به نام خدا

# پروژه دوم درس NLP: **پیاده سازی تولید متن با استفاده از بازیابی (RAG) بر روی گزارشهای سازمانی**

## نام اعضای گروه: فرزان رحمانی - 99521271 محمد حسین عباسپور - 99521433 محمد اصولیان - 99521073

### دانشگاه علم و صنعت ایران دانشکده مهندسی کامپیوتر

# مقدمه

پروژه‌ای که در اینجا توضیح داده می‌شود، با هدف بهبود فرآیند پاسخ‌دهی به سوالات سازمانی بر اساس اسناد و مدارک موجود در سازمان طراحی شده است. در این پروژه، از تکنولوژی Augmented-Retrieval Generation (RAG) استفاده می‌شود که ترکیبی از بازیابی اطلاعات و تولید متن با کمک مدل‌های هوش مصنوعی است. در این پروژه، هدف اصلی استفاده از تکنولوژی RAG برای پاسخ‌دهی به سوالات مربوط به مدارک سازمانی است. تکنولوژی RAG ترکیبی از دو فرآیند اصلی است:

1. **بازیابی اطلاعات (Retrieval)**: یافتن مرتبط‌ترین اسناد و مدارک با توجه به سوال مطرح شده.
2. **تولید متن (Generation)**: تولید پاسخ‌های طبیعی و دقیق بر اساس اطلاعات بازیابی شده.

این پروژه تلاش می‌کند تا با استفاده از این تکنولوژی، به سوالات کاربران سازمانی به شکلی موثر و کارآمد پاسخ دهد.

این پروژه دارای اهمیت ویژه‌ای برای سازمان‌ها است. با افزایش حجم اطلاعات و اسناد در سازمان‌ها، پاسخ‌دهی به سوالات و دسترسی به اطلاعات دقیق و مرتبط به یکی از چالش‌های اصلی تبدیل شده است. این پروژه با بهره‌گیری از تکنولوژی RAG، می‌تواند به اهداف ذیل دست پیدا کند:

1. **بهبود دسترسی به اطلاعات**: افزایش سرعت و دقت در دسترسی به اطلاعات مرتبط با سوالات کاربران.
2. **افزایش کارایی سازمان**: کاهش زمان مورد نیاز برای جستجو و بازیابی اطلاعات.
3. **بهبود تصمیم‌گیری**: فراهم کردن اطلاعات دقیق و به‌روز برای تصمیم‌گیری‌های بهتر در سازمان.
4. **افزایش رضایت کاربران**: ارائه پاسخ‌های سریع و دقیق به سوالات کاربران سازمانی.

# پیش زمینه تکنولوژیکی

در این پروژه، از ترکیبی از مدل‌ها و تکنولوژی‌های پیشرفته هوش مصنوعی و پردازش زبان طبیعی (NLP) استفاده شده است. مهم‌ترین مدل‌ها و تکنولوژی‌های مورد استفاده عبارتند از:

## LLaMA (Large Language Model Meta AI)

**LLaMA** یک مدل زبانی بزرگ است که توسط شرکت Meta (فیسبوک) توسعه یافته است. این مدل مبتنی بر معماری‌های پیشرفته یادگیری عمیق است و برای درک و تولید متن‌های طبیعی در مقیاس بزرگ طراحی شده است. LLaMA توانایی‌های زیر را دارد:

* **تولید متن**: تولید پاسخ‌های طبیعی و متنی که شبیه به نوشته‌های انسانی است.
* **پردازش زبان طبیعی(NLP):** قابلیت انجام وظایف مختلف NLP مانند ترجمه زبان، خلاصه‌سازی متن، و پاسخ‌دهی به سوالات.
* **قابلیت یادگیری از متن‌های طولانی**: به دلیل معماری بهینه، این مدل قادر است متن‌های طولانی را به خوبی پردازش کند و از تمام اطلاعات موجود در آن‌ها برای تولید پاسخ استفاده کند.

