Universitat de Girona
Escola Politècnica Superior

Treball Final de Màster

Estudi: Màster en Ciència de Dades

Títol: Ajustament d'un model generatiu de llenguatge per a la creació de xatbots personalitzats per administracions públiques

Document: Memòria

Alumne: Martí Mas Fullana

Tutor: Josep Suy Franch
Tutor: Miquel Tarragona Margarit

Departament: Departament d'Informàtica, Matemàtica Aplicada i Estadística
Àrea: Intel·ligència Artificial

Convocatòria (mes/any): Setembre 2024

Universitat de Girona
Escola Politècnica Superior

TREBALL FINAL DE MÀSTER

Ajustament d'un model generatiu de llenguatge per a la creació de xatbots personalitzats per administracions públiques

Autor:
Martí MAS FULLANA

Setembre 2024

Màster en Ciència de Dades

Tutors:
Josep SUY FRANCH
Miquel TARRAGONA MARGARIT

Resum

Agraïments

Per començar vull agrair molt especialment a ...

Índex

Índex de figures

Índex de taules

CAPÍTOL 1

Introducció

1.1 Antecedents

1.1.1 Introduction to Dialogue Systems

Dialogue systems, also known as chatbots, have experienced a significant step-change in the last few years. Initially these systems were based on predefined rules and decision trees [1, 2], limiting their capacity for understanding and answering user queries in a natural and flexible manner. These rudimentary systems, commonly referenced as rule-based chatbots, might have been enough for simple tasks, but could not have managed the full complexity and variability of natural language.

1.1.2 Towards Language Models

As the first machine learning-based language models appeared, such as the Sequence-to-Sequence (Seq2Seq) model [3], and more recently the transformer-based models such as GPT (Generative Pre-trained Transformer) [4, 5], the capacity of chatbots to understand and generate natural language has improved significantly. These models are trained on large datasets of text, learning the complex patterns and structures of language, and are able to generate text that is coherent and contextually relevant.

1.1.3 GPT and its Contribution

The GPT model [5], developed by OpenAI, has been one of the most notable advances in this field. GPT uses the transformer architecture [4], which is a type of neural network that is particularly well-suited for processing sequences of data, such as text. Its capacity for generating coherent and contextually relevant responses has been leveraged in a wide range of applications, from virtual assistance to automated content generation.

1.1.4 Retrieval Augmented Generation (RAG)

One of the most recent advances in the integration of language models has been the use of retrieval augmented generation (RAG) [6]. RAG combines the strengths of information retrieval from databases with the generative capacity of language models. In this context, when a user query is received, the system first retrieves relevant information from a database, and then the language model generates a coherent and precise response based on this information. This approach has been shown to improve the accuracy and relevance of the responses generated by chatbots [6].

1.1.5 Applications and Benefits of RAG-based Chatbots

RAG-based chatbots offer a variety of benefits compared with more traditional systems. They are able to generate responses that are more coherent and contextually relevant. In this way users are both less frustrated and more satisfied. These systems also allow the chatbots to have access to newer, more up to date information than the data the model was originally trained on, as the data provided to the information retrieval component can be updated by simply adding new entries to the database. This makes the chatbot more adaptable and flexible, and allows it to provide more accurate and relevant information to users, reducing the necessity of performing full or partial retraining of the model, which can be prohibitively expensive.

1.2 Objectives

The main goal of this project is to develop an advanced chatbot system that uses GPT (Generative Pre-trained Transformer) technology and RAG (Retrieval Augmented Generation) to provide responses to user queries based off of the content of a database. This general goal can be broken down into the following specific objectives:

1. **Pick an appropriate GPT Model**
    - Choose a GPT model that is well-suited for the task of generating responses to user queries based on the content of a database.
2. **Integrate RAG Technology**
    - **Information Retrieval:** Develop and implement a system for retrieving relevant information from a database based on user queries.

