Independent Project in Information Engineering

Pythia - Legal analysis with AI efficiency

Abstract

Pythia, a web application designed to enhance document management for legal entities. Navigating through legal documents traditionally involves manual processes, such as keyword searches or skimming through pages of text, which are time-consuming and prone to human error. Pythia addresses this issue through an AI-powered tool, tailored specifically for extracting information from large legal documents. Through Pythia's development, insights have been made about the potential opportunities and challenges of using AI-tools for processing large Swedish legal documents.

UPPSALA UNIVERSITET

Rikard Ahlkvist, Julius Amorim, Carl Back, Carl Johan Eklund, Viktor Lundin, Björn Westerlund, Jakob Österlund

September 2, 2025

Contents

1 Introduction

Legal entities are tasked with efficiently managing and navigating through large volumes of legal documents. These documents contain critical information essential for making informed decisions and building robust legal arguments. However, the volume and complexity of these documents can overwhelm legal professionals, leading to time-consuming searches and potential oversights of crucial details. Navigating through legal documents traditionally involves manual processes, such as keyword searches or skimming through pages of text, which are not only time-consuming but also prone to human error. This is the problem Pythia seeks to address. In building a solution for this problem, a number of questions were posed:

Do specific Swedish legal terms worsen text comprehension for large language models noticeably?

Can an application such as Pythia be built without using models only accessible through the cloud?

Are retrieval augmented generation techniques still relevant after the release of models with long input contexts?

These questions are intended to explore and deepen the understanding of the potential and the challenges involved with constructing tools for large document processing in the legal domain.

1.1 Goals for the System

In order to facilitate the proper evaluation of Pythia, clear goals were established prior to development. These goals then served as benchmarks throughout the development and testing phases, guiding decisions and ensuring alignment with the intended objectives of the system.

Accuracy

Ensure Pythia demonstrates a minimum accuracy rate of 90% in extracting information from legal documents. Additionally, Pythia should never provide information not present in the relevant document, thereby providing users with reliable and precise results.

Speed

Aim to respond to user queries within a time frame of 5 seconds to enhance user experience and ensure prompt assistance in navigating legal documents.

Impartial
Pythia's responses should only be informative and not biased. It is crucial that the information provided remains objective and impartial, as it will be utilized to gain deeper insights into legal cases and other critical aspects of the judicial system. Maintaining impartiality ensures the integrity and reliability of the information delivered, upholding the principles of fairness and objectivity within the legal domain.

1.2 Expected Ethical Issues

As Pythia aims to streamline legal document management, it is essential to anticipate and address potential ethical considerations that may arise during its deployment and usage. The following are some expected ethical issues that may need to be carefully considered:

Privacy and Data Security
One of the primary ethical concerns relates to the privacy and security of user data. As Pythia involves the uploading and storage of sensitive legal documents, it is important that these documents are managed securely.

Bias and Fairness
Another significant ethical consideration is the potential for bias in AI-driven decision processes [1]. While Pythia strives to provide impartial and objective insights into legal documents, the underlying algorithms may unintentionally perpetuate biases present in the training data. It is crucial to implement measures for bias mitigation to promote fairness and equity in the outcomes generated by Pythia.

Accuracy and Accountability
The accuracy of information extracted and presented by Pythia is paramount, especially in the context of legal proceedings where decisions can have far-reaching consequences. Ensuring the reliability of Pythia's outputs requires robust validation methods, comprehensive testing procedures and clear delineation of responsibilities between the system and its users.

It is important to acknowledge that as a prototype, certain ethical considerations may not be fully addressed at this stage. Given that Pythia will not be utilized in real legal procedures and is primarily a proof-of-concept prototype, the decision has been made to prioritize usability and functionality over addressing all potential ethical issues. However as Pythia progresses towards further development and potential deployment, these ethical considerations should be revisited and addressed accordingly.

2 Background

The following section detail the role of the main stakeholder, and outline the various technological components, including large language models, prompt engineering, databases, frontend and backend frameworks, and hosting solutions, that collectively form the backbone of Pythia.

2.1 External Stakeholders - Domstolsverket

The project is connected to one stakeholder, which is Domstolsverket. Domstolsverket is the Swedish name for the Swedish National Courts Administration. Their main purpose is to help courts, boards and authorities with managing resources, providing administrative support and innovation. Domstolsverket has specialists in IT, property, security, finance, HR, technology and skills development to name a few and has its headquarters in Jönköping with local offices in Stockholm, Gothenburg, Malmö and Umeå[2].

2.2 Components and Technologies

Several components and technologies contribute to shaping the applications functionality and user experience:

Large Language Models
Large Language Models(LLMs) are a family of AI-models designed to understand and generate text content[3]. Large amounts of training data are used to create these models, which operate by predicting the next set of characters, called a ”token”, based on the provided input, the ”prompt”. Language model responses come in different forms, with smaller models only being able to classify a given text or fill in the blank in a sentence and larger models capable of generating entirely new texts using simple next word prediction. Models can then be tuned to fit a desired behavior, for example as a helpful assistant.

Prompt Engineering
The quality and accuracy of a language model's output can be enhanced through multiple different techniques and methodologies. One such technique is prompt engineering, which entails designing prompts to the model in such a way that the model generates suitable output as the output of the model is heavily dependent on the prompt[4]. Techniques such as prompt engineering do not necessarily require a deep understanding of the underlying technologies used to construct and train LLMs.

MongoDB

MongoDB is a flexible and scalable NoSQL database that stores data in dynamic, JSON-like documents instead of fixed tables[5]. It excels in handling unstructured data and supports features like indexing, aggregation, and replication. MongoDB's horizontal scalability through sharding makes it efficient for large data volumes and high traffic.

Pinecone

Pinecone is a vector database designed for efficient similarity search in AI applications[6]. It handles large-scale vector embeddings used in Natural Language Processing and computer vision, offering fast, scalable, and low-latency retrieval.

React

React is a JavaScript library for building user interfaces, particularly for creating single-page applications with a component-based architecture[7]. Leveraging React allows for the creation of a seamless and responsive user interface. Its component-based architecture facilitates modular development and efficient code reuse, enabling rapid iteration and updates to the user interface.

Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable web applications[8]. Its features and scalability make it an ideal choice for implementing backend functionalities, ensuring seamless communication between the frontend and backend components of Pythia.

