بسم الله الرحمن الرحیم

شرکت مهندسی نرم افزاری هلو

گزارش تحقیق در خصوص RAG

کاری از امیرعلی نسیمی

فهرست

مقدمه‌ای بر هوش مصنوعی و بازیابی-تولید (RAG)

مقدمه

هوش مصنوعی (AI) به سرعت در حال تغییر و تحول در بسیاری از صنایع است. یکی از تکنیک‌های جدید و موثر که توجه محققان و متخصصان را به خود جلب کرده است، تکنیک "بازیابی-تولید" یا به اختصار RAG می‌باشد. RAG به عنوان یک روش ترکیبی از بازیابی اطلاعات و تولید متن، در پاسخ به سوالات پیچیده و کاربردهای مرتبط با پردازش زبان طبیعی (NLP) به‌کار گرفته می‌شود.

در این تکنیک، سیستم ابتدا اطلاعات مرتبط با پرسش را از منابع بزرگ اطلاعاتی، مانند پایگاه‌های داده یا اینترنت، بازیابی کرده و سپس از یک مدل تولید زبان استفاده می‌کند تا پاسخی مناسب و دقیق ایجاد کند. این ترکیب از بازیابی داده‌های دقیق و تولید محتوا، به سیستم امکان می‌دهد تا نه تنها اطلاعات را از منابع موجود بازیابی کند بلکه بتواند آن را به شیوه‌ای طبیعی و انسانی بازنویسی و تولید کند.

اهمیت RAG

در بسیاری از سیستم‌های سنتی پاسخ‌گویی به سوالات، فرآیند به دو بخش مجزا تقسیم می‌شد: بخش اول به بازیابی اطلاعات مرتبط با سوال اختصاص داشت و بخش دوم وظیفه تولید پاسخ بر اساس اطلاعات موجود را بر عهده داشت. این روش‌ها اغلب با مشکلاتی از جمله دقت پایین در بازیابی اطلاعات یا ناتوانی در تولید پاسخ‌های جامع مواجه بودند.

RAG این مشکلات را با ادغام بازیابی و تولید برطرف می‌کند. این تکنیک به‌ویژه در حوزه‌هایی که نیاز به بازیابی دقیق اطلاعات و تولید پاسخ‌های منعطف و متناسب با سوال وجود دارد، بسیار مفید است. به عنوان مثال، در صنعت

پزشکی، RAG می‌تواند اطلاعات حیاتی در مورد بیماری‌ها یا درمان‌ها را از پایگاه‌های داده پزشکی بازیابی کند و سپس به زبان ساده‌ای به کاربران ارائه دهد.

تفاوت RAG با دیگر سیستم‌های بازیابی و تولید

در حالی که بسیاری از سیستم‌های سنتی تنها به یک بخش از فرآیند پاسخ‌گویی متمرکز بودند، RAG از یک مدل دوگانه بهره می‌برد که نه تنها اطلاعات را به‌طور موثر بازیابی می‌کند بلکه آن‌ها را به شیوه‌ای طبیعی و قابل فهم تولید می‌کند. در واقع، یکی از اصلی‌ترین تفاوت‌های RAG با دیگر سیستم‌ها این است که از دو رویکرد مکمل یعنی بازیابی اطلاعات و تولید زبان طبیعی به‌طور همزمان استفاده می‌کند.

مدل‌های قدیمی‌تر ممکن است تنها به بازیابی دقیق اطلاعات تکیه کنند، اما این اطلاعات بازیابی‌شده ممکن است به‌طور کامل نیاز کاربر را برآورده نکند. RAG با استفاده از مدل‌های زبان بزرگ مانند BERT یا GPT، اطلاعات بازیابی‌شده را در قالبی مفهومی و معنادار ترکیب و تولید می‌کند.

نقش این تکنیک در پاسخ‌گویی به سوالات

یکی از کاربردهای اصلی RAG در سیستم‌های پاسخ‌گویی به سوالات است. این سیستم‌ها در حوزه‌های مختلفی از جمله خدمات مشتریان، سیستم‌های جستجو، و حتی دستیارهای مجازی به کار گرفته می‌شوند. در یک سیستم سنتی، پاسخ‌دهی به سوالات اغلب به بازیابی مستقیم از یک منبع متنی وابسته بود که می‌توانست محدود به اطلاعات موجود باشد. اما RAG، با ترکیب داده‌های بازیابی‌شده با توانایی تولید محتوای جدید، می‌تواند پاسخ‌های جامع‌تری ارائه دهد.

