**داکیومنتیشن کد چت بات ایرانسل**

**حمیدرضا اسدیان راد**

**400615004**

**مقدمه:**

**انواع چت بات و مجموعه داده های آموزشی:**

چت بات ها انواع مختلفی دارند که هر کدام نقاط قوت و محدودیت های خود را دارند. انتخاب نوع چت بات و مجموعه داده آموزشی به کاربرد خاص و در دسترس بودن داده ها بستگی دارد.

**انواع چت ربات:**

1. چت ربات های مبتنی بر rule که با پیروی از یک سری قوانین از پیش تعیین شده عمل می‌کنند. این چت‌بات‌ها برای سناریوهایی که تعریف کاملی از آنها شده بسیار کارآمد هستند اما برای سازگاری با داده‌های جدید محدودیت‌هایی دارند.

2. چت‌بات‌های یادگیری ماشین: از الگوریتم‌های یادگیری ماشینی برای یادگیری از روی داده‌های متنی استفاده می‌کنند و پاسخ‌های طبیعی‌تری ایجاد می‌کنند. این مدل‌ها سازگاری بیشتری با داده‌های جدید دارند و می‌‌توانند بصورت همه‌کاره‌تر و کاربرپسندتر عمل کنند.

**مجموعه داده های آموزشی برای چت بات ها:**

1. مجموعه داده های مبتنی بر متن: این داده‌ها متشکل از مجموعه بسیار بزرگی از داده‌های متنی هستند که امکان مکالمه در حیطه‌*های مختلفی را برای کاربر فراهم می‌کنند و می‌توانند مفهوم صحبت کاربر را متوجه شده و بصورت خلاقانه پاسخ‌هایی برای آن تولید کنند.*

2. مجموعه داده های مبتنی بر embedding: در این مجموعه داده، کلمات و جملات بصورت بردارهایی از اعداد نشان داده می‌شوند. با استفاده از این بردارها می‌توان روابط معنایی میان کلمات و جملات را کشف کرد که می‌توان از آن برای کاربردهای مختلفی استفاده کرد.

**چت‌بات‌های مبتنی بر embedding برای کاربردهای با دامنه‌ی خاص:**

این چت‌بات‌ها برای اینکه بتوانند درباره حیطه‌ی خاصی صحبت کنند، بر اساس داده‌های محدود و مربوط به هدف خاصی که دنبال می‌کنند آموزش داده می‌شوند. برای موقعیت‌هایی که الزامات روشنی دارد و می‌توان کمی از انعطاف مدل چشم‌پوشی کرد، گزینه‌ی خوبی محسوب می‌شوند. این مدل‌ها براساس گفتگویی که با کاربر دارند، از پایگاه داده خود برای پاسخ دادن به کاربر استفاده می‌کنند. مثلا برای موردی که در این پروژه دنبال می‌شود، پایگاه داده می‌تواند مجموعه داده‌ای از سوالات و پاسخ‌هایی باشد که احتمالا کاربر خواهد پرسید. بر این اساس که پایگاه داده چگونه بروزرسانی شود و چه embedding هایی استفاده شوند، این مدل می‌تواند بسیار مقیاس‌پذیر و سازگار با اطلاعات جدید باشد که پاسخ‌هایی درخور سوالات کاربر را ارائه می‌کند.

**انتخاب نوع چت بات و مجموعه داده مناسب:**

با توجه به حیطه‌ای که این پروژه دنبال می‌کند که یک مجموعه اطلاعات درباره شرایط و ضوابط محصولات ایرانسل است، آموزش یک مدل با پیچیدگی و پارامترهای زیاد گزینه نسبتا ناکارآمدی محسوب می‌شود. مدل‌های مولد برای تولید داده از روی متن، نیاز به مجموعه داده‌ای بزرگ دارند تا بتوانند خودشان را با حیطه‌ی مورد پژوهش تطبیق دهند. این امر از آنجایی که این پروژه به منبع اندکی از اطلاعات محدود می‌شود، راه حل مناسبی نیست و استفاده از روش‌های جایگزین بهره‌ی بالاتری ایجاد می‌کند. ما در این پروژه از روش embedding استفاده کرده‌ایم. این روش بر اساس مجموعه داده‌ای که برای آن مهیا می‌شود بردارهایی را برای آن مجموعه داده می‌سازد و براساس کاربرد، از این بردارها استفاده می‌کند. برای مثال در این پروژه که هدف ساخت یک چت‌بات پرسش و پاسخ بر اساس متون شرایط و ضوابط ایرانسل است، ما دیتاست نسبتا بزرگی با 154 سوال و پاسخ آماده کردیم و با استفاده از ترانسفورمرهای جمله، این پرسش و پاسخ‌ها را به بردارهایی تبدیل کردیم. این مجموعه داده به خوبی از عهده سوالاتی که کاربر از حیطه شرایط و ضوابط پرسش می‌کند، بر می‌آید و پاسخ‌هایی درخور و سریع می‌دهد. مجموعه‌داده بر اساس جدول‌های html که در یک فایل txt ذخیره شده‌اند تولید می‌شود. هدف از این پارامترها بهبود و تسریع زمان ساخت مجموعه داده‌ها بوده است و می‌توان با سایر روش‌ها نیز مجموعه داده را تولید کرد.

