چت‌بات دانا

شرکت عصر گویش پرداز

گزارش کارآموزی

نگارش
سهند نوعی کردکندی

سرپرست کارآموزی
دکتر سروش گوران

شهریور ۱۴۰۲

چکیده

در دو دهه آخر، هوش مصنوعی رشد بسیاری کرده است، به‌طوری که کاربردهای مختلف زیرشاخه‌های آن در صنعت، آموزش و کاربردهای دیگر مشاهده می‌شود. پردازش زبان طبیعی به عنوان یکی از زیرشاخه‌های هوش مصنوعی، با ظهور مدل‌های زبانی بزرگ مانند GPT-3.x, BERT, T5, LLaMA-2 و ... پیشرفت زیادی داشته است. هم اکنون کاربرد این مدل‌ها در زمینه یک مدل چت‌بات، Text-to-Text ، آنالیز احساساتی متون، تشخیص اسامی خاص موجودیت‌ها، ترجمه زبان‌ها، خلاصه‌سازی متون و ... را می‌توان مشاهده کرد.

هم اکنون نیز، در کشور ما، از یادگیری ماشین در کاربردها و محصولات مختلف استفاده می‌شود. در این گزارش قصد دارم تجربیاتم را به عنوان کارآموز پردازش زبان طبیعی در شرکت عصر گویش پرداز بیان کنم و پروژه و مسائلی که بنده روی آنها کار می‌کردم را تشریح کنم.

واژه‌های کلیدی:

هوش مصنوعی، یادگیری ماشین، پردازش زبان طبیعی، مدل‌های زبانی بزرگ

فهرست مطالب

صفحه

فصل اول

مقدمه

مقدمه

هوش مصنوعی در دهه‌های اخیر به صورت چشمگیری رشد کرده است، به‌طوری که امروزه در صنعت در کاربردهای بسیاری استفاده می‌شود. از کاربردهای نظامی، پزشکی، اقتصادی و مالی تا کاربردهای در زمینه پردازش زبان طبیعی و بینایی ماشین. و البته حائز اهمیت است اشاره شود که همه این پیشرفت‌ها به لطف پیشرفت‌های صورت‌گرفته در ساخت واحدهای پردازشی و محاسباتی مخصوص مانند واحد پردازش گرافیکی[1] واحد پردازش تنسور[2]و ... و همچنین وجود داده‌های دیجیتال بسیار می‌باشد که توان محاسباتی کامپیوترهای امروزی چند برابر شده و یادگیری الگوهای پیچیده توسط مدل‌ها امکان پذیر شده است.

پردازش زبان طبیعی، به عنوان یکی از شاخه‌های هوش مصنوعی، با تمرکز بر تعامل بین انسان و کامپیوتر، تلاش بر درک زبان طبیعی انسان دارد. در واقع هدف آن درک زبان طبیعی، تفسیر و بازتولید زبان انسان می‌باشد. پردازش زبان طبیعی، کاربردهای بسیاری در زندگی روزمره ما دارد. از جمله، دستیارهای صوتی و چت‌بات‌ها، آنالیز احساساتی متون (درک احساسات با توجه به متن‌نوشته‌ها)، ترجمه زبان‌ها، بازیابی اطلاعات و موتورهای جستجو، و بسیاری کاربردهای دیگر. همه این موارد باعث شده است که پردازش زبان طبیعی یکی از شاخه‌های مهم هوش مصنوعی باشد.

با توسعه یافتن پردازش زبان طبیعی، امروزه در کشور ما، سعی شده است که در محصولات صنعتی، آموزشی و ... استفاده شود. یکی از بارزترین کاربردهای آن در ایران، استفاده از آن برای تحلیل نظرات مشتریان شرکت‌های بزرگ و تلاش برای ایجاد مکانیزم‌هایی برای فروش بیشتر محصولات، با توجه به بازخوردهای مدل می‌باشد.

من در این گزارش سعی کرده ام که تجربیات و فعالیت‌های خود در دوره کارآموزی خود در شرکت عصر گویش پرداز را شرح دهم و چالش‌ها و راهکارهای استفاده شده در پروژه خود را تشریح کنم.

[1] Graphics Processing Unit
[2] Tensor Processing Unit

فصل دوم
فعالیت‌ها و تجربیات کارآموزی

پروژه چت‌بات دانا

۳-۱- مقدمه

چت بات‌ها به عنوان یکی از مهم‌ترین تکنولوژی‌های هوش مصنوعی در دنیای امروز شناخته می‌شوند. امروزه استفاده از ربات‌های چت بات بسیار مورد استقبال قرار گرفته است. این سیستم‌ها هوش مصنوعی توانایی برقراری ارتباط و ارائه خدمات به کاربران از طریق پیام‌رسان‌ها و وبسایت‌ها را دارا هستند. یکی از ویژگی‌های بارز چت بات‌ها، توانایی پردازش و درک زبان طبیعی است. این به معنای آن است که آنها می‌توانند پیام‌ها و سوالات کاربران را درک کرده و به صورت منطقی و جواب‌های منطبق با آن ارائه دهند. این توانایی‌ها باعث شده است که چت بات‌ها به عنوان یک ابزار قدرتمند برای افزایش کارایی و اثربخشی در تعاملات انسانی و مشتری‌مداری در شرکت‌ها و سازمان‌ها شناخته شوند.

علاوه بر این، چت بات‌ها می‌توانند به صورت پیشرو در تجربه کاربری[1]بهبودهای مهمی ایجاد کنند. آنها به کاربران این امکان را می‌دهند که به سرعت و به راحتی اطلاعات مورد نیاز خود را دریافت کنند، سوالات خود را مطرح کنند، و حتی در برخی موارد، تراکنش‌های مالی را انجام دهند.

با توجه به توانمندی‌های چت بات‌ها و پتانسیل آنها در بهبود ارتباطات و ارائه خدمات به کاربران، این فناوری به سرعت در حال گسترش و توسعه است. از این رو، به طور کلی می‌توان گفت که چت بات‌ها نقش مهمی در آینده تکنولوژی و تعاملات انسانی خواهند داشت.

