Report
Final project: chat bot with RAG and finetune
Name: Tung Dinh
Date: 24.11.2023

Target: build a chatbot to answer question about bad food.

1. Description:

TODO 1: Prepare dataset

- Crawl data based on the topic obesity, from high-reputation resources
- Divide into sub-topics: processed food, obesity, fast food, diet
- In each sub-topcis, there are 4 writings
- Use OpenAI to generate question-answer pairs:
  - OpenAI API is costly => use chatgpt 3.5 with prompt

You are an examiner, you create 10 pairs of question and answer on the given text.
If possible, there should be 1 listing question.

Question: should be max 50 word long, and it should be generalized like an average human question, not so detailed. It is named <question>

Answer: should be max 200 words long, do not make up the information, only use information in the text to answer

Output format should be a list of json, in which json contains question and answer as follow: [{"question":<question>, "answer":<answer> }]

Text

  - Some question need to be twisted so that they are human-liked questions => quality is not as good as human expert
- Augmentation:
  - For each QA pair, other 5 augmented versions are generated by chatgpt3.5

TODO 2:
The scope of the project is to answer questions related to the topic obesity, prompting works well up to now. However, further development related to nutrients calculation and conduct continous conversation, then few-shot and chain of thought will be useful.

For demonstration purpose, I created a case study in order to create a continuous conversation, where zero shot, few shot and chain of thought can be compared.
Case study: I want to understand the command of user, is it a yes or a no; does user want to do in the same page, does user give a set of pages

- Prompting:
  - Zero shot: zero shot works well on simple task, not on complicated tasks

```
def has_page_numbers(question):
    question_full = f"""
    Does the below input contain page numbers? If yes, which page, answer as a list of integer. If not, return an empty list
```

```
    ###
    Question: {question}
    """
    return answer_default(question_full)
```

- Few-shot: few shot works well in almost all of the test, for example

```
def is_same_page(question):
    question_full = f"""
    Does the question contain 'the same page' or in the 'current context' or anything similar? Answer only True or False
    Example:
    what is a loss function, using the same page => Return True
    Can you look for the answer to what is a neural network in page 1, 2 and also 3 => Return False
    ###
    Question: {question}
    """
    return answer_default(question_full)
```

- Chain of thought: in this study case, change of thought does not result consistently and correctly => not reliable in this studycase

```
def get_question_info(question):
    question_full = f"""
    Extract information from the question. Follow the steps:
    Q1. Is it a question/request? Answer <A1> True or False
    Q2. Does user mention about the same context, same page? Answer <A2> True or False
    Q3. Does user mention about page number? Answer <A3> if user mentions page numbers, get them as a list of page numbers, else return an emtpy list
    ###
    Question: {question}
    """
    return answer_default(question_full)
```

QA chatbot using prompting outputs as expected in this scope of answering question about obesity

TODO 3:

2. Answer 2 in 3

- Analyze the performance of different techniques: RAG vs Finetune
  - RAG can perform well on a small dataset of text, it can search fast, and the answer is similar to the context, *it remembers the text not much creative*, even with temperature = 0.7. And the amount of token needed to be sent to API is bigger => cost higher in using, cheaper in development
  - Finetune can perform well in different contexts, given that it has large + diverse enough training dataset, recommended 1000QA pairs. The answer could be creative, however, it stays close to what it is expected, and it is consistent. It send minimal amount of text to the API => cheaper in using, expensive in development. Finetune is a costly process.
- Change the amount of data in finetune:
  - Test using only 50 different pairs QA => result bad, model doesnt learn, the answers are wrong in comparison with the training answers.

    - Test using 50 pairs QA about processed food only => model performs well on the topic of processed food, it can answer similar to the answer, sometimes, it remembers and answers the same training answers.
    - Test using 450 pairs pairs QA (150 original and 300 augmentation) => model can performs pretty good on the 4 topics. Reason: in each topics, there are 4 writing with close context, for example: 1. difference-between-processed-and-ultra-processed-foods 2. nova-classification-ultra-processed-foods,.. Because of this, the QA pairs generated from the text stay close in their context => acceptable performance.
- For RAG, changing chunk_size and chunk_overlap observation: with grid search, try chunk_size from 128 to 2000, chunk_overlap is 10% of chunk_size => best perform at chunk_size 256 and chunk_overlap 20

2. Future work:

- Software engineering
    - Store user information
    - Collect user information
    - Collect nutrients database
    - Set reminder to user about food
- Product development
    - Friendly UI
    - Has animation
    - Can recommend food and analyze food user inputs
    - Make into an app
- ML engineering
    - More training data, diverse, and from human expert, not QA from models
    - Perform chat context with history
    - Filter politics and hate speech
    - Work on body shaming
    - Work on chatting style => motivation
    - Flag topics that the chatbot doesnt know

3. Challenges:

- Software engineering:
    - Fully develop an friendly UI
- Operation
    - Data generation is a costly process
    - Finetuning is a costly process
    - It is difficult to control the quality of the chatbot