به عنوان مثال، در یک سیستم خدمات مشتریان، RAG می‌تواند به سرعت اطلاعات مربوط به محصولات یا خدمات را از چندین منبع بازیابی کند و سپس آن را به زبانی ساده و قابل فهم برای کاربر نهایی تولید کند. این روش باعث بهبود کیفیت پاسخ‌ها و افزایش دقت و کارآیی سیستم‌های پاسخ‌گویی می‌شود.

ساختار RAG (بازیابی-تولید)

توضیح فرآیند بازیابی اطلاعات

بازیابی اطلاعات در سیستم‌های RAG به عنوان نخستین مرحله از زنجیره تولید پاسخ، نقشی کلیدی ایفا می‌کند. هدف از بازیابی اطلاعات، یافتن اسنادی است که پاسخ صحیح به سوال مطرح‌شده را در خود دارند. برای این کار، سیستم باید به‌سرعت از میان حجم زیادی از داده‌ها یا اسناد، بهترین و مرتبط‌ترین اطلاعات را استخراج کند.

در سیستم‌های سنتی جستجو مانند مدل‌های TF-IDF یا BM25، واژه‌ها به‌طور ساده از متن استخراج و بر اساس فراوانی و اهمیتشان در یک سند رتبه‌بندی می‌شدند. با این حال، این روش‌ها به دلیل ناتوانی در درک معنایی عمیق‌تر از متن و هم‌بستگی معنایی میان کلمات، نتایج دقیقی ارائه نمی‌دادند. در مقابل، مدل‌های جدید بازیابی اطلاعات برداری (Vector-based Retrieval) مانند DPR (Dense Passage Retrieval)، با استفاده از شبکه‌های عصبی و نمایش برداری (Embeddings) توانایی تحلیل عمیق‌تری از معنای سوال و اسناد دارند.

در این روش‌ها، ابتدا سوال کاربر به بردارهای چندبعدی تبدیل می‌شود که این بردارها نمایانگر ویژگی‌های معنایی آن سوال هستند. سپس، سیستم اسناد موجود در پایگاه داده یا منابع مختلف را نیز به‌صورت بردارهای مشابهی تبدیل می‌کند. در مرحله بعد، سیستم با محاسبه شباهت‌های برداری (Vector Similarity) میان سوال و اسناد، مرتبط‌ترین اسناد را انتخاب و بازیابی می‌کند.

مدل‌های بازیابی برداری به دلیل توانایی درک و مدل‌سازی معنایی بهتر از متن، توانایی بالاتری در پیدا کردن اطلاعات دقیق دارند. به‌ویژه در سیستم‌های پیچیده‌ای که نیاز به درک عمیق از مفهوم سوالات و اسناد دارند، مانند سوالات مربوط به مباحث علمی یا حقوقی، این مدل‌ها عملکرد بهتری از خود نشان می‌دهند. این فرآیند، مرحله اساسی و اولیه‌ای است که زمینه را برای تولید محتوای نهایی فراهم می‌کند.

نحوه ترکیب بازیابی و تولید در RAG

سیستم RAG توانایی منحصر به فردی در ترکیب بازیابی و تولید دارد که باعث می‌شود خروجی‌های آن از سیستم‌های سنتی پاسخ‌گویی بسیار متفاوت باشد. ترکیب این دو مرحله به‌طور هم‌زمان و بهینه، باعث ایجاد سیستم‌های پیشرفته‌ای می‌شود که قادرند نه تنها اطلاعات دقیق و مرتبط را بازیابی کنند، بلکه از آن‌ها برای تولید پاسخ‌هایی با زبان طبیعی استفاده کنند.

در مرحله تولید، مدل زبانی مانند GPT یا BERT از اطلاعات بازیابی‌شده به عنوان ورودی استفاده می‌کند. برخلاف سیستم‌های سنتی که بازیابی و تولید را به‌صورت جداگانه انجام می‌دادند، RAG این دو مرحله را در یک فرآیند هماهنگ و هم‌زمان ادغام می‌کند. این بدان معناست که مدل زبانی مستقیماً از اسناد بازیابی‌شده برای تولید پاسخ استفاده می‌کند، و نه از دانشی که از پیش در مدل ذخیره شده است. به این ترتیب، پاسخ‌ها دقیق‌تر و منطبق بر آخرین اطلاعات به‌روز موجود در منابع بازیابی‌شده هستند.