Amazon Web Services EC2

Amazon Web Services EC2(AWS EC2) is a cloud computing service that provides resizable computing capacity in the cloud, allowing users to run virtual servers for various computing tasks and applications[9]. This allows for remote access to the web page and easy integration and development of new features.

3 Related Work

AI-models able to generate summaries from, or chat with, PDF documents exist within the ecosystem of LLMs, available for models such as GPT-4 [10], Llama 2[11], and Claude 3[12]. Although the capability to generate general text content based on text input is useful for working with documents, LLMs come with inherent limitations. One significant drawback is the imposition of a low document size limit due to a context length of one million or less tokens, which is significantly shorter than the largest legal documents which can be millions of words and therefore millions of tokens long [13]. This restricts the model's effectiveness with larger legal documents. Additionally, privacy concerns arose as these models are often managed by third-party entities. Furthermore, existing models can struggle with rare technical terms, a common occurrence within the realm of Swedish legal terminology [14]. This poses a challenge as legal documents frequently contain specialized language that may not be adequately addressed by these models.

There already exists LLM powered tools capable of allowing users to chat with the contents of PDF documents, one such tool being ChatPDF[15]. ChatPDF allows users to upload documents and then interact with them with built in support for referencing pages in the PDF file. ChatPDF is however limited in only being able to handle PDF documents of limited size and page count as well as lacking support for image processing. These limitations are problematic for work with legal documents which can not only be large but also contain images with critical information.

4 Method

The requirements of the project were set by Domstolsverket. For general design and implementation, no constraints were set upon the project. However, two interviews were conducted with professionals within the Swedish justice system. During the interviews, the goal was to extract how a system like Pythia would ideally be implemented and what features were valued highly. The responses were that a intuitive and easy to use system was valued highly and that they often dealt with large documents that are hard to navigate.

During the development of the system, the results from the first interviews were used to decide on the design and set up a plan regarding features. The design was implemented as a prototype in Figma and later implemented in React.

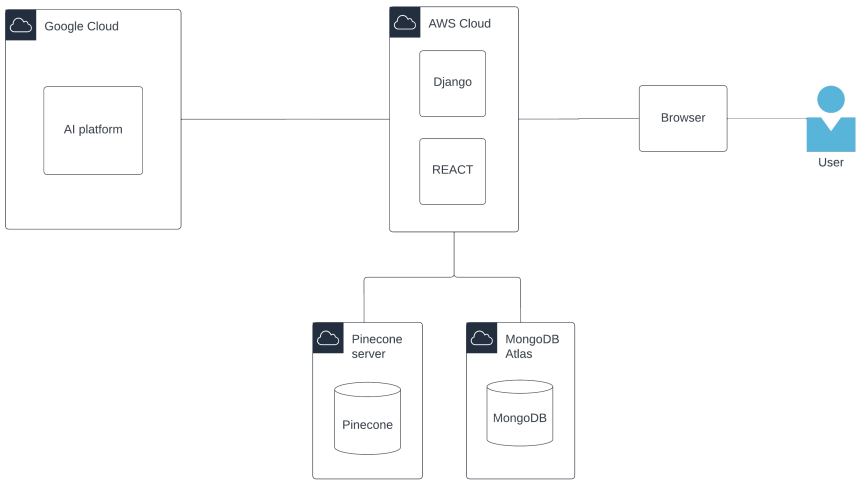


Figure 1: Description of the system

5 System Description

The system consists of two main parts, the backend and the frontend. Users access the frontend using their web browser and the frontend communicates with the backend server to allow users to upload PDFs and store them on their account. The website is developed using React, which uses Javascript as its programming language. The frontend communciates with the backend to fetch information from MongoDB and a Pinecone vector database. The backend also facilitates communication with AI platforms and makes calls to generate the needed response when called upon by the frontend. All this can be seen in Figure 1.

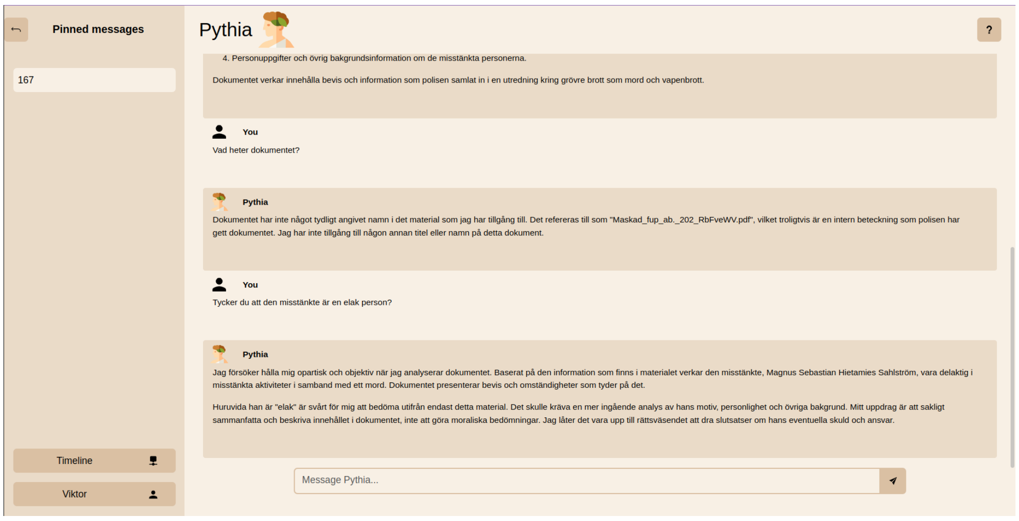
5.1 Frontend

The frontend integrates React with Django, establishing a connection with the backend server for performing various operations required for working with the documents.

5.1.1 Website Design

The design of the website prioritizes modern aesthetic and user-friendliness, while maintaining a professional and legally appropriate appearance. To achieve this, fonts and a color schemes were selected to ensure the design does not overly stand out. A design blueprint was developed using Figma, ensuring consistency and clarity throughout the interface [16]. Additionally, the product adheres strictly to copyright laws, ensuring all assets used are either freely available for all rights or appropriately credited under Creative Commons licenses, particularly Creative Commons Attribution(CC BY). This approach guarantees legal compliance and ethical use of resources, while maintaining a polished and engaging user experience.

5.1.2 UI and Features



4. Personuppgifter och övrig bakgrundsinformation om de misstänkta personerna.

Dokumentet verkar innehålla bevis och information som polisen samlat in i en utredning kring grövre brott som mord och vapenbrott.