۳-۲- معرفی پروژه چت بات دانا

در این پروژه سعی کردیم که مسائل زیر را برای کاربر حل کنیم:

- چگونه می‌توان یک چت بات همیشه در دسترس داشت؟
- با توجه به دقت بالای مدل‌های GPT-3.5 و GPT-4 نسبت به سایر مدل‌های زبانی موجود، همچنین با توجه به محدودیت تعداد توکن‌های قابل استفاده در این مدل‌ها، نمی‌توان متن‌های طولانی برای پردازش فرستاد. چگونه می‌توان این مسئله را رفع کرد؟
- چگونه می‌توان با فرستادن متون طولانی، از چت بات خواست که به صورت خلاصه در چند خط آن را توضیح دهد.
- چگونه می‌توان با فرستادن مقاله، کتاب یا متون طولانی تا حدود ۱۵۰ صفحه، خلاصه‌ای از آن در حد چند خط یا خلاصه‌ای جامع که دربرگیرنده تمامی نکات مهم آن باشد، در چندین صفحه داشت؟
- چگونه می‌توان به تمامی مطالب اشاره شده در رابطه با یک موضوع خاص در یک مقاله، کتاب یا یک متن دسترسی داشت؟
- چگونه می‌توان در رابطه با محتوای یک مقاله، کتاب یا یک متن سوال پرسید و پاسخی دقیق با توجه به فقط محتوای آن مقاله دریافت کرد؟

[1] UX (User Experience)

- آیا می‌توان به‌جای استفاده از پیام متنی، پیام خود را به صورت صوتی به مدل GPT بدهیم؟
- آیا می‌توان پاسخ توسط مدل GPT را به صورت انواع مختلف فایل، از قبیل پیام صوتی، فایل TXT یا PDF دریافت کرد؟
- آیا می‌توان یک پیام متنی به GPT داد و از آن خواست که عکس را با توجه به آن پیام تولید کند و به عنوان خروجی به کاربر بدهد؟
- متاسفانه GPT فقط به اطلاعاتی که تا سپتامبر ۲۰۲۱ در وب وجود داشته است، دسترسی دارد. چگونه می‌توان اطلاعات به روز را از چت بات دریافت کرد؟

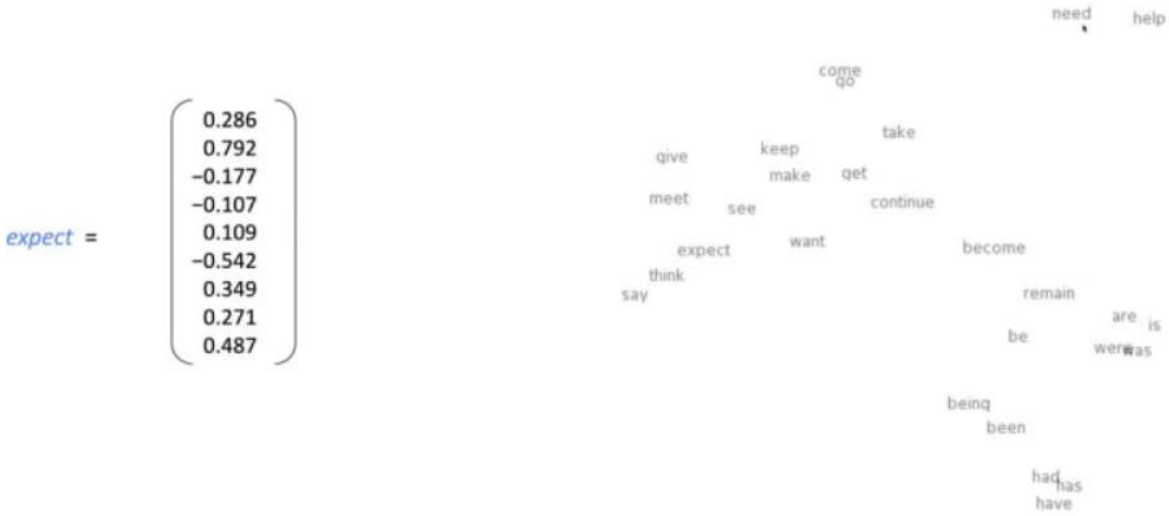
۳-۳- توضیحات اولیه

معماری Transformer یک معماری یادگیری عمیق برای پردازش زبان طبیعی و ترجمه ماشینی[1]است که توسط Google در مقاله‌ای به نام"Attention Is All You Need" معرفی شد و به سرعت به یکی از معماری‌های مهم و اصلی در حوزه هوش مصنوعی و پردازش زبان طبیعی تبدیل شد. این معماری در سال ۲۰۱۷ توسط Vaswani و همکارانش توسعه داده شد و به واسطه قابلیت‌های منحصر به فرد خود، به عنوان پایه‌ای برای ساخت مدل‌های زبانی مانند BERT، GPT، و T5 شناخته می‌شود. Transformer از یک مفهوم اصلی به نام "توجه" استفاده می‌کند که به مدل امکان می‌دهد روابط میان کلمات در یک جمله را دریافت کرده و در ترجمه ماشینی یا تولید متن در محتوای متن تاثیر بدهد. در واقع، Transformer از مکانیزم توجه برای مدیریت وزن‌دهی به ورودی‌های مختلف مدل در هر مرحله استفاده می‌کند.

مدل‌های زبانی بزرگ[3] به یک دسته از مدل‌های هوش مصنوعی اشاره دارند که برای درک و تولید زبان انسانی طراحی شده‌اند. این مدل‌ها قادرند متن به زبان انسانی را پردازش کرده و تولید کنند و برای انجام وظایفی مانند ترجمه زبان، خلاصه‌نویسی متن، پاسخ به سوالات[4]و موارد دیگر آموزش داده‌شده‌اند. یکی از مثال‌های معروف از مدل‌های زبانی بزرگ GPT-3[5] است که توسط OpenAI توسعه داده شده است. GPT-3 و مدل‌های مشابه به نتایج قابل توجهی در وظایف مختلف پردازش زبان طبیعی دست یافته‌اند و به دلیل توانایی آنها در تولید متن منطقی و مرتبط، توجه زیادی را به خود جلب کرده‌اند. LLMها معمولاً بر پایه معماری Transformer توسعه می‌یابند که به آنها این امکان را می‌دهد وابستگی‌های دور و نزدیک در متن را دریافت کرده و بسیاری از وظایف مربوط به زبان طبیعی را مدیریت کنند. این مدل‌ها پتانسیل استفاده در برنامه‌هایی مانند چت‌بات، تولید محتوا، ترجمه زبان و حتی به عنوان دستیار در ویرایش متون را دارند.