RAG دو روش کلیدی برای ترکیب بازیابی و تولید دارد:

۱. RAG-Sequence:

در این روش، ابتدا سیستم تمامی اسناد مرتبط با سوال را بازیابی می‌کند و سپس مدل تولید زبان از آن‌ها به‌عنوان یک بسته کامل ورودی برای تولید پاسخ استفاده می‌کند. این روش به مدل تولید زبان اجازه می‌دهد که تمامی

اطلاعات موجود را یک‌جا دریافت و پردازش کند و پاسخ نهایی را به‌طور کامل بر اساس آن تولید کند. این روش برای سوالات پیچیده که نیاز به تجمیع اطلاعات از چندین منبع دارند مناسب است.

۲. RAG-Token:

این روش پیچیده‌تر است و در آن مدل تولید زبان در حین تولید هر قسمت از پاسخ، به‌طور پویا اطلاعات بیشتری از منابع بازیابی‌شده دریافت می‌کند. این روش به مدل تولید زبان انعطاف بیشتری می‌دهد تا بر اساس هر نشانه (Token) یا بخش از پاسخ، از اطلاعات جدیدتر و دقیق‌تر استفاده کند. به عبارتی، سیستم به‌طور مداوم در حین تولید پاسخ، اسناد بازیابی‌شده را مرور می‌کند و از آن‌ها برای تولید هر بخش از پاسخ استفاده می‌کند. این رویکرد به تولید پاسخ‌های دقیق‌تر و منطقی‌تر کمک می‌کند، زیرا مدل زبانی می‌تواند در صورت نیاز به جزئیات بیشتری دست یابد.

نقش مدل‌های زبان بزرگ (LLMs) در تولید پاسخ

مدل‌های زبان بزرگ (LLMs) مانند GPT (Generative Pre-trained Transformer)، BERT (Bidirectional Encoder Representations from Transformers) و T5 (Text-to-Text Transfer Transformer) از پیشرفته‌ترین مدل‌های پردازش زبان طبیعی هستند که توانایی تولید متون طبیعی و شبیه به انسان را دارند. این مدل‌ها در RAG برای مرحله تولید متن پس از بازیابی اطلاعات مورد استفاده قرار می‌گیرند.

GPT یکی از مشهورترین مدل‌های زبانی است که توسط شرکت OpenAI توسعه یافته است. این مدل از معماری Transformer بهره می‌برد و بر اساس یک پایگاه داده بزرگ از متون عمومی و تخصصی آموزش دیده است. مدل‌های مبتنی بر GPT می‌توانند بر اساس ورودی‌های داده‌شده (مانند اسناد بازیابی‌شده)، متنی جدید و طبیعی تولید کنند که شباهت زیادی به نوشتار انسانی دارد. در RAG، GPT اطلاعات بازیابی‌شده را تجزیه و تحلیل کرده و سپس بر اساس آن‌ها یک پاسخ تولید می‌کند که از نظر ساختار زبانی و معنایی بسیار شبیه به متون نوشته‌شده توسط انسان است.

BERT نیز یکی دیگر از مدل‌های مهم در حوزه پردازش زبان طبیعی است که برخلاف GPT از رویکرد دوسویه برای درک متن استفاده می‌کند. این ویژگی به BERT اجازه می‌دهد که متن را با توجه به بافت کل جمله و معنای کلی آن بهتر درک کند. در سیستم RAG، از BERT به عنوان یک مدل پایه برای بازیابی اطلاعات دقیق و متناسب استفاده می‌شود. این مدل می‌تواند سوالات پیچیده را بهتر درک کند و به پاسخ‌های دقیق‌تری دست یابد.

T5 نیز یکی دیگر از مدل‌های زبانی است که تمرکز اصلی آن بر تبدیل وظایف مختلف زبانی به فرمت‌های متنی مشابه است. این مدل قادر است تا وظایف مختلف مانند ترجمه، خلاصه‌سازی، و پاسخ‌گویی به سوالات را به‌طور یکپارچه انجام دهد. در سیستم RAG، T5 می‌تواند برای تولید پاسخ‌هایی که نیاز به تحلیل و تفسیر دقیق متن دارند به کار گرفته شود.