در این پروژه، از مدل زبانی LLaMA برای تولید پاسخ‌های دقیق و طبیعی بر اساس اسناد بازیابی شده استفاده می‌شود.

## LaBSE (Language-agnostic BERT Sentence Embedding)

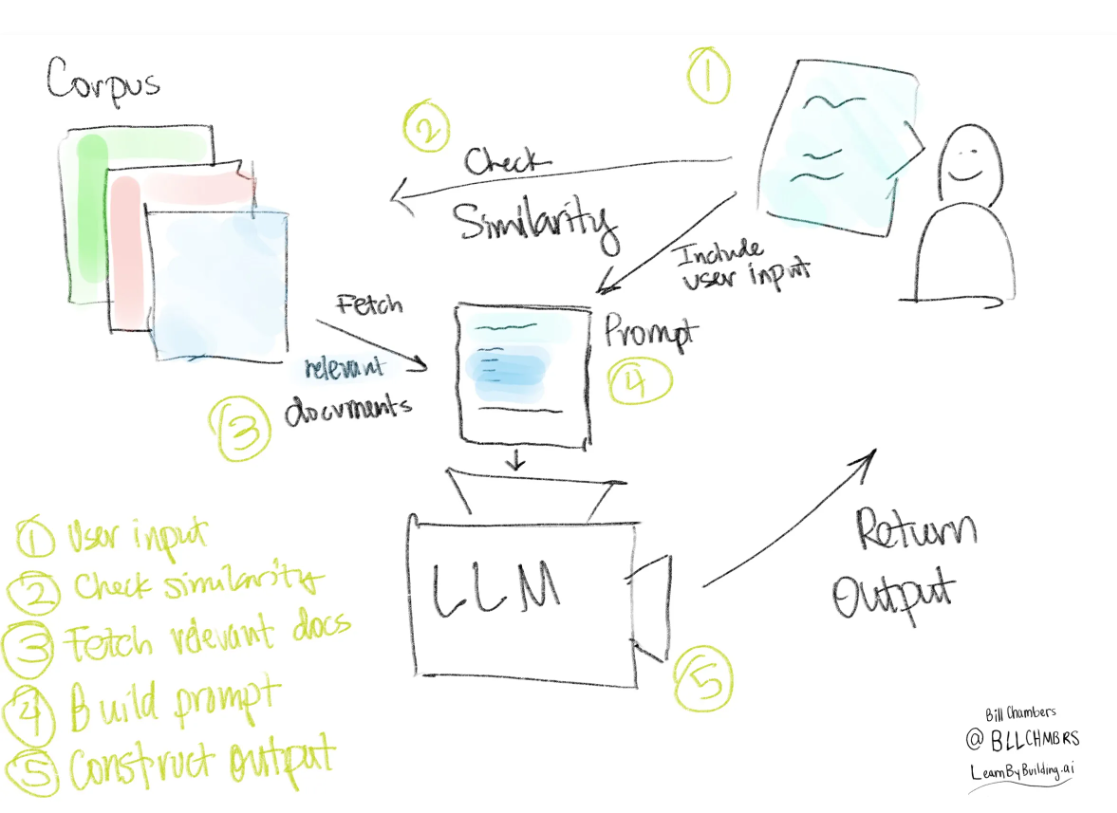
LaBSEیک مدل جاسازی جمله است که بر اساس معماری BERT ساخته شده است و قادر است جملات را در چندین زبان مختلف به یک فضای جاسازی مشترک تبدیل کند. این ویژگی به برنامه‌هایی که به جستجو و بازیابی متن نیاز دارند، کمک می‌کند تا اطلاعات را به صورت کارا و دقیق از منابع مختلف جمع‌آوری کنند.ویژگی‌های اصلی LaBSE عبارتند از:

* **پشتیبانی از چند زبان**: توانایی پردازش و تولید بردارهای معنایی برای جملات به زبان‌های مختلف، بدون نیاز به تنظیمات خاص برای هر زبان.
* **بردارهای معنایی**: تولید بردارهای عددی که معانی جملات را به شکلی فشرده و قابل مقایسه نمایه‌سازی می‌کنند.
* **استفاده در بازیابی اطلاعات**: به دلیل قابلیت تولید بردارهای معنایی، LaBSE به طور خاص در بازیابی اطلاعات و مقایسه متون بسیار مفید است.

