استاد:
دکتر اعتمادی

درس:
پردازش زبان‌های طبیعی

دانشجو:
محمدامین چینی‌فروشان اصفهانی – ۴۰۱۷۲۲۲۷۱

تمرین Prompt Engineering

تسک انتخاب شده

برای انجام این تمرین تسک Relation Prediction را انتخاب کرده‌ام. مطابق آنچه در سایت https://nlpprogress.com آمده است تعریف این تسک بدین صورت است:

Relation Prediction is the task of recognizing a named relation between two named semantic entities. The common test setup is to hide one entity from the relation triplet, asking the system to recover it based on the other entity and the relation type.
For example, given the triple <*Roman Jakobson*, *born-in-city*, *?*>, the system is required to replace the question mark with *Moscow*.

برای ارزیابی از متریک MRR استفاده کرده ام. متریک MRR به معنای "میانگین معکوس اولویت" است و یک معیار ارزیابی استفاده می‌شود تا عملکرد یک سیستم جستجو یا مرتب‌سازی نتایج را اندازه‌گیری کند.
در واقع، MRRمعکوس میانگین رتبه نتیجه معتبر اولین نتیجه برای همه درخواست‌ها است. به عبارت دیگر، اگر نتیجه معتبر اولین نتیجه برای درخواست اول، اول باشد، MRRبرابر با 1 خواهد بود؛ اگر نتیجه معتبر اولین نتیجه برای درخواست دوم، دوم باشد، MRRبرابر با 0.5 خواهد بود و به همین ترتیب.
همچنین برای ارزیابی از تست ست دیتاست WN18RR استفاده کرده‌ام که نتایج state of the art بر روی این دیتاست به‌صورت زیر است:

Model	H@10	H@1	MRR	Paper / Source	Code
Max-Margin Markov Graph Models (Pinter & Eisenstein, 2018)	59.02	45.37	49.83	Predicting Semantic Relations using Global Graph Properties	Official

آزمایشات انجام شده:

دو سری آزمایش برای انجام این تمرین انجام دادم که سری اول به دلیل اینکه متریک ارزیابی را در ابتدا به درستی متوجه نشده بودم آزمایش درستی نبود. در ادامه این دو سری آزمایش را شرح می‌دهم.

آزمایش‌های سری اول (اشتباه):

در این سری آزمایشات که شامل ۵ آزمایش است، به دلیل اینکه درک درستی از مسئله و معیار ارزیابی آن نداشتم، نتایج به دست آمده چندان قابل ارزیابی نبوده ولی با توجه به اینکه برای این آزمایشات از prompt های مختلفی استفاده کردم در این گزارش به طور مختصر توضیح می‌دهم.

اشتباه بنده در این بخش این بود که به ازای هر ایتم از دیتاست از ChatGPT API یک جواب می‌خواستم و بدین صورت متریک MRR که پیشتر توضیح داده شده قابل محاسبه نبود و لازم بود که به ازای هر ایتم، ۱۰۰ پیشنهاد از API بگیرم(عدد ۱۰ عددی است که در مقاله‌ایی که به state of the art رسیده است به آن اشاره شده).

اولین prompt ایی که بعد از تست با UI خود ChatGPT و بررسی promptهای مختلف به آن رسیدم، به‌صورت زیر است:

Consider the relation prediction task, in the following lines first the entity and second the relation are separated by space. Give me all the missing entities in a single array in your answer:

که بعد از آن در هر prompt ۵۰ مورد از دیتاست آورده، انتیتی اول و ریلیشن اش آورده می‌شد.

در ادامه با تغییر دادن propmt به صورت زیر سعی در بهبود دادن نتایج داشتم:

```
prompt = "I have relation prediction task, consider these entities:"
for triplet in triplet_batch:
    prompt += f'\n{triplet[0]}'
prompt += "\nAnd also these relation respectively"
for triplet in triplet_batch:
    prompt += f'\n{triplet[1]}'
prompt += "Find just missing entities (NOT THE RELATION OR OTHER FALSY WORDS!)
respectively and give them to me in a single array with this format:\nmissing entities:  ['', '', ...]"
```

این نحوه promptingباعث بهبود در format پاسخ‌های دریافتی شد ولی خود جواب‌ها درست نبود.

تا اینجا موفق به گرفتن جواب درستی از API نشدم درصورتی که در UI بعضا جواب درست با همین prompt ها به دست می‌آمد. چت مربوطه در اینجا قابل مشاهده است.

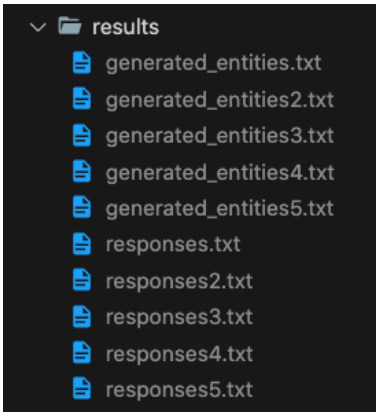
در ادامه با افزودن عبارت زیر به prompt خود توانستم از مجموع ۳۱۳۴ ایتم موجود در دیتاست به ۷ مورد صحیح در پاسخ ها برسم:

Notice: DO NOT GIVE ME THE RELATION OR OTHER FALSY WORDS IN YOUR RESPONSE!

و در ادامه با دادن ۳ مثال از دیتاست به ۱۱ مورد صحیح برسم(few shot).