- **Combine Retrieval and Generation:** Integrate the information retrieval system with the GPT model to generate coherent and contextually relevant responses to user queries.

3. **Facilitate User-Chatbot Interaction**

   - **UI Design** Develop a user interface that allows users to interact with the chatbot in a natural and intuitive way.
   - **UX Design** Ensure that the user experience is smooth and seamless, and that users are able to easily access the information they need.

4. **Accessibility**

   - **Multilingual Support** Implement support for multiple languages to make the chatbot accessible to a wider range of users.
   - **Accessibility Features** The system must be designed to be accessible to users with visual or motor impairments. As such, it should support voice input. The voice input feature must be able to be activated through a voice command.

5. **Evaluate and Validate the System**

   - **User Testing** Conduct user testing to assess the usability and effectiveness of the chatbot system.
   - **Results Analysis** Analyze the results of the different tests to identify areas for improvement and optimization.

1.3 Methodology

1.3.1 Data Collection and Preparation

- **Data Collection:** We are given a set of files documenting the laws related to social rights in Catalonia. These files are in PDF format, and contain a variety of information, including the text of the laws, the dates they were enacted, and the organizations responsible for enforcing them.

- **Data Preparation:** We will extract the text from the PDF files and convert it into a format that can be used by the information retrieval system. This will involve cleaning the text and removing any extraneous characters. This will be done using the LlamaIndex Python library.

1.3.2 RAG Implementation

- **Information Retrieval:** Develop a system capable of retrieving relevant information from a database based on user queries. This will involve creating an index of the laws related to social rights in Catalonia, and implementing a search algorithm that can return the most relevant laws based on the user query. In practise this will be done using the LlamaIndex Python library, which chunks the text into smaller pieces and indexes them.

- **Combining Retrieval and Generation:** Integrate the retrieval system with the GPT model to generate coherent and contextually relevant responses to user queries. This will involve passing the retrieved information to the GPT model, which will generate a response based on this information.

1.3.3 User Interface Design

- **UI Design:** Develop a user interface that allows users to interact with the chatbot in a natural and intuitive way. This will involve creating a chat interface that allows users to input queries and receive responses from the chatbot.

- **UX Design:** Ensure that the user experience is smooth and seamless, and that users are able to easily access the information they need. This will involve testing the user interface with a group of users to identify any areas for improvement, as well as demoing the system to the stakeholders.

1.4 Architecture

Tenim una Frontend UI de conversa (rollo whatsapp) que comunica am un Backend que gestiona les crides als models de llenguatge i a la base de dades. El Backend també s'encarrega de la gestió de la base de dades i de la indexació de la informació.

1.4.1 Hotword/Wakeword Detection

En el Frontend es realitza també el tema dels Hotwords: la màquina client escolta i processa localment al micròfon, de manera constant i en temps real. Si sent la paraula clau "xat"o "xatbot"activa el microfon i comença a gravar. La gravació s'envia al un server de reconeixement de veu, que retorna el text reconegut. Aquest text es passa al Backend per a que el chatbot pugui respondre. Per fer aixo s'utilitza la nostra versió de la llibreria EfficientWord-Net, que ha sigut