در این پروژه، LaBSE برای بردار‌سازی جملات از اسناد سازمانی و سوالات کاربران استفاده می‌شود. این بردارها سپس برای مقایسه و بازیابی اسناد مرتبط به کار می‌روند.

## RAG (Retrieval-Augmented Generation)

RAG یک مدل تولید متن است که از اطلاعات به دست آمده از جستجوی متن (retrieval) برای بهبود فرایند تولید متن استفاده می‌کند. به طور کلی، RAG دو بخش اصلی دارد: یک مدل بازیابی که برای جستجوی متن استفاده می‌شود و یک مدل تولید که برای ایجاد متن استفاده می‌شود. این دو بخش با همکاری تولید متن بهتر و معنادارتری که از منابع خارجی به دست آمده، کمک می‌کنند.



ویژگی‌های کلیدی RAG عبارتند از:

* **بازیابی اطلاعات** (**Retrieval**): در این مرحله، سیستم مرتبط‌ترین اسناد و داکیومنت‌ها را با توجه به سوال مطرح شده بازیابی می‌کند.
* **تولید متن (Generation):** پس از بازیابی اسناد، سیستم با استفاده از مدل‌های زبانی، پاسخی مناسب و طبیعی بر اساس اطلاعات موجود در اسناد بازیابی شده تولید می‌کند.
* **ترکیب دو مرحله**: RAG به طور همزمان از قابلیت‌های بازیابی و تولید استفاده می‌کند، که این امر منجر به تولید پاسخ‌های دقیق‌تر و مرتبط‌تر می‌شود.

در این پروژه، RAG برای ترکیب مراحل بازیابی اطلاعات از اسناد سازمانی و تولید پاسخ‌های طبیعی و دقیق به سوالات کاربران به کار می‌رود.

# مراحل پیاده سازی

## جمع‌آوری داده‌ها

**جمع‌آوری داده‌ها**: ابتدا تمامی اسناد و مدارک سازمانی از منابع مختلف مانند پایگاه‌های داده، فایل‌های متنی و ایمیل‌ها جمع‌آوری می‌شود.

**پیش‌پردازش داده‌ها**: داده‌ها به منظور پاکسازی، حذف نویزها، و تبدیل به فرمت‌های قابل پردازش، پردازش می‌شوند. این شامل نرمال‌سازی متون، حذف کاراکترهای اضافی و تقسیم‌بندی متن به جملات و پاراگراف‌ها می‌شود. همچنین فایل ورد questions 1.docx را تبدیل به questions 1.xlsx کردیم تا همگی در یک فرمت باشند. به علاوه، فایل های input 1.docx و input 2.doc را تبدیل به فرمت pdf کردیم.

# questions\_df = pd.read\_excel('questions 2.xlsx')

# Read the Excel file without using the first row as the header

questions\_df = pd.read\_excel('questions 2.xlsx', header=None)

# Drop the first row which contains irrelevant information

questions\_df = questions\_df.drop(0)

# questions\_df = questions\_df.drop(-1)

# Reset the index

questions\_df.reset\_index(drop=True, inplace=True)

# Set the new first row as the column names

questions\_df.columns = questions\_df.iloc[0]

# Drop the row that is now used as the header

questions\_df = questions\_df.drop(0)

