



Building a Question-Answering Chatbot using Hugging Face Transformers

Yago Botella Barbeito
Jaime Villarrubia Fernández

INDEX

- 1 Project Objective
- 2 Tools and Technologies Used
- 3 WHAT IS Hugging Face AND
- 4 Results

PROJECT OBJECTIVE



The main goal of this project was to develop an intelligent chatbot capable of answering user questions based on a custom input text, known as the context. Unlike typical chatbots that rely on pre-defined knowledge or open-ended models, this chatbot focuses specifically on extractive question answering, meaning it searches for the answer within the given text.

To achieve this, we used the transformers library by Hugging Face, which provides access to a wide variety of pre-trained models specialized in natural language processing tasks. Our objective was not only to create a functional chatbot, but also to compare different models in terms of their accuracy, response time, and resource consumption, in order to find a practical balance for real-time usage.

Additionally, we implemented a confidence score mechanism to help detect uncertain or potentially incorrect answers. If the model's confidence in the result is too low, we inform the user that the answer might not be reliable. This adds an extra layer of trust and transparency to the chatbot's responses.

In summary, the goal was to combine powerful NLP models with an interactive interface, allowing users to explore how machine learning can be applied to understand and extract information from texts.

TOOLS AND TECHNOLOGIES USED

To carry out this project, we worked entirely in **Python 3.13**, which is a widely-used and flexible language for machine learning and natural language processing tasks. Python allowed us to integrate powerful libraries and run our models smoothly in a local environment.

For the core of the chatbot, we used the **transformers library by Hugging Face**. This library provides a very high-level API to load, use, and even fine-tune state-of-the-art language models with minimal code. It was essential for creating the question-answering pipeline.

We also installed **torch**, which serves as the backend engine for many models in Hugging Face. Without PyTorch or TensorFlow, models cannot be loaded into memory, so this was a necessary component to make the pipeline work.

The entire development was done using **Visual Studio Code**, which is a lightweight yet powerful code editor. We ran all scripts and debugging tasks through the **Zsh terminal**, which gave us control over the Python environment, including the creation and activation of a virtual environment.

The main model we selected for deployment was **distilbert-base-cased-distilled-squad**. This model offers a good trade-off between performance and speed. It is smaller and faster than traditional BERT models, while still being accurate enough for extractive question answering tasks. It is pre-trained and fine-tuned on the SQuAD dataset, which makes it ideal for our chatbot's purpose.

WHAT IS HUGGING FACE AND TRANSFORMERS

The main model we selected for deployment was **distilbert-base-cased-distilled-squad**, which belongs to the DistilBERT family of models. This model offers an excellent **trade-off between performance and computational efficiency**. It is significantly smaller and faster than the original BERT models, which makes it ideal for projects where speed is important but we still want a good level of accuracy.

distilbert-base-cased-distilled-squad is not only pre-trained on a large amount of general language data, but it is also **fine-tuned on the SQuAD dataset** — a benchmark dataset for extractive question answering. This means the model is already optimized to find answers inside a given context, which fits perfectly with the goal of our chatbot.

Its compact size reduces resource usage and speeds up inference, making it especially suitable for local development or lightweight applications, such as terminal-based chatbots. Overall, this model helped us achieve an effective balance between accuracy and responsiveness, which was one of the main objectives of our work.

RESULTS

- The chatbot allows the user to enter a custom context and then ask questions related to that context in natural language.
 - It uses a pre-trained transformer model to extract and generate answers based on the input text.
 - The response is fast (within seconds) and includes a confidence score, which helps assess the reliability of the answer.
- Here's a practical example of how it works:

```
(venv) jaime@Ordenador-portatil-de-Jaime project % python3 chatbot_qa.py

=====
👋 Welcome to the Question-Answering Chatbot
=====
Device set to use mps:0

📄 Please enter the context (the background text):
> Hugging Face is a company specialized in natural language processing technologies. It was founded in 2016 by Julien Chaumond, Clément Delangue, and Thomas Wolf.

✅ Context loaded. You can now ask questions.
Type 'exit' to end the chat.

? Ask a question: When was Hugging Face founded?
💬 Answer: 2016 (confidence: 0.98)

? Ask a question: Who founded Hugging Face?
💬 Answer: Julien Chaumond, Clément Delangue, and Thomas Wolf (confidence: 0.79)

? Ask a question: What is the name of the company?
💬 Answer: Hugging Face (confidence: 1.00)
```

The background is a light cream color with abstract, wavy shapes in blue and yellow. In the top left, a blue shape with a white outline curves downwards. In the top right, a yellow shape with white outlines curves downwards. In the bottom left, a yellow shape with white outlines curves upwards. In the bottom right, a blue shape with a white outline curves upwards.

THE END