زهرا مشکوتی 994023040

Question Answering Implementation

امروزه به دلیل وجود حجم زیادی از داده های متنی در حوضه های مختلف و نیاز برای پردازش آنها متدهای پردازش متنی بسیار مورد توجه قرار گرفته اند. به طور کلی به این حوضه nlp یا پردازش زبان های طبیعی گفته می شود.

این مبحث زیرمجموعه های متعددی دارد که در اینجا به مبحث question aswering می پردازیم.

در بسیاری از داده های متنی علمی و غیر علمی ممکن است به دنبال پاسخ سوالی در میان داده ی متنی با حجم بالا باشیم حال این متد به ما کمک می کند که پاسخ سوالمان را از سرتاسز مقاله و داده ی متنی مورد نظرمان با یک انالیز و سرچ معنایی، استخراج کرده و در نهایت یک پاسخ منسجم human_like دریافت کنیم.

در پیاده سازی فایل متنی به صورت یه pdf در نظر گرفته شده.

- extract text from pdf:

ابتدا متن را با استفاده از کتابخانه ی PuMuPDF از فایل pdf استخراج می کنیم.

- preprocess_text:

این function متن را دریافت کرده و عملیات preprocessing را با استفاده از کتبخانه ی re انجام می دهد که شامل حذف کردن header و , footerنشانه های خاص و تبدیل چندینspace به یکی می باشد.

- سپس برای انالیز کردن متن preprocess شده نیاز داریم آن را tokenize کنیم .هر token میتواند طبق نیاز ما کلمه ،جمله و یا ... باشد .در اینجا از یک tokenizer به نام sent_tokenizer که از کتابخانه ی nltk فراخوانی کردیم استفاده می کنیم. متن را به توکن های تقسیم میکنیم که هر توکن یک جمله است.

حال لیستی متشکل از جملات داریم.

- در ادامه برای اینکه بتوانیم عملیت سرچ را در متنمان انجام دهیم نیاز داریم که هر جمله را به مقادیر عددی تبدیل کنیم تا بتوان عملیت مقایسه و یافتن شباهت میان بردارهای عددی را انجام داد.با استفاده از Tfidfvsctorizerکه یک کلاس از scikit_learn می باشد ،لیستی از رشته های موجود را به یک ارایه ی دو بعدی تبدیل کنیم که هر سطر متناظر با یک جمله و هر ستون متاظر با یک term تعریف شده توسط کتابخانه است.به هر term در هر جمله یک مقدار عددی بسته به اهمیت آن کلمه در آن جمله نسبت داده می شود.

در ادامه سوال را نیز به همین شکل به یک ارایه ی عددی تبدیل می کنیم.

- در ادامه دو معیار شباهت یکی cosine-similarity و دیگری sentiment-similarity برای سرچ کردن پاسخ سوال در متن تعریف شده اند.در معیار شباهت کسینوسی میزان شباهت دو vector با محاسبه ی کسینوس زاویه میانشان محاسبه می شود.

و در sentiment-similarity با استفاده از یک مدل pre-trained میزان شباهت جملات را براساس شباهت معنایی میانشان میسنجیم. هر دو معیار ضعف ها و قوت هایی دارند. در cosine similarity خیلی تاکید بر وجود کلمات مشابه در دو جمله است و جملاتی که ممکن است کلمات مشابهی نداشته باشند ولی مربوط به سوال باشند نادیده گرفته می شوند ولی در شباهت معنایی بر اساس context دو جمله میزان شباهتشان سنجیده میشود.که ترکیب این دو باهم میتواند کاربردی باشد.

برای شباهت معنایی از مدل distilbert base uncased finetuned sst 2 english استفاده شده.

- در cosine similarity شباهت میان سوال و هر جمله محاسبه شده و ارایه ای به فرم (n,) که شامل امتیاز هر جمله است خروجی میدهد.با مرتب کردن امتیاز ها اندیس صد جمله ی اولی که بیشترین شباهت را به سوال داشتند ذخیره می کنیم سوس جملات متناظر را در relavant_sentences ذخیره میکنیم.

- سپس در تابع search by sentiment با پارامتر ها سوال ، relavant sentencesکه در مرحله ی قبل استخراج کردیم و مدل مشخص شده برای سرچ معنایی جملات مشابه از لحاظ context به سوال را از میان جملات مشابهی که در مرحله قبل یافتیم استخراج می کنیم.

- در اخرین مرحله میخواهیم با جملات relavant یک پاسخ مناسب با گفتار مانند انسان تولید کنیم که با استفاده ار مدل train شده ی distilgpt2 این متن را تولید میکنیم که حداکثر طول آن همان طول متن تولید شده قرار داده شده.

باید توجه کنیم که هر دو مدل از قبل اموزش داده شده یک محدودیت دریافت token دارند و برای متن های بزرگ تر نیاز است که متن به چند قسمت تقسیم شده و relavant sentences در هر پارت یافته شده و در نهایت پاسخ با استفاده از همه ی آنها تولید شود.

در فایل پایتون مشاده می شود فایل متنی مربوط به clustering و kmeansبه پیاده سازی همراه با سوال داده شده که هر دو با رعایت limitation tokens قابل تغییر هستند و در نهایت پاسخی برای سوال what is?" " kmeans algorithmتولید می شود.