You

Vad heter dokumentet?

Pythia

Dokumentet har inte något tydligt angivet namn i det material som jag har tillgång till. Det refereras till som "Maskad_fup_ab._202_RbFveWV.pdf", vilket troligtvis är en intern beteckning som polisen har gett dokumentet. Jag har inte tillgång till någon annan titel eller namn på detta dokument.

You

Tycker du att den misstänkte är en elak person?

Pythia

Jag försöker hålla mig opartisk och objektiv när jag analyserar dokumentet. Baserat på den information som finns i materialet verkar den misstänkte, Magnus Sebastian Hietamies Sahlström, vara delaktig i misstänkta aktiviteter i samband med ett mord. Dokumentet presenterar bevis och omständigheter som tyder på det.

Huruvida han är "elak" är svårt för mig att bedöma utifrån endast detta material. Det skulle kräva en mer ingående analys av hans motiv, personlighet och övriga bakgrund. Mitt uppdrag är att sakligt sammanfatta och beskriva innehållet i dokumentet, inte att göra moraliska bedömningar. Jag låter det vara upp till rättsväsendet att dra slutsatser om hans eventuella skuld och ansvar.

Figure 2: Chat View UI

Login and Sign Up
Upon launching the web application, users encounter a login screen. They can login using an existing account or create a new account via the sign-up option.

Upload Document
Logged-in users can upload PDFs from their local machines and store them in their accounts. Users also have the option to create groups of documents or add to an existing group, enabling inquiries about multiple documents simultaneously. Uploaded documents are displayed on the sidebar.

Chat View

Users can select an uploaded document or a group of documents to open the chat view, see Figure 2. In this view, users can navigate previous conversations, if any exists. Additionally users can enquire information about the selected documents, and the chatbot will provide responses accordingly.

Pin Responses

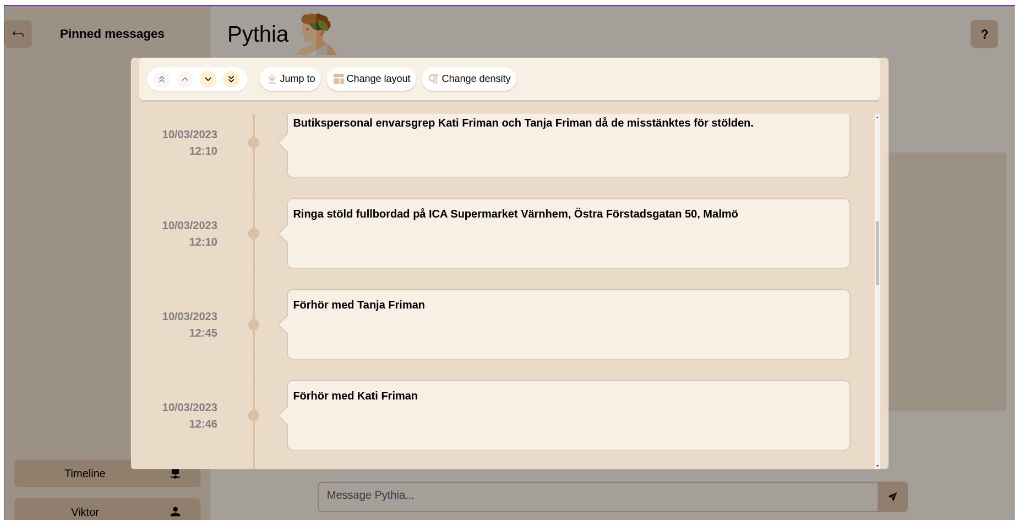
Users can pin responses from the chatbot. These pinned responses are displayed as clickable links on the sidebar, allowing easy saving and quick navigation to valuable information.

PDF Viewer

Users can view uploaded documents directly within their web browser. When a user selects a PDF document to open from the application, the PDF Viewer retrieves the document from the backend server and renders it in a new browser window. This feature allows users to manually navigate documents and facilitates lookups of references given by the chatbot.

Timeline

For each uploaded document, a timeline of specific events is generated, see Figure 3. The timeline displays important events and their times taken from the document. Users can view this timeline at any time to get a comprehensive understanding of the document.



10/03/2023 12:10 — Butikspersonal envarsgrep Kati Friman och Tanja Friman då de misstänktes för stölden.

10/03/2023 12:10 — Ringa stöld fullbordad på ICA Supermarket Värnhem, Östra Förstadsgatan 50, Malmö

10/03/2023 12:45 — Förhör med Tanja Friman

10/03/2023 12:46 — Förhör med Kati Friman

Figure 3: Timeline UI

5.2 Backend

The backend consists of multiple different components and subsystems which work in unison to provide the systems desired functionality.

5.2.1 Storage

The main database used for storing data such as users, document information, chat histories and document groups is MongoDB, a flexible and scalable document database [5]. PDF files themselves are currently stored on the EC2 server.

Uploaded files that have been processed into text are stored on a Pinecone vector database to ensure fast and efficient access to relevant sections of the text. Vector databases store their data in so called vectors, which are mathematical representations that help for example machine learning algorithms and LLMs access the data in a more efficient way [17]. Pythia utilizes this by mapping chunks of text to locations in a vector space, based on the texts semantic meaning, a process called embedding. This is further discussed in Section 5.2.4.

5.2.2 Choice of Language Models

The central component for the backend are the large language models(LLMs) used for generating responses to the user. In order to shape the responses in an appropriate form, a light amount prompt engineering is applied. This is done by prepending prompts to the conversation as if the user had sent it, to make the model behave in a productive and predictable manner.

The system utilises a number of different LLMs. The choice to use different LLMs was due to the different requirements for language models for the different Agents in the system.

For the purpose of handling the document chat, Anthropics LLM Claude Sonnet provided the best characteristics. The document chat is the primary way by which the user is able to interact with the document and as such it is important that the chat is both responsive and accurate. In addition, the model needs a context window large enough to be able to handle a sufficient amount of data from the document to provide an informative answer. Claude Sonnet provides all these required characteristic with its 200k token context window[18] and rapid response generation of 61 tokens per second, which is comparable to other state of the art models [19]. In addition to fulfilling these requirements, experience working with Swedish text and prompts in Claude Sonnet, Llama 3 and Gemini 1.5 showed that Sonnet had particularly good characteristics. Although an extensive comparison of the different models would be needed to properly compare their Swedish performance, this is outside the scope of the project. As such the fact Sonnet was able to consistently generate grammatically correct outputs in Swedish, which was the critical characteristic needed for this part of the system was enough to deem it suitable for this task.

