CheatApp

We are going to develop a cheating app for open domain question answering systems through a notebook. In this app, we would like to suggest users of the wikipedia page with the relevant answers for given questions. To further stretch the challenge, we would like to suggest the best paragraphs having the answers of the questions in the corresponding wikipedia page. Below are few examples -

Question: how are glacier caves formed ?
wikipedia page - Glacier cave - Wikipedia
*paragraph : 'A **glacier cave** is a cave formed within the ice of a glacier. Glacier caves are often called ice caves, but the latter term is properly used to describe bedrock caves that contain year-round ice' (summary of the page).*

Question - how much is 1 tablespoon of water ?
wikipedia page -https://en.wikipedia.org/wiki/Tablespoon
paragraph is - It has multiple answers. It could like -
'In most places, except Australia, one tablespoon equals three teaspoons—and one US tablespoon is 14.8 ml (0.50 US fl oz; 0.52 imp fl oz) or 15 ml (0.51 US fl oz; 0.53 imp fl oz).'
Or
'In nutrition labeling in the U.S. and the U.K., a tablespoon is defined as 15 ml (0.51 US fl oz).[7] In Australia, the definition of the tablespoon is 20 ml (0.70 imp fl oz)' etc.

Question - how did anne frank die
wikipedia page - https://en.wikipedia.org/wiki/Anne_Frank
Paragraph - 'Following their arrest, the Franks were transported to concentration camps. On 1 November 1944,[2] Anne and her sister, Margot, were transferred from Auschwitz to Bergen-Belsen concentration camp, where they died (probably of typhus) a few months later. They were originally estimated by the Red Cross to have died in March, with Dutch authorities setting 31 March as the official date. Later research has suggested they died in February or early March.'

Expectation

Given this is an open problem, we don't expect a particular level of correctness. What we are mainly looking for - how you approach and quickly prototype crappy solutions. Then you keep adding complex logic in iterations to achieve some satisfactory levels. While doing that journey, we expect that you may generate following artifacts -

1. Hypothesis and motivations for choosing different modeling techniques.
2. How you measured the model performance.
3. Data curation, training/evaluation data generations, model performance measurements etc.
4. end 2 end machine learning pipeline in python notebook including above steps.

5. Also, what constraints you felt which led you not to try the things you wanted to do to solve this problem is an awesome way.

** - If you use an already available model/code/library from the web, we expect that you have a full understanding of motivation and why you are using it. Ex:- if you use entity linking library, we expect that you understand - pros and cons of that model. This includes - Why do you think your chosen entity linking library is good for your problem? When do you expect your chosen model may behave poorly?

Resources

You are free to use open source resources including already available annotated training data on the web. Also, free to use already trained models & libraries existing in open source. What we mainly expect is - how you approach the problems and journey.

You are not allowed to use llm libraries like Langchain and LammaIndex.

Wikipedia text data is available in Kaggle at - wikidata-text
Also added sample open questions and expected answers - wikipedia_question_similar_answer.tsv . The answers added here are not exact wikipedia graphs, but it may be super helpful for your modeling techniques.

Other open source resources that can be used are - https://paperswithcode.com/dataset/wikiqa (questions in wikipedia_question_similar_answer.tsv is taken from this data set).

Notes

- Please create a loom video explaining all solutions/approaches.