Universität St.Gallen

NLP Project Report
Eduatar Chatbot

Authored by:

Aydin Kaan & Schnydrig Sven

A report submitted in partial fulfillment for the Natural Language Processing with Deep Learning Course in the Master of Computer Science (MCS)

Natural Language Processing with Deep Learning - 8,862,1
Institute of Computer Science (ICS-HSG)

June 2023

"Natural Language Processing lies at the very intersection of artificial intelligence, computer science, and linguistics. The power of NLP enables us to communicate with machines almost as easily as we do with each other, opening a world of limitless possibilities." - Yoav Goldberg

1 Introduction

One of the biggest pillars of the Swiss education system is the apprenticeship program. More than half of young people move on to apprenticeships after concluding secondary education [1], typically at the age of 16. Although this system has significant advantages on a socio-economic basis, the pressure on the apprentices is visible. More often than not, many apprentices and their parents have questions regarding the apprenticeships, including legal questions. Topics revolve around working hours, compensations, and contracts, for instance, "Can I work more than 9 hours daily" or "Can my boss request that I work every Saturday".

To answer those questions, apprentices usually reach out to their local secondary education and vocational training office via phone or email. Employees of such offices spend a lot of time trying to answer those questions and might take a few days to receive an answer. On the one hand, this increases manual work for those employees. On the other hand, the time to receive a response is lengthened. This indicates that there is a need to automate that process to enhance productivity and reduce the time to answer.

Inspired by the recent technological advances in natural language processing, such as large language models (LLMs) like ChatGPT, we partnered up with the vocational training office in Bern to build Eduatar, an intelligent chatbot designed to assist users, such as apprentices, teachers, or parents, in finding information related to apprenticeships in Switzerland. By using the technology behind LLMs, Eduatar aims to simplify access to essential information and to serve as a reliable resource for users to navigate the Swiss vocational education system.

We also want to note that these new technologies are currently in the evaluation phase and so far best practice guidelines have not been established. Thus, this project aims to demonstrate a viable proof of concept that requires further investigation and optimization.

The remainder of this report is structured as follows. Section 2 introduces the reader to the methods leveraged in this project. In particular, we review our project approach and dive into each step in more detail, addressing questions on how the dataset was generated, which models were investigated, how the pipeline was set up, and which evaluation methodologies were used. Section 3 presents the overview of the evaluation result and provides a sneak peek into answers that the models have generated on selected questions. This report closes with a discussion of the project and provides an outlook for further improvement in subsequent project phases.

2 | Methods

In this section, we will examine our project approach and provide an overview of each step. Broadly speaking, our approach can be categorized into the following four key steps:

1. Dataset generation
2. Model selection & Fine-Tuning
3. Pipeline Setup
4. Model Evaluation

2.1 Dataset generation

Current LLMs, such as GPT, work quite well, even without fine-tuning due to the large amount of data, on which they have been trained. However, there are still some benefits that come from fine-tuning an LLM, such as better answer quality and enhanced model performance [2]. It is important to note that the benefits of fine-tuning depend on the quality and quantity of the data [3]. To fine-tune an LLM, we decided to build our dataset, which we will explain in more detail.

At the beginning of our project, we mainly looked at two models that we wanted to build and evaluate in the scope of this project: BERT and OpenAI's GPT. As OpenAI's GPT models are not open-source and provide no opportunities for fine-tuning, we focused our data generation efforts on BERT-style models. We added VICUNA at a later stage to our scope, thus our dataset was prepared to be more suited for BERT models, and not VICUNA.

To better explain our approach, we split this subsection into three steps: data collection, data generation, and data cleaning.

2.1.1 Data Collection

In the first step, we started with collecting relevant data sources for our task at hand. Overall, we used two primary data sources: (i) collection of previous Q&As provided by the vocational training office and (ii) fact sheets and legal documents manually collected and downloaded from "Berufsbildung.ch". Those documents are used for two purposes: (i) dataset generation for fine-tuning and (ii) document database for the retrieval of relevant passages.