یکپارچگی و هماهنگی بازیابی و تولید در RAG

یکی از ویژگی‌های کلیدی سیستم RAG این است که به‌جای اتکا به تنها یک بخش از پردازش، از یکپارچگی میان بازیابی اطلاعات و تولید پاسخ بهره می‌برد. این روش دو مزیت اصلی دارد:

۱. دقت در اطلاعات بازیابی‌شده: با استفاده از تکنیک‌های برداری و مدل‌های عصبی، سیستم می‌تواند اسنادی را بازیابی کند که با معنای سوال کاملاً همخوانی دارند.

۲. طبیعی‌بودن پاسخ‌ها: مدل‌های زبانی بزرگ مانند GPT و BERT باعث می‌شوند که پاسخ‌های تولیدشده از نظر ساختار زبانی و معنایی کاملاً قابل درک و طبیعی باشند.

این یکپارچگی به RAG اجازه می‌دهد که در صنایع مختلف و برای کاربردهای مختلف، از جمله پرسش و پاسخ در سیستم‌های خدمات مشتریان، تحقیق و جستجوی اطلاعات علمی، و تولید محتوای تعاملی به کار گرفته شود.

تحقیقات انجام شده

جدول مقایسه مدل‌های خودرگرسیو (GPT Medium، GPT Large، GPT Extra Large و LLaMA)

در این بخش، به مقایسه مدل‌های خودرگرسیو (Autoregressive) مختلف شامل GPT در اندازه‌های Medium، Large، Extra Large و مدل LLaMA (Large Language Model Meta AI) پرداخته خواهد شد. این مقایسه بر اساس معیارهای کلیدی مانند تعداد پارامترها، معماری، عملکرد، و کاربردهای اصلی ارائه می‌شود.

مدل	تعداد پارامتر	معماری	حجم داده‌های آموزش	قدرت پردازش و عملکرد	کاربردها	ویژگی‌های کلیدی	ارزیابی جمله فارسی
GPT Medium	345میلیون پارامتر	Transformer-based	داده‌های متنی عمومی و علمی	عملکرد خوب در تولید متن	تولید محتوا، تکمیل متن، چت‌بات	سرعت پردازش خوب، مناسب برای کاربردهای ساده و عمومی	خوب
GPT Large	762میلیون پارامتر	Transformer-based	داده‌های متنوع‌تر و گسترده	قدرت پردازش بیشتر نسبت به مدل Medium	کاربردهای تخصصی‌تر مانند ترجمه، تحلیل زبان طبیعی	دقت بیشتر در تحلیل و تولید متن	به علت حجم زیاد ارزیابی نشد
GPT Extra Large	1.5میلیارد پارامتر	Transformer-based	داده‌های بسیار گسترده و متنوع	عملکرد بسیار قوی در تولید متن و تکمیل	پردازش زبان طبیعی پیشرفته، سوال و جواب‌های پیچیده	مناسب برای کاربردهای پیچیده و تخصصی	به علت حجم زیاد ارزیابی نشد

(7B) LLaMA	7میلیارد پارامتر	Transformer-based	داده‌های گسترده علمی، عمومی، و تخصصی	عملکرد بسیار قوی در تولید متن، سرعت بالا	تحقیق و توسعه، تولید محتوا در مقیاس بزرگ، سوال و جواب پیشرفته	مدل کوچک‌تر و کارآمدتر نسبت به دیگر مدل‌ها با تعداد پارامتر بالا	عالی
(13B) LLaMA	13میلیارد پارامتر	Transformer-based	داده‌های گسترده و تخصصی‌تر	قدرت پردازش بسیار بالا	کاربردهای پیچیده و تحقیقاتی، پردازش داده‌های حجیم	کارایی بالاتر نسبت به نسخه‌های قبلی، مناسب برای کاربردهای پیشرفته	به علت حجم زیاد ارزیابی نشد

۱. GPT Medium:

این مدل با ۳۴۵ میلیون پارامتر، به عنوان یک مدل خودرگرسیو در اندازه متوسط، عملکرد خوبی در تولید متن و پردازش زبان طبیعی دارد. این مدل برای کاربردهای عمومی مانند تولید محتوا و پاسخ‌گویی به سوالات ساده مناسب است. از نظر سرعت پردازش و کارایی، مدل Medium گزینه‌ای اقتصادی برای کارهای روزمره است.

