Design Document: WikiCheck

Group Number: 31

Fabian Kaltenegger	BackEnd and Pipelines
Bastian Kandlbauer	FrontEnd and Chrome Extension
Florian Fellner	Preprocessing and Evaluation

November 28, 2025

Abstract

WikiCheck aims on providing users with an efficient tool for checking and searching for facts within the knowledge base of Wikipedia. The tool is specialized to accept queries and provide and display results from and for users with different language skills. It simply focuses on giving everyone the same opportunity to retrieve the same knowledge from one of the biggest knowledge bases of the modern internet.

1. Idea

The idea of the project is to develop a system, which efficiently enables users to check and verify several facts, formulated via a natural language query by matching the query against the knowledge base of Wikipedia. Since users live in different countries, and therefore also speak different languages, we want to equip our tool with a multilingual transformer model, which enables users to query the system in their chosen language.

Goal

The main goal of the project is to provide users with an efficient multilingual tool for matching facts with one of the most important knowledge bases of the internet.

Research Question

How can we efficiently use information sources like Wikipedia to help users quickly assess and inform about established factual knowledge given some short external text (query), in order to oppose the spread of fake knowledge and independent of any language differences between the query and the articles?

Problem Statement

In the latest era of the internet, users often encounter the problem of retrieving fake news while searching for facts throughout the web. It can also be very time consuming to back-check the facts found throughout the search process. Also for people with different backgrounds and different language skills, it can often be very challenging to find and understand facts formulated in a different language.

2. Main Task

In order to tackle the problem formulated in section 1, our focus lays on accessibility and efficiency throughout the search and back-check process of facts.

The goal is to develop a web app or/and a browser extension which enables users to select any short text, which can be a statement, claim or phrase, which will be used as query and therefore a check based on presence and context is executed within the scope of Wikipedia. The results of such process are going to be links to the related and relevant Wikipedia articles. Another main focus point is to provide users with the ability to formulate their query and retrieve their results in their chosen language.

3. Dataset & Processing

The used dataset is provided by WikiMedia (`https://huggingface.co/datasets/wikimedia/wikipedia`) and hosted on the platform Hugging Face.

Dataset

The used dataset is basically a snapshot of several Wikipedia articles in multiple languages. Those articles include the title of the article, the text, as well as the link towards the article itself.

Processing

The dataset already contains cleaned versions of the Wikipedia articles, which means the cleaning of text from tags, tables and lists is not necessary. However most articles contain section headings which probably have to be removed or at least used to structure the articles in subsections. So, in order to better manage any articles which appear to be very long, we have to test and evaluate the option to split those up into more manageable text-chunks.

4. Methods/Models

Model

For the model we are planning to use mBERT, which is the multilingual model of BERT, in order to encode both the user queries as well as the documents (Wikipedia articles). For now we decided to only support one language, which the model is already capable of handling.

Pipeline

User Query:
User formulates a query in natural language → query will be be tokenized and processed into vectors using the BERT model → vectors are then matched / compared with the vectors of our already processed articles → relevant results are returned by the system.

Articles:
Initially the articles are getting preprocessed and divided into corresponding subsections → articles are processed and vectorized by the BERT model

Extension/Web App

We are planning to give users the option to use our tool as a simple web app, where the users can just formulate their own query in natural language, and also to maybe offer the user a google chrome extension, where simply the text of a website could be selected and via right click → WikiCheck the check is performed and the answers are retrieved and displayed within a small popup window in the browser.

5. Evaluation

Test Set

Based on our dataset, we will create a test set of query-article pairs and use it as groundtruth. To evaluate the models, we are going to use the test queries as input and compare the output to the expected results.

Metrics

In order to quantify the output, we are going to use Recall@k and Mean Reciprocal Rank (MRR) for both the tf-idf pipeline as well as for the mBERT pipeline.

Visual Depiction

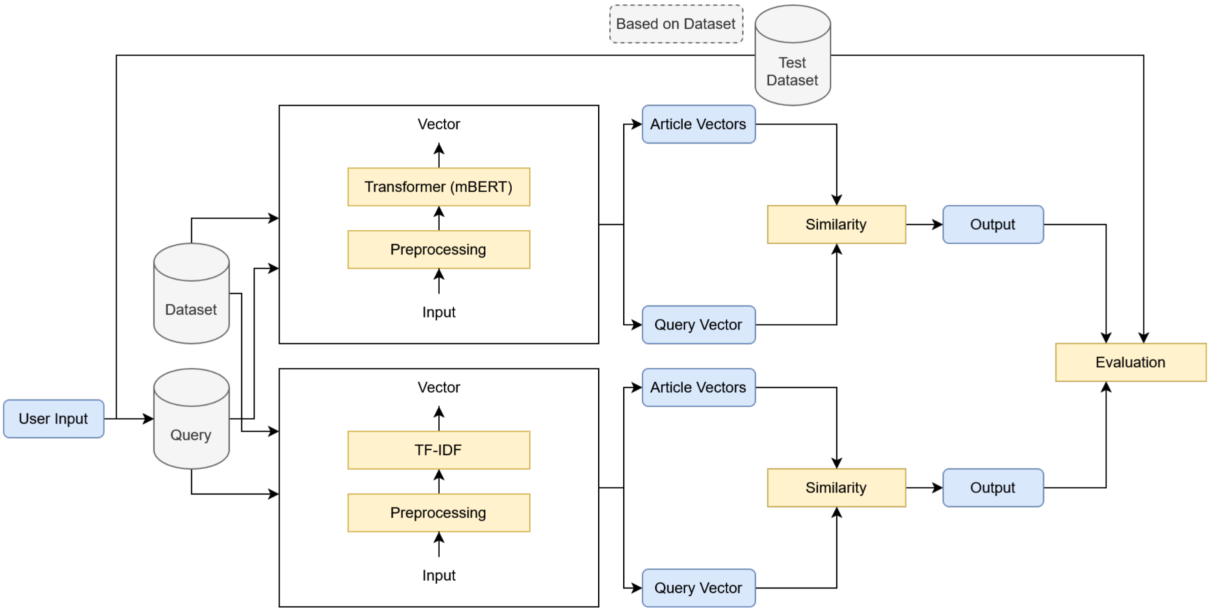

Figure 1: Visual depiction of the project