2.1.2 Data Generation

In data science projects, including ones focusing on LLMs, data generation is considered one of the more work-intensive and expensive tasks. To reduce our manual efforts, we leveraged GPT-4 (via ChatGPT and OpenAI's API) to create the dataset. One limitation of this approach is that the dataset might reflect the biases and limitations of the base model. GPT-4, like its predecessors, can carry forward biases present in the data it was trained on, and the model also has limitations in terms of understanding context or generating logical responses in all situations.

BERT-based LLMs require for fine-tuning for the question-answering domain a specific input: a triple of context, question, and answer. Furthermore, the answer must be extracted exactly (i.e., character-by-character) from the context. As we are using GPT-4 for generating those context-question-answer triples, we carefully developed a prompt that did well in doing so. Thus, in the second step, we set up a process that allowed us to generate our target output.

As we only had access to the GPT-4 API later during our project, we first used the ChatGPT web interface to generate our dataset based on the collection of previous Q&As (s. first primary data source above). We manually copied the relevant paragraphs from the collection (i.e., the context) and queried ChatGPT to generate the question and the corresponding answer from the paragraph. To do so, we used the prompt in the footnote[1] For our second primary data source (i.e., the fact sheets), we used the API access and wrote a script that executed the process automatically, enabling us to save more time and generate a larger dataset.

After combining all the output into a single file, we reached a data set consisting of approximately 1'200 Q&A pairs.

2.1.3 Data Cleaning

While GPT-4 generally performed satisfactorily in generating context-question-answer triples, there were instances where it struggled to precisely extract answers on a character-by-character basis from the context. In these cases, GPT-4 occasionally reformulated the answers into complete sentences, resulting in the addition or substitution of certain words. Approximately 7-8% of the generated triples exhibited this behavior, necessitating manual correction on our part.

For the generation of our test set, we manually selected 25 high-quality question & answer pairs.

2.2 Model selection and Fine-Tuning

As mentioned above, early in our project, we mainly considered two models: BERT and OpenAI's GPT 3.5 model. Due to rising interest in LLMs, various companies and research institutions have started to develop and open-source their models, for instance, Meta with LLaMA [4] or the Technology Innovation Institute with Falcon [5] . In order to incorporate the latest advancements in this rapidly evolving field, we decided to add the VICUNA model to our project scope.

2.2.1 BERT / GELECTRA

For our BERT-based model, we decided to use the "GELECTRA-large-germanquad" model that was published by deepset [6] on April 21st, 2021, and consists of approximately 335M parameters. This model leverages the "GELECTRA-large" model, a German version of the ELECTRA model, as its foundation. Furthermore, it was fine-tuned on a subset of 11'518 German question-answer pairs from the collection of the GermanQuAD data [7].

We fine-tuned the pretrained GELECTRA model with our generated dataset with the following hyperparameters:

Hyperparameter	Value
Number of Epochs	3
Optimizer	AdamW

Table 2.1: Hyperparameters GELECTRA FINETUNING

2.2.2 OpenAI GPT-3.5

For our GPT-based model, we decided to use the TEXT-DAVINCI-003 model. The maximum token size of this model is 4'097 tokens with training data cut-off until June 2021 [8]. We also considered two other alternatives: the GPT-3.5-TURBO and the GPT-4. We decided against those because TEXT-DAVINCI-003 delivers qualitatively better output compared to GPT-3.5-TURBO and the access to GPT-4 is still limited and much more expensive.

2.2.3 VICUNA

VICUNA [9] is an LLM developed by the Large Model System Organization (LMSYS Org), an

[1] Please create as many high quality question/answer pairs as possible to fine-tune a BERT model in a similar style as the Squad dataset based on the following German context text. The answer should be copied 1:1 from the context text. Give your output in that format: "question -|- answer -|-". Give me only the output in that format and no other comments. Here is the context text: placeholder for context text

open research organization founded by students and faculty from UC Berkeley. It is an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. For our experiments, we used the smaller version of the VICUNA that consists of approximately 7B parameters. There are also a larger models available that consist of 13B, 33B, or 65B parameters. We finetuned the last two layers of the VICUNA model with 7B parameters training for a total of 3 epochs. To finetune the model we used the approximately 1'200 question/answer pairs we initially generated for the finetuning of the BERT model and converted it into the required format for the finetuning of VICUNA. To fit the training into our RAM we used half-point precision. Here is an overview of the hyperparameters we used for the finetuning:

Hyperparameter	Value
Number of Epochs	3
Batch size	1
Gradient Accumulation Step	16
Learning Rate	2e-5
Learning Rate Scheduler	Cosine
Model length	2048
FP	16
Warmup Ratio	0.03
Lazy preprocess	True

TABLE 2.2: Hyperparameters VICUNA FINETUNING

2.3 Pipeline Setup

This sub-section delves into the details of the pipeline setup. For our pipeline setup, we leveraged LangChain, a recently developed framework for developing applications powered by language models. In the following, we will dive into three aspects: the retriever-reader model, the ConversationalRetrievalChain, and the memory buffering.

2.3.1 Retriever-Reader Setup

Typically, question-answering systems leverage a retriever-reader setup. Figure 2.1 shows a conceptual overview of the retriever-reader model, which works as follows: In step 1, the question is fed into the retriever, which builds a query vector based on the given question. In step 2, the retriever retrieves relevant context passages from the database of all documents. In the last step, the retrieved passages and the question are fed into the reader (i.e., the LLM) that generates the final answer. This setup has major advantages, such as scalability (i.e., the reader model does not need to be re-trained when new documents are added) and interpretability (i.e., separation of document retrieval and answer generation).

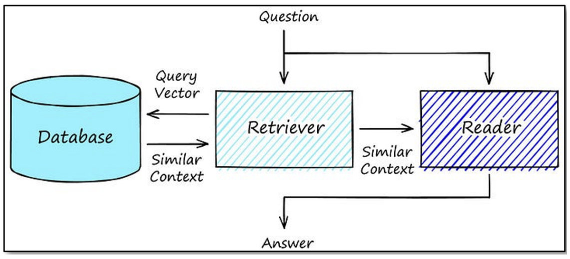

FIGURE 2.1: Retriever-Reader Setup

The retriever setup has a significant impact on the performance of question-answering chatbots as retrieval-based LLMs are only as good as the retrieved passage. After loading our documents (i.e., the raw format of the documents gathered during data collection), we used a test splitter function, to split our entire document base into paragraphs of certain sizes. To do so, we used a chunk size of 400 tokens, with an overlap of 50 tokens to ensure that we do not separate any relevant text passages. After splitting our documents, we chose our embedding and database to build our vector store.

For the embedding type, we selected the embedding type from OpenAI. There are also other alternatives, such as HuggingFace or Cohere. Through some initial experimentations, we observed that when using the OpenAI embeddings, our retriever showed better performance as more relevant paragraphs were retrieved.

For the database, we chose Chroma, an open-source embedding database. Again, there are other alternatives (e.g., Pinecone), which we did not look further in detail.

2.3.2 ConversationalRetrievalChain

The ConversationalRetrievalChain is a chain within the LangChain framework that allows users to build a retrieval-based conversational chatbot. It takes the following three arguments as input: (i) an LLM to be queried (e.g., GPT, VICUNA), (ii) a vector store from where the relevant passages can be retrieved, and (iii) a memory buffer, where the chat history is saved. The memory buffer is discussed in more detail in the next subsection. Prompt Engineering is an important part of state-of-the-art language models. For that reason, the LangChain team has already integrated an appropriate prompt[2] within the ConversationalRetrievalChain. Using this prompt, we ask the model to only answer the question provided given the retrieved document passages, and not with information from this inherent memory. This is important to ensure that only answers based on the vector store will be generated.

2.3.3 Memory Buffer

A key consideration for chatbots is their ability to respond to multiple queries in a chat-like manner, enabling a coherent conversation by considering past interactions. By default, LLMs are stateless. This means that each query is processed independently of previous interactions. A functionality called "conversational memory" allows us to gap this issue. In the following, we explain how this works.