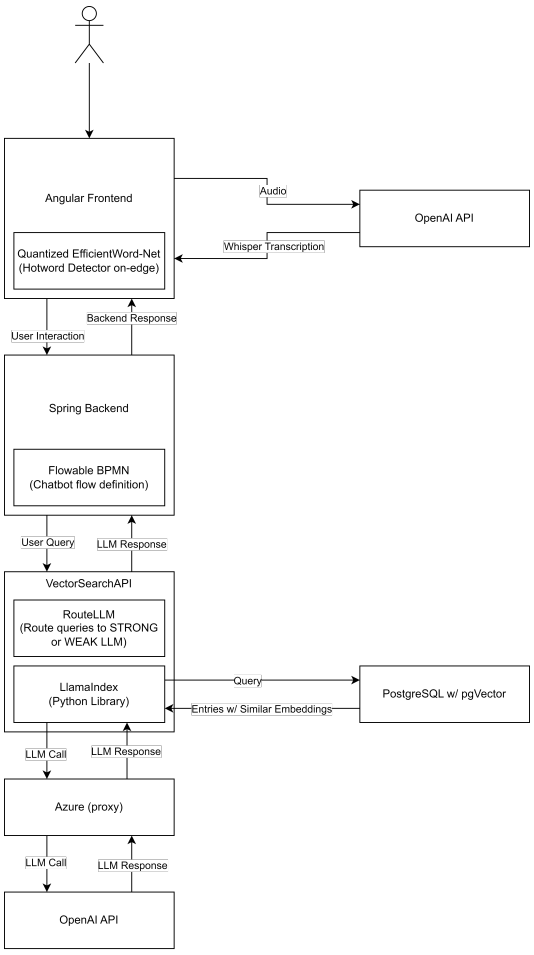

Figura 1.1: Arquitectura del sistema

modificada per tal de reduir el temps de resposta, els requeriments de la màquina client i el pes dels fitxers descarregats. S'ha aconseguit una millora d'un 100% en el temps de resposta, i d'un 300% en la reducció del pes dels fitxers descarregats (abans pesaven 80MB i ara en pesen 20MB). Aquestes millores s'han aconseguit sense sacrificar perceptiblement la precisió del reconeixement de veu, gràcies als mètodes de Quantització. Per fer anar aquest model en el navegador de manera eficient s'utilitza la llibreria WebAssembly (wasm).

1.4.2 Backend

1.4.2.1 Conversation Flow

The Backend implements, among others, the conversation flow followed by the chatbot. It implements multiple stages and follows a state machine to manage the conversation. User queries at each stage are routed to the appropriate model.

Conversations flows are implemented using Flowable. Here are the flow diagrams for the conversation stages:

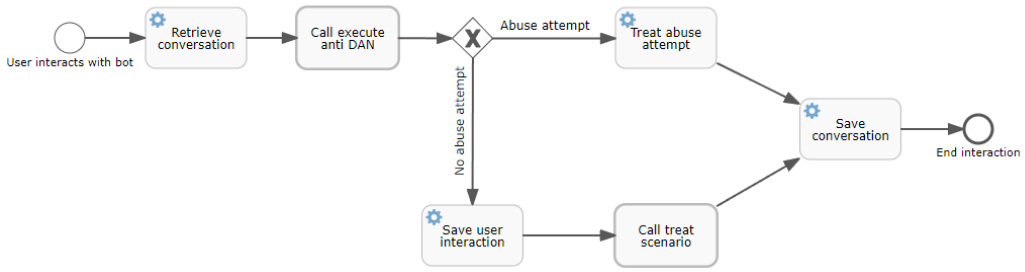

Figura 1.2: Top level flow diagram for the conversation

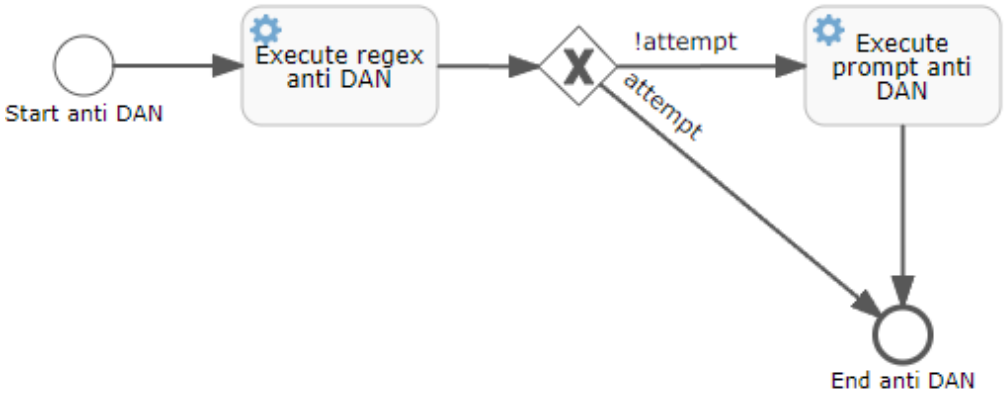

Figura 1.3: Flow diagram for the Anti DAN stage

1.4.2.2 Model Routing

We use a slightly tweaked version of the fairly new RouteLLM python library. This allows us to route a certain percentage of queries to a "stronger", more expensive model and the rest to a "weaker", less expensive model. With this system we are able to achieve X% of the performance of the stronger model at a fraction of the cost. The RouterLLM system uses a small BERT classifier model that decides which queries should be routed to the stronger model and which to the weaker model. This classifier was trained by the original library authors on a dataset of human preferences augmented with synthetic data generated using GPT-4. They report good generalization performance, so we apply the system on a pair consisting of GPT-4o and GPT-3.5-Turbo.

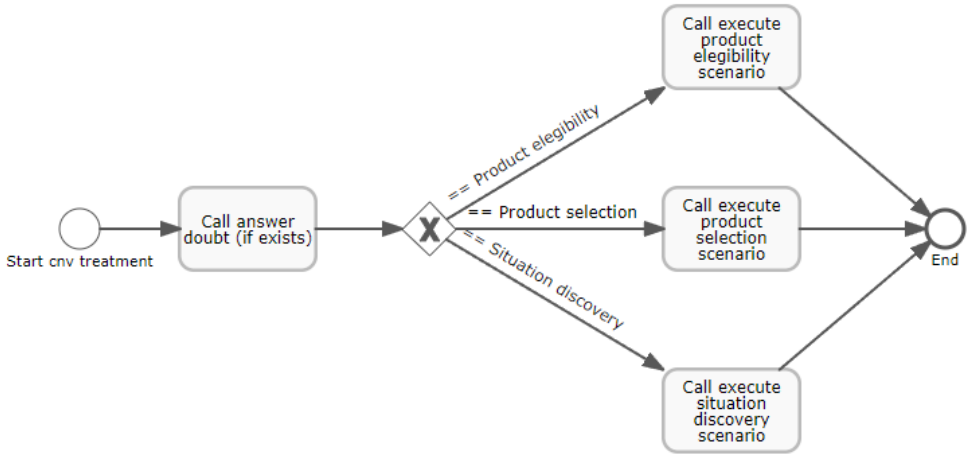

Figura 1.4: Flow diagram for the Treat Scenario stage

// TODO: PUT HERE EXAMPLES OF CONVERSATIONS BEING ROUTED TO DIFFERENT MODELS DEPENDING ON WHAT IS BEING ASKED.

1.4.2.3 Information Retrieval

We do RAG using the LlamaIndex python library. We implement a custom RAG algorithm that uses hierarchical embeddings of chunks, allowing us to capture both fine-grained and coarse-grained information. This was implemented by Joan Oller.

