Natural Language Processing

استاد : سرکارخانم دکتر فهمیه قاسمیان

گردآورنده: سید محمد معین حسینی دهمیری

بهار ۱۴۰۵

بخش مهندسی کامپیوتر ، دانشگاه شهید باهنر کرمان ، کرمان ، ایران

گزارش پروژه: طراحی و پیاده‌سازی سامانه هوشمند قوانین آموزشی با استفاده از معماری RAG و الگوریتم MMR

۱. مقدمه و بیان مسئله

پاسخگویی به سؤالات مکرر دانشجویان در خصوص آیین‌نامه‌ها و قوانین آموزشی، همواره یکی از چالش‌های بخش آموزش دانشگاه‌ها بوده است. هدف از این پروژه، توسعه یک سامانه هوشمند مبتنی بر هوش مصنوعی است که با دریافت پرسش به زبان طبیعی، بتواند مستندات دانشگاهی را تحلیل کرده و پاسخی دقیق، مستند و تشریحی ارائه دهد. در این سامانه از معماری RAG (Retrieval-Augmented Generation) بهره گرفته شده است تا از تولید پاسخ‌های غیرواقعی (Hallucination) توسط مدل زبانی جلوگیری شود.

۲. معماری سیستم و ماژول‌های توسعه‌یافته

این سامانه در بستر زبان برنامه‌نویسی پایتون و در قالب سه ماژول اصلی توسعه یافته است:

الف) ماژول تنظیمات پیکربندی (settings.py)

برای مدیریت یکپارچه مسیرها و کلیدهای دسترسی، فایل تنظیماتی ایجاد شده است. در این فایل، مسیر استقرار اسناد (pdf_docs)، مسیر ذخیره پایگاه داده (knowledge_db) و مدل Embedding با نام intfloat/multilingual-e5-large که در درک مفاهیم زبان فارسی بسیار قدرتمند است، تعریف شده است.

ب) پردازشگر و وکتورایزر داده‌ها (data_processor.py)

این ماژول وظیفه ساخت پایگاه دانش سیستم را بر عهده دارد که شامل گام‌های زیر است:

- **تبدیل تصاویر و استخراج متن:** فایل‌های PDF با استفاده از ابزار poppler به تصویر تبدیل شده و سپس توسط موتور Tesseract OCR متون فارسی آن‌ها استخراج می‌شود.
- **نرمال‌سازی (Normalization):** برای یکسان‌سازی فاصله‌ها و نیم‌فاصله‌ها در زبان فارسی، از کتابخانه hazm استفاده شده است.
- **تقطیع متون (Chunking):** متون استخراج شده به بلوک‌های ۱۵۰۰ کاراکتری با ۲۰۰ کاراکتر هم‌پوشانی شکسته می‌شوند تا جملات ناقص نمانند.
- **تولید بردار (Vectorization):** بلوک‌های متنی توسط مدل Embedding به بردار تبدیل شده و در وکتور دیتابیس ChromaDB (با استفاده از فاصله Cosine) نمایه می‌شوند.

ج) رابط کاربری و موتور استنتاج (web_interface.py)

این بخش واسط ارتباطی بین کاربر و مدل زبانی را فراهم می‌کند:

- **تکنیک جستجوی پیشرفته (MMR):** برخلاف جستجوهای ساده، در این سامانه برای جلوگیری از استخراج متون تکراری و افزایش تنوع اطلاعات، از الگوریتم Maximal Marginal Relevance (MMR) استفاده شده است. این الگوریتم ابتدا ۲۰ قطعه مرتبط را پیدا کرده و سپس ۸ قطعه‌ای که بیشترین ارتباط و در عین حال کمترین شباهت به یکدیگر را دارند، گلچین می‌کند.
- **تولید پاسخ استدلالی:** متون گلچین شده به همراه پرسش کاربر (پس از نرمال‌سازی) به مدل زبانی llama-3.3-70b-versatile از طریق Groq API ارسال می‌شوند. پرامپت (Prompt) سیستم به گونه‌ای تنظیم شده است که پاسخی تشریحی و کامل ارائه دهد و در صورت نیافتن ارتباط معنایی، از پاسخ دادن خودداری کند.
- **طراحی بصری (UI):** محیط **گرافیکی** با استفاده از Streamlit توسعه یافته و با کدهای سفارشی CSS، محیطی تیره (Dark Mode)، کاملاً راست‌چین و کاربرپسند برای نمایش پاسخ‌ها و ارجاعات (شماره سند و صفحه) ایجاد شده است.