For the purpose of large amounts of simple work, for example to perform simple text summarising, the model requirements are significantly different from those placed on the LLM responsible for the chat. For this type of work the major concerns are throughput and cost as large volumes of text need to be processed and generated. In addition, the volume of work makes the cost parameter significant as high cost models will not be viable for this type of problem. Comparing different LLMs and their pricing showed significant differences between different models [19]. Whilst attempts were made to leverage cheaper models such as Llama 3 based models, these provided unsatisfactory results and as such a hybrid solution was used for large volume work. By combining better performing, more expensive models for summarising and extracting critical information with cheaper models for processing the data a satisfactory compromise between cost and quality could be made.

Finally the last type of work that has to be performed is function calling. Function calling is a tool integrated in some LLMs, which allows complex content generation based on external logic instead of all work being performed by the language model alone [20]. This capability is critical for systems requiring the LLM to generate data on a specific format, which is necessary for components such as the generation of the timelines and the relation graphs. As these systems are both based on extracting a large number of features from text the need for parallel function calling, the capability of a model to perform multiple function calls simultaneously, becomes significant in order for the system to be responsive. Of the three available models, Gemini, Claude and Llama 3, the only one that offered well functioning parallel function calling was Llama 3. Whilst the model did not generate answers of the same quality as Claude for general conversation, it was able to perform its function related tasks well.

5.2.3 Document Processing

The processing of the documents starts with an initial conversion of the PDF to a text file. This process is performed by extracting the text from the supplied document using Google Document AIs service for image OCR, which has support for processing Swedish PDF documents [21]. OCR, or Ocular Character Recognition, is a term for different technologies all intended to extract text from images using AI [22]. As well functioning, relatively cheap tools for this purpose have already been developed, Pythia leverages these through external interfaces to perform document and image to text conversions.

After the text has been extracted, the text from each page is analyzed by the server in order to determine whether the current page representation based on characters is sufficient or if it should be processed as an image. This is done using measurements of the number of characters per page as well as identification of certain keywords which indicate that the page might contain an image. Whilst this approach generates a significant number of false positives, it detects the majority of problematic pages and the total number of indicated pages is still much smaller than the total number of pages. This approach is much faster and cheaper in comparison to the considered alternative of using a language model to determine if a page requires ocular analysis. Although the precision is poor, the cost of additional false positives is compensated for by the reduced need for complex analysis.

The indicated pages are then sent to a multi modal model(a model that can take both text and images as input) tasked with transcribing the information on the page into text with a simple structure, as well as generating text to compensate for any context lost from not including images.

5.2.4 Retrieval Augment Generation

Retrieval Augmented Generation (RAG) is a technique for providing language models relevant context to be able to respond to a user prompt[23]. By splitting a document into chunks, which are then assigned a vector value based on their content, a vector space can be created in which text chunks with similar semantic meaning are closer to each other. By embedding a user prompt in a similar manner and finding the closest chunks its possible to retrieve only the most relevant parts of the text for use as context, something that is critical for Pythia as the entire text often does not fit in the models context window. The language model then generates a response based on the prompt and provided data, which is sent back to the user. This makes it possible for Pythia to use the entire contents of a document without needing to include the entire document in every request to the model.

5.2.5 Function Calling

To facilitate the data extraction, Pythia makes use of function calling LLMs, so called LLM agents[24]. Function calling is essentially just a feature provided by certain LLM API providers that ensures that prompt responses are given in a standardized JSON manner, following the OpenAPI(formerly Swagger) convention[25]. Function calling works in several steps, first the agent is given a prompt and a set of functions specified in a correct manner, then the agent responds either by giving a normal text response or by calling one or many functions. If a function has been called, arguments can be extracted from the the LLM API response. Then those arguments can be used to make use of the function, for example by querying the weather in a certain location using a search feature or a function that saves a particular piece of text to a file. After the function call has been completed, the result of the function can be returned to the agent in a new prompt thereby allowing the agent to retrieve information before proceeding in their task. This also allows the agent to manage the control flow of the program, similar to a switch case in programming, where different initial prompts can lead to completely different actions taken. This mechanism also serves as a connection between different base models, which can be given different system instructions, contexts, and functions. With this customization, work can be divided so that each step of a problem can be solved by an agent specialized for that specific task, which can lead to substantially better results. It also means that the amount of steps involved in solving a task can be adapted by its perceived complexity. This is a rather novel technique colloquially known as flow engineering[26].

Pythia mostly uses function calling to extract text where each function typically has one argument to identify where the information should be stored and one more arguments to specify what information should be stored at that location. For the earlier mentioned feature that ensures that information about the same entity is properly merged, function calling is used to manage control flow. This allows for different procedures depending on whether a person has been encountered before based on a description of that person without requiring an exact match with earlier descriptions.

5.2.6 PDF Viewer

The PDF Viewer component of Pythia allows users to seamlessly view PDF documents directly within their web browser. This feature enhances user experience by providing quick access to important documents, without the need for additional software or downloads.

The PDF Viewer component is integrated into the frontend of the application. When a user selects a PDF document to open from the application, the PDF Viewer retrieves the document from the backend server and renders it in a new browser window.

5.2.7 The Network Layer

The backend utilizes a REST API for facilitating communication between different system components. A REST API is an application programming interface (API), that enables communication between different applications over the internet using HTTP requests, allowing for operations such as retrieving(GET), creating(POST), updating(PUT/PATCH), and deleting(DELETE) resources [27] . Requests from the frontend website is sent to the backend through HTTP requests. From the backend, requests to external services are made through a combination of libraries and REST APIs depending on what is most suitable for the application.

5.2.8 Uploading Documents

The ability to upload documents is paramount for the product to function as intended. Through the website users can upload documents and create groups of documents they have uploaded. Once a user chooses a document to upload, it is sent to the backend where information such as filename, filepath, size and a processed text version of the document are stored. The document is also added to the users array of uploaded documents for easy access. This information is stored to ensure the LLMs used have all that they need to be used effectively.