Besides the input parameter (i.e., our query with context), we pass to the LLM another parameter, the conversational memory, which keeps track of the chat history. In this way, we ask the LLM to return the predicted continuation of the conversation. In the LangChain framework [10], there are a few approaches how to keep track of the chat history, three of which we discuss in more detail:

1. **ConversationBufferMemory:** It keeps track of the conversation in its raw (i.e., the entire chat history). The advantages of this approach are that storing everything gives the LLM the maximum amount of information, but there might be slower response times and higher costs.
2. **ConversationSummaryMemory:** As the name suggests, this approach summarizes the conversation history before it is passed as the conversational memory parameters to the LLM. The advantages of this approach are that longer conversations are possible, shortening the number of tokens for long conversations. However, the performance is dependent on the summarization ability of the LLM.
3. **ConversationBufferWindowMemory:** This approach is similar to the first one, but we add a window to the memory, allowing the pipeline to forget interactions that are further away than the specified window size. While this also allows for longer conversations, the disadvantage is the LLM does not know anything about the interaction outside of the window.

For our implementations, we used the ConversationBufferMemory approach.

2.4 Model evaluation

In this sub-section, we will review which evaluation approaches are employed to evaluate and compare the performance of our chatbots.

[2] Use the following pieces of context to answer the question at the end. If you don't know the answer, just say that you don't know, don't try to make up an answer.

Overall, evaluating chatbots is challenging and requires examining language understanding, reasoning, and context awareness. Thus, it remains an open question requiring further research in academia [9]. However, to provide some kind of comparison of our models, we will take two approaches: manual evaluation and GPT-4 evaluation. The following paragraphs dive into those two approaches in more detail.

2.4.1 Manual evaluation

As the name already suggests, in this approach, we evaluate all answers to our models manually based on whether the answer is correct and pertinent to the original query. To simplify, we classify each answer into "correct" and "incorrect".

2.4.2 GPT-4 Evaluation

This approach was originally suggested by the VICUNA team [9]. In the original approach, the VICUNA team used GPT-4 to evaluate and compare the answers from their own VICUNA model with other models, such as Alpaca, LLaMa, GPT-3.5, and Bard. We took their prompt and adjusted it slightly. Instead of comparing two models, we asked GPT-4 to evaluate the answer of our model with a prompt[3]. This approach allows us to retrieve a relatively objective score and qualitative justification of the score for each answer and model and subsequently compare the performance.

3 Results

This section shows the overview of the results based on the evaluation methodologies as explained in the previous section.

3.1 Numerical Results

3.1.1 High-level Results

Table 3.1 presents a comparison of five different models, including GELECTRA, GELECTRA FINETUNED, GPT-3.5, VICUNA, and VICUNA FINETUNED. The metrics used to assess these models are from the manual evaluation and the GPT-4 evaluation (i.e., automatic), both of which capture different aspects of the model performance. In the manual evaluation we did a binary classification into correct (1) and incorrect (0) answers using our human judgment. The table shows the overall accuracy of that classification. In the GPT-4 evaluation, we asked GPT to rate the answer on a scale of 1-10 with the prompt described in section 2.

In the manual evaluation, it is notable that GPT-3.5 demonstrates the highest score at 76%. This implies that based on human evaluation, GPT-3.5 is perceived to be more accurate or effective in its output compared to the other models. Following GPT-3.5, VICUNA FINETUNED, and VICUNA score 56% and 60%. While the fine-tuning for Vicuna improved the answers to some questions, it also worsened the answers to other questions so there was not a significant difference in overall performance in the manual evaluation. Although lower than GPT-3.5, this score still highlights VICUNA models as highly competitive given that the model size of the chosen VICUNA model is considerably lower. On the other hand, the

[3] You are a helpful and precise assistant for checking the quality of the answer. Question:'question' Response: 'answer' We would like to request your feedback on the performance of the AI assistants in response to the user question displayed above. Please rate the helpfulness, relevance, level of details of the responses. Each answer should receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. Please first output the score on a scale from 1 to 10 followed by '-|-' as a separator and a comprehensive explanation of your evaluation, avoiding any potential bias

GELECTRA models show the least effectiveness in the Manual Evaluation, with the original version scoring 36% and the fine-tuned version further dropping to 24%.

The GPT Evaluation, tells a somewhat different story. While GPT-3.5 still leads the pack with a score of 9.04, the gap to the other models is noticeably smaller than in the manual evaluation. The original VICUNA model takes the second position with a score of 8.0, closely followed by its fine-tuned version at 7.06. The GELECTRA models again come in last, but with a less drastic difference from the other models, scoring 6.74 for the original and 5.26 for the fine-tuned version.

