Contents

1 Building Data - A Chatbot for the Data Science Study Programme at FHNW 1
1.1 Introduction . . . 1
1.2 Concept . . . 2
1.2.1 Goal . . . 2
1.2.2 Bot Capabilities . . . 2
1.2.3 Knowledge Base . . . 3
1.2.4 Design . . . 3
1.2.5 Architecture & Tech Stack . . . 3
1.2.6 Target Audience . . . 6
1.2.7 Privacy . . . 7
1.2.8 Evaluation . . . 7
1.3 Methodology . . . 8
1.3.1 Step 1: Classification . . . 8
1.3.2 Step 2: Question Answering . . . 9
1.3.3 Step 3: Concern Path . . . 11
1.3.4 Step 4: Assemblying the Chatbot . . . 11
1.4 Future Work . . . 12
1.5 Conclusion . . . 13
1.5.1 Key Learnings and Achievements . . . 13
1.5.2 Comparative Performance . . . 13
1.5.3 Future Directions and Procedure for Improvement . . . 13
1.6 References . . . 13

1 Building Data - A Chatbot for the Data Science Study Programme at FHNW

Authors: Tobias Buess, Alexander Shanmugam, Yvo Keller

1.1 Introduction

This documentation will go over how we built Data, the Chatbot for the Data Science Study Programme at FHNW. It will cover the following topics:

- Concept
- Methodology
  - Step 1: Classification
  - Step 2: Question Answering
  - Step 3: Concern Path
  - Step 4: Assemblying the Chatbot
- Future Work
- Conclusion
- References

We refer to the specific implementations of the individual components for more details, which are liked here. Each repository has its own README, which explains the implementation in detail, and further references relevant notebooks and python scripts if you want to dive deeper into the code.

1.2 Concept

Our project started based on the definition of the Chatbot Concept and the Chatbot Architecture. The concept outlined planned features and the architecture, and is described in this section. It was written before we started work on the project, and thus might be slightly different from the final implementation.

1.2.1 Goal

The goal of this challenge is to develop a chatbot named Data. It is intended to assist students of the Data Science program by answering their questions about the content of the Spaces related to the modules. The chatbot should have a personality and follow predefined ethical guidelines. Data will be able to access the content of the module Spaces and answer standard queries using a knowledge base.

The bot should also recognize users' problems and respond in a morally appropriate way, such as with encouraging words or by referring them to a contact person. Additionally, it should contribute to motivating the students.

Our focus in the challenge is to build a version of the bot that functions well in German (the language of most content in the knowledge base). While it is not excluded that the bot could also function in English, we will not explicitly focus on this.

1.2.2 Bot Capabilities

The bot should be able to provide the following information upon request:

- Explain its capabilities
- Details about the structure of the degree program (concept, handbook, curriculum, regulations)
- Details about the module
  - Subject experts
  - Language
  - ECTS
  - Type
  - Level
- Contents of the "Portrait" tab of the modules
- *optional*:
  - Provide knowledge from PDFs in the module learning materials
  - Suggest learning materials
  - Tab "Tasks"

In its behavior, the bot should consider the following personality aspects:

- Use informal language with users (as is customary in the SG Data Science)
- Speak ethically and morally correctly (fair, appreciative)
- Have a motivating, humorous, and empathic personality
- Be able to conduct a dialogue with the user and know about its own history
- Adequately respond to the user's problems when recognized (e.g., stress in studies, dissatisfaction, depressive phases) and provide contact information for points of contact

1.2.2.1 What the Bot Should NOT Do

- Explicitly support multilingualism
- Access the content of learning materials (e.g., PDFs or external links)
- We will not invest direct effort in preventing prompt injection and similar issues within the scope of the challenge

1.2.3 Knowledge Base

The bot's knowledge base should be based on multiple sources. These include the necessary information for providing its capabilities, from both the Spaces DB dump and the PDFs about the degree program (concept, handbook, curriculum, and regulations).

The bot should prioritize the context provided when answering questions. If a question is asked that the bot cannot answer based on the existing context from the knowledge base, it declares this and either uses the knowledge in the LLM to answer the request (e.g., "What is linear regression?") or declines to answer the request.

The content can be in both German and English, which we will consider in development.

1.2.4 Design

We will create an avatar for "Data" to be displayed in the chat interface.

- Avatar image
- *optional*: Synthetic voice

1.2.5 Architecture & Tech Stack

The planned architecture can be seen in the following sketch. The individual components are briefly explained below, with changes to the architecture or the technologies used not being excluded during the challenge.