[1] Machine Translation
[2] Attention
[3] Large Language Model (LLM)
[4] Question Answering (QA)
[5] Generative Pre-Trained Transformer

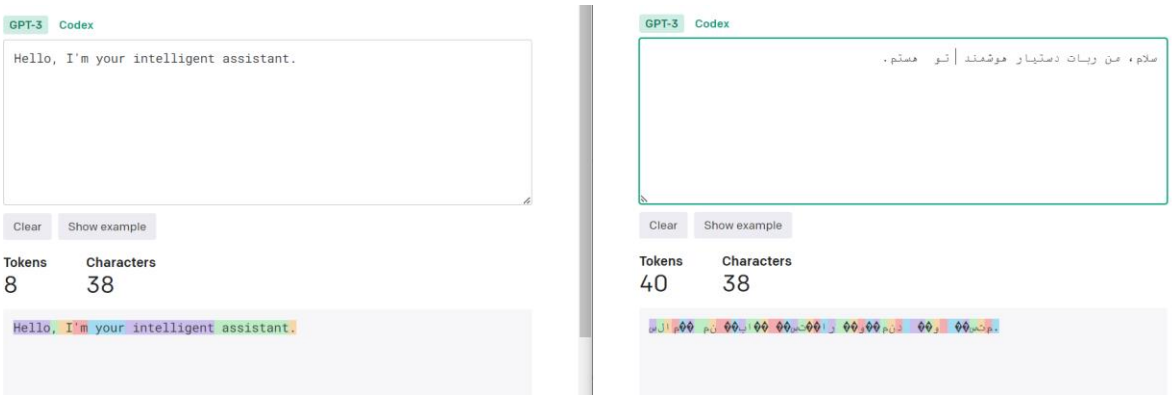
مدل‌های زبانی بزرگ، از روشی به نام token بندی کاراکترها برای پردازش کلمات، حروف، علائم و غیره استفاده می‌کنند. با این روش تعداد بسیار زیادی از کاراکترهای موجود در زبان‌های مختلف که بیش از ۹۹درصد آنها را تشکیل می‌دهند، بردار دگرنمایی[1]منحصر به خود را دارند که هر کدام اقلا ۳۰۰ بعد دارند و اگر این بردار را در فضای دو بعدی نشان دهیم(هرچند که این کار باعث می‌شود بسیاری از ویژگی‌هایی که بردار می‌تواند داشته باشد، از دست برود)، خواهیم دید که کلمات یا کاراکترهایی که بیشترین ارتباط را با هم دارند، در مکان‌های نزدیکتر به یکدیگر قرار دارند که این امر با استفاده از احتمال به دست‌آمده برای هر کدام به آنها نسبت داده می‌شود. با این روش هنگام ساخت جملات، ترجمه یا هر فرایند دیگری که با کلمات انجام شود، نتایج بسیار دقیق‌تری به دست می‌آید. به طور مثال در شکل زیر قسمتی از بردار دگرنمایی کلمه "expect" نمایش داده شده است که ملاحظه می‌کنیم کلمات "see"، "meet"، "think" و "say" به "expect" نزدیک‌ترند که یعنی عدد احتمال آنها به یک نزدیک‌تر است و تشابه بیشتری با یکدیگر دارند یا مثلا کلمات "is"، "are"، "was" و "were" در مختصات دو بعدی بسیار به یکدیگر نزدیک هستند و این به خاطر شباهت آنهاست.



[۱]

یکی از مشکلاتی که در زبان‌های غیر انگلیس در این روش وجود دارد، توکن بندی با استفاده از کاراکتر به جای کلمات است که این باعث می‌شود بارمحاسباتی و تعداد توکن‌های مورد استفاده در زبان‌های دیگر بسیار افزایش پیدا کند.
مثلا در صورتی که به آدرس [۲] مراجعه کنید و یک متن با تعداد کاراکترهای مشخص به زبان انگلیسی و فارسی در آن وارد کنید، خواهید دید که در تعداد کاراکتر مساوی، تعداد توکن‌های متن به زبان فارسی دو تا شش برابر متن به زبان انگلیسی است که در شکل زیر آورده شده است. روشی برای اصلاح این روش وجود دارد که در مقاله [۳] آورده شده است و برای پروژه‌های بعدی با همکاری شرکت عصر گویش پرداز به آن خواهیم پرداخت.

[1] Vector Embedding



[۲]

از مهم‌ترین frameworkها و کتابخانه‌های مورد استفاده در این پروژه می‌توان به numpy، scikit-learn، pytorch، langchain، openai و pytelegrambotapi اشاره کرد.

langchain یک framework است که ساخت برنامه‌ها با استفاده از مدل‌های زبانی بزرگ را ساده‌تر می‌کند. کاربردهای langchain به طور عمده در کلیه حوزه‌های پردازش زبان طبیعی از جمله تحلیل و خلاصه‌سازی فایل‌ها، چت‌بات‌ها و تحلیل کد است. این framework با استفاده از هر دو زبان javascript و python قابل استفاده است. در اینجا ما زبان python را به دلیل قابلیت‌های بیشتر در توسعه مدل‌های هوش مصنوعی انتخاب کرده‌ایم. برای نصب langchain در محیط برنامه‌نویسی خود، باید علاوه بر دستور عمومی pip install langchain، از دستور pip install langchain[llms] هم استفاده کنیم تا برنامه‌های لازم برای استفاده از llmها در سیستم شما نصب شود.

openai یک کتابخانه است که به زبان python توسط شرکت OpenAI توسعه داده شده است و می‌توان با آن به API شرکت OpenAI دسترسی پیدا کرد.

با توجه به همه گیری استفاده از شبکه اجتماعی تلگرام[1] توسط مردم عزیزمان، برخورداری از API نسبتا غنی و با قابلیت‌های متعدد، تصمیم بر آن شد که برای ارتباط کاربر با پروژه از یک ربات تلگرامی استفاده کنیم. تلگرام دارای دو API به نام python-telegram-bot و pytelegrambotapi در python است که ما از pytelegrambotapi برای ارتباط با ربات تلگرامی استفاده می‌کنیم.