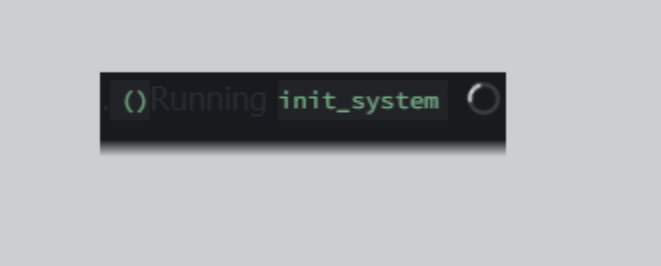
۳. نمایش محیط برنامه و سناریوهای کاربری

رابط کاربری این سامانه با تمرکز بر سادگی، کارایی و تجربه کاربری (UX) بهینه طراحی شده است. استفاده از تم تیره (Dark Mode)، کادرهای تفکیک‌شده و قالب‌بندی کاملاً راست‌چین (RTL)، فضایی مدرن و خوانا را برای کاربر فارسی‌زبان فراهم کرده است. در ادامه، عملکرد سامانه در قالب سناریوهای مختلف اجرایی بررسی شده است:

۳-۱. نمای اولیه و صفحه ورود پرسش

هنگامی که کاربر سامانه را اجرا می‌کند، با یک محیط یکپارچه مواجه می‌شود که شامل عنوان برنامه و یک کادر متنی ساده برای دریافت پرسش است. کاربر می‌تواند سؤال خود را به زبان محاوره یا رسمی در این کادر تایپ کرده و روی دکمه «جستجو» کلیک کند. بلافاصله پس از کلیک، یک نشانگر در حال چرخش (Spinner) با پیام "لطفا صبر کنید... سیستم در حال تحلیل مستندات است..." ظاهر می‌شود که نشان‌دهنده آغاز فرآیند

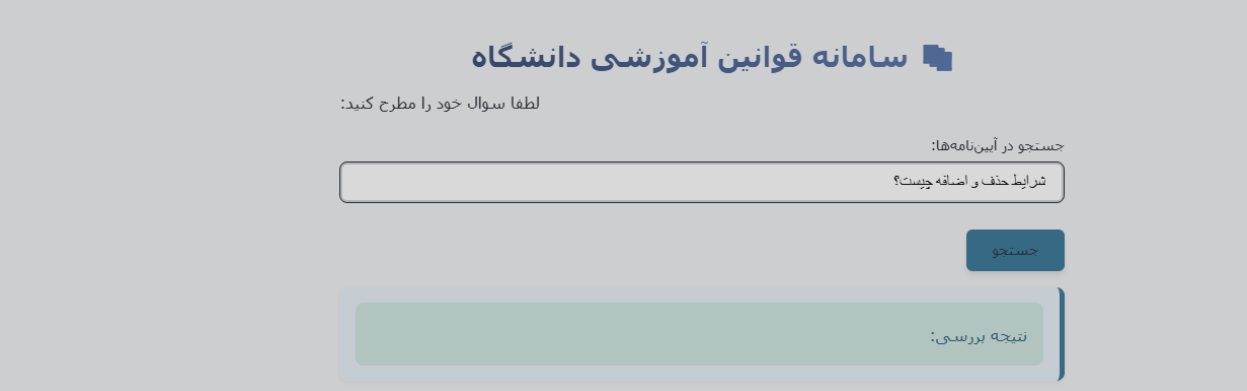
جستجوی برداری و ارتباط با API مدل زبانی است.



۳-۲. سناریوی اول: پاسخگویی موفق و استخراج قوانین

اگر کاربر سؤالی مرتبط با آیین‌نامه‌ها بپرسد (به عنوان مثال: "شرایط حذف و اضافه چیست؟" را وارد کند)، سیستم پس از تحلیل، پاسخی جامع و دسته‌بندی شده را در یک کادر سبز رنگ (Success Box) نمایش می‌دهد.

