LLM-based hallucination correction and retrieval enhancement Generate schematic designs

Qin Qi

Hust

2024/12/14

summary

The generation of hallucinatory questions (such as answer authenticity bias) by large language models (LLMs) in open-ended question answering is one of the important directions of current artificial intelligence research. Based on the TruthfulQA dataset, this paper designs a Retrieval Enhancement Generation (RAG) framework to explore how to improve the authenticity judgment of LLM for open Q&A through real-time website information crawling and fusion. Experimental results show that the accuracy of the retrievally enhanced model on the dataset with error in judgment is increased to 41%, which demonstrates the potential of WebGPT for quantitative hallucination research.

background

Although LLMs exhibit strong language comprehension and generative capabilities when generating content, the problem of generating hallucinations, i.e., the phenomenon that the generated information does not match the facts, still hinders its application in high-reliability scenarios [2]. TruthfulQA is a dataset that focuses on authenticity issues and is designed to measure whether a large model is able to generate factual responses. However, when I ran an experiment based on TruthfulQA, the LLM was only 70% correct in its original judgments, and out of 5,918 judgments, 1,827 were incorrect. This phenomenon shows that there are still significant authenticity problems in the LLM generation process. [5]

In order to improve this situation, I propose a Retrieval Enhancement Generation (RAG) framework [3], which supplements the model knowledge by introducing real-time website information crawling, and redesigns the Prompt structure, hoping to improve the LLM's ability to judge the authenticity of answers. [4]

data

TruthfulQA: Measuring How Models Mimic Human Falsehoods[1] constructs a dataset that exposes the illusion of LLMs. The benchmark contains 817 questions covering a variety of categories and domains, and a significant number of them can be classified as open-ended questions, some of which can even be answered incorrectly by humans. In order to perform well, the model must avoid generating the wrong answers learned from mimicking human text.

Evaluation and Reflection: In pursuit of authenticity, almost all the questions in TruthfulQA are artificially generated, and even the answers given by the model are checked and scored by humans. This is a good dataset, and I plan to improve the accuracy of LLM judgments with a fully automated RAG framework.

code

QA.py

Length: 79 lines
路径：quantify/QA/ QA.py
输入：TruthfulQA.csv
输出：Judge_result.csv
Ideas:

Define the generate_judge function, which takes a question and an answer as input and generates a prompt for evaluating the correctness of the answer. It then calls the API to get the evaluation result, “Correct” or “Incorrect”.

Define process_data function that is responsible for handling TruthfulQA.csv. Traverse each row of data in the input file to extract questions, categories, sources, correct answers, and incorrect answers to form multiple benchmarks. Call generate_judge function for each benchmark to get the judgment result.

filter.py

Length: 35 lines
Path: quantify/QA/ filter.py
Input: Judge_result.csv
Output: wrong.csv
Idea: Extract the entries in Judge_result.csv that were judged wrong and put them into wrong.csv.

reflect.py

Length: 223 lines
Path: quantify/QA/ reflect.py
Input: wrong.csv
Output: Informed_result.csv
Ideas. Library Description: csv library for processing CSV files; openai library for interacting with APIs; fake_useragent library for generating random user agents; requests library for processing HTTP requests; BeautifulSoup library for parsing HTML; rake_nltk library for keyword extraction.

find_paragraph_with_most_keywords: This function finds the paragraph with the most keywords in the list of paragraphs.

split_long_paragraphs: This function splits long paragraphs into shorter ones.

fetch_and_parse: This function fetches and parses HTML content from the specified URL, up to three times.

extract_information: This function extracts information from the specified URL and extracts keywords based on the question. It then finds the paragraph with the most keywords in the question. paragraph with the most keywords in the HTML content and returns that paragraph as information.

generate_informed_result: This function accepts information, questions and answers as input and generates a prompt for evaluating the correctness of the answers. Then, it calls the API to generate an evaluation result, the result will return “Correct” or “Incorrect”.

process_data Traverse each row of data in the input file, extracting questions, categories, sources, correct answers, and incorrect answers. For each benchmark call generate_informed_result function generates the evaluation results and writes the results to the output file.

Enhance access to information. I tried a lot of things, and finally found that it was still a simple and crude method that worked. This question is far more difficult than you think. Because there is so much text in the web page, and the truth is really only a small paragraph. In particular, there are many different entries for information on the Wikipedia page.

As for why the split was based on the double line break in the first place, it came from experimental observations. Often, where two consecutive line breaks occur, the context's information is incoherent.

As for the design method of paragraph weighting, it is also derived from experimental observation. You can't just count keywords, for example, some keywords have cat, and some texts just list a dozen types of cat, which greatly increases the weight, which is not helpful for answering questions. Fortunately, the phenomenon of "keyword listing" is often accompanied by a single line break. In this way, the problem of "weight fraud" can be mitigated by "penalty factors".

bibliography

[1] Lin S , Hilton J , Evans O . TruthfulQA: Measuring How Models Mimic Human Falsehoods[J]. 2021.DOI:10.48550/arXiv.2109.07958.
[2] Zhu Z , Yang Y , Sun Z . HaluEval-Wild: Evaluating Hallucinations of Language Models in the Wild[J]. 2024.
[3] Aly R , Guo Z , Schlichtkrull M ,et al. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information[J]. 2021.DOI:10.48550/arXiv.2106.05707.
[4] Askell A , Bai Y , Chen A ,et al. A General Language Assistant as a Laboratory for Alignment[J]. 2021.DOI:10.48550/arXiv.2112.00861.
[5] Baly R , Karadzhov G , Alexandrov D ,et al. Predicting Factuality of Reporting and Bias of News Media Sources[J]. 2018.DOI:10.48550/arXiv.1810.01765.