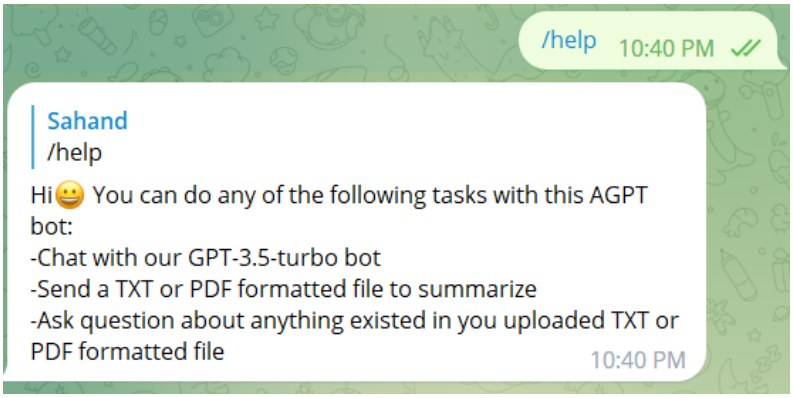
از کتابخانه‌های numpy، scikit-learn و pytorch هم برای کار با مدل‌های هوش مصنوعی استفاده می‌کنیم.

[1] Telegram

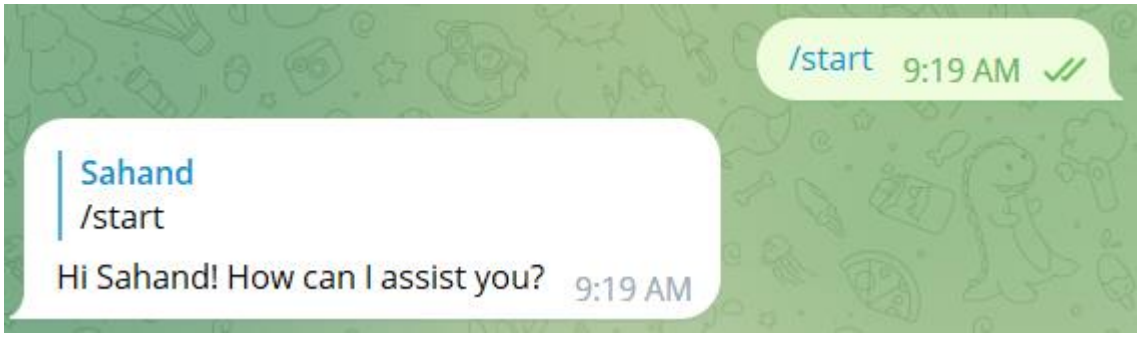
علاوه بر این کتابخانه‌ها، بسیاری پکیج‌های دیگر هم باید نصب شوند که به دلیل اهمیت کمتر آنها در اینجا به آنها نمی‌پردازیم. در صورت تمایل می‌توانید به صفحه گیتهاب پروژه [۴] مراجعه کنید و در فایل requirements.txt این پکیج‌ها را هم مشاهده کنید.

۳-۴- مراحل توسعه برنامه و چالش‌های موجود

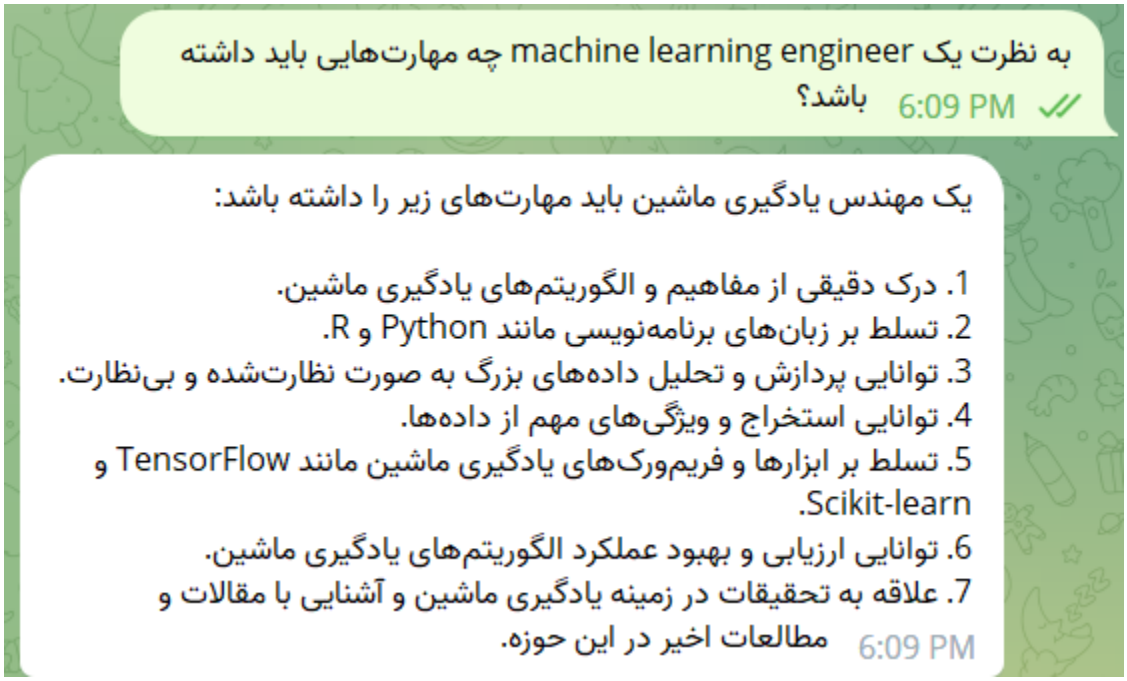
- ابتدا تمام پیش‌نیازهای ذکر شده در فایل requirements.txt را در محیط برنامه نصب می‌کنیم.
- یک حساب کاربری جدید در openai ایجاد می‌کنیم تا اعتبار لازم برای استفاده از API آن را داشته باشیم. سپس از همین سایت توکن حساب کاربری خود را دریافت کرده و در openai.api_key لحاظ می‌کنیم.
- با استفاده از ربات BotFather در تلگرام ربات جدیدی برای پروژه ایجاد می‌کنیم و از توکن آن برای استفاده در پروژه استفاده می‌کنیم.
- با استفاده از تابع bot.message_handler() و آرگومان‌های commands و content_type می‌توان دستورات و نوع ورودی که کاربر به ربات داده است را کنترل کرد.
- در ربات شش دستور start/، help/، summarize/، comprehensive_summarize_english/، comprehensive_summarize_persian/ و question_answer/ وجود دارد که در ادامه پیاده سازی هر یک را توضیح می‌دهیم.
- دستور help/ برای شروع ربات است و توضیحاتی در رابطه با عملکرد ربات و قابلیت‌های آن به کاربر می‌دهد.



