rules (Réglement d'Ordre Intérieur), and multiple Charters, as described in the Section 1.1). The user can display the document of its

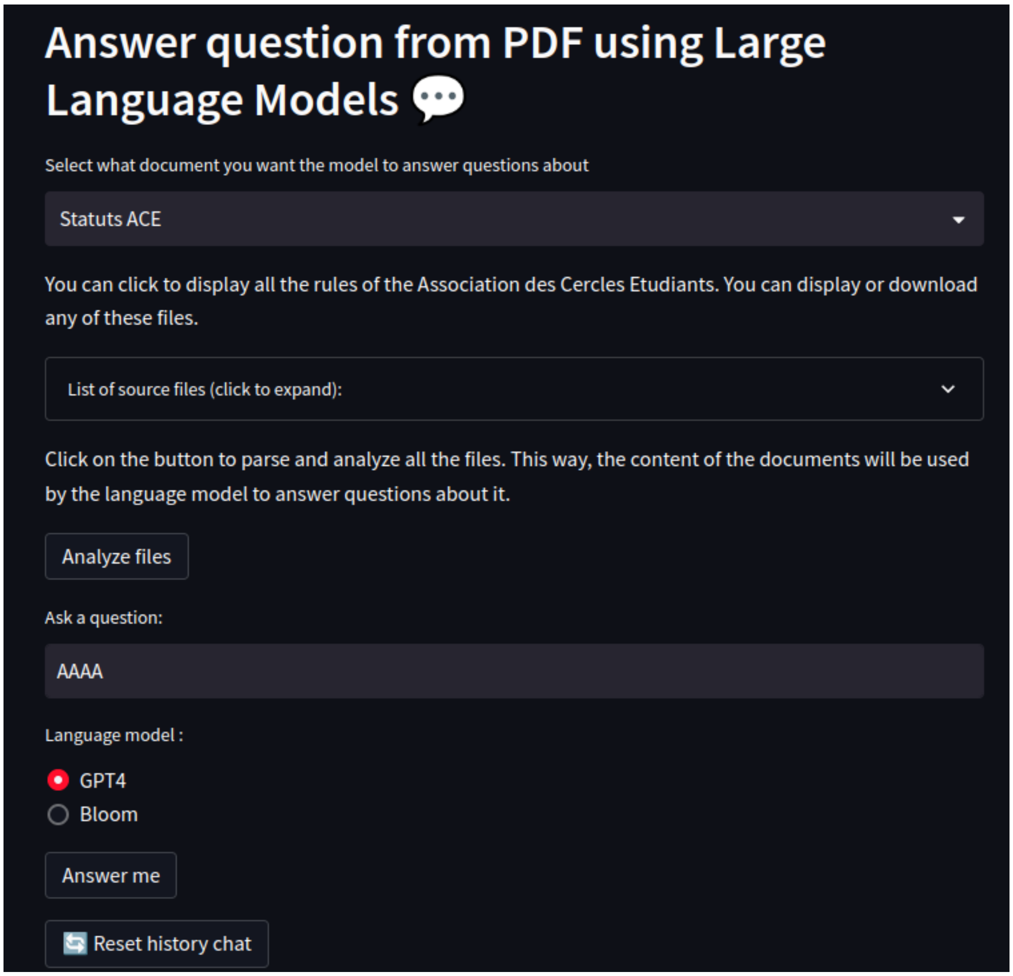

Figure 2: Main page of the application. The selected option is *Statuts et règlements de l'ACE*. The list of documents can be displayed by expanding the `List of source files`. In the bottom part, a question can be asked, and the user can select the language model of its choice (GTP4, Bloom, or Vicuna). The question and the answer are displayed in a chat-like interface.

choice in an embedded PDF viewer, or download any document he or she wants. The user must then click on the button "Analyze files". The text will then be extracted from the PDFs, parsed and cleaned, and then used to create a knowledge database.

