Chatbox sale of trotec laser machines

Jhonathan De La Torre jhdavidd@udistrital.edu.co
Cod : 20222020033
Ing Sistemas.

Abstract-- **This document consists of collecting relevant information on the development of a specialized chatbox for sales assistance of Trotec Laser brand machines. The aim is to provide the client with a tool for their own information that allows them to decide on what type of machine. meets your business expectations.**

Introducción

The development of the specialized Chatbot will begin with a systematic analysis of the basic requirements of customers looking to buy Trotec laser cutting machines. Once the essential requirements have been determined, the technical documents with the specifications of the products will be personally requested. the machine models chosen according to the requirements analysis

During the requirements analysis it was found that the vast majority of customers ask specifically for a very small group of models of laser cutters and engravers. Particularly the R series with its R400 and R500 models, SP Series with its SP500, SP2000 and SP3000 models., and the SpeedMarker industrial series with its SpeedMarker100, 300, 700 and 1300 models.

After the analysis, the corresponding pdf of said machine models is requested from the Trotec Laser technical service, thus achieving great consistency in the data since they would be endorsed by the company that manufactures the machines.
Once the PDFs are in possession, the necessary information is filtered to be presented to the various types of Trotec Laser clients. During the filtering of information, it is grouped into paragraphs, highlighting in said information components that are highly asked about by clients, such as:
What is the power of the laser machine?
What is the work area?
What is the power of the laser?
What is the type of laser?
What is the price of the machine? etc.

.

.

[*] Revista Argentina de Trabajos Estudiantiles. Patrocinada por la IEEE.

I. DEVELOPMEN

Based on this, the size of the original PDFs is reduced to establish the system limits.
Once with the information organized and polished, we proceed to propose the development of the chatbox, it is decided to use, on the recommendation of the teacher, Lang chain, and llama-cpp, all with the purpose of using the contexts of the sentences using an established language model. , to find a grammatically correct way to answer a question based on the information provided in the PDFs.

Step-by-Step Explanation of the Chatbot

1. **Introduction to the Chatbot:**
   - The chatbot is designed to assist customers interested in Trotec laser cutting machines by providing technical and functional information about the machines, including their prices.
2. **Loading PDF Data:**
   - The first step is to load PDF files that contain information about the laser cutting machines. This is done using the load_pdf_data function, which reads all PDFs from a specified directory and combines their content into a single text string.
3. **Splitting Text into Chunks:**
   - The large text string from the PDFs is then split into smaller, manageable chunks. The split_chunks function takes the text string and divides it into chunks of 10,000 characters each, with an overlap of 20 characters to ensure continuity.
4. **Getting Embeddings:**
   - Next, the get_embeddings function generates semantic embeddings for each chunk of text. These embeddings are numerical

representations that capture the meaning of the text, making it easier for the chatbot to understand and retrieve relevant information.

5. **Creating a Vector Store:**
    - The get_chunk_embeddings function then creates a vector store using FAISS (a library for efficient similarity search). This store contains the embeddings of the text chunks and allows the chatbot to quickly find relevant chunks when a question is asked.
6. **Loading the LLM Model:**
    - The load_llm function loads a pre-trained LLM (Large Language Model) from the Mistral family, configured to provide responses to user queries. The model is optimized for efficiency, balancing response speed and accuracy.
7. **Interacting with the Chatbot:**
    - The main function starts an infinite loop where the user can ask questions. When a question is entered, the agent_answer function uses the LLM and the vector store to find and generate a relevant answer. The chatbot then displays the answer to the user.
8. **User Communication:**
    - If the user wants to exit the chat, they can type "exit", and the program will terminate. If there are no inputs, the loop continues to wait for the user's next question.

THE COMPLETE CODE WILL BE SHOWN IN THE ANNEXED DOCUMENT

II. Results

The results of the chatbot using the lightweight language model were grammatically correct, the failure rate remained low, providing coherent and real information 8 out of 10 times

III. Analisis of results

When entering a series of prompts related to the sensitive information of laser cutting machines, such as the one shown below, it is observed under the eyes of the Trotec technician (in that case my person), that the information provided is correct, for example:

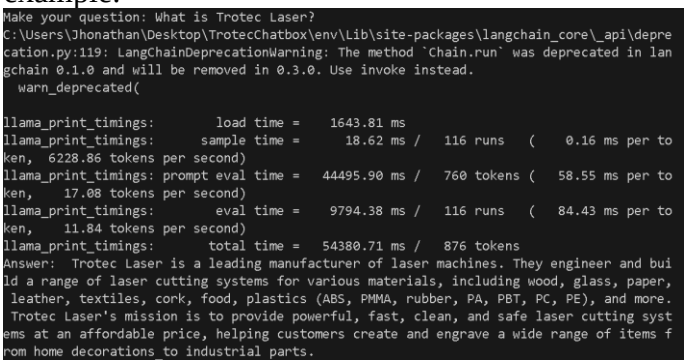

Image 1. Prompt: What is Trotec Laser?

The information obtained after the prompts is not only completely correct but also the execution time is quite low, this is due to reducing the amount of PDF loading to only 24 chunks (the aim was to have less than 30), since it was tested with 30 and the average execution time exceeded 10 minutes (with 16 GB of RAM)

After performing a second prompt, with a much more technical approach, which consists of knowing the work area of one of the specific machines, not only is the answer correct but it provides additional information about the machine, as shown below:

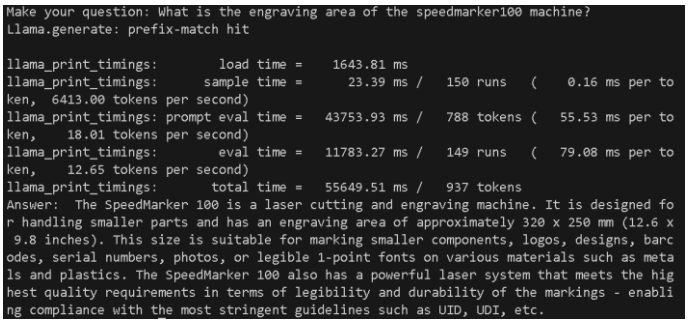

Image 2. Prompt: What is the engraving area of the speedmarker100 machine?

It is found that with the parameters: temperature=1.0, top_p=0.9,n_ctx=4096, the answers are 8 out of 10 times correct or partially correct.

Conclusions

The chatbot is a powerful tool that will facilitate the sale of Trotec laser cutting machines,

helping customers to obtain true information, at all times and faster than being provided by a person.

A balance is found between the performance necessary to run the model and the set of CPU hardware where it is executed, resulting in low execution times (initially they exceeded hours).

REFERENCES

[1] GitHub, https://github.com/ggerganov/llama.cpp.
[2] huggingface., *https://huggingface.co/models?other=llama.cpp*
[3] stackoverflow.com.