Case Study *Biography Generator* V2-

Fernuniversität in Hagen

Executive Summary

Biographical interviews are an important source in the humanities. Usually, they feature complex narratives that span a biography from birth until the day of the interview. Often, these interviews are conducted to answer specific research questions and after finalizing the research project they are conveyed to an archive. Usually, writing a short biography about the interviewee is part of the Oral History routine. Nevertheless, often collections are archived without biographies. In order to help archivists and researches to gather information about the interviewee, AI should be used to automatically generate short biographies out of an interview.

Introduction

Biographical interviews cover the perception of social and historical phenomena and are an established technique in contemporary history, qualitative social sciences, psychology, pedagogy and marketing. The *Institut für Geschichte und Biographie (IGB)* at the *FernUniversität in Hagen* is a pioneering institution for Oral History in Germany and influenced theory and practice of the biographical interview in Germany significantly. The research data archive of the IGB, the *Archiv "Deutsches Gedächtnis" (ADG)* holds several thousand interviews as digital transcripts, audio and video files which originated as well in research projects conducted by the IGB as in external projects. Usually, after conducting a biographical interview, the researcher writes a short biography of roughly 500 words to provide information about the interviewee for secondary analysis. Nevertheless, from time to time short biographies are missing in the collections of interviews and creating them afterwards by hand is a time-consuming procedure. In the light of recent developments in generative AI, the task is to setup a pipeline for automatically summing up a biographical interview while taking care of the GDPR as the interviews are sensitive data. Preceding tasks have confirmed the good performance of the LLM Mistral 7B. In this follow-up, different prompting approaches and different chunk sizes should be evaluated to optimize the pipeline. Chunking (splitting the interview transcript into smaller pieces) is necessary to deal with the huge documents.

General Idea of the Proposed System

We have some experience with Python und Jupyter Notebooks, so the pipeline could be set up either as Python Code to be run locally or – and this is the preferred solution – as a Jupyter Notebook.

EPICS

1. Evaluating text understanding and text generation in German.
2. Setting up a GDPR-conform and user-friendly pipeline that takes a biographical interview with up to 80.000 words as input and creates a short biography of about 500 words.

User Stories

EPIC 1: Evaluating text understanding and text generation in German.

US1.1: As a researcher, I want to find all recent papers on prompt-engineering and pre-processing for generative AI for German, so that I understand the state-of-the-art.

US1.2: As a data scientist, I want to find the most appropriate workflow for chunking interviews and prompting in order to get a good biography of the interviewee.

EPIC 2: Setting up a GDPR-conform and user-friendly pipeline that takes a biographical interview with up to 80.000 words as input and creates a short biography of about 500 words.

US 2.1 As a data scientist, I want to set up a CoLab with a GDPR-conform and user-friendly pipeline for generative AI based on the results of EPIC 1.

US 2.2: As a historian, I want to find a CoLab, where I can start a pipeline that takes as input the transcript of a biographical interview of up to 80.000 words and returns a biography of the interviewee with a length of 500 words.

Expected Outcomes of this Case Study

The anticipated pipeline should provide historians with the ability to use generative AI for automatically creating short biographies of the interviewee out of a biographical interview.

Detailed Architecture of the Existing System

- Google Colab
- Python

Detailed Architecture of the Proposed System

- Google Colab
- Python
- LLMs like LlaMa or GPT