# Reset the index again

questions\_df.reset\_index(drop=True, inplace=True)

questions\_df



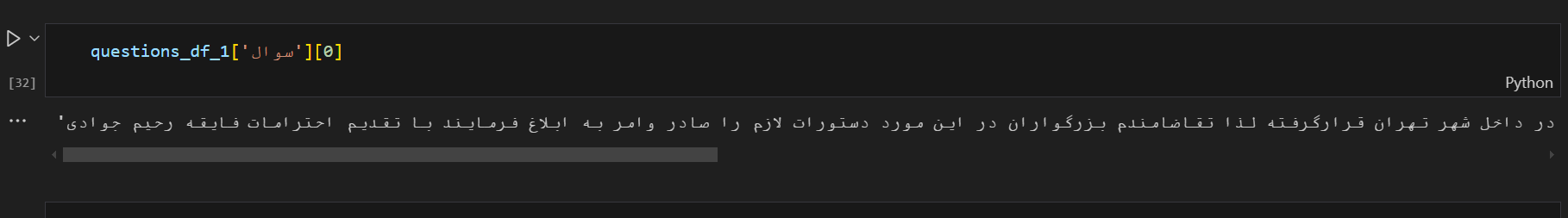
questions\_df\_1 = pd.read\_excel('questions 1.xlsx')

questions\_df\_1

questions\_df\_1 = pd.DataFrame(questions\_df\_1.dropna())

questions\_df\_1.reset\_index(drop=True, inplace=True)

questions\_df\_1



## لود کردن مدل‌های لازم

در این مرحله مدل های Llama و LaBSE لود می‌شوند تا از آنها برای ایجاد نمایه و تولید پاسخ استفاده شود.

* **ایجاد نمایه با LaBSE:** اسناد سازمانی با استفاده از **LaBSE** ایندکس می‌شوند. **LaBSE** از بردارهای معنایی برای نمایه‌سازی جملات و پاراگراف‌ها استفاده می‌کند که به تسریع فرآیند بازیابی کمک می‌کند.
* **آموزش مدل LaBSE** : مدل LaBSE با استفاده از داده‌های موجود در اینترنت از قبل آموزش داده شده ولی برای بهترشدن نتایج میتوان آن را با نمونه اسنادی که در اختیار داشت، تنظیم دقیق کرد که در این پروژه در نظر گرفته نشده است.
* **آموزش مدل LLaMA** : مدل Llama با استفاده از داده‌های موجود در اینترنت از قبل آموزش داده شده ولی برای بهترشدن نتایج میتوان آن را با نمونه اسنادی که در اختیار داشت، تنظیم دقیق کرد. ما به علت کمبود وقت این کار را نکردیم.

llm = HuggingFaceLLM(

    model\_name="NousResearch/Meta-Llama-3-8B-Instruct",

    generate\_kwargs={

        "do\_sample": True,

        "temperature": 0.6,

        "top\_p": 0.9,

        "pad\_token\_id": tokenizer.pad\_token\_id

    },

    tokenizer\_name="NousResearch/Meta-Llama-3-8B-Instruct",

    stopping\_ids=stopping\_ids,

    # The quantization line (for T4 GPU)

    # comment it for using TPU

    model\_kwargs={"torch\_dtype": torch.bfloat16, "load\_in\_4bit": True},

    device\_map="auto",

)

## تلفیق مدل‌ها برای کارکرد RAG

در این مرحله، از مدل LaBSE به عنوان مدل بازیابی استفاده می‌شود که برای جستجو و بازیابی اطلاعات مورد نیاز برای تولید متن، به کار می‌رود. این مدل معمولاً یک سیستم جستجوی متن است که قادر به بازیابی مطالب مرتبط با یک موضوع خاص از مجموعه‌ای از متن‌ها یا پایگاه‌های داده است

با استفاده از مدل بازیابی، مطالب مرتبط با موضوع مورد نظر از مجموعه‌ای از متن‌ها یا پایگاه‌های داده بازیابی می‌شود. این مطالب به عنوان ورودی برای مرحله بعدی (مدل تولید) استفاده می‌شود.