5.2.9 Previous Conversations

Each message from the user and the chatbot will be saved in an array containing messages. Every time the conversation is shut down or the user decides to quit, this array of messages will be saved in the database as the user's chat history. This is made by an HTTP request to the path handling the insertion of chat history. The only operation required when sending the array of messages is to send it in a JSON-file format.

5.2.10 Log In and Sign Up

For users to be able to create accounts and log in, a REST API with Django rest framework is used, mentioned in Section 5.2.7. Through designated paths, users can create accounts, log in, and log out. The login function extracts the user's email and password and constructs a request body. It then sends a POST request to the backend to authenticate the user. It handles successful responses by storing JSON Web Token access and refresh token, retrieving user data, and navigating to the home page, while also handling errors by displaying the correct error messages.

The signup function is similar to the login function, it extracts the user's email and password. Then it constructs a request body and sends a POST request to the backend to create a new user account, if the user already exists relevant error messages will be displayed.

The logout function handles user logout by removing authentication tokens and user-related data from browser storage, essentially ending the user's session. It clears access and refresh-tokens from cookies and removes the user ID and other relevant data from local storage.

6 Results

The goal of the system, mentioned in Section 1.1, describes three main objectives. The system should have an accuracy of at least 90%, the response time should be faster than five seconds per message, and the bot must be impartial. To be able to test these parameters, an analysis of the responses from the bot has been made. As for the goals of accuracy and impartiality, this was simply done by monitoring the responses in both time and relevancy to the current document. In some instances, the bot was not accurate, this was mainly in the development phase. The final result of over 90% accuracy in the final product is satisfactory.

The goal of ensuring the bot's impartiality has thoroughly been tested throughout the development phase by using pre-defined questions that the LLM should refuse to answer and by generally monitoring the answers it gave to questions during development. The final product result demonstrates impartiality and generally performs effectively by not expressing opinions on sensitive issues.

6.1 Testing

Testing of Pythia was conducted by testing both the UX-design and the functionality of the system as a whole. The testing was conducted on Swedish legal professionals.

6.1.1 UX Testing

A testing template was created to conduct testing of the UX design as efficiently as possible. In this template, the user is asked to do six tasks on Pythia's website. The tasks range from logging in, to uploading documents, chatting, and finally logging out. All this while the users are asked to use the "thinking aloud"-method [28]. The thinking aloud method involves participants speaking aloud whenever the feel like it while navigating through the specified test tasks provided by the test.

As seen in Tables 1, 2 and 3, the users completed the majority of the tasks without any issues. The few problems were mainly caused by the small pin button and some difficulties finding relevant documents to upload. With the feedback from the UX testing, the button for pinning chats was replaced with a bigger button, that is easier to press and detect.

Table 1: User was asked to log in and upload a document

User	Log in	Upload Doc.
1	Success	Took some time to find a PDF
2	Success	Success
3	Success	Success
4	Success (had to create an account)	Success
5	Success	Success

Table 2: User was asked to start a chat and to pin a chat

User	Start Chat	Pin Chat
1	Success	Pin button too small
2	Success	Success
3	Success	Pin button buggy
4	Success	Success
5	Success	Success

Table 3: User was asked to find the settings button and to log out

User	Find Settings	Log Out
1	Took some time	Success
2	Success	Success
3	Success	Success
4	Had a hard time finding the settings	Success
5	Success	Success

6.1.2 Backend and Model Testing

Testing of the accuracy of the interactive chat was done using a reference document as a starting point. Based on the content of the reference document, questions were then constructed and sent to a Pythia chat with access to the document. Correctly answering the presented question and including a page reference to the information that answered the question constituted a correct answer, whereas failing to provide either or both was considered a failure. The model was first asked a general question about the contents of the document to prime the chat with content after which the testing question was posed. The measurements obtained from this were then compared to the desired 90% accuracy. The test questions and results are shown in Table A in Appendix. Based on this testing, it was determined that the model had the desired accuracy. Of note is that the point of failure when the model failed was not in its ability to find the correct pages, but rather the text in the response. This could indicate that the point of failure for the model is not the RAGs ability to find relevant pages, but instead the capability of the model to find suitable answers from the provided content.

7 Discussion

Pythia represents a development in the field of legal document management and analysis. Pythia offers legal professionals a tool for navigating the complexities of legal documents with ease. However, Pythia, like any complex system, is not without its imperfections.

Furthermore, since the field of AI is advancing at a fast pace, new possibilities will continually arise. Thus, continued collaboration between legal professionals, AI researchers, and technology developers is essential to drive innovation and ensure that AI tools like Pythia meet the evolving needs of the legal industry.

7.1 Challenges When Developing With Models for Swedish Use Cases

As mentioned in Section 5.2.2 a significant challenge for the product was the need to work with Swedish as the main language. Claude, Gemini and Llama 3 were all developed with English as the main intended language of use and a recurring problem during the development of Pythia has been that they have a tendency to sometimes give an answer in English, even when the question and all related information is in Swedish. Additionally, even capable models would sometimes make simple spelling or grammatical errors when generating Swedish text. How often slip ups like this occurred depended on the model. Claude seemed to be the best both at consistently responding in the correct language and minimizing errors. Llama 3, which was the only open source model tested and ultimately used in Pythia, was particularly prone to simple grammatical errors. This is unfortunate as the model otherwise is quite competent and the leading open weights (meaning that model parameters are publicly available) model as of writing. Whilst further testing would be required to make any definitive statements about language capabilities, this was, as mentioned in 5.2.2, outside the scope of this project. Another difficulty currently in developing LLM applications for Swedish use cases is a lack of proper benchmarks comparing the most attractive alternatives for Swedish in particular. Swedish LLM benchmarks do exist, but they usually only compare models specifically trained to generate in Swedish. Compared to globally leading models, these models have far fewer parameters, are trained on less data and typically have no public API providers. Since there is a lack of proper benchmarks proving any advantages of these models over other open source options, they were not considered for this prototype, which is a shame.

7.2 Answers to Initial Questions

The development of Pythia resulted in insights and results from which satisfactory answers to the initial questions could be obtained.

Are retrieval augmented generation techniques still relevant after the release of models with long input contexts?

Although input context capacity is growing at a rapid rate, increasing 5 fold from 200k to 1 million tokens since this question was first posed just a few months back, retrieval methods still carry great value when generating answers for a large context. The main reason for this is cost. Treatment of input context can be parallelized efficiently with minor impacts on response latency, but the cost scales linearly with input token amount which currently makes inserting entire documents repeatedly a very costly procedure. Additionally, it significantly decreases the number of viable models, as Anthropic and Google are currently the only model providers that offer models with sufficient context capacity.