- دستور start/ برای شروع کار با ربات است که کاربر فقط برای اولین بار استفاده از ربات لازم است روی start کلیک کند تا بتواند با ربات تعامل کند.



- برای ایجاد یک چت بات تعاملی با استفاده از openai API، مدلی که چت‌بات تحت وب استفاده می‌کند gpt-3.5 هست که دارای محدودیت ۴۰۹۶ توکن است. ما به جای این مدل، از مدل gpt-3.5-turbo-16k استفاده می‌کنیم که محدودیت توکن آن ۱۶۳۸۴ عدد است. بنابراین حافظه چت‌بات در این حالت ۴ برابر می‌شود و کاربر می‌تواند متون بسیار طولانی‌تری را به صورت پیام متنی به چت‌بات بدهد و درخواست‌های خود را در آن مشخص کند.

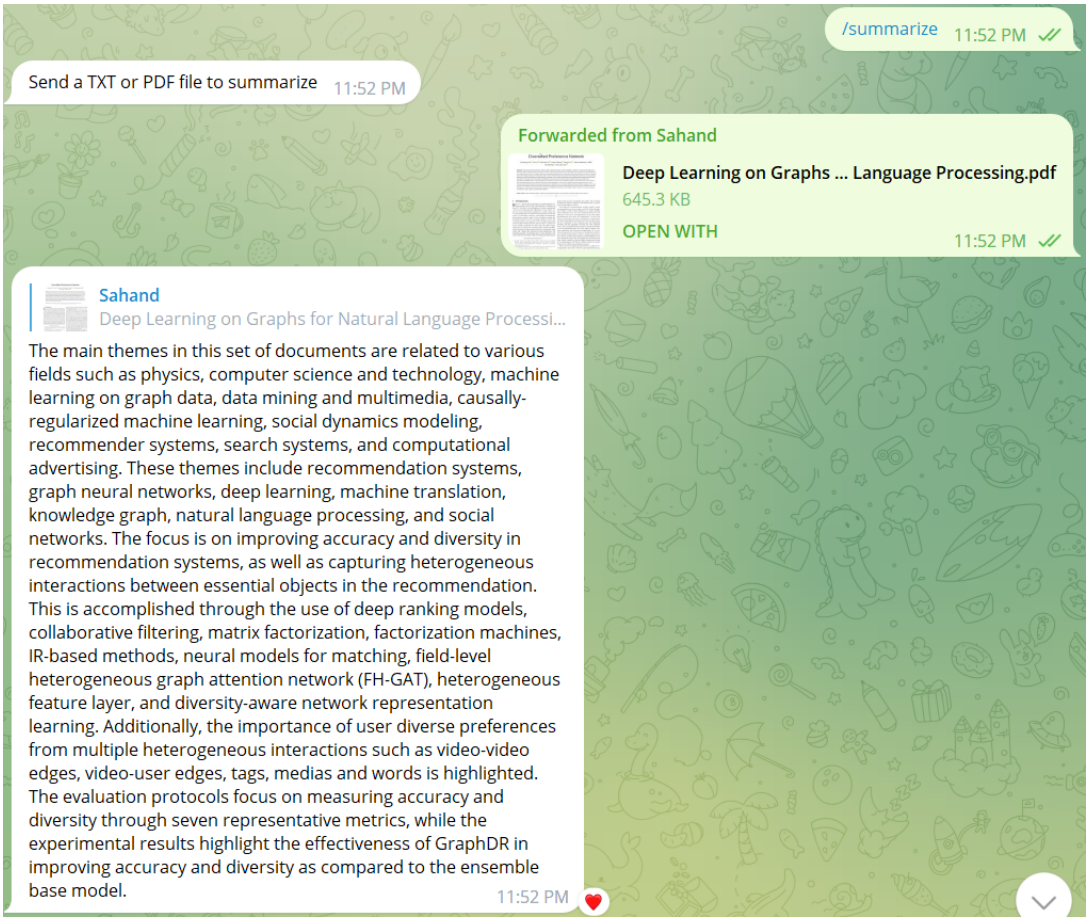


- یکی از قابلیت‌های این ربات، امکان استفاده از آن در گروه‌های تلگرامی است. در این حالت همانند شکل زیر، کافیست کاربر ربات را tag کند و درخواست خود را در ادامه بنویسد. برای این کار لازم است ربات را به گروه اضافه کرده و دسترسی مدیر به آن داده شود.

- یکی از قابلیت‌هایی که به چت‌بات اضافه کردیم، امکان فرستادن درخواست به صورت صوتی است که کاربر می‌تواند درخواست خود را در قالب صوت که به صورت پیش فرض با پسوند ogg. هست به بات بدهد تا پاسخ

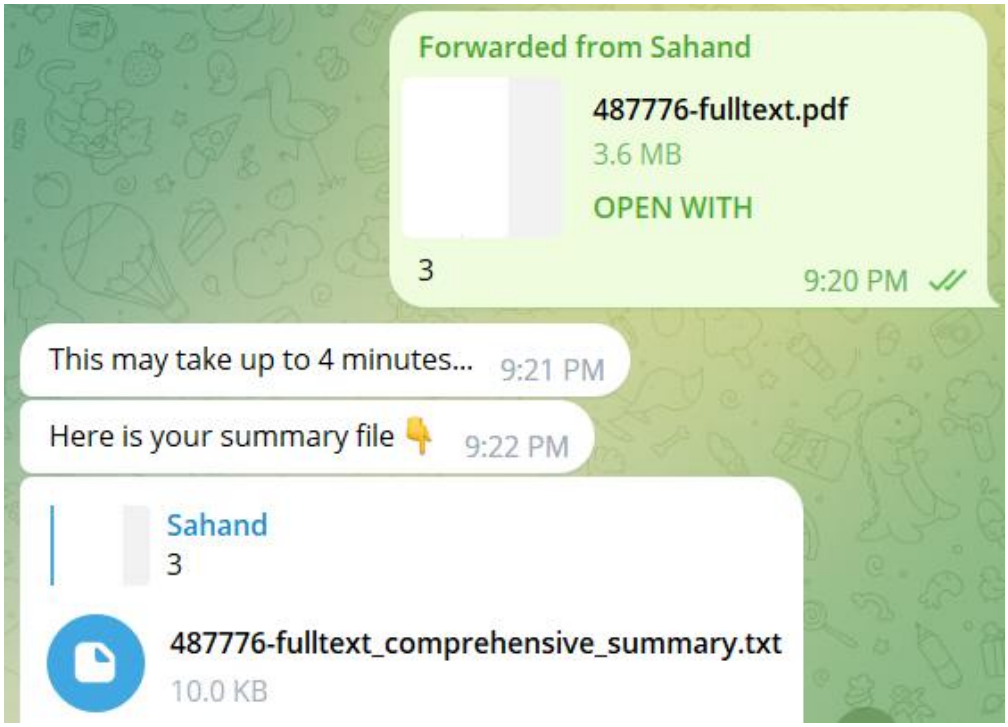
آن را دریافت کند. همچنین فایل صوتی می‌تواند پسوندهای دیگری هم از قبیل .mp3، .wav، .m4a و .mpeg3 باشد. برای پیاده‌سازی این مورد از مدل Whisper استفاده می‌کنیم که یک مدل تشخیص گفتار خودکار[1] است و با حدود ۶۸۰هزار ساعت داده آموزش داده شده است که این مورد هم از طریق openai API در دسترس است.