1.2.5.1 Chat Interface The bot is available to users in a simple web chat interface. We will build the logic for processing requests by the bot using Python, possibly using LangChain to communicate with the various LLMs.

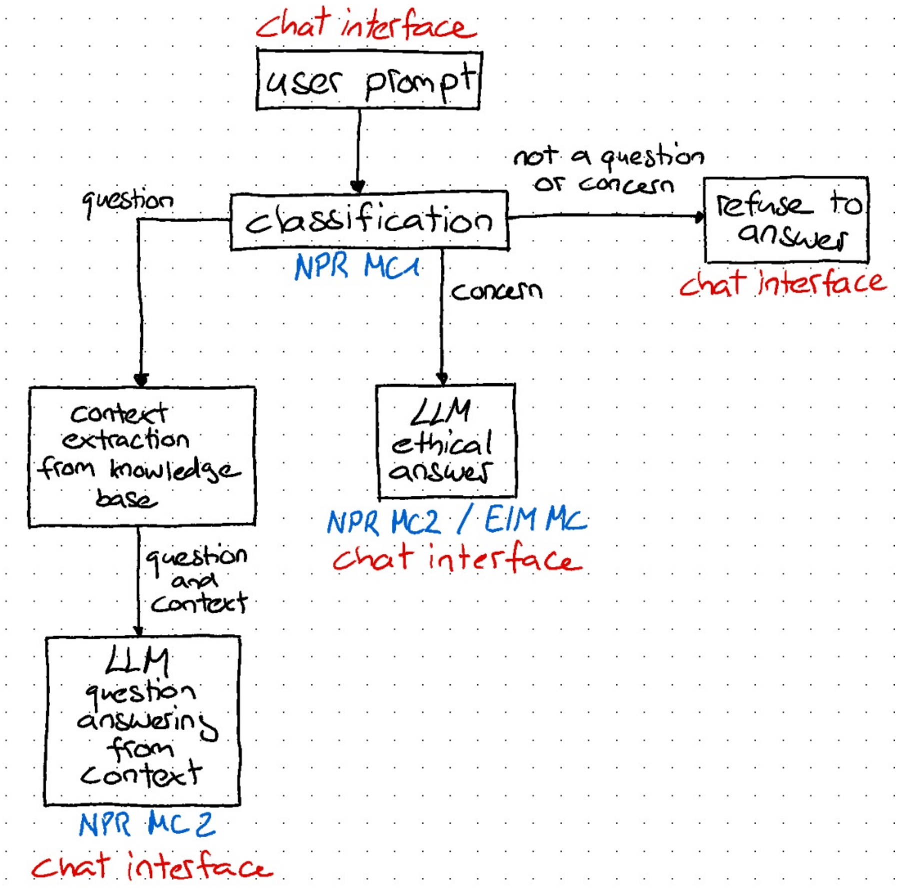

Figure 1: Architecture Sketch

Tech Stack:

- Chat interface with Streamlit (alternatively Gradio from HuggungFace)
- *likely*: LangChain (For embedding and communication with LLMs)

1.2.5.2 Prompt Classification (also: npr MC1) The bot initially distinguishes between 3 types of requests:

- question
- concern
- not a question or concern (harm)

Possible approach:

- Evaluate in an experiment how many training data are needed until the model achieves good performance in classification
- Fine-tuning a BERT model for user prompt classification

Tech Stack:

- HuggingFace Transformers library
- LLM: BERT Base Multilingual or similar

1.2.5.3 LLM for "concern" (also: npr MC2 and eim MC) The bot addresses the user's concerns when recognized. It should respond empathetically and motivatingly, and provide the user with contact information for points of contact if needed.

Possible approach:

- Fine-tuning a LLAMA2 model (or similar) on the ethical guidelines for advising and supporting the user regarding their concerns.

Tech Stack:

- HuggingFace Transformers library
- LLM: LLAMA2-13B-Chat or similar

1.2.5.4 LLM for "question" (also: npr MC2) The bot answers the user's question if it understands it and the answer is available in the knowledge base. Otherwise, it declines to answer the question, or informs the user that there is nothing available in the knowledge base, and tries to help further with internal LLM knowledge.

Possible approach:

- Chunking and embedding the context in the knowledge base.
- Fine-tuning/Instruction-tuning of a LLAMA2 model on answering questions from given context.