۲. GPT Large:

با ۷۶۲ میلیون پارامتر، مدل GPT Large دقت و توانایی پردازش بیشتری نسبت به مدل Medium دارد. این مدل برای کاربردهای تخصصی‌تر مانند ترجمه زبان، تحلیل متن‌های پیچیده، و چت‌بات‌های پیشرفته مناسب است. در مقایسه با مدل Medium، GPT Large در پردازش داده‌های متنوع و حجم بالاتر عملکرد بهتری دارد.

۳. GPT Extra Large:

این مدل با ۱.۵ میلیارد پارامتر، به‌طور قابل‌توجهی از نظر اندازه و توانایی پردازش قوی‌تر از مدل‌های قبلی است. این مدل برای سوالات پیچیده، تولید محتوا در مقیاس بزرگ، و تحلیل‌های پیشرفته در زمینه پردازش زبان طبیعی استفاده

می‌شود. GPT Extra Large به دلیل قدرت پردازش بالا و دقت زیاد، برای کاربردهای پیچیده و تحقیقاتی بسیار مناسب است.

۴. LLaMA (7B):

مدل LLaMA با ۷ میلیارد پارامتر، یکی از مدل‌های پیشرفته‌ای است که توسط Meta AI توسعه یافته است. با وجود تعداد پارامترهای بسیار بالاتر از GPT Extra Large، این مدل بهینه‌سازی‌های بیشتری در کارایی و پردازش ارائه می‌دهد. این مدل برای تحقیقات پیشرفته، تولید محتوای پیچیده، و سوال و جواب‌های پیچیده بسیار مناسب است.

۵. LLaMA (13B):

با ۱۳ میلیارد پارامتر، نسخه پیشرفته‌تر مدل LLaMA است که عملکرد بسیار قوی‌تری در پردازش حجم‌های عظیم داده دارد. این مدل برای پردازش داده‌های پیچیده‌تر و تحقیقاتی مانند تحلیل متون تخصصی و علمی استفاده می‌شود و از کارایی بالایی برخوردار است. به‌عنوان یکی از مدل‌های بزرگ زبان، LLaMA 13B به‌ویژه در حوزه‌های تحقیق و توسعه پیشرفته مورد استفاده قرار می‌گیرد.

جدول مقایسه مدل‌هایEmbedding

در این بخش، به مقایسه مدل‌های برجسته embedding مانند Zephyr, MPNet-all, LaBSE و دیگر مدل‌های معروف پرداخته می‌شود. این مدل‌ها ابزارهای قدرتمندی برای نمایش معنایی کلمات و جملات به صورت برداری در فضای عددی هستند و به ویژه در پردازش زبان طبیعی (NLP) استفاده می‌شوند.

مدل	تعداد پارامترها	معماری	حجم داده‌های آموزش	قدرت پردازش و عملکرد	کاربردها	ویژگی‌های کلیدی
Zephyr	Sentence Embedding	512	تحلیل احساسات، ترجمه متون	زبان‌های مختلف	قابلیت استفاده برای چندین زبان و	Zephyr

					کاربردهای پیچیده پردازش زبان طبیعی	
MPNet-all (all-mpnet-base-v2)	Sentence Transformer	768	مشابه‌سازی جملات، جستجوی معنایی	چند زبانه، پشتیبانی از زبان‌های مختلف	دقت بسیار بالا در مشابه‌سازی جملات و مفاهیم، ترکیب دو روش ماسک و ترتیب‌گذاری	MPNet-all (all-mpnet-base-v2)
LaBSE (Language-agnostic BERT Sentence Embedding)	Sentence Transformer	768	جستجوی چند زبانه، ترجمه	بیش از ۱۰۰ زبان	مدل چند زبانه با توانایی بسیار بالا در ترجمه و یافتن معانی در زبان‌های مختلف	LaBSE (Language-agnostic BERT Sentence Embedding)
USE (Universal Sentence Encoder)	Transformer-based, Deep Averaging Network	512	پردازش زبان طبیعی، جستجوی متون	انگلیسی و چند زبان دیگر	مدل عمومی برای نمایش جمله‌ای، بهینه‌سازی شده برای متن‌های عمومی و محتوای جستجو	USE (Universal Sentence Encoder)
SBERT (Sentence-BERT)	Sentence Transformer	768	مقایسه جملات، رتبه‌بندی متن	چند زبانه، زبان‌های مختلف	مدل سفارشی‌سازی شده برای دقت بالاتر در مشابه‌سازی و جستجوی معنایی	SBERT (Sentence-BERT)
DistilBERT	Transformer (distilled)	768	نمایش معنایی و طبقه‌بندی	انگلیسی و برخی زبان‌های دیگر	مدل سبک‌تر با کارایی بالا برای کاربردهای زمان واقعی و بهینه در مصرف حافظه	DistilBERT