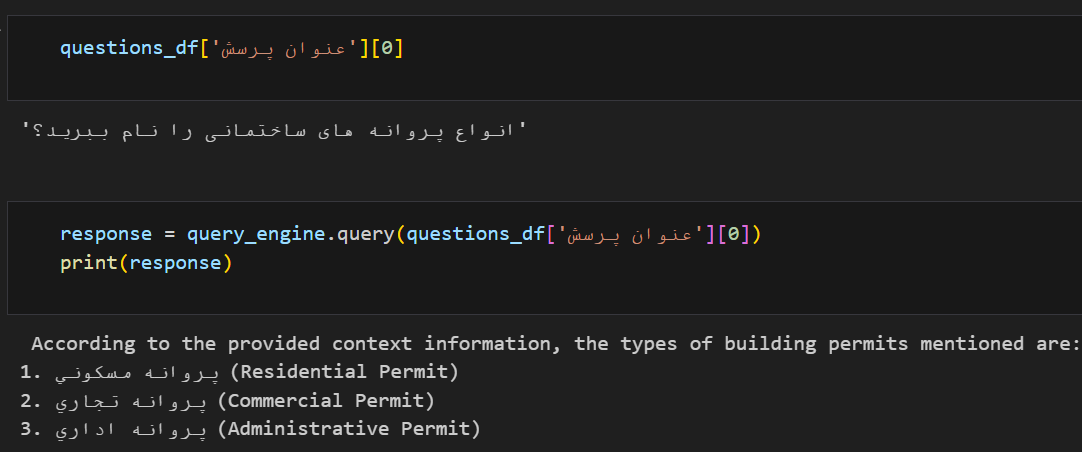
در این مرحله، از مدل Llama به عنوان مدل تولید متن برای تولید متن نهایی بر اساس مطالب بازیابی شده به کار می‌رود. این مدل مسئول ایجاد جملات و متونی است که منطقی و مفهومی باشند و با محتوای بازیابی شده همخوانی داشته باشند.

در نهایت مطالب بازیابی شده و توسط مدل تولید تولید شده، به هم ادغام می‌شوند تا متن نهایی تولید شود. این مرحله می‌تواند شامل فرآیندهایی مانند انتخاب و ادغام جملات یا پاراگراف‌های مختلف باشد تا یک متن جامع و کامل تر تولید شود.

# بررسی نتایج و مشاهدات

در این بخش، نتایج به‌دست‌آمده از پروژه با استفاده از تکنولوژی RAG با تلفیق مدل‌های LLaMA و LaBSE به‌طور دقیق‌تر بررسی می‌شود:

* **دقت و مرتبط‌بودن پاسخ‌ها**: استفاده از مدل LaBSE برای ایجاد بردارهای معنایی و استفاده از LLaMA برای تولید جملات معنی‌دار باعث بهبود دقت و مرتبط‌بودن پاسخ‌ها به سوالات مطرح شده توسط کاربران می‌شود. این امر به کاربران اطمینان می‌دهد که پاسخ‌هایی که دریافت می‌کنند، دقیق و مرتبط با نیازهای آن‌ها است.
* **سرعت پاسخ‌دهی**: با استفاده از تکنیک‌های بازیابی اطلاعات به‌وسیلهLaBSE ، زمان مورد نیاز برای بازیابی اطلاعات کاهش می‌یابد که منجر به افزایش سرعت پاسخ‌دهی به کاربران می‌شود. البته متوجه هستیم که در مقایسه با عدم به کارگیری بازیابی اطلاعات و فقط اتکا به LLM سرعت کمتری داریم.
* **کیفیت پاسخ‌ها**: استفاده از LLaMA برای تولید جملات معنی‌دار و طبیعی بهبود محسوسی در کیفیت پاسخ‌ها ایجاد می‌کند که بازخوردهای مثبت بیشتری از سوی کاربران دریافت می‌شود.



# چالش‌های فنی و نحوه غلبه

## سنگین بودن مدل ها

مدل‌های مانند LLaMA و LaBSE دارای تعداد پارامترهای بسیار بالا هستند که نیازمند منابع محاسباتی قدرتمند برای آموزش و استفاده هستند. این امر می‌تواند به مشکلاتی نظیر افزایش هزینه‌ها و نیاز به مدیریت دقیق منابع منجر شود. همچنین چون Kaggle فیلتر بود و به ناچار از کارت گرافیگ T4 کولب استفاده کردیم، مجبور شدیم مدل LLaMA را به شکل کوانتیزه لود کنیم.