3.3 Parsing

One of the main challenges faced by data scientists and engineers working with Large Language Models (LLMs) is acquiring high-quality data for training. These models are typically trained using documents like Wikipedia entries and scientific articles. However, most of the content available on the internet is not suitable for this purpose. For example, posts on platforms like Reddit often contain inappropriate language or harmful views, such as racist or sexist opinions, making them unsuitable for use in training LLMs.

Even in the context of our relatively small-scale project, we've encountered related challenges. It's not that the ACE's statutes and internal rules contain unsuitable content, but our primary data source - PDF files - come with their own formatting issues. Headers and footers intrude into the main body of the text on each page. Additionally, hyphenation at the end of lines often results in subwords that do not exist, further complicating the data preparation process. It is important to identify these words and to recombine them. We use the following regular expression (regex) `(\w)-\n(\w)` to identify these cut words, and merge the two parts together. Another problem is the header and foot of each page, which appear in the middle of the text. Those vary from one page to another, but also from one document to another (such as the date, the page number, and the type of document). We had to create additional regex to locate them and remove them automatically.

3.4 New PDF option

If the user selected the option "upload new PDF", then he or she can upload a new PDF of its choice. In order for the program to work efficiently, the PDF must contain text, and must contain a text such that questions can be asked about (*e.g.* an article, a report, a book, etc). As the structure and layout of the uploaded document are unpredictable, it's not feasible to implement custom modifications as we do for the ACE option. Therefore, there is a risk that some interfering text causes a lower quality of answers. However, we can still perform some rudimentary cleanup like reformatting hyphenated words or substituting newline characters (`\n`) with standard spaces.

We decided to propose this option because it is an extension of the actual project. In reality, the underlying process is exactly the same, so addressing queries related to any PDF is essentially an expanded application of handling queries about the ACE files, with the former serving as a general case.

3.5 Analyze the files

Refining the models specifically for text comprehension from the ACE documents was not feasible due to the limited size of the documents (over 80 pages), which is not enough for fine-tuning. Furthermore, repetitive content is necessary for effective concept retention. Therefore, to process queries, it becomes necessary to include the text, or at least parts of it, as part of every query input. The text, once extracted from the PDFs (and cleaned, in the case of the ACE option), still isn't immediately usable by the models. Simply inputting the entire text to the models isn't a viable option either, as the text's length exceeds the models' memory capacity. By "memory," we refer to the allowable length of the prompt provided to the model. The precise maximum length varies across different models but usually falls within a few thousand tokens (equivalent to one or several pages). Importantly, computational effort (and consequently, cost) is directly proportional to the length of the input and corresponding output, making brevity an important advantage.

Our solution to this problem lies in identifying the most pertinent parts of the text for each query. These sections (only) are then sent to the models, along with the user's question, allowing the models to analyze the content and respond appropriately. The method of identifying these sections involves the use of semantic similarities.

We split the entire text into manageable chunks of approximately 1000 tokens each. In the ACE option, we group together rules articles as far as possible, since they form meaningful clusters. We then utilize embeddings (a method of capturing word meanings) to interpret each word. These embeddings are sourced from OpenAI. The meanings of each chunk are stored in a 'knowledge base' or 'vector of meaning', facilitated by Langchain's FAISS library (Facebook AI Similarity Search, a tool designed for efficient similarity search and clustering of dense vectors).

3.6 User question

The user is then invited to enter a question in an input zone. After that, he or she can select a language model, between the three propositions (GPT4, Bloom, and Vicuna). The choice of the proposed language models has been largely discussed in Section 2. Then, a click on the `Answer me` button will use the selected language model to generate an answer, that will appear after a few seconds. The list of questions and associated answers from the model are displayed in the form of a chat. The chat history can be reset at any time by clicking on the reset button.

3.7 Generate an answer

When the user activates the `Answer me` button, the FAISS library from Langchain, which was used to create our "knowledge base" or "vector of meaning" (see above), is now used to locate the chunks with the highest semantic similarity to the question asked. This selection step is crucial as it helps to identify the most relevant segments of the entire text. These selected chunks are integrated into an input prompt, structured to guide the model's response. This prompt comprises not only the semantically closest chunks of the original text but also the user's question. Additionally, we may include some extra sentences to provide further context for the model, enhancing its ability to generate a relevant and accurate response.