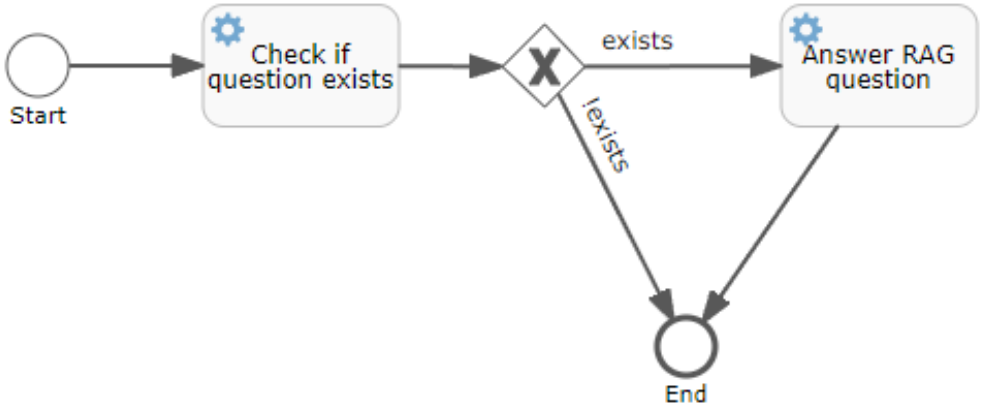

Figura 1.5: Flow diagram for the Answer RAG Question If Exists stage

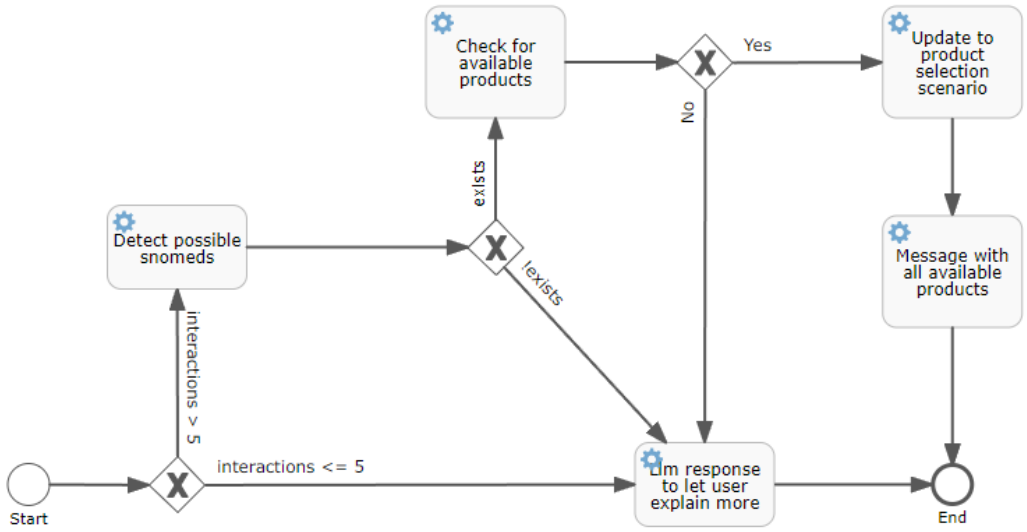

Figura 1.6: Flow diagram for the scenario where we need to discover the user's situation

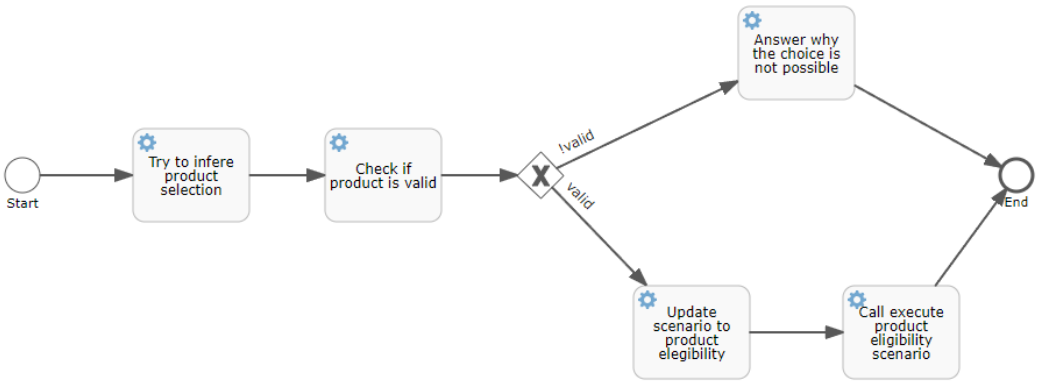

Figura 1.7: Flow diagram for the scenario where we need to select a product from the kSocial catalog given the conversation we have had with the user