Tech Stack:

- Embeddings (BERT/Open AI)
- Vector Storage PGVector
- HuggingFace Transformers library
- LLM: LLAMA2-13B-Chat or similar

1.2.6 Target Audience

The target audience for the bot are the students of the Data Science program. We have defined two personas to represent the target group.

1.2.6.1 Persona 1: Anna, the Diligent Student

1.2.6.1.1 Demographic Data

- Age: 23
- Gender: Female
- Professional Background: Apprenticeship as a computer scientist

1.2.6.1.2 Personality

- Ambitious and focused
- Detail-oriented
- Loves to plan early

1.2.6.1.3 Needs and Goals

- Wants to have the best overview of her modules
- Always looking for additional resources for better learning success
- Wants to stay informed about changes in the curriculum

1.2.6.1.4 Usage Scenarios

- Asks the bot about the assessments in specific modules
- Wants to know which subject experts are responsible for a module
- Plans the next semester and asks the bot about modules in the curriculum
- Is praised by the bot for her good work and feels motivated

1.2.6.2 Persona 2: Markus, the Working Student

1.2.6.2.1 Demographic Data

- Age: 29
- Gender: Male
- Professional Background: Works part-time in accounting

1.2.6.2.2 Personality

- Pragmatic and goal-oriented
- Values work-life-study balance
- Somewhat prone to stress due to many commitments

1.2.6.2.3 Needs and Goals

- Looks for an efficient way to consult study information
- Wants to spend as little time as possible searching for basic information
- Seeks a quick way to get his questions answered to focus on his work and studies

1.2.6.2.4 Usage Scenarios

- Wants to quickly know how many ECTS a module has
- Wants to know what to expect in a specific module
- Uses the bot's motivating and empathetic functions to reduce stress

These personas can serve as a foundation for the development of the chatbot "Data." They represent the needs and goals of the target group and can help to optimally align the functionality and behavior of the bot.

1.2.7 Privacy

We use our own fine-tuned models, so no sensitive data leaves our system. The chatbot is only accessible to students of the Data Science program, and therefore only information that students already have access to is available.

1.2.8 Evaluation

A chatbot that gives incorrect answers or does not address the user's questions is not helpful. Therefore, it should be evaluated in various ways.

1.2.8.1 Prompt Classification Quantitatively:

- F1-Score
- Accuracy

Qualitatively:

- Test individual examples on several models for comparison

1.2.8.2 LLM for "concern" Quantitatively:

- Not planned

Qualitatively:

- Individual examples to test whether the bot follows the ethical guidelines

1.2.8.3 LLM for "question"

Quantitatively:

- Retrieval (Are the relevant chunks identified)
- BLEU Score (How good is the answer)

Qualitatively:

- Individual examples to test the usefulness of the answers

These evaluation methods will help ensure that the chatbot "Data" is effective, reliable, and adheres to the intended ethical and operational standards, catering effectively to the specific needs and scenarios of the target student audience in the Data Science program.

1.3 Methodology

1.3.1 Step 1: Classification

Find the implementation and all details on the Classification here: Text-Classification

1.3.1.1 Objective

The primary objective was to enable our chatbot, named Data, to classify user prompts into one of three categories: questions, harm, or concerns. This classification is vital for the chatbot to respond appropriately to user interactions. For details on why this is vital to our architecture, see Concept.

1.3.1.2 Data Selection and Preparation

We selected three datasets for creating a comprehensive training dataset for our classifier. These included:

- **GermanQUAD for Questions**: This dataset provided a robust set of question-answer pairs in German, ideal for training our model to recognize user queries.
- **GermEval for Harm**: This dataset was chosen to help the classifier identify harmful or inappropriate content.
- **Stress-Annotated Dataset (SAD) for Concerns**: To enable the chatbot to recognize and appropriately address user concerns, especially those of a sensitive or personal nature.

A qualitative assessment using a BERT classifier suggested the feasibility of our approach. We then proceeded with a train-test split to prepare the datasets for model training.

1.3.1.3 Model Selection and Training

We chose three model architectures:

1. **LinearSVC with TF-IDF**: Selected as a baseline model to compare against more advanced deep learning approaches. LinearSVC was chosen

based on its superior performance among seven tested machine learning classifiers.

2. **LSTM-CNN**: This model was considered due to the LSTM's ability to handle sequential data, potentially learning relationships between word semantics. The CNN was employed for initial feature extraction before feeding data into the LSTM. Pre-trained embeddings (bert-base-german-cased) were utilized, anticipating a reduction in overfitting risk and less need for extensive training data.
3. **BERT Fine-Tuning as Classifier (bert-base-multilingual-cased)**: This model was selected for its support of both English and German sentence structures, coupled with the expectation that its pre-training would require less training data.

