Agenda

01 Summary of 1st Term
02 Objective of 2nd Term
03 Experiment & Analysis on ChatGPT
04 In-context Learning
05 Prompt Design for Historical Bias
06 Conclusion
01 Summary of 1st Term
Objective

- Closed-domain Question-Answering
  - Process a question → output an answer
  - Focus on a specific domain
- Advantages:
  - Accurate
  - Efficient
Architecture

- A question decomposition model
  - decompose the compositional questions
- An extractive QA model
  - find the answers to the questions in the documents
- A ranking model
  - rank all candidate answers and decide the most possible one
Test result

Test with 2 datasets extracted manually by documents from different field:
- NLP
- Bank Service

We test whether it can:
- Decompose the sentence correctly.
- Find the context that contains the answer.
- Output the correct format of the answer, especially for those containing terms or special phrases.
Objective of 2nd Term
Objective

Goal: Study chatbots

How do we learn chatbot?
• Learn the theories related to chatbots and build one ourselves (Term 1)
• Analyze and improve existing chatbots (Term 2)
  • Our focus: ChatGPT
ChatGPT

- AI chatbot built on top of OpenAI's GPT-3.5 and GPT-4 families of large language models [1]
- Can generate high-quality, human-like responses for various NLP tasks
- Gains immense popularity due to its ability to produce high-quality responses
- Studies revealed its tendency to produce factual and reasoning errors
  - Our focus: QA capability
Experiment & Analysis on ChatGPT
Experiment Setup

• Evaluate ChatGPT on five QA benchmarks: SQuAD [2], HotpotQA [3], Common-senseQA [4], TruthfulQA [5], SuperGLUE [6], to test different aspects of reasoning capabilities.
Introductions on Datasets

- SQuAD: a reading comprehension dataset that need to extract the answer from the document.
- HotpotQA: a muti-hop QA dataset that need to generate an answer among many paragraphs.
- SuperGLUE: it contain many tasks. We select 3 of them that related to QA:
  - BoolQ: yes/no question about a paragraph
  - MultiRC: predict whether the possible answer is correct or not.
  - ReCoRD: select one from possible entities to fill the blank.
- CommonsenseQA: select the most reasonable answer to questions about common sense.
- TruthfulQA: select the correct answers from many misleading answers. We only use its MC1 and MC2 dataset, which are single selection and multiple selection.
Experiment Design

• Steps:
  • Randomly pick 100 samples from each datasets
  • Design appropriate prompts to make it better understand the tasks.
  • Summarize failed cases to find its weakness and problems.
  • Analyse the factors which cause these problems and try to avoid them.
Result

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>SOTA</th>
<th>ChatGPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD2.0</td>
<td>EM F1</td>
<td>90.578</td>
<td>39.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92.978</td>
<td>45.35</td>
</tr>
<tr>
<td>HotpotQA (Distractor set)</td>
<td>EM F1</td>
<td>67.46</td>
<td>40.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80.52</td>
<td>55.30</td>
</tr>
<tr>
<td>SuperGLUE (BoolQ)</td>
<td>Acc</td>
<td>92.4</td>
<td>73.0</td>
</tr>
<tr>
<td>SuperGLUE (MultiRC)</td>
<td>EM F1</td>
<td>65.8</td>
<td>61.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>89.6</td>
<td>88.421</td>
</tr>
<tr>
<td>SuperGLUE (ReCoRD)</td>
<td>EM F1</td>
<td>95.9</td>
<td>78.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.4</td>
<td>79.73</td>
</tr>
<tr>
<td>CommonsenseQA</td>
<td>Acc</td>
<td>78.2</td>
<td>80.0</td>
</tr>
<tr>
<td>TruthfulQA(MC1)</td>
<td>Acc</td>
<td>-</td>
<td>69.0</td>
</tr>
<tr>
<td>TruthfulQA(MC2)</td>
<td>Acc</td>
<td>-</td>
<td>73.9</td>
</tr>
</tbody>
</table>
Observation

• ChatGPT may output different answers to the same question
• ChatGPT always tend to output a long and detailed answers without restrictions
• ChatGPT could remember chat history but may forget it after many turns
• History may affect its output
Improving strategy