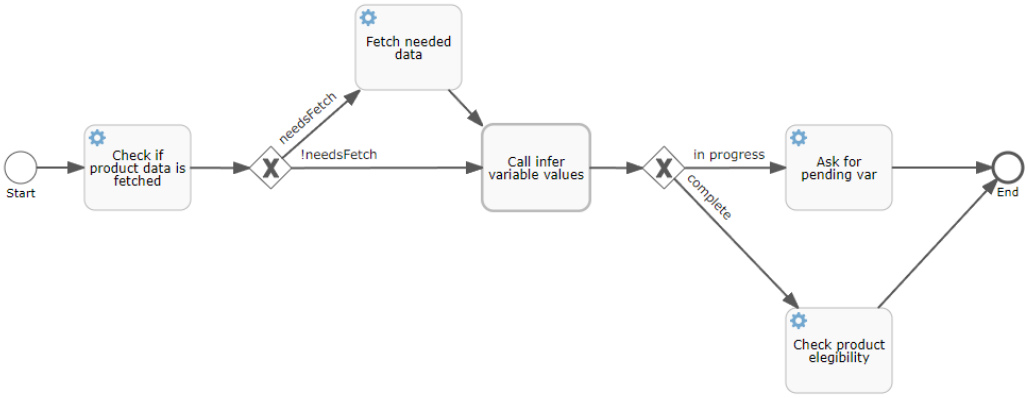

Figura 1.8: Flow diagram for the scenario where we need the user to select one of the products from the kSocial catalog

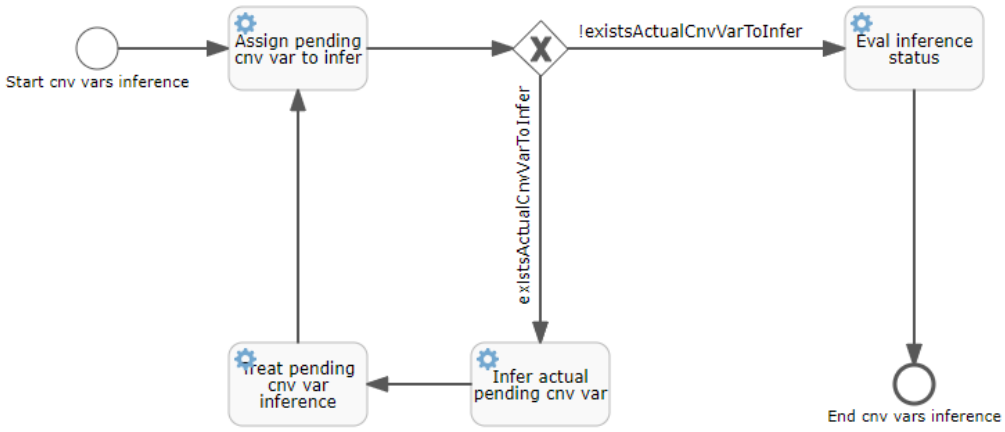

Figura 1.9: Flow diagram for inferring necessary variables for the current scenario based on the conversation

CAPÍTOL 2

Estat de l'art

2.1 Secció

2.1.1 Subsecció

Capítol 3

Preliminars

Capítol 4

Planificació i Metodologia

Capítol 5

Contribució Metodològica

Capítol 6

Resultats

Capítol 7

Conclusions i treball futur

Bibliografia

[1] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, p. 36–45, jan 1966. (Cited on page 1.)

[2] B. Abushawar and E. Atwell, "Alice chatbot: Trials and outputs," *Computación y Sistemas*, vol. 19, 12 2015. (Cited on page 1.)

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014. (Cited on page 1.)

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. (Cited on page 1.)

[5] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. (Cited on page 1.)

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. (Cited on page 2.)