تا اینجا بهترین نتیجه‌ایی که گرفتم همین ۱۱ مورد بود ولی خب متوجه شدم که اساسا تا اینجا درست پیش نرفتم و نتایج به دست آمده قابل ارزیابی با متریک MRR نیست.

نتایج به دست آمده در نوتبوک مشخص است و همچنین پاسخ‌های دریافتی در فایل های زیر ذخیره شده‌اند:



آزمایش‌های سری دوم:

برای انجام این آزمایش‌ها چت‌های زیر را انجام دادم:

- https://chat.openai.com/share/738e89b7-3977-41f6-a2df-e63b18a5e6fb
- https://chat.openai.com/share/e72556f0-f045-4a67-88cf-c104a4a0e71d
- https://chat.openai.com/share/10ca194f-f8b8-4e5e-9255-cf6bb5d58339
- https://chat.openai.com/share/ac453ac1-5ec5-417d-af3d-ca627ea0d18b

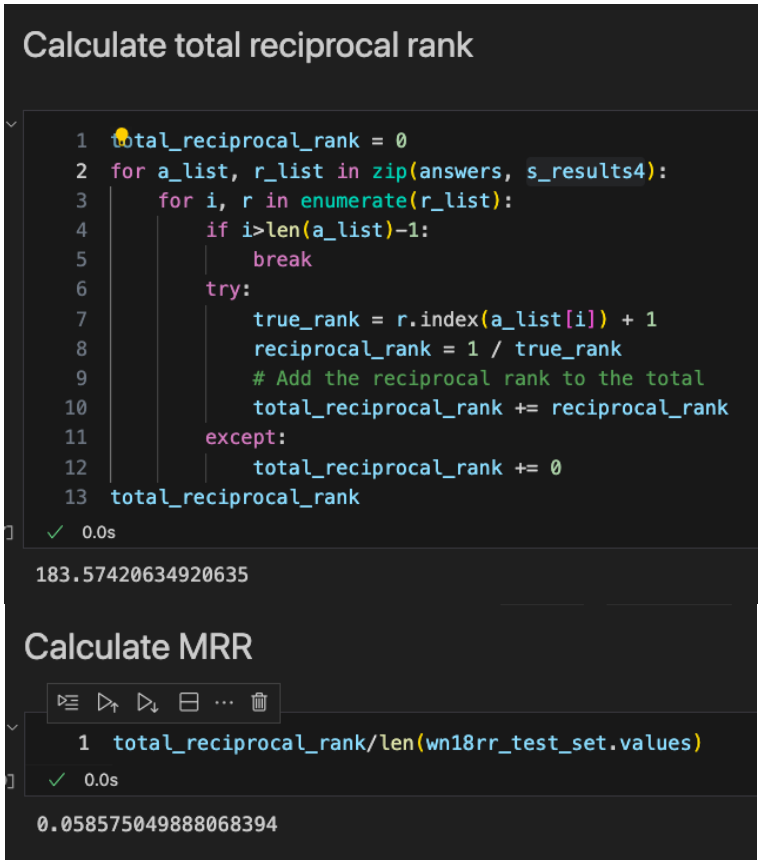
تا اینکه به propmt زیر رسیدم، البته چند آزمایش انجام دادم تا به این نتایج رسیدم که در فایل نوتبوک قابل مشاهده است.

```
    prompt = f"""
Consider the relation prediction task and I want you to behave as a relation prediction model according to the following rules:
1. Data is provided in this format: <target_word> <relation>\\n
2. You should  predict 10 sorted candidate words for each item
3. Your answer should be a nested array that the length of the array is the count of given data, and each array contains 10 sorted candidate words
4.
5. The output MUST be a nested array with this format:
  [[<candidate1>, <candidate2>, <candidate3>, <candidate4>, <candidate5>, <candidate6>, <candidate7>, <candidate8>, <candidate9>, <candidate10>], [<candidate1>, <candidate2>, <candidate3>, <candidate4>, <candidate5>, <candidate6>, <candidate7>, <candidate8>, <candidate9>, <candidate10>], ...]
  Example: [['example1.n.01', 'example2.n.01', 'example3.n.01', 'example4.n.01', 'example5.n.01', 'example6.n.01', 'example7.n.01', 'example8.n.01', 'example9.n.01', 'example10.n.01']]
6. Don't generate code, JUST give me the nested list
7. Give output like this and no more explanation: result : <nested_list>
8. Don't use \\n or in your response
9. Don't give me the example i gave you in rule number 5, give me candidate words!
10. The out put should contains {len(triplet_batch)} arrays each contains 10 candidate, DO NOT make larger arrays and keep array format!
11. An example of <target_word> <relation> <missed_entity> would be {sample[0]} {sample[1]} {sample[1]}. NOTE: it is just an example, don't give me this example as response!

data:
"""
    for triplet in triplet_batch:
        prompt += f'\n{triplet[0]} {triplet[1]}'
```

در آزمایشاتی که در این بخش انجام دادم علاوه بر خود prompt مقدار temprature را نیز تست کردم که در حالتی که صفر ست می‌شد جواب ها stable تر بودند.

بهترین نتیجه به دست‌آمده به صورت زیر است:



و این درحالی است که در مقاله: https://arxiv.org/abs/1808.08644 که state of the art این مسئله است به دقت ۴۹.۸۳ رسیده است.