## محدودیت زمانی

برای تولید ایندکس‌های مورد نیاز برای بازیابی اطلاعات با استفاده از LLaMA، زمان زیادی صرف می‌شود. این ایندکس‌ها نیازمند پیش‌پردازش دقیق و پردازش مجدد بر روی مجموعه داده‌ها هستند تا اطلاعاتی که توسط مدل‌ها به اشتراک گذاشته می‌شوند به شکلی قابل استفاده باشند. این موضوع می‌تواند به تاخیر در پاسخ‌دهی به کاربران منجر شود.

## منابع محاسباتی

استفاده از GPU برای آموزش و استفاده از مدل‌های بزرگی مانند LLaMA و LaBSE اساسی است. کمبود منابع مناسب مانند GPU می‌تواند محدودیت‌های جدی برای اجرای پروژه ایجاد کند. به دلیل کمبود منابع، امکان آموزش و استفاده همزمان از مدل‌های بزرگ در محیط‌های توسعه ممکن است محدود شود.

## راه حل چالش ها

برای حل این مشکلات میتوان از منابع محاسباتی ابری استفاده کرد. این منابع به صورت رایگان یا با دریافت هزینه در اختیار افراد قرار میگیرند. برای انجام این پروژه از فضای google colab استفاده شد.

# آموخته‌ها و پیشنهادات

با توجه به تجربیات به‌دست‌آمده در اجرای پروژه با استفاده از تکنولوژی RAGو مدل هایLLaMA و LaBSE، می‌توان به نکات زیر اشاره کرد:

**آموخته‌ها**

* **اهمیت پیش‌پردازش داده‌ها**: پیش‌پردازش داده‌ها به‌طور کامل و دقیق قبل از ورود به مدل‌ها بسیار حیاتی است. این کار می‌تواند به بهبود دقت و کیفیت پاسخ‌ها کمک کند.
* **نیاز به منابع محاسباتی**: استفاده از مدل‌های پرمصرف نیازمند منابع محاسباتی قدرتمند است، بنابراین باید قبل از شروع پروژه اطمینان حاصل شود که منابع مورد نیاز در دسترس هستند.
* **افزایش بالقوه دقت مدل با به کارگیری از تکنولوژی RAG**: این مهم به این دلیل است که اطلاعات بازیابی شده درخور سوال مربوطه هستند و مدل احتمالا آنها را در طی آموزش ندیده یا اگر هم دیده به علت حجم داده های زیاد به خوبی به یاد ندارد.

**پیشنهادات برای تیم‌های آینده**

* **شروع با پروژه‌های پایلوت**: شروع با یک پروژه پایلوت کوچک و تست مدل‌ها قبل از اجرای پروژه به مقیاس بزرگتر می‌تواند به کاهش ریسک‌ها و بهبود استراتژی اجرایی کمک کند.
* **مدیریت منابع محاسباتی**: مدیریت دقیق منابع محاسباتی و برنامه‌ریزی استفاده از آن‌ها به‌طور هوشمند برای کاهش هزینه‌ها و بهره‌وری بیشتر از منابع بسیار حیاتی است.

این نکات می‌توانند به تیم‌هایی که قصد دارند پروژه‌های مشابهی را در آینده اجرا کنند، کمک کنند تا بهره‌وری و عملکرد بهتری داشته باشند.

# منابع استفاده شده

## مقالات و مستندات علمی:

* "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Jacob Devlin et al.
* "LaBSE: Language-agnostic BERT Sentence Embedding" by Feng Gao et al.
* "RAG: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" by Patrick Lewis et al.

## وب‌سایت‌ها و ابزارها:

* [Hugging Face](https://huggingface.co/)
* [Google AI](https://ai.google/)
* [Meta AI](https://ai.facebook.com/)
* [Google Colab](https://colab.research.google.com/)

## ابزارهای توسعه:

* Hugging Face Transformers Library
* PyTorch
* LLaMA
* llama-index
* Google Colab

پایان