1- Zephyr:

مدل Zephyr یک مدل جدید برای ایجاد نمایش‌های معنایی از جملات است. ابعاد برداری آن ۵۱۲ است و برای کاربردهایی مانند تحلیل احساسات و ترجمه متون به کار می‌رود. این مدل از قابلیت‌های چندزبانه‌ای برخوردار است و می‌تواند برای کاربردهای پیچیده پردازش زبان طبیعی استفاده شود.

2- MPNet-all (all-mpnet-base-v2):

این مدل یکی از بهترین مدل‌های embedding برای مشابه‌سازی معنایی جملات است MPNet .از روش‌های ماسک و ترتیب‌دهی استفاده می‌کند و با ابعاد برداری ۷۶۸، دقت بسیار بالایی در مشابه‌سازی متون ارائه می‌دهد. این مدل برای جستجوی معنایی و مقایسه جملات به خوبی عمل می‌کند و پشتیبانی چندزبانه‌ای دارد.

3 - LaBSE (Language-agnostic BERT Sentence Embedding):
مدل LaBSE یک مدل BERT چندزبانه است که برای جستجوهای معنایی و ترجمه متون استفاده می‌شود. این مدل با ابعاد ۷۶۸، از بیش از ۱۰۰ زبان پشتیبانی می‌کند و به دلیل قابلیت‌های چندزبانه و قدرت پردازش در یافتن معانی در زبان‌های مختلف، در کاربردهای جهانی محبوب است.

4- USE (Universal Sentence Encoder):
مدل Universal Sentence Encoder (USE) یکی از مدل‌های ساده‌تر و کاربرپسندتر برای ایجاد embeddingهای جملات است. با ابعاد ۵۱۲، این مدل برای کاربردهای عمومی پردازش زبان طبیعی و جستجو مناسب است. استفاده از USE ساده و سریع است، و برای بسیاری از کاربردهای پایه‌ای توصیه می‌شود.

5- SBERT (Sentence-BERT):
مدل SBERT بر پایه BERT ساخته شده است و برای دقت بالاتر در مشابه‌سازی جملات به کار می‌رود. این مدل با ابعاد ۷۶۸، توانایی بالایی در مقایسه جملات و رتبه‌بندی متن دارد SBERT .به خصوص برای کاربردهایی که نیاز به جستجوی معنایی دقیق دارند مناسب است.

6- DistilBERT:
DistilBERT یک نسخه سبک‌تر و سریع‌تر از BERT است. این مدل با ابعاد ۷۶۸، برای کاربردهایی که نیاز به سرعت و مصرف بهینه حافظه دارند بسیار مناسب است. DistilBERT به دلیل کاهش تعداد پارامترها همچنان دقت خوبی دارد و برای کاربردهای زمان واقعی (real-time) استفاده می‌شود.

خط لوله RAG با استفاده از Embedding و GPT

برای ایجاد یک خط لوله مبتنی بر RAG (تولید مبتنی بر بازیابی اطلاعات) با استفاده از یک مدل embedding و GPT، باید نحوه عملکرد RAG را درک کنیم و همچنین نحوه تعامل اجزای آن یعنی مدل embedding (برای بازیابی اطلاعات) و GPT (برای تولید متن) را بفهمیم. هدف اصلی این خط لوله، بازیابی اطلاعات مرتبط از یک پایگاه داده یا دانش و سپس تولید پاسخ‌هایی طبیعی و دقیق بر اساس اطلاعات بازیابی شده است.