یکی از مهم‌ترین ویژگی‌های این بخش، کادر «مستندات و منابع ارجاعی» در انتهای پاسخ است. در این قسمت، سیستم دقیقاً اعلام می‌کند که این پاسخ را از کدام فایل‌های PDF و دقیقاً از چه صفحاتی استخراج کرده است تا کاربر بتواند برای اطمینان، به اصل آیین‌نامه مراجعه کند.



سامانه قوانین آموزشی دانشگاه

لطفا سوال خود را مطرح کنید:

جستجو در آیین‌نامه‌ها:

شرایط حذف و اضافه چیست؟

جستجو

نتیجه بررسی:

شرایط حذف و اضافه در دانشگاه بر اساس مقررات و آئین‌نامه‌های آموزشی به شرح زیر است:

حذف واحد درسی:

1. دانشجو می‌تواند در صورت موافقت دانشگاه، یک هفته قبل از شروع امتحانات پایان نیمسال، صرفا یک درس نظری را در بازه زمانی تعیین‌شده توسط دانشگاه حذف کند (ماده ۱۸).
2. درخواست حذف ترم و واحد درسی تنها در صورتی قابل بررسی است که دانشجو در هیچ امتحانی شرکت نکرده باشد و استاد نیز برای آن درس و یا دروس غیبت کلاسی ثبت ننموده باشد (ماده ۷).
3. در صورتی که دانشجو در جلسه امتحان درسی حاضر شده باشد، امکان بررسی حذف پزشکی آن درس پس از امتحان وجود نخواهد داشت (ماده ۷).
4. برای حذف واحد درسی، دانشجو باید مدارک لازم را مانند عکس پرونده بستری، گواهی متخصص، نسخه دارویی، و نتایج پارا کلینیک را بارگذاری کند (متن مرجع: الف، ب، ج، د).

اضافه واحد درسی:

1. دانشجو می‌تواند در هر نیمسال تحصیلی، حداقل ۸ واحد درسی تا حداکثر ۱۴ واحد درسی را اخذ کند (ماده ۳۰).
2. در صورتی که دانشگاه به هر دلیل با حذف واحدهای درسی دانشجو موافقت کند و این امر منجر به کاهش تعداد واحدهای دانشجو از حد نصاب شود، آن نیمسال به عنوان یک نیمسال کامل جزو سنوات تحصیلی وی محسوب می‌شود (تبصره ۱ ماده ۱۴۶).
3. دانشجو در نیمسال آخر تحصیلی از شرط اخذ حداقل واحد آموزشی معاف است (تبصره ۱ ماده ۳۰).

حذف پزشکی:

1. در صورتی که دانشجو بدلیل بیماری در جلسه امتحان غیبت نماید، باید دلایل خود را حداکثر تا ۷۲ ساعت بعد از تاریخ امتحان با ارائه درخواست حذف پزشکی در سامانه گلستان ثبت نماید (ماده ۹).
2. عدم اعلام علت غیبت، عدم تأیید گواهی پزشکی، و یا غیبت غیر موجه در جلسه امتحان، غیبت غیر مجاز محسوب و نمره درس صفر منظور خواهد شد (ماده ۹).
3. ارائه گواهی استعلاجی نمی‌تواند دلیلی برای عدم شرکت در امتحان و حذف پزشکی درس باشد (تبصره ۲).

در مجموع، شرایط حذف و اضافه در دانشگاه بر اساس مقررات و آئین‌نامه‌های آموزشی و با توجه به شرایط خاص هر دانشجو تعیین می‌شود.