The model uses the prompt as input of its transformers, which are a neural network type, with billions of parameters. The input is the next token in the sequence. This process, described in Section 2 is repeated after each token until the answer is complete.

3.8 Input prompt

It is important to provide a well-constructed prompt to large language models, as those generate responses based on the input they receive. A clear, concise prompt helps guide the model toward the kind of response we are looking for. Large language models don't have access to real-world context or knowledge beyond their training data cut-off. They don't know the specific context in which they're being used, nor the previous interactions. So it is needed to provide the context in the prompt. In particular, when the language model is not trained for question-answering (as it is the case for BLOOM) but is used as a chatbot, it is especially important to guide the direction.

GPT4 : For GPT4, we use the concept of Chain proposed by Langchain. In essence, chains allow to combine multiple components together to create a single, coherent application. We create a chain that uses the GPT4 models and passes as input the chunks and the user question. It will then format the prompt internally.

Bloom : As Bloom is not a trained as chatbot, it is important to format the prompt in order to get an answer inspired by the provided chunks. We use the following prompt : *"Dialogue entre Alan et Marie. Marie a accès aux informations suivantes dans le réglement: Début des informations : " + chunks + "-fin des informations à disposition de Marie. Alan pose la question : " + question + " Marie lui répond naturellement : "*
However, the results are not always satisfying. Sometimes the model provides an accurate answer but then continues asking a new question and providing the answer as if it was continuing the dialogue. Sometimes, it may also just repeat textually the content of the more relevant chunk.

Vicuna : Vicuna is trained as a chat, so we can give it instructions. But these instructions must be well formulated. *"Tu es mon avocat. Je te fournis un texte de réglement. Sur base de ce texte, réponds à ma question que je te poserai après. Voilà mon texte : " + chunks + "Sur base de ces informations, réponds à cette question, en citant les articles dont il relève : "* + question

4 Results

The results provided by the large language models may highly vary from one model to another. Here are some example of execution, on different queries for the different models.

4.1 Example of execution

4.1.1 GPT4

Question : *Comment demander une carte ACE ?*
Answer : "La demande d'une carte ACE doit être effectuée auprès du Bureau du Conseil d'Administration de l'ACE afin que celle-ci soit examinée et votée en Assemblée Générale ou en Réunion ACE."

Question : *Que fait la comission cantus ?*
Answer : "La Commission Cantus est compétente pour coordonner l'organisation d'activités liées au folklore chantant sur les campus de l'Université libre de Bruxelles, en particulier celles qui se déroulent dans la salle Eric Schelstraete, attribuer et distribuer des dates de location pour les Guildes membres, accepter ou refuser des demandes de location de la salle Eric Schelstraete par des organisations autres que les Guildes membres, voter tout changement de prix lié aux fûts et gobelets de la salle Eric Schelstraete, et réviser la Charte Cantus."

4.1.2 Bloom

Question : *Comment demander une carte ACE ?*
Answer : "La demande de carte ACE se fait par le biais du formulaire de demande de carte ACE, qui est disponible sur le site interne de l'ACE. Ce formulaire est à renvoyer au Bureau du Conseil d'Administration de l'ACE, par courrier postal ou électronique, au plus tard le 15 septembre de chaque année. Le Bureau du Conseil d'Administration de l'ACE se réserve le droit de refuser une demande de carte ACE, sans avoir à motiver sa décision. Le refus est notifié."

Question : *Que fait la comission cantus ?*
Answer : "La Commission Cantus a pour but de coordonner l'organisation d'activités liées au folklore chantant sur les campus de l'Université libre de Bruxelles, en particulier celles qui se déroulent dans la salle Eric Schelstraete. Art. 14. De la composition de la Commission. §1er. La Commission Cantus est présidée par la·le Délégué·e Cantus de l'ACE. Elle est composée des représentant·e·s des Guildes membres, ainsi que des éventuell·le·s délégué·e·s guilde des Cercles membres de l'ACE. Chaque Guilde membre dispose d'un droit de vote.