اجزای خط لوله RAG

خط لوله RAG از دو بخش اصلی تشکیل شده است:

۱. بازیابی اطلاعات (Document Retriever) با استفاده از مدل Embedding

این بخش وظیفه پیدا کردن مرتبط‌ترین اسناد یا پاراگراف‌ها از یک پایگاه داده بزرگ را دارد. این کار با استفاده از یک مدل embedding که هم پرسش‌ها (ورودی‌های کاربر) و هم اسناد را به صورت بردارهایی در فضای معنایی نمایش می‌دهد انجام می‌شود. مراحل اصلی در این بخش شامل موارد زیر است:

- مرحله ۱: تولید بردار پرسش: زمانی که یک پرسش مطرح می‌شود، مدل embedding این پرسش را به یک بردار چگال (dense vector) تبدیل می‌کند. این کار معمولاً توسط مدل‌هایی مانند MPNet، LaBSE، USE، یا SBERT انجام می‌شود که جملات را به بردارهای چندبعدی تبدیل می‌کنند.

- مرحله ۲: تولید بردار اسناد: تمامی اسناد موجود در پایگاه دانش به صورت پیش‌پردازش شده و به صورت embedding ذخیره می‌شوند. این کار امکان مقایسه کارآمد آن‌ها با بردار پرسش را فراهم می‌کند.

– مرحله ۳: جستجوی شباهت (بازیابی): پس از تولید بردار پرسش، سیستم جستجوی شباهت (مانند شباهت کسینوسی یا ضرب داخلی) را برای یافتن مرتبط‌ترین بردارهای سند انجام می‌دهد. مرتبط‌ترین اسناد انتخاب و به مرحله بعد منتقل می‌شوند.

۲. تولید زبان (Language Generator) با استفاده از GPT

پس از بازیابی اسناد مرتبط توسط embedder، مرحله بعد شامل تولید پاسخ طبیعی زبان بر اساس آن اسناد است. در این مرحله از مدل GPT استفاده می‌شود. مراحل شامل:

– مرحله ۴: ورودی اسناد به GPT: اسناد بازیابی‌شده به همراه پرسش اصلی کاربر به عنوان زمینه (context) به مدل GPT داده می‌شوند. این کار به GPT اطلاعات لازم برای تولید پاسخ‌های معنادار را فراهم می‌کند.

– مرحله ۵: تولید پاسخ: GPT بر اساس پرسش کاربر و زمینه اسناد بازیابی شده، پاسخ طبیعی تولید می‌کند. چون GPT یک مدل خودرگرسیو است، کلمات بعدی را براساس کلمات قبلی پیش‌بینی می‌کند و در نتیجه پاسخ منسجم و روان تولید می‌شود.

مراحل خط لوله RAG

۱. ورودی پرسش:

– کاربر یک سوال یا درخواست مطرح می‌کند

۲. تبدیل پرسش به بردار:

– پرسش از طریق یک مدل embedding (مثل MPNet یا LaBSE) به یک بردار چگال تبدیل می‌شود.

۳. بازیابی اسناد:

– سیستم جستجوی شباهت بین بردار پرسش و embeddingهای از پیش محاسبه‌شده اسناد را انجام داده و مرتبط‌ترین اسناد را بازیابی می‌کند.

۴. ایجاد زمینه اسناد:

– اسناد بازیابی‌شده با هم ترکیب شده و به عنوان زمینه به مدل GPT داده می‌شوند.

۵. تولید پاسخ:

– GPT براساس ورودی‌های ارائه‌شده، پاسخ طبیعی و منسجمی تولید می‌کند که به پرسش کاربر پاسخ می‌دهد.

۶. خروجی:

– پاسخ تولید شده توسط GPT به عنوان پاسخ نهایی به کاربر بازگردانده می‌شود.

موارد کلیدی برای ساخت خط لوله RAG

۱. انتخاب مدل Embedder مناسب:

– انتخاب مدل مناسب برای embedding می‌تواند تأثیر زیادی بر عملکرد بخش بازیابی اطلاعات داشته باشد. برای کاربردهای چندزبانه، LaBSE ممکن است مناسب‌تر باشد، در حالی که برای مشابه‌سازی دقیق جملات، MPNet یا SBERT گزینه‌های بهتری هستند.

۲. بهینه‌سازی کارایی:

– محاسبه و ذخیره از پیش embeddingهای اسناد و استفاده از تکنیک‌های جستجوی سریع (مثل FAISS) می‌تواند سرعت بازیابی اطلاعات را حتی برای مجموعه داده‌های بزرگ افزایش دهد.

۳. تنظیم GPT برای دامنه‌های خاص:

– برای کاربردهای خاص، ممکن است نیاز به تنظیم دقیق مدل GPT روی داده‌های مرتبط با دامنه خاص باشد. این کار باعث می‌شود پاسخ‌ها دقیق‌تر و مناسب‌تر باشند.