• In-context Learning
• Prompt Design for Historical Bias
In-context Learning

Step [7]:
1. Design a demonstration context containing a few examples
2. Take the demonstration and a query as the input
3. LLMs are responsible for making predictions
In-context Learning

Problem to be focused on:
- Poor performance on unanswerable questions in SQuAD 2.0
- Unable to identity unanswerable questions
Experiment Setup

Step:
1. Give an instruction on the problem
2. Randomly select 2 samples in training set to be demonstration examples
3. Explain the answer in the samples and repeat the instruction again
4. Input the question and use "Edit" for inputting next question
<table>
<thead>
<tr>
<th></th>
<th>Answer include *</th>
<th>Manually</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>-</td>
<td>14/50</td>
</tr>
<tr>
<td>ICL</td>
<td>20/50</td>
<td>24/50</td>
</tr>
<tr>
<td>SOTA</td>
<td>-</td>
<td>45/50</td>
</tr>
</tbody>
</table>
Further Exploration

Step:
1. Give an instruction on the problem
2. Randomly select 2 samples in training set to be demonstration examples
3. Explain the answer in the samples and repeat the instruction again
4. Input the question and use "Edit" for inputting next question
## Result

<table>
<thead>
<tr>
<th></th>
<th>Answer include *</th>
<th>Manually</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>-</td>
<td>14/50</td>
</tr>
<tr>
<td>ICL</td>
<td>20/50</td>
<td>24/50</td>
</tr>
<tr>
<td>ICL (without demonstration examples)</td>
<td>22/50</td>
<td>28/50</td>
</tr>
<tr>
<td>SOTA</td>
<td>-</td>
<td>45/50</td>
</tr>
</tbody>
</table>
Possible Reasons

- ChatGPT outputs different answers
- Demonstration examples make ChatGPT forget the instruction
- Design of demonstration examples
  - Number of samples is not enough
  - Better prompt can be used
Prompt Design for Historical Bias
What is Historical bias?

• Histories can impact the responses generated by ChatGPT, leading to potential bias.
Dataset used: gsm8k[8]

- contains 8.5K high quality, linguistically diverse grade school math word problems
- most problems are multi-step mathematical reasoning questions created by humans
- test how history impact mathematical reasoning ability of ChatGPT
Experiment setup:

Ask questions under different histories

- History1
- Questions

- History2
- Questions

2 independent conversation windows
Recap: ChatGPT may generate different response under the same input
Recap: ChatGPT may generate different response under the same input

• Find a question that ChatGPT may generate correct or wrong answers. Asking questions under to get a "Correct Case" and a "Wrong Case"
• Using "edit" options to keep all followed questions next to the "Correct Case" or "Wrong Case". (So that it won't forget the history)
Edit questions: example
Result

ChatGPT generates more wrong answers after the "Wrong Case" history

<table>
<thead>
<tr>
<th>History</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Case</td>
<td>88%</td>
</tr>
<tr>
<td>Wrong Case</td>
<td>68%</td>
</tr>
</tbody>
</table>
Further exploration

What if we ask ChatGPT to ignore the history?

Example:

Ignore the dialog history and answer the question: Dana can run at a rate of speed four times faster than she can walk, but she can skip at a rate of speed that is half as fast as she can run. If she can skip at 3 miles per hour, how many miles can she travel in six hours if she spends one-third of the time running and two-thirds of the time walking?

Dana can run at a rate of speed four times faster than she can walk, but she can skip at a rate of speed that is half as fast as she can run. If she can skip at 3 miles per hour, how many miles can she travel in six hours if she spends one-third of the time running and two-thirds of the time walking?
Result of "Ignore history":

18 of 32 wrong answers becomes correct. Ignore incorrect history could improve its performance!
Conclusion
Conclusion

01 Conducted a comprehensive study on chatbots

02 Built a QA system that is capable of handling question-answering on professional topics

03 Studied existing chatbots and evaluated their performance to identify areas for improvement
Q&A
Thank You!
Reference