Model	Manual Evaluation	GPT Evaluation
GELECTRA	36%	6.74
GELECTRA FINETUNED	24%	5.26
GPT-3.5	76%	9.04
VICUNA	56%	8.0
VICUNA FINETUNED	60%	7.06

TABLE 3.1: Results overview

Overall, it is evident that GPT-3.5 outperforms the other models in both manual and GPT evaluations, suggesting it might be the most robust choice of model. Meanwhile, the VICUNA models show consistent performance, scoring comparably in both evaluations. The GELECTRA models, particularly the fine-tuned version, may require improvement or better fine-tuning strategies, as they consistently rank the lowest in both evaluations. We can conclude that both the GELECTRA and VICUNA models would benefit from better data for the finetuning in terms of both quality and quantity as the finetuning actually decreased the accuracy of the model for GELECTRA and did not make a noticeable difference for VICUNA.

3.1.2 Comparison by Embedding

As mentioned previously, the impact of the embedding, and thus which passages were retrieved by the pipeline, can have a significant impact on the result. To show the impact, we also evaluated the results from GELECTRA once with the OpenAI embeddings and once with the HuggingFace embeddings.

GELECTRA FINETUNED	ACCURACY
OPEN AI embeddings	24%
HuggingFace embeddings	12%

TABLE 3.2: Comparison between HuggingFace and OpenAI embeddings

From Table 3.2, it can be observed that the OpenAI embeddings allow for better passage retrieval, as the accuracy based on the manual evaluation is lower when we use the HuggingFace embeddings.

3.2 Qualitative Results

3.2.1 Answers on selected questions

Figure 3.1 shows an overview of answers generated by the models on selected questions.

Example 1: An wen müssen Probleme aus dem Lehrverhältnis gemeldet werden?

GELECTRA	das Berufsbildungsamt
GELECTRA FINETUNED	das Berufsbildungsamt
GPT	Probleme aus dem Lehrverhältnis müssen an den Berufsinspektor resp. die Ausbildungsberaterin gemeldet werden.
VICUNA	1. Als Berufsbildner/in 2. Eine Drittperson (z.B. Vorgesetzte/r, Praxisbildner/in, Personalabteilung usw.) 3. Der Lehrling selbst (falls er noch nicht 18 Jahre alt ist, inklusive gesetzlicher Vertretung) 4. Keine Antwort
VICUNA FINETUNED	1. Den Ausbildungsberater/in des Berufsbildungsamts.

Example 2: Ist die Krankentaggeldversicherung in der Schweiz gesetzlich vorgeschrieben?

GELECTRA	Antwort
GELECTRA FINETUNED	Regel 720 Tage zu 80 Prozent).
GPT	Nein, die Krankentaggeldversicherung ist in der Schweiz nicht gesetzlich vorgeschrieben. Sie ist jedoch in vielen Branchen üblich und wird in einigen Gesamtarbeitsverträgen (GAV) vorgeschrieben.
VICUNA	Ja, die Krankentaggeldversicherung ist in der Schweiz gesetzlich vorgeschrieben. Die Krankentaggeldversicher
VICUNA FINETUNED	1. Ja, es

FIGURE 3.1: Overview of example answers

In Example 1, concerning the query, "To whom should problems from the apprenticeship relationship be reported?" (translated from German), both the fine-tuned and standard versions of GELECTRA accurately respond with "das Berufsbildungsamt". The GPT model not only gives the correct answer but expands upon it, providing a more detailed response. Interestingly, the effect of fine-tuning VICUNA for this question is evidently positive. The fine-tuned version delivers the correct response, whereas the standard version presents an incorrect and poorly structured answer.