۴. مدیریت طول زمینه در GPT:

– تعداد اسناد بازیابی‌شده که به GPT داده می‌شوند باید در چارچوب محدودیت‌های طول توکن (مثلاً ۴۰۹۶ توکن برای GPT-3) باشند. انتخاب بیش از حد اسناد طولانی ممکن است منجر به حذف بخش‌هایی از زمینه شود، بنابراین تعادل بین مرتبط بودن اسناد و دقت پاسخ ضروری است.

پروژه انجام شده – تبدیل سوالات مربوط به بانک

در این پروژه، ما به بررسی یک سیستم پرسش و پاسخ (Q&A) مبتنی بر مدل‌های یادگیری عمیق و استفاده از LangChain می‌پردازیم:

پردازش داده‌ها

در مرحله اول، داده‌ها از یک فایل JSON بارگذاری می‌شوند. داده‌ها شامل سوالات و پاسخ‌های مختلف است که به فرمت مورد نیاز برای استفاده در مدل‌های یادگیری ماشین تبدیل می‌شود. بخشی از این سوال و جواب ها به شرح زیر است:

۱. ایجاد حساب کاربری:

پرسش: چگونه می‌توانم یک حساب کاربری ایجاد کنم؟

پاسخ: برای ایجاد حساب کاربری، روی دکمه 'Sign Up' در گوشه بالا سمت راست وبسایت کلیک کنید و دستورالعمل‌های ثبت‌نام را دنبال کنید.

۲. روش‌های پرداخت:

پرسش: چه روش‌های پرداختی را قبول می‌کنید؟

پاسخ: ما کارت‌های اعتباری اصلی، کارت‌های بدهی و PayPal را برای سفارش‌های آنلاین قبول می‌کنیم.

۳. پیگیری سفارش:

پرسش: چگونه می‌توانم سفارش خود را پیگیری کنم؟

پاسخ: با ورود به حساب کاربری و مراجعه به بخش 'Order History' می‌توانید سفارش خود را پیگیری کنید.

۴. سیاست بازگشت کالا:

پرسش: سیاست بازگشت کالاهای شما چیست؟

پاسخ: ما اجازه می‌دهیم تا محصولات را ظرف ۳۰ روز از تاریخ خرید بازگردانید و مبلغ کامل را بازپرداخت کنیم، مشروط بر اینکه محصولات در شرایط اصلی و بسته‌بندی خود باشند.

۵. لغو سفارش:

پرسش: آیا می‌توانم سفارش خود را لغو کنم؟

پاسخ: بله، اگر سفارش هنوز ارسال نشده باشد، می‌توانید آن را لغو کنید.

۶. مدت زمان ارسال:

پرسش: مدت زمان ارسال چقدر است؟

پاسخ: مدت زمان ارسال به مقصد و روش ارسال انتخاب شده بستگی دارد. ارسال استاندارد معمولاً ۳-۵ روز کاری طول می‌کشد، در حالی که ارسال فوری ممکن است ۱-۲ روز کاری زمان ببرد.

با توجه به تعداد زیاد سوال ها و جواب ها، از ادامه آن ها خودداری شده است.

آماده‌سازی مستندات

LangChain برای پردازش و جستجو در این مستندات استفاده می‌شود لذا ÷پس از پردازش داده ها، کلیه این موارد در قالب مشخص به این فریم ورک داده می شود.

بارگذاری مدل‌ها

مدل‌های یادگیری عمیق، به ویژه مدل‌های بزرگ زبانی (مثل Zephyr-7b)، بارگذاری می‌شوند. این مدل‌ها برای پردازش و تولید پاسخ‌ها استفاده می‌شوند. تنظیمات مختلفی برای بهینه‌سازی عملکرد مدل‌ها و کاهش مصرف حافظه مورد استفاده قرار گرفته اند.

ساخت پایپ‌لاین پرسش و پاسخ

یک پایپ‌لاین برای پرسش و پاسخ ایجاد می‌شود که شامل مدل زبانی و سیستم جستجوی مستندات است. این پایپ‌لاین به کاربران امکان می‌دهد تا سوالات خود را بپرسند و پاسخ‌های مناسب را دریافت کنند. جهت ساخت این پایپ لاین از langchain استفاده شده است.