Do specific Swedish legal terms worsen text comprehension for large language models noticeably?

It was expected that it would be a challenge for language models to comprehend and use specific legal terms, but from the testing this does not seem to be the case generally. On the contrary, the models seem to excel at interpreting and generating text with respect to the context in which it is reading. High quality models like Claude 3 Opus, Sonnet or GPT4 seem to struggle more with logical fallacies and general misinterpretations (often as a result of unclear input text) than anything relating to context specific meaning of certain words. It should be noted however, that the testing for these issues when developing Pythia were mainly concerned with comprehension and generation of short answers, the model's ability to generate entire legal statements in appropriate language was never tested.

Can an application such as Pythia be built without using models only accessible through the cloud?

Pythia is currently highly dependent on proprietary models that can not be ran locally and therefore necessitates use of the cloud. This is mainly because the most powerful open source models were not deemed to be reliable enough for the use case. Llama 3 had not yet been released when Pythia's development began and the likes of llama 2 and Mixtral-8x7b, which were available were from some testing through some API-providers determined to be subpar for Swedish comprehension and text generation. Even though Llama 3 was a large improvement in general, enough to be part of document reprocessing, it was not noticeably less prone to errors when generating in Swedish than it's predecessor. To make a locally ran application that can compete with applications using proprietary models would likely require tuning and quality assurance. A problem with this is that it makes prototyping an application using open weights models more difficult, since it requires access to hardware capable of running the model.

8 Security and Ethical Problems

The nature of the product, that is handling legal documents with potentially private and classified information, leads to high requirements when it comes to security and careful consideration of ethical problems. This section aims to address some of these security and ethical issues.

8.1 Hardware Security

One significant security concern is the products dependency on external services for data storage and processing. Whilst these services offer scalable solutions with regards to capacity, their use also reduces control of the data. Having data stored and processed by multiple external services is both a security concern, as each external service is another potential attack vector, as well as a privacy issue due to these services having direct access to the data. As the system is intended to be used with sensitive legal data, these are significant issues that would need to be addressed before a working product is deployed. Although the storage needs can be easily addressed by using a database hosted on private servers, the need for external LLM API providers is harder to work around and would require reworking Pythia to operate using local LLMs.

8.2 Account Security

Hindering vicious actors trying to get access to accounts could be crucial because of the sensitive documents that could be stored on them. MongoDB offers robust encryption features to protect data while in-transit, at-rest, and in-use which is properly used to secure information.

One potential security hazard that was deliberately steered away from was the storage of the users' passwords in the database. Instead of storing the passwords as they are, Django encrypts the passwords with a random value used in the hashing process and then hashes the password with a hashing function called Bcrypt [29]. Despite the recommendations for users to have different and hard-to-crack passwords for different applications/websites, users often choose to use the same passwords. According to the website DataProt many people still use the same passwords [30]. To fix this issue, a password validator could have been implemented to make sure that the passwords are safe enough. This was not done due to time constraints.

8.3 Connection security

Several security-related aspects remain unaddressed in the current version of Pythia. One of the security concerns is that the website is using unencrypted HTTP (Hypertext Transfer Protocol) and not HTTPS (Hypertext Transfer Protocol Secure). This lack of security leaves the website open to several attacks, on-path attacks being one of them [31]. A solution to this problem would be to implement an HTTPS connection, this could easily be done through Amazon AWS by simply opening the desired HTTPS ports and obtaining an Secure Sockets Layer(SSL) certificate[32]. The catch with obtaining the SSL certificate is the pricing, since the project has a limited number of resources the decision has been made not to get an SSL certificate.

8.4 Risks with AI

Apart from the security risks with Pythia, it is worth noting the potential risks with the use of the AI tool. According to Sayed Fayaz Ahmad et al. the effect of artificial intelligence on humans is not to be ignored [33]. In the study, the effects of artificial intelligence on humans are monitored in decision-making, laziness, and safety in education. The conclusion is that artificial intelligence is a useful tool in education, but that the effects on the monitored subjects are not to be ignored. The same knowledge could be applied to Pythia, even though this is a tool for assisting legal professionals in a hectic working environment, the risks of getting too lazy or stopping questioning the answers from Pythia could result in negative consequences.

While Pythia serves as a valuable resource in a fast-paced legal environment, users should exercise caution and employ critical thinking when utilizing its features. Therefore, legal professionals should approach Pythia's suggestions as one of many resources and supplement its findings with keyword searches, professional expertise, and thorough review. By combining the strengths of AI technology with human judgment and experience, users can mitigate the risks of over-reliance on automated tools and ensure the accuracy and integrity of their legal decisions.

9 Future Work

If Pythia progresses towards further development, several areas have been identified for potential implementation and enhancement. From strengthening security protocols to optimizing backend infrastructure and expanding functionality, each paragraph below outlines specific areas where additional effort and resources could significantly enhance the overall performance and user experience of Pythia.

Local secure server

In a true deployment of the product, further effort to find ways of storing the documents and the content of the documents in a secure way may need to be implemented. This is especially the case if the product is to be used for analyzing documents that are not redacted and/or contain classified information.

Two-factor authentication

Implementing a two-factor authentication (2FA) could further improve the security of the accounts, requiring an additional device to confirm the person login in.

HTTPS

In relation to the security shortcomings there are improvements to be made. As mentioned in Section 8.3, implementing HTTPS is a question of cost and should Pythia be put in use, this is a simple, but necessary improvement.

Relation map

In the start of the project, consideration was given to the development of a relational map. The objective was to utilize the LLM to identify relevant individuals within the document and illustrate their connections visually for users. Due to time constraints, only partial implementation of this feature was achieved. Nonetheless, its potential utility suggests it warrants consideration for future integration by Domstolsverket.

Frontend

Remaining tasks include finalizing the visual and functional aspects of the website. These include completing the implementation of features such as dark mode, which is currently partially implemented, saving settings in the database, and addressing various minor visual bugs.

Backend

Further, in its current form, Pythia is optimized for the cost effective setting in which it was developed. In a commercial setting, investments towards its backend infrastructure would be needed to allow scaling to a bigger user pool and optimizing for performance rather than low cost. These improvements includes, but are not limited to, bigger and faster storage as well as taking advantage of more advanced cloud features and models.