Example 2 asks, "Is daily sickness benefit insurance legally required in Switzerland?". Unfortunately, both the standard and fine-tuned versions of GELECTRA and VICUNA fall short of providing the correct answer, with GPT being the only model to accurately address the query. An interesting observation here is the varied impact of fine-tuning on the different models. In the case of GELECTRA, fine-tuning demonstrates a modestly positive impact on the output. The original model's output with the term "Antwort" does not make sense at all, whereas the output of the fine-tuned model, although still incorrect, conveys a more sensible meaning. However, for VICUNA, the process seems counterproductive, as the answer from the fine-tuned model abruptly ends, suggesting a negative effect of the fine-tuning in this particular instance.

3.2.2 Explanations from Automatic Evaluation

With the help of GPT-4, we are able to do a qualitative evaluation of the generated answers automatically. Figure 3.2 shows one example of such an evaluation.

Question	Wer bleibt bei Schulversäumnissen zur Mitunterzeichnung der Entschuldigungen verpflichtet?
Generated answer	Die Schule ist verpflichtet, den Berufsbildner/innen zu berichten, wenn die Schulleistungen einen Prüfungsmisserfolg befürchten lassen.
Explanation	The response provided by the AI assistant is not directly relevant to the question asked. The question is about who is obligated to co-sign excuse notes for missed school days, while the response discusses the school's obligation to report poor academic performance to vocational trainers. The answer does not address the specific query and therefore receives a low score for helpfulness, relevance, and level of detail.

FIGURE 3.2: Example of qualitative justification for GPT score

The GPT-4 model accurately assesses that the answer provided is not capturing the essence of the question and also explains why this is not the case.

4 Discussion

This project aimed to build an intelligent chatbot designed to assist users in finding information related to apprenticeships in Switzerland. With the advent of state-of-the-art language models such as GPT-3.5/4 and VICUNA, a new and promising avenue for achieving this has opened up. As mentioned in the introduction, given that these technologies are still in the evaluation phase and best practice guidelines have not yet been established, our work here can be regarded as pioneering. Therefore, this project serves as an essential first step, demonstrating a viable proof of concept that lays the foundation for further development and optimization. In the following, we will discuss the main insights gleaned from this venture, along with considerations that should guide subsequent research and development efforts. In total, we outline five considerations to be considered.

Language Models: BERT-based vs. State-of-the-art LLMs

Our evaluation has shown that the state-of-the-art LLMs developed in recent times significantly outperform older LLMs, such as the BERT-based GELECTRA model. Besides the different architectures, we believe that the model size and the bigger dataset size on which the newer models have been trained are the most critical reasons. Our observation also validates the hypothesis that the state of the models can power effective question-answering chatbots. Furthermore, when comparing GPT-3.5 and VICUNA, it was also observed that GPT-3.5 outperforms VICUNA. Again, we believe this is due to the model size as GPT-3.5 is significantly larger than the VICUNA version we have selected.

Due to the high development pace in the LLM space, there have been many new language models developed since the beginning of the project. Amid our project, we decided to include Vicuna to capture another state-of-the-art LLM besides GPT. However, in the future, we would suggest testing larger and even more of the newer models, such as larger versions of Vicuna (i.e., 65B parameters) or the Falcon model.

Impact of Retriever and Embedding

As mentioned, the retriever has a strong impact on the performance of question-answering chatbots as LLMs are only as good as the retrieved passages. We analyzed the impact of the embeddings and observed that when using the HuggingFace embeddings, the accuracy (i.e., manual observation) dropped significantly compared to using the OpenAI embeddings. This observation underlines the importance of embedding systems and more so the one of retriever models.

In the future, the retriever pipeline could be further investigated. One approach could be to fine-tune your retriever model based on one's documents leveraging models from, e.g., Haystack. Unfortunately, we have had limited success with this approach. However, we believe that optimizing the retriever is an important consideration that should not be neglected.

Fine-Tuning of LLMs

In section 3.1.3, we compared the results from the non-finetuned GELECTRA / VICUNA model vs. the one from the fine-tuned model. Even though our dataset is comparably low, it was observed that fine-tuning had limited (for VICUNA), or even negative (for GELECTRA) impact on the performance. We believe the performance decrease stems from two reasons (i) the limited dataset quality and (ii) using BERT-based question & answer pairs.