مستندات و منابع ارجاعی:

منبع 1: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 18

منبع 2: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 20

منبع 3: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 104

منبع 4: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 84

منبع 5: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 115

منبع 6: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 17

منبع 7: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 83

منبع 8: سند *Majmooeh ghavanin va ayin-nameh-haaye amoozeshi 1402.pdf* | **صفحه:** 114

۳-۳. سناریوی دوم: مدیریت پرسش‌های نامرتبط یا خارج از مستندات

یکی از الزامات اصلی معماری RAG، جلوگیری از دروغ‌پردازی هوش مصنوعی (Hallucination) است. اگر کاربر سؤالی بپرسد که جواب آن در هیچ‌کدام از فایل‌های PDF آپلود شده وجود نداشته باشد (مثلاً سؤالی درباره اخبار روز یا قوانین یک دانشگاه دیگر)، الگوریتم سیستم به جای حدس زدن یا ارائه اطلاعات غلط، فرآیند را متوقف کرده و پیام هشداری با مضمون "پاسخ این سؤال در اسناد موجود یافت نشد." را به کاربر نمایش می‌دهد. این ویژگی اعتبار و اتکاپذیری سامانه را در محیط‌های رسمی دانشگاهی تضمین می‌کند.

۴. راهنمای نصب و راه‌اندازی

برای استقرار و اجرای سامانه بر روی محیط محلی، مراحل زیر باید انجام شود:

۱. **نصب وابستگی‌ها:** نصب کتابخانه‌های مورد نیاز از جمله streamlit، langchain، chromadb، hazm، pdf2image و pytesseract با استفاده از مدیر بسته‌ی pip.

۲. **پیش‌نیازهای سیستمی:** نصب نرم‌افزار Tesseract (به همراه داده‌های زبان fas) و Poppler روی سیستم‌عامل میزبان و افزودن مسیر اجرای آن‌ها به متغیرهای محیطی سیستم (PATH).

۳. **ساخت پایگاه دانش:** قرار دادن فایل‌های آیین‌نامه در پوشه‌ی pdf_docs و سپس اجرای اسکریپت python data_processor-3.py در ترمینال جهت ساخت وکتور دیتابیس در پوشه‌ی knowledge_db.

٤. **اجرای سرویس وب:** اجرای دستور streamlit run web_interface.py در ترمینال. پس از اجرای این دستور، سامانه در مرورگر (آدرس localhost) در دسترس خواهد بود.

۵. نتیجه‌گیری

این پروژه نشان داد که ترکیب تکنیک‌های پردازش تصویر (OCR)، نرمال‌سازی متون فارسی، و الگوریتم‌های پیشرفته بازیابی نظیر MMR در کنار مدل‌های زبانی عظیم (LLM)، می‌تواند سامانه‌ای قابل‌اطمینان برای تفسیر و پاسخگویی به متون پیچیده اداری ایجاد کند.

از منظر فنی، به‌کارگیری ترکیبی از تکنیک‌های پردازش تصویر (Tesseract OCR) برای استخراج متون فارسی از فایل‌های PDF اسکن‌شده، و همچنین نرمال‌سازی داده‌ها با استفاده از ابزارهای پردازش زبان طبیعی (NLP)، کیفیت داده‌های ورودی را به شدت افزایش داد. علاوه بر این، پیاده‌سازی الگوریتم پیشرفته‌ی MMR (Maximal Marginal Relevance) در فاز بازیابی اطلاعات، باعث شد تا سیستم بر مشکلاتی نظیر استخراج متون تکراری غلبه کرده و متنوع‌ترین و مرتبط‌ترین بندهای قانونی را برای تحلیل به مدل زبانی قدرتمند Llama-3.3 ارسال کند.

اتکاپذیری و جلوگیری از تولید اطلاعات غلط:
یکی از مهم‌ترین دستاوردهای کاربردی این سامانه، جلوگیری کامل از پدیده توهم (Hallucination) در هوش مصنوعی است. سیستم به گونه‌ای برنامه‌ریزی شده است که در صورت عدم وجود اطلاعات در مستندات پایگاه دانش، با شفافیت کامل از ارائه اطلاعات ساختگی خودداری کرده و کاربر را مطلع می‌سازد. همچنین، قابلیت ارجاع‌دهی دقیق به نام فایل و شماره صفحه مستندات، امکان راستی‌آزمایی را برای دانشجویان و کارشناسان آموزش فراهم کرده و اعتماد به این دستیار هوشمند را در محیط‌های رسمی افزایش می‌دهد.