- با استفاده از دستور /summarize، کاربر می‌تواند فایل متنی تا ۵ صفحه را به ربات بدهد و یک خلاصه کوتاه نیم صفحه‌ای از محتوای آن داشته باشد. فایل ارسالی کاربر می‌تواند TXT یا PDF باشد. در این بخش برای خلاصه‌سازی مطالب می‌توانیم از چهار روش ارائه شده توسط langchain استفاده کنیم. یکی از این روش‌ها stuff نام دارد که در این روش، با توجه به محدودیت ۱۶هزارتایی تعداد توکن‌های مدل gpt-3.5-turbo-16k و با توجه به این که فایل متنی نهایتا ۵ صفحه‌ای به احتمال بسیار بالا در هر زبانی باشد کمتر از ۱۵هزار توکن دارد و خلاصه تولید شده توسط مدل هم کمتر از ۱هزار توکن خواهد داشت در نتیجه مدل می‌تواند با نهایت دقت تمام فایل متنی را پردازش کرده و خلاصه کوتاهی از آن را ارائه دهد. این دستور، خلاصه فایل را می‌تواند فقط به زبان انگلیسی تولید کند.



[1] ASR (Automatic Speech Recognition)

- با استفاده از دو دستور /comprehensive_summarize_english و /comprehensive_summarize_persian همانند شکل زیر کاربر می‌تواند به صورت میانگین یک فایل متنی انگلیسی تا سقف ۱۵۰ صفحه یا یک فایل متنی غیر انگلیسی (از جمله فارسی) تا سقف ۵۰ صفحه را به ربات بدهد و در عنوان[1] آن تعداد صفحات مورد نیاز برای خلاصه‌سازی فایل را مشخص کند. دستور /comprehensive_summarize_english خلاصه را به زبان انگلیسی برمی‌گرداند و دستور /comprehensive_summarize_persian خلاصه را به زبان فارسی به کاربر ارائه می‌دهد. در هر دو دستور فایلی که کاربر به ربات می‌دهد می‌تواند PDF یا TXT باشد و خلاصه تولید شده توسط مدل، در قالب فایل TXT به کاربر ارسال می‌شود. در شکل زیر یک مقاله ۷۰ صفحه‌ای به زبان فارسی به ربات داده شده است و از آن خواسته شده است که فایل را در ۳ صفحه خلاصه کند.



فایلی که کاربر به ربات می‌دهد می‌تواند تا ۱۵۰هزار توکن داشته باشد و چالشی که پیش می‌آید این است که چگونه با مدلی که محدودیت توکن ۱۶هزارتایی دارد، می‌توان این فایل را پردازش و خلاصه کرد. علاوه بر این استفاده از مدل gpt-3.5-turbo-16k با استفاده از کتابخانه OpenAI هزینه‌ای مطابق با شکل زیر دارد که به صورت رسمی توسط همین شرکت ارائه شده است. با توجه به این شکل، فایلی که ۱۵۰هزار توکن دارد هزینه ای حدود ۰٫۵ دلار برای پردازش دارد که هزینه قابل توجهی است.

[1] Caption

GPT-3.5 Turbo

GPT-3.5 Turbo models are capable and cost-effective.

`gpt-3.5-turbo` is the flagship model of this family and is optimized for dialog.

`gpt-3.5-turbo-instruct` is an Instruct model and only supports a 4K context window.

Learn about GPT-3.5 Turbo ↗

Model	Input	Output
4K context	$0.0015 / 1K tokens	$0.002 / 1K tokens
16K context	$0.003 / 1K tokens	$0.004 / 1K tokens

برای رفع این چالش‌ها ما فایل متنی ارسالی را ابتدا با استفاده از تابع text_splitter به فایل‌های با تعداد توکن‌های ۱۰هزارتایی با حداکثر ۳هزار توکن همپوشانی[1] تقسیم می‌کنیم. سپس با استفاده از تابع embed_documents محتوای این فایل‌های متنی را در بردارهای ۱۵۳۶ بعدی embed می‌کنیم. با استفاده از روش k-means که یکی از روش‌های خوشه‌بندی[2]در یادگیری ماشین بدون نظارت[3]است، این فایل‌ها را با توجه به بردار embedding آنها به خوشه‌های متعددی تقسیم می‌کنیم. سپس ابعاد این خوشه‌ها را با استفاده از روش t-SNE[4] به ۲ کاهش می‌دهیم تا بتوانیم آنها را در صفحه دو بعدی تجسم[5]کنیم. با این کار می‌توانیم فاصله هر یک از خوشه‌ها تا نقطه مرکزی[6]را حساب کنیم و بردارهای embedding با کمترین فاصله از نقطه مرکزی را بدست بیاوریم و آنها را به عنوان بردارهای embedding برگزیده برای خلاصه‌سازی متن مورد نظر انتخاب کنیم. با این کار فقط بخشی از متون فایل اولیه را به مدل gpt-3.5-turbo-16k می‌دهیم که بهترین نتایج را بدست می‌دهند. حال می‌توان prompt مورد نظر خود را برای مدل ایجاد کرد و خلاصه‌سازی را انجام داد. در شکل زیر متن prompt داده شده به مدل در پروژه آورده شده است که با توجه به دستوری که کاربر می‌دهد، می‌تواند به زبان فارسی یا انگلیسی خلاصه‌سازی را انجام دهد.