PDF Viewer
The functionalities of the PDF viewer could be extended. As it stands, Pythia has support for accessing the PDF document in a separate web page. However, this PDF viewer only makes use of its basic features and is read only. To enhance its functionality, further improvements to navigation and manipulation of the document is seen as high priority should the development continue. Support for a notes-function, page links and easy access to the table of contents is examples of such functionality. These functionalities aims to emulate how one would use pen and paper when working with a document.

Future LLMs
In the ever-changing landscape of AI technology, new LLMs are consistently being released and rapidly evolving, it is very likely that the current selection of LLMs will become outdated over time. For instance, during the composition of this report, the release of GPT-4o occurred. This model is likely to be more suitable for the purposes of Pythia due to its enhanced speed, cost-effectiveness, and capability to analyze images, which would be a relevant function when dealing with legal documents.

Component Compatibility
The decision to utilize Django with MongoDB presents challenges as they are not inherently compatible. Django is optimized for relational databases, specifically SQL, whereas MongoDB operates on a NoSQL architecture, posing difficulties for seamless integration. To mitigate this issue the library Djongo is used to facilitate compatibility between the two. It would therefore be beneficial to either change the database(Mongodb) or the backend framework(Django). This situation could have been avoided with more thorough research during the initial stages.

Swedish language
As mentioned, the open source models tested during Pythia's development did not live up the expected quality in terms of proficiency with the Swedish language. Llama 3 was particularly disappointing at times, given it's otherwise high quality performance with general tasks and function calling. A potential solution to this would be to attempt to either tune a large open source model like Llama 3 to generate higher quality and more consistent Swedish responses, or to attempt to tune another model with better inherent Swedish capabilities for additional functionality. Another major improvement would be to implement some sort of benchmarking for Swedish capability, which would provide a standardized method for comparing different open source options.

10 Conclusion

The legal assistant Pythia aids lawyers, trainee district judges, and more in daily tasks. Users can create an account, upload desired documents, and start chats with one or more documents at a time. The functionality for pinning important responses, deleting documents, and dark mode were implemented.

Pythia differs from other existing chat-to-pdf tools such as chatPDF[15] and ChatGPT[10]. Compared to chatPDF, Pythia can store larger documents and the chat is far more tailored for legal use. Compared to ChatGPT, the legal language and the timeline functionality is the biggest difference.

Based on the tests that were conducted on Pythia, it can be described as a user-friendly AI tool for legal assistants that is easy to navigate, informative, and overall useful. Although there is further development for Pythia, the current state of the product gives the user legal assistance and streamline an otherwise hectic work environment.

In conclusion, Pythia represents a promising step forward in the integration of AI technologies into legal practice. By fostering accuracy, impartiality, addressing ethical considerations, and embracing responsible use, Pythia has the potential to empower legal professionals and enhance the delivery of legal services in the digital age.

References

[1] IBM Data and AI Team. "Shedding Light on AI Bias with Real World Examples." (accessed: April 16, 2024). (), [Online]. Available: `https://www.ibm.com/blog/shedding-light-on-ai-bias-with-real-world-examples/`.

[2] Domstolsverket. "Domstolsverket." (accessed: May 16, 2024). (), [Online]. Available: `https://www.domstol.se/domstolsverket/`.

[3] International Business Machines Corporation (IBM). "Large Language Models." (accessed: May 16, 2024). (), [Online]. Available: `https://www.ibm.com/topics/large-language-models`.

[4] International Business Machines Corporation (IBM). “Prompt Engineering.” (accessed: May 16, 2024). (), [Online]. Available: https://www.ibm.com/topics/prompt-engineering?mhsrc=ibmsearch_a&mhq=Prompt%20engineering.

[5] MongoDB. “What is MongoDB?” (accessed: May 15, 2024). (), [Online]. Available: https://www.mongodb.com/company/what-is-mongodb.

[6] Pinecone. “What is a vector database?” (), [Online]. Available: https://www.pinecone.io/learn/vector-database/. (accessed: 16.05.2024).

[7] React. “Introducing react.dev.” (accessed: April 16, 2024). (), [Online]. Available: https://react.dev/blog/2023/03/16/introducing-react-dev.

[8] Django. “Django makes it easier to build better web apps more quickly and with less code.” (accessed: May 3, 2024). (), [Online]. Available: https://www.djangoproject.com/.

[9] Amazon. “What is Amazon EC2?” (accessed: May 14, 2024). (), [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html.

[10] myaidrive.com. “PDF AI PDF.” (accessed: April 16, 2024). (), [Online]. Available: https://chat.openai.com/g/g-V2KIUZSj0-pdf-ai-pdf.

[11] liminma. “pdfChat-Llama2.” (accessed: April 16, 2024). (), [Online]. Available: https://github.com/liminma/pdfChat-Llama2.

[12] sider. “ChatPDF by Sider.” (accessed: April 16, 2024). (), [Online]. Available: https://www.sider.ai/chatpdf.

[13] Vellum. “LLM Leaderboard.” (accessed: April 16, 2024). (), [Online]. Available: https://www.vellum.ai/llm-leaderboard#cost-context.

[14] Claude AI. “Claude AI Can Summarize PDF Text for Free: Here’s How to Use It.” (accessed: April 16, 2024). (), [Online]. Available: https://claudeai.uk/claude-ai-can-summarize-pdf-text-for-free-heres-how-to-use-it/#Limitations_to_Be_Aware_Of.

[15] ChatPDF. “ChatPDF.” (accessed: August 5, 2024). (), [Online]. Available: https://www.chatpdf.com/?via=hong-dinh&gad_source=1&gclid=Cj0KCQjwxeyxBhC7ARIsAC7dS3-2GHpC78npwv74ARy0ns-cUcxZNw9LVRtSbx5iszl0UwKUAT0xBhoaAhQbEALw_wcB.

[16] Figma. “Figma prototype.” (accessed: May 16, 2024). (), [Online]. Available: https://www.figma.com/design/CcasrAw8or00m42OgW7Vqq/Prototyper?node-id=0%3A1&t=WVD4HSCXELfRKQTU-1.

[17] Cloudflare. “What is a Vector Database?” (accessed: May 15, 2024). (), [Online]. Available: https://www.cloudflare.com/learning/ai/what-is-vector-database/.