In the future, we suggest investing more time in the dataset generation. While the automatic generation of the dataset was time-efficient, manual review and refinement is needed to ensure quality and reliability. This highlights the critical balance between automation and human review in the dataset creation process [11]. Additionally, we also suggest investing more time in generating a dataset specific for newer LLMs by collecting high-quality user conversations about Swiss apprenticeships in German.

Evaluation Approaches

As mentioned previously, there are currently no established methodologies on how to evaluate chatbots, a question that remains to be addressed by research. Besides the manual evaluation, we also used an automatic evaluation approach designed by the VICUNA team. In many cases, the manual evaluation and the scores of the automatic evaluation seem to correlate. More interestingly, it can be observed the score of the automatic evaluation for GPT 3.5 is higher compared to VICUNA and also the differences between different models with GPT evaluation are less drastic than with the manual evaluation. This is in line with the findings of a recent paper [12], that found that the approach of GPT evaluation might be biased because imitation models like VICUNA are adept at mimicking ChatGPT's style but not its factuality. They found that ChatGPT and their best imitation models produce answers with a similar style, but while ChatGPT's answer is mostly correct, the imitation model's answer is completely inaccurate in some cases despite sounding authoritative and ended up getting good scores in the automated GPT evaluation despite being factually wrong.

In the future, it is important to explore other evaluation methodologies that are not leaning towards one specific model.

Scalability of Document Database

While we spent a significant amount of time engineering a dataset for fine-tuning, we spent limited time scaling up our document database, which ultimately consisted of factsheets and the collection of Q&As we received from the partner. The limitedness and specificity of the information content presented an issue at times. For instance, when we asked GPT-3.5, the question of whether it is possible to shorten the vacation days of apprentices due to sick leave, the model provided the answer specifically for poly-mechanics. In case there are more general information chunks available in the document database, a well-designed retriever setup or prompt can alleviate the issue of specificity, but less so of limitedness.

In the future, especially when going to production, we suggest integrating more general documents that help answer a variety of questions from users. This will increase the usefulness of the tool by being able to answer more questions at the right level of detail.

Conclusion

In conclusion, our extensive evaluation and comparison of various language models, including GELECTRA, GELECTRA FINETUNED, GPT-3.5, VICUNA, and VICUNA FINETUNED, have yielded several noteworthy insights that can significantly inform the future of language model application in chatbots in the realm of information retrieval overall, but also with a focus on apprenticeships in Switzerland.

Primarily, this project has reaffirmed the efficacy of advanced language models like GPT-3.5 in generating accurate and relevant responses, as judged both by manual human assessment and automated evaluation using GPT itself. Additionally, the results show that the small version of the VICUNA model is outperformed by GPT-3.5, but by a significantly smaller margin than compared to GELECTRA. This indicates that there is further potential to improve by fine-tuning with better data or using a larger version of VICUNA.

Overall, we outline five recommendations along different dimensions that should be considered in future project steps, such as the retriever setup, document store, and evaluation approaches. We hope that our findings and insights serve as a robust springboard for future endeavors, sparking more innovative solutions for the challenges that still lie ahead.

Bibliography

[1] Federal Statistical Office. Vocational and professional education and training (vpet), 2023. URL https://www.bfs.admin.ch/bfs/en/home/statistics/education-science/pupils-students/upper-secondary/vocational-training-apprenticeships.html.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Ettinger, Suchin Goyal, Ellie Pang, Neoklis Polyzotis, Steven Skiena, and Kathleen McKeown. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[5] Technology Innovation Institute. Introducing falcon llm, 2023. URL https://falconllm.tii.ae.

[6] Branden Chan, Stefan Schweter, and Timo Möller. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.598. URL https://aclanthology.org/2020.coling-main.598.

[7] Timo Möller, Julian Risch, and Malte Pietsch. Germanquad and germandpr: Improving non-english question answering and passage retrieval, 2021.

[8] OpenAI. Openai api, 2023. URL https://platform.openai.com/docs/guides/gpt.

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[10] James Briggs and Francisco Ingham. *Conversational Memory for LLMs with Langchain*. 2023.

[11] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[12] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

Declaration of Authorship

We declare that this report, and the work presented in it as our own. We confirm that:

- This work was done wholly while in candidature for a degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signatures:

Schnydrig

Aydin Kaan　　　　Schnydrig Sven