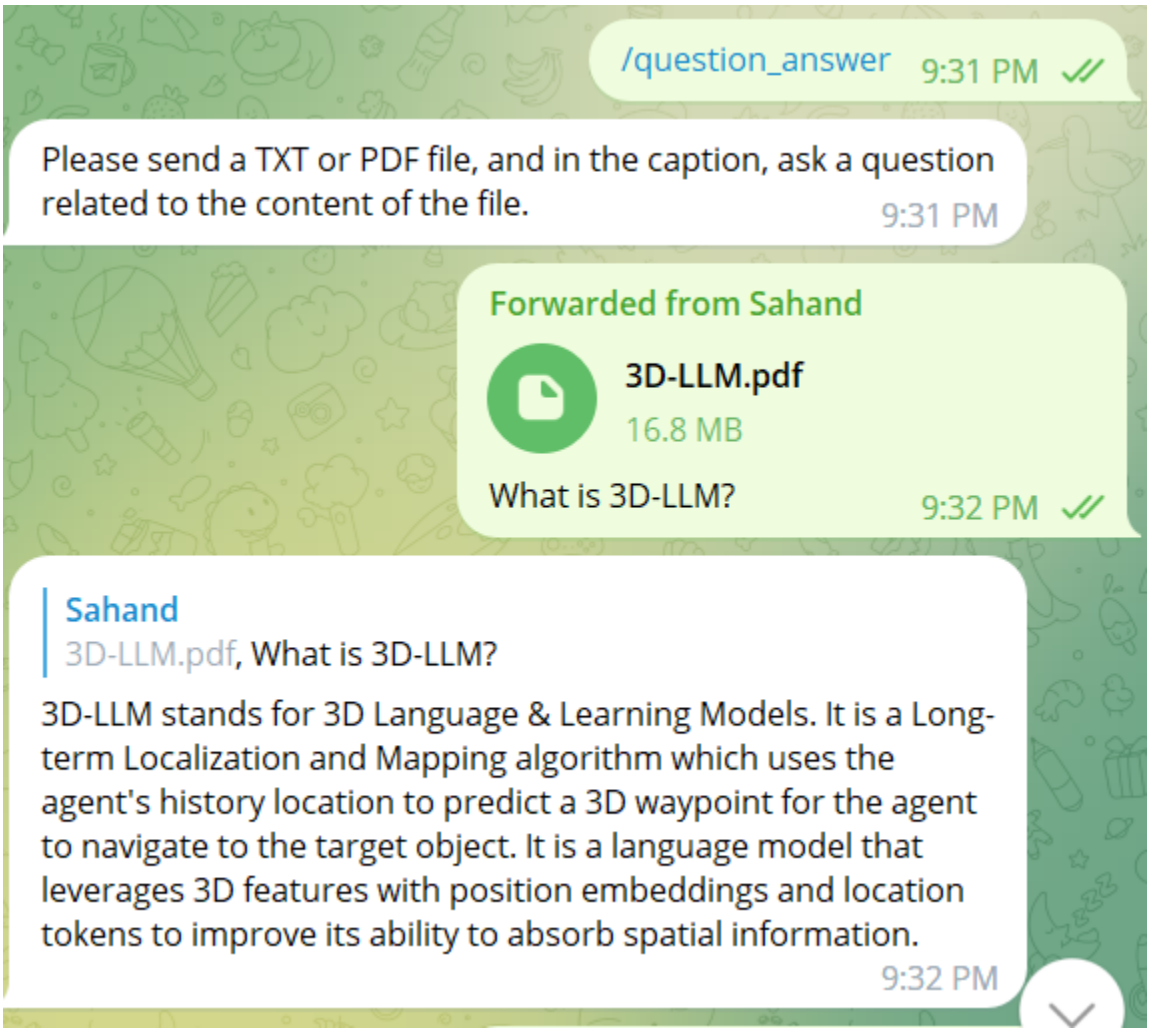
```
language = ""
if state == 4:
    language = " in persian language"
map_prompt = """
    You will be given a single passage of a book. This section will be enclosed in triple backticks (```)
    Your goal is to give a summary of this section""" + language + """ so that a reader will have a full
    understanding of what happened.
    Your response should be at least three paragraphs and fully encompass what was said in the passage.

    ```{text}```
    FULL SUMMARY:
    """
```

تعداد صفحات خلاصه‌سازی هم به این شکل مدیریت می‌شود که هر خوشه می‌تواند حدود ۲۵۰کلمه را به صورت خلاصه تولید کند و با استفاده از ورودی گرفته شده از کاربر و در نظر گرفتن این مورد که هر صفحه حدود ۵۰۰کلمه دارد، تعداد خوشه‌ها در ابتدا تعیین می‌شود.