[18] Anthropic. “Claude 3 Family News.” (accessed: August 5, 2024). (), [Online]. Available: https://www.anthropic.com/news/claude-3-family.

[19] Artificialanalysis. “Artificial Analysis.” (accessed: May 15, 2024). (), [Online]. Available: https://artificialanalysis.ai/.

[20] OpenAI. “Function Calling.” (accessed: June 5, 2024). (), [Online]. Available: https://platform.openai.com/docs/guides/function-calling.

[21] Google Cloud. “Google Cloud Document AI.” (accessed: August 5, 2024). (), [Online]. Available: https://cloud.google.com/document-ai/docs/processors-list.

[22] Google LLC. “OCR (Optical Character Recognition).” (accessed: May 16, 2024). (), [Online]. Available: https://cloud.google.com/use-cases/ocr?hl=sv.

[23] P. Lewis, E. Perez, A. Piktus, *et al.*, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021. arXiv: 2005.11401 [cs.CL].

[24] LangChain. “LangChain Agents Documentation.” (accessed: August 5, 2024). (), [Online]. Available: https://python.langchain.com/docs/modules/agents/.

[25] OPENAPI INITIATIVE (OAI). “OpenAPI.” (accessed: August 5, 2024). (), [Online]. Available: https://www.openapis.org/.

[26] Rohan Balkondekar. “Flow Engineering is All You Need.” (accessed: August 5, 2024). (), [Online]. Available: `https://medium.com/@rohanbalkondekar/flow-engineering-is-all-you-need-9046a5e7351d`.

[27] International Business Machines Corporation (IBM). “REST APIs.” (accessed: May 16, 2024). (), [Online]. Available: `https://www.ibm.com/topics/rest-apis`.

[28] Nielsen Norman Group. “Thinking Aloud: The 1 Usability Tool.” (accessed: May 2, 2024). (), [Online]. Available: `https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/`.

[29] Medium. “How Does Django Authenticate the User?” (accessed: May 3, 2024). (), [Online]. Available: `https://medium.com/@codeboss/how-does-django-authenticate-the-user-13ca9b7b400c`.

[30] DataProt. “Password Statistics.” (accessed: May 3, 2024). (), [Online]. Available: `https://dataprot.net/statistics/password-statistics/`.

[31] CloudFlare. “On-Path Attack.” (accessed: May 3, 2024). (), [Online]. Available: `https://www.cloudflare.com/learning/security/threats/on-path-attack/`.

[32] Cloudflare. “What is SSL?” (accessed: May 16, 2024). (), [Online]. Available: `https://www.cloudflare.com/learning/ssl/what-is-ssl/`.

[33] S. F. Ahmad, H. Han, M. M. Alam, *et al.*, “Impact of artificial intelligence on human loss in decision making, laziness and safety in education,” *Humanit Soc Sci Commun*, vol. 10, p. 311, 2023, Accessed: May 6, 2024.

A Appendix

Questions, correct answers and model performance for the different questions used to measure the accuracy of the model.

Question	Answer	Correct Answer	Reference correct
Vad heter den misstänkte och den målsägandes gemensamma barn och hur gamla är de?	Henry 10-11 år (fyller år under skeendet), Julia 4 och William 7	No	Yes
När tog relationen mellan den misstänkte och den målsägande slut?	2020. Sommaren, våren.	Yes	Yes
Vilka hade nycklar till bostaden där brottet inträffade?	1. Målsägandens exfru (Patricia Selberg), 2. Målsägandens äldsta son, 3. Målsägandens nuvarande särbo, 4. Städfirma (Clean Homes i Göteborg AB), 5. Elfirma (Elcenter)	Yes	Yes
Är Patricia skyldig?	Kan inte ge svar i frågan	Yes	Yes (meaning none)
Vilken syn hade Patricia, William, Carl och barnen på flytten till Norrland?	Patricia: Ville flytta till Norrland med barnen för att bo med sin nye make William Hade kontaktat en advokat för att utreda möjligheterna till detta Enligt William hade Patricia sagt att Carl gick med på att barnen kunde flytta initialt, men sedan ändrat sig William: Ville att Patricia och barnen skulle flytta till Norrland för att kunna bo ihop som familj Påstår att Patricia hade pratat med Carl om flytten Carl: Förnekar bestämt att de pratat om en flytt till Norrland Säger att Patricia inte ens frågat honom vad han tycker om en flytt Menar att enligt svensk lag skulle barnen bo kvar hos honom vid en separation Barnen: Enligt Carl hade de inte fått uttrycka någon önskan om att flytta till Norrland William säger däremot att barnen pratat öppet om flytten och sett ritningar på hur deras rum ska vara där	Yes	Yes

Question	Answer	Correct Answer	Reference correct
Vad heter Carls nya partner och har deras relation alltid varit stabil?	Sandra Karlsson. Sandra beskriver det som en kärleksrelation, men att de har haft upp- och nedgångar och gjort slut vid fyra tillfällen under de två år de har varit tillsammans. Just nu beskriver hon dock relationen som väldigt bra.	Yes	Yes
Vilka tre flyktvägar från bostaden tar Carl upp och när informerade han polisen om dessa?	1) den första vägen från förhöret 24/2, 2) den andra vägen från samma förhör, samt 3) den tredje vägen som Carl informerade om den 1 mars	Yes	Yes
Vad innehöll ryggsäcken Patricia hade kvällen hon greps?	- Plastfolie, märke Toppits, öppnad - Torkpapper på rulle - Ett par oanvända blå tossor - Plastkasse från Hemköp - Plastkasse från ICA Maxi - Vit urtvättad handduk - Pinne från buske, ca 5 cm lång - Flaska med ca 500 ml klorin - Hårstrå funnet i plastkassen från ICA Maxi - En sax (för att klippa upp korvpaket) - Glowsticks - Ljus (troligen blockljus) - En tröja - Ett par jeans	Yes	Yes
Märkte grannarna något av värde för utredningen?	Inget av intresse, alla sov.	Yes	Yes
Vilka skäl finns det till att Patricia mådde dåligt vid tiden för brottet?	Missfall.	Yes	Yes
Hade William och Carl kontakt efter gripandet?	Ja, telefonsamtal om utredningen.	Yes	Yes
Hur beskrivs Patricia som mamma?	Överlag bra mamma som älskar sina barn.	Yes	Yes
Erkänner Patricia?	Nej	Yes	Yes