1.3.1.4 Evaluation and Improvement All models performed well on the test dataset, with BERT achieving the highest accuracy (98%). However, when tested on a benchmark dataset comprised of examples created by our team, all models showed a significant drop in performance (Accuracy: SVC 0.60, LSTM-CNN 0.62, BERT 0.67). Based on these results, we decided to continue with BERT as our primary classifier model.

To enhance model performance, we created a synthetic dataset using GPT-4 with tailored prompting. This dataset, comprising queries from all three classification categories, led to a marked improvement in BERT's performance on our benchmark dataset.

We also recognized some misclassifications in basic queries, such as "Hallo was kannst du" being incorrectly categorized as harm. To address this, we extended our synthetic dataset with a "human feedback" component, that can be continually updated based on flagged results from production use, to further refine accuracy.

1.3.2 Step 2: Question Answering

Question answering of the chatbot is based on the RAG process. This consists of two main parts: Retrieval and Fine-tuning of the LLM.

- In the retrieval part, we chunk the documents and store them in a vector store. We then query the vector store to retrieve relevant chunks based on the user's question, together with metadata such as the source of the chunk.
- In the fine-tuning part, we fine-tune an open-source LLM (base model) to answer questions based on the retrieved chunks (passed as context) and refer to sources.

Find the implementation and more details on the Question Answering here: Implementation

1.3.2.1 Objective The aim was to develop a large language model (LLM) capable of providing context-informed, abstractive answers to user queries. The context is provided by the relevant chunks retrieved from the knowledge base, and the answers generated by the LLM.

1.3.2.2 Data Selection and Preparation We again worked with the **GermanQUAD dataset**, which includes questions, contexts, and corresponding answers in German. This dataset was used to train the LLM in extracting answers from the provided context.

1.3.2.3 Initial Approach and Realization Initially, we aimed for the model to learn answer extraction from context by fine-tuning it with examples from the GermanQUAD dataset. However, we observed that the answers generated were extractive, mirroring the dataset's nature. To achieve our goal of abstractive response generation, we modified our approach by using GPT-3.5 to create a dataset of abstractive answers from questions, contexts, and the original answers.

1.3.2.4 Context Swapping and Source Referencing Context swapping was incorporated into our methodology, as we recognized the importance of the model learning to not provide answers when relevant context was unavailable. We also trained the model to refer to sources at the end of its abstractive responses by splitting the available context into chunks and adding synthetically generated sources to these chunks.

1.3.2.5 Model Training and Selection We selected and trained three different LLMs:

- Baseline: meta-llama/Llama-2-13b-hf
- German-optimized Llama-2-13b: flozi00/Llama-2-13b-german-assistant-v7
- German-optimized Mistral-7b: VAGOsolutions/SauerkrautLM-7b-v1-mistral

We opted for base models over chat variants for the learning experience, despite potentially lower overall performance.

1.3.2.6 DVC Stages Using a DVC pipeline allowed us to automate the retrieval and fine-tuning stages of our methodology. This pipeline was designed to be run on a GPU-enabled machine.

1.3.2.6.1 Retrieval Stages The retrieval stages aimed to create chunks from relevant documents (Data Science study program documents, course materials, etc.) and save them in a ChromaDB vector store. This store was later queried for relevant documents in response to user queries.

1.3.2.6.2 Fine-tuning Stages The fine-tuning stages were designed to enhance the language model's question-answering abilities using our modified GermanQUAD dataset. The resulting fine-tuned models can then be used to generate abstractive answers to user queries.

1.3.2.7 Dataset, Models, and Experiments

1.3.2.7.1 Fine Tuning Dataset The dataset, derived from GermanQUAD, was modified to include both answerable and unanswerable questions, with abstractive answers generated using GPT-3.5. The dataset was split into training, validation, and test sets.

1.3.2.7.2 Fine Tuning Models We conducted experiments with three different language models:

- Baseline: meta-llama/Llama-2-13b-hf
- German-optimized Llama-2-13b: flozi00/Llama-2-13b-german-assistant-v7
- German-optimized Mistral-7b: VAGOsolutions/SauerkrautLM-7b-v1-mistral

1.3.2.8 Evaluation We performed quantitative and qualitative evaluations to ascertain the best-performing model in terms of generating context-informed, abstractive answers.