embedding:

embedding نمایش عددی از اطلاعات است که به عنوان مثال می‌تواند متن، تصاویر صدا و غیره را به برداری از اعداد تبدیل کند. این بردار می‌تواند بگونه‌ای معنای جمله یا کلمه را مشخص کند. برای مثال می‌توان متن "ایرانسل چیست؟" را با استفاده از embedding در یک فضای برداری و مثلا با لیستی از 384 عدد نشان داد به عنوان مثال: [0.84، 0.42، ...، 0.02[. از آنجایی که این بردار نشان‌دهنده بار معنایی جمله است، می‌توانیم از آن برای درک سوالات کاربر و پاسخ به آنها استفاده کنیم. برای اینکه بتوانیم بردارهای embedding را تولید کنیم، کتابخانه‌های مختلفی ارائه شده‌اند که یکی از بهترین آنها کتابخانه Sentence Transformers است. این کتابخانه به ما این قابلیت را می‌دهد تا متون را در یک بردار با سایز مختلف embed کنیم. برای استفاده از این کتابخانه، ابتدا باید مدلی را که از پیش آموخته شده است را وارد کنیم و سپس بر اساس پارامترهای مدل انتخاب شده، داده‌های دلخواهمان را embed کنیم.

ما مدل sentence-transformers/all-MiniLM-L6-v2 را انتخاب می‌کنیم. بر اساس شواهد تجربی، این مدل بهینگی بیشتری دارد و همچنین از امتیاز بالاتری در پلتفرم Hugging Face برخوردار است. این مدل، مدل کوچکی است که از داده‌ها، بردارهایی با طول 384 می‌سازد که برای هدفی که ما در این پروژه دنبال می‌کنیم، گزینه‌ی مناسبی محسوب می‌شود که در عین سادگی، بسیار قدرتمند است.

تمام کارکردهایی که در این پروژه انجام می‌شود بصورت محلی است و درصورت قطع ارتباط اینترنتی هیچ خللی در اجرای این کدها وارد نمی‌شود.

کد و مستندات:

این پروژه از سه بخش تشکیل شده است. این بخش‌ها عبارتند از یک فایل جوپیتر برای تولید دیتاست و تست آن، یک فایل پایتون که بعنوان هسته برای برگرداندن پاسخ سوالات کاربر از مجموعه داده استفاده می‌شود و یک فایل پایتون که کد کتابخانه streamlit را برای رابط کاربری اجرا می‌کند.

بدلیل اینکه این حیطه کمی با زبان فارسی بیگانه است و ممکن است در یافتن برخی معادل‌ها دچار مشکلاتی شویم، مستندات کد را در ادامه بصورت انگلیسی می‌آوریم:

**Overview**

This code provides a simple question-answering system using sentence embeddings. It uses the Sentence Transformers library to encode questions and answers into embeddings, and then performs semantic search to find the most relevant answers to a given question.

**Functions**

* model\_generator()

This function initializes and returns a pre-trained Sentence Transformer model. The specific model used here is "sentence-transformers/all-MiniLM-L6-v2".

* dataset\_generator(text, csv\_write=True)

This function takes a text containing HTML tables, extracts questions and answers from the tables, and returns them as separate lists. It also provides an option to write the extracted data to a CSV file. The default output file is "dataset\_english.csv".

* embedding\_func(questions, model, csv\_write=True)

This function takes a list of questions and a pre-trained model, encodes the questions into embeddings, and returns them as a PyTorch tensor. It also provides an option to write the embeddings to a CSV file. The default output file is "embeddings\_english.csv".

* answer\_to\_question(question, answers, questions)

This function takes a user-inputted question, performs encoding, and then finds and prints the most relevant answers using semantic search on the pre-embedded dataset. If the system is uncertain about the question, it suggests a similar question from the dataset.

**Code Execution**

The code starts by generating a pre-trained model using model\_generator().

It then reads questions and answers from an HTML file using dataset\_generator(). If the csv\_write parameter is set to True, it writes the extracted data to a CSV file.

The questions are encoded into embeddings using embedding\_func(). If the csv\_write parameter is set to True, it writes the embeddings to a CSV file.

The user is prompted to input a question, and the system uses answer\_to\_question() to find and print relevant answers.

نحوه کارکرد کد به این صورت است که پس از اجرای کد app.py رابط کاربری برنامه لود می‌شود. سپس کاربر می‌توان در محیط چت با مدل تعامل کند. سوالات کاربر از طریق فایل trained\_model.py تولید می‌شوند. این فایل صرفا به مجموعه داده و بردارهای embedding که با استفاده از Chatbot.ipynb ساخته می‌شوند دسترسی دارد.