[1] Overlap
[2] Clustering
[3] Unsupervised machine learning
[4] t-distributed Stochastic Neighbor Embedding
[5] Visualize
[6] Centroid
```

- همانطور که می‌دانید در صورتی که از chatgpt در رابطه با مطلبی سوالی پرسیده شود، پاسخ آن با توجه به تمام داده‌هایی که با آن آموزش دیده است، داده می‌شود. اگر کاربر بخواهد می‌تواند با استفاده از دستور /question_answer یک فایل با فرمت PDF یا TXT با هر تعداد صفحه دلخواه را به ربات بدهد و در رابطه با محتوای آن فایل در عنوان آن، از ربات پرسش کند. در این حالت ربات فقط با توجه به محتوای آن فایل متنی پاسخ کاربر را می‌دهد و در صورتی که مطلبی در رابطه با آن سوال در فایل متنی وجود نداشته باشد، پاسخ می‌دهد که اطلاعاتی در رابطه با این سوال در فایل ذکر نشده است.



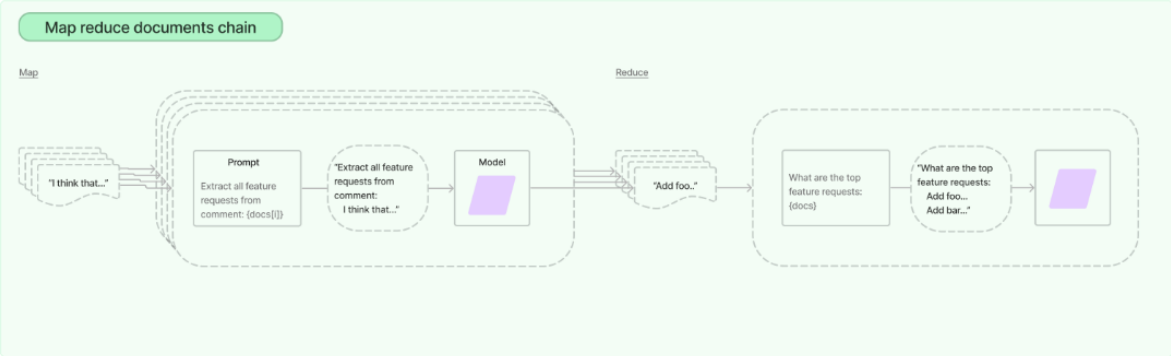
پرسش و پاسخ[1] یکی از زمینه‌های مهم در زمینه پردازش زبان طبیعی است که در این پروژه از مدل mpnet که در وبسایت رسمی huggingface در دسترس است و برای پرسش و پاسخ تنظیم-دقیق[2] شده است، استفاده می‌شود. برای استفاده از این قابلیت از کلاسی در کتابخانه langchain به نام SVMRetriever استفاده می‌کنیم که برای بازیابی اسناد مربوطه با توجه به پرسش[3] داده شده استفاده می‌شود. در این روش، مجموعه‌ای

[1] Question answering (QA)
[2] Fine-tune
[3] Query

از روش‌های یادگیری تحت نظارت[1] تحت عنوان SVM[2] که برای طبقه‌بندی[3] و رگرسیون[4] استفاده می‌شود، به کار می‌رود.

از این مدل برای جمع آوری محتوای موجود در فایل ورودی کاربر استفاده می‌شود و برای جواب نهایی از مدل gpt-3.5-turbo-4k استفاده می‌شود. برای ارائه جواب نهایی از روشی به نام map_reduce استفاده می‌شود که در این روش خروجی‌های بدست آمده از مدل mpnet به عنوان docهای جداگانه به مدل داده می‌شود و این docها به صورت جداگانه پردازش می‌شوند و جواب هر یک از آنها در مرحله با یگدیگر ادغام می‌شود.

در شکل زیر این فرایند را می‌توانید مشاهده کنید.



[۵]

promptهای مورد استفاده برای مدل mpnet و gpt3.5 به ترتیب با نام‌های question_prompt_template و combine_prompt_template در شکل زیر آورده شده است.

```
question_prompt_template = """Answer the question delimited by triple backquotes as precise as possible using the
provided context delimited by triple backquotes. If the answer is not contained in the context, say "answer not
available in context".
 \n Context: ``` {context}? ```
 \n Question: ``` {question} ```
 \n Answer:   """
combine_prompt_template = """Given the extracted content and the question delimited by triple backquotes , create a
final answer. If the answer is not contained in the context, say "answer not available". \n\n
Summaries: {summaries}?
Question: ``` {question} ```
Answer:
"""
```

[1] Supervised Learning
[2] Supported Vector Machines
[3] Classification
[4] Regression

۳-۵- فعالیت‌های آینده

- یکی از چالش‌های موجود در استفاده از مدل‌های زبانی بزرگ در زبان فارسی، مشکل توکن‌بندی هست که توضیح داده شد. در آینده قصد داریم که با استفاده از روش گفته شده در مقاله [۳]، یک tokenizer مخصوص زبان فارسی ایجاد کنیم و آن را به دایره لغات مدل‌های منبع‌باز موجود - از جمله مدل LLaMA-2 که توسط گروه Meta عرضه شده است - اضافه کنیم، تا علاوه بر افزایش سرعت پردازش متون فارسی، تعداد توکن‌های آن را تا یک ششم کاهش دهیم. از جمله مزایای دیگر این کار، درک عمیق‌تر مدل نسبت به کلمات فارسی است که می‌تواند نتایج بهتری را ارائه دهد.
- همانطور که در شکل زیر مشاهده می‌کنید، درصد بالایی از داده‌های آموزشی مدل‌های زبانی بزرگ منبع‌باز از جمله LLaMA-2، به زبان انگلیسی است و این باعث می‌شود که توانایی تحلیل، استنتاج و تولید متون فارسی در این مدل‌ها نسبت به متون انگلیسی کمتر باشد. یکی از راه‌های بهبود این مدل‌ها تنظیم-دقیق[1]کردن آنها با استفاده از مجموعه‌داده‌های به زبان فارسی است. بدیهی است این روش هزینه مالی و زمانی بسیار کمتری از ساختن مدل‌های محلی است.

[1] Fine-tune

فصل سوم
نتیجه‌گیری

نتیجه‌گیری

در طی کارآموزی در شرکت عصر گویش با برخی از مفاهیم مورد نیاز در حوزه پردازش زبان طبیعی آشنا شدم و با استفاده از کتابخانه‌ها و frameworkهای متعددی از جمله langchain، OpenAI، pytelegrambotapi و با بهره‌گیری از مدل‌های زبانی بزرگ gpt-3.5 و mpnet سعی کردم رباتی طراحی کنم که با استفاده از فناوری‌های پیشرفته و به‌روز در حوزه پردازش زبان طبیعی، بتواند به برخی نیازهای کاربران در هنگام استفاده از چت‌بات‌ها، از جمله پرسش و پاسخ از متون حجیم، خلاصه‌سازی متون تا ۱۵۰صفحه، تعامل و گفت‌وگو با چت‌بات با متون تا اندازه ۱۶هزار توکن به جای ۴هزار توکن پاسخ دهم. بدیهی است که پروژه جای پیشرفت زیادی دارد، محدودیت‌ها و نقص‌های بسیاری همچنان باقی است و قابلیت اضافه کردن ویژگی‌های متعدد دیگری هم دارد که سعی می‌کنیم در آینده این موارد را بهبود بخشیم.

منابع و مراجع

[1] https://web.stanford.edu/class/cs224n/

[2] https://platform.openai.com/tokenizer

[3] Kudo, Taku, and John Richardson. "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing." *arXiv preprint arXiv:1808.06226* (2018)

[4] https://github.com/LexemeAI-Internship/LLM-driven-chatbot-DANA/tree/master

[5] https://python.langchain.com/docs/modules/chains/document/map_reduce