4.1.3 Vicuna

Question : *Que fait la comission cantus ?*
Answer : "La commission cantus a pour but de coordonner l'organisation d'activités liées au folklore chantant sur les campus de l'Université libre de Bruxelles, en particulier celles qui se déroulent dans la salle Eric Schelstraete. Elle est composée des représentants des guildes membres, ainsi que des délégués des cercles membres de l'ACE. Chaque guilde membre dispose d'un droit de vote. Les anciennes représentantes des

guildes membres peuvent également assister aux réunions de la commission en tant que membres consultatifs, mais elles n'ont pas le droit de vote."

Question : *Quelles sont les règles de ventes d'alcool ?*
Answer : "A.Les cercles peuvent vendre des bières spéciales contenant jusqu'à 6° d'alcool lors de leurs événements festifs, mais doivent les proposer pour 2 à 3 fois le prix de la bière. B. Les cercles peuvent vendre des bières allégées contenant jusqu'à 3° d'alcool pendant leurs événements festifs. C. Les cercles sont interdits de vendre des boissons à base d'alcool fort ou de spiritueux contenant plus de 12° d'alcool lors de leurs événements festifs. D. Les cercles peuvent vendre des boissons énergisantes lors de leurs événements festifs. E. Les cercles sont interdits de vendre des boissons énergisantes lors de leurs événements festifs"

4.2 Analyze of the results

We can see that the results of GPT4 are really accurate. From small chunks of text, it extracts the meaning and is able to answer correctly the vast majority of asked questions. Bloom, on the other hand, is less efficient. In the first example, the answer is mostly correct but contains inaccuracies. First, the demand must be expressed by mail, and not through a form. Secondly, the request must be done 15 days before the next General Assembly, and not before the 15 of September. These facts that are not based on the provided data are called *hallucinations.* The model generates texts that make sense semantically, but the meaning is inaccurate. In the second example, the model simply generates an extract from the provided chunks with almost no reformulation.

4.3 Limitation

If this application, when it uses GPT4 model, provides accurate answers that are very well the vast majority of the time, it is important to keep in mind that such models work as black-box (meaning that there is no way to visualize how the models got to the output), and therefore they can't be considered as 100% trustful. They can be a useful tool to get quickly an answer to a question, but in the context of law and legal text, their answer should have no legal value, at least with the current development.

5 Conclusion

We have successfully developed an application that leverages the content of PDF files to generate answers to inquiries. The performance of the application is dependent on the model employed. In the case of GPT4, the responses are often accurate, making the application a valuable tool to simplify the research of information on such documents. As a result, GPT4 is the chosen model for the simplified `Teste-moi` page.

On the other hand, the BLOOM and Vicuna models have got variable outcomes. The project asks them to retrieve information from a bunch of sentences, and then answer questions accordingly. These are not their initial purpose and, therefore, the results are limited. This highlights the need for such models to be used for tasks they have been specifically trained for. The use case of ACE is a concrete example that AI can help people to better understand legal texts, and help them to efficiently research information.

References

[1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[2] J. Vincent, "Meta's powerful ai language model has leaked online - what happens now?," Mar 2023.

[3] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023.

[4] B. Huang, "Vigogne: French instruction-following and chat models." `https://github.com/bofenghuang/vigogne`, 2023.

[5] B. Workshop, :, T. L. Scao, A. Fan, C. Akiki, and . more, "Bloom: A 176b-parameter open-access multilingual language model," 2023.

[6] T. C. SambaNova Systems, "BLOOMChat: a New Open Multilingual Chat LLM."

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[8] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, (Online), pp. 2898–2904, Association for Computational Linguistics, Nov. 2020.

[9] S. Paul, A. Mandal, P. Goyal, and S. Ghosh, "Pre-trained language models for the legal domain: A case study on indian law," in *Proceedings of 19th International Conference on Artificial Intelligence and Law - ICAIL 2023*.