دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پروژه درس پردازش زبان های طبیعی
مهندسی کامپیوتر

سیستم RAG با پشتیبانی از داده‌های چندوجهی در حوزه‌ی مشاهیر

نگارش

گروه ۸ و ۱۹

استاد

دکتر احسان‌الدین عسگری

شهریور ۱۴۰۴


بسم الله الرحمن الرحیم

چکیده

امروزه سیستم‌های پرسش و پاسخ مبتنی بر مدل‌های زبان بزرگ با چالش‌هایی نظیر تولید اطلاعات نادرست (توهم) و محدودیت دانش مواجه هستند. معماری بازیابی‌ـافزایش‌ـتولید (RAG) با اتصال مدل به یک پایگاه دانش خارجی، راهکاری مؤثر برای این مشکلات ارائه می‌دهد. با این حال، بسیاری از این سیستم‌ها به اطلاعات متنی محدود هستند و از دانش غنی موجود در تصاویر غفلت می‌کنند. این پژوهش بر طراحی و پیاده‌سازی یک سیستم پرسش و پاسخ چندوجهی (Multimodal-RAG) برای زبان فارسی، با تمرکز بر دامنه دانش شخصیت‌های ایرانی، متمرکز است. برای این منظور، ابتدا یک پایگاه دانش شامل بیوگرافی‌های متنی و تصاویر مرتبط جمع‌آوری شد. سپس، یک مدل Encoder دوگانه با استفاده از یادگیری تقابلی (Contrastive-Learning) آموزش داده شد تا بتواند بازنمایی‌های برداری هم‌تراز برای جفت‌های متن‌ـتصویر تولید کند. این مدل به عنوان بخش بازیاب (Retriever) در یک پایپ‌لاین کامل RAG به کار گرفته شد تا اسناد و تصاویر مرتبط با سوال کاربر را بازیابی کرده و در اختیار یک مدل زبان بزرگ (Generator) قرار دهد. ارزیابی نهایی بر روی یک مجموعه سوالات چهارگزینه‌ای نشان داد که مدل چندوجهی پیشنهادی، با بهره‌گیری از اطلاعات تصویری، به دقت بالاتری نسبت به مدل‌های صرفاً متنی دست می‌یابد و توانایی سیستم در پاسخگویی دقیق‌تر را اثبات می‌کند. دستاوردهای این پروژه شامل یک دموی تحت وب و انتشار مدل Encoder آموزش‌دیده برای استفاده عمومی است.

کلیدواژه‌ها: پرسش و پاسخ چندوجهی، معماری RAG، بازیابی اطلاعات، یادگیری تقابلی، پردازش زبان طبیعی فارسی، مدل‌های متن‌ـتصویر.

فهرست مطالب

فصل ۱

مقدمه

توانایی انسان در درک جهان، حاصل تلفیق اطلاعات از حواس مختلف، به ویژه بینایی و زبان است. سیستم‌های هوش مصنوعی نوین نیز برای دستیابی به درکی عمیق‌تر، نیازمند فرارفتن از مرزهای متنی و ورود به دنیای چندوجهی (Multimodal) هستند. در حوزه پردازش زبان طبیعی، مدل‌های زبان بزرگ (LLMs) با وجود توانمندی‌های چشمگیر در تولید متن، به دو ضعف عمده مشهورند: تمایل به تولید اطلاعات نادرست یا *توهم* (*Hallucination*) و عدم آگاهی از اطلاعات جدیدی که پس از تاریخ آموزش آن‌ها رخ داده است.

معماری **بازیابی‌ـ افزایش‌ـ تولید**[۱] به عنوان یک راهکار پیشرو برای مقابله با این چالش‌ها ظهور کرده است. ایده کلیدی در RAG، مجهز کردن مدل زبان به یک پایگاه دانش خارجی است. پیش از تولید پاسخ، سیستم ابتدا اسناد مرتبط با پرسش کاربر را از این پایگاه بازیابی کرده و به عنوان "زمینه" (Context) در اختیار مدل قرار می‌دهد. این فرآیند، پاسخ‌ها را به سمت اطلاعات واقعی و مستند هدایت می‌کند.

با این حال، دانش ما تنها در قالب متن خلاصه نمی‌شود. یک تصویر می‌تواند حاوی جزئیاتی باشد که توصیف آن در هزاران کلمه نیز دشوار است. برای مثال، درک کامل یک شخصیت تاریخی بدون دیدن تصویر او ناقص است. این واقعیت، انگیزه اصلی این پژوهش برای حرکت به سوی **RAG چندوجهی** است؛ سیستمی که قادر است به طور همزمان از منابع اطلاعاتی متنی و تصویری برای پاسخ به پرسش‌ها بهره ببرد.

---

[۱] Generation Retrieval-Augmented

۱-۱