1.3.3 Step 3: Concern Path

For the concern path, we built a prompt upon ethics theory. Find the comprehensive report including evaluation here: Concern Path

The functionality is built directly into the Chatbot Dashboard, see dashboard.py.

1.3.4 Step 4: Assembling the Chatbot

After the individual components were developed, we assembled the chatbot using the Hugging Face Gradio framework. It allowed us to create a simple, user-friendly interface for the chatbot, which can be accessed via web browser. It has a Chat Tab, a Retrieval Tab and a Classification Tab. This way, each component can be tested individually, which is immensely helpful for debugging and improving the chatbot.

The Chat Tab is the main interface for the user. It first uses the classifier to determine the type of the user input. If the input is a question, it is passed to the Question Answering module, which retrieves relevant contex and generates the answer. If the input is a concern, the Concern Path is triggered, which generates an appropriate messasge to continue the conversation. If the input is harm, the chatbot responds with a predefined message that denies the user's request.

The implementation can be found in dashboard.py.

For setup instructions, see README.md.

1.4 Future Work

- When we would move towards deploying "Data" in a production environment, our focus would be on continual retraining and improvement of the classifier. This will be achieved by leveraging new human feedback to refine the model's accuracy and adaptability to real-world interactions.
- Our next steps would involve continuous improvement of the question-answering module based on real-world interactions and user feedback.
- Another important improvement would be building a history into the Question Answering module, so that the chatbot can remember previous questions and answers and use them to generate more context-informed answers.
- Instead of opting for solely a classifier, we would include the help of an LLM in that task, and instead of having either the Question Answering model or the Concern model responding, we would opt for one model with both functionalities built into it. This could for example be achieved by the combined model using a response generated by a separate model focused on QA, and using that output in it's response, or fine-tuning a model to have both capabilities.
- During evaluation of retrieval, we discovered that some document chunks produced are really small and offer no value. The questions produced for retrieval evaluation on these where therefore also pointless, and would have distorted our retrieval evaluation, which led us to remove them in the evaluaiton to make it more representative. We would make sure to remove all irrelevant short chunks from the vectorstore altogether.
- Currently we apply a really simple retrieval strategy of just using the user's chat input as query. There are many additional techniques that we could experiment with and apply here, like self-querying, query re-writing, or performing re-ranking on the retrieved top-k chunks, that could leed to better retrieval performance.
- We could try other sampling strategies during model inference like beam-search, top-k or top-p during generation (question answering and concern path).
- Fine-tuning the QA LLM with data more specific to the domain of the chatbot (more technical than Germanquad)
- Another idea is to experiment with different instruction tuning prompts.
- We would correct the instances in the fine-tuning dataset where `can_be_answered=False` samples that were nevertheless answered were included in the fine-tuning, which possibly degraded model performance because of the inconsistency introduced.
- Perform more sophisticated hyperparameter fine-tuning on neftune alpha, lora rank, lora alpha, and try to apply lora target adapters not only on query and value, but also on MLP layers.

1.5 Conclusion

The development of the Data chatbot has been an enlightening journey, filled with numerous learning opportunities and challenges. This project not only enhanced our understanding of Natural Language Processing (NLP) and Large Language Models (LLMs) but also delved into the ethical aspects of implementing such technologies in real-world applications.

1.5.1 Key Learnings and Achievements

- **NLP and LLMs**: We gained profound insights into the intricacies of NLP and the capabilities of LLMs, exploring many aspects from data retrieval to fine-tuning these models for specific tasks.
- **Ethical Considerations**: Implementing ethical considerations in the development process was rewarding. We ensured that our chatbot adhered to ethical guidelines.
- **Retrieval System**: Our experience identified the retrieval system as a key area with significant potential for enhancement. Improving the accuracy of information retrieval will directly impact the chatbot's effectiveness.

1.5.2 Comparative Performance

- **Fine-Tuned vs. Vanilla LLM**: The fine-tuned version of our LLM showed a marked improvement over the vanilla model. This validates our approach of customizing the model to fit the specific needs of the Chatbot.

1.5.3 Future Directions and Procedure for Improvement

- **Informed Decision-Making**: Going forward, a more data-driven and informed decision-making process could lead to even better outcomes.

In conclusion, the "Data" chatbot project has been a remarkable endeavor, pushing the boundaries of our understanding and capabilities in NLP and LLMs. The lessons learned and experiences gained will be really helpful in our future journey as Data Scientists.

1.6 References

For the references and resources used in our work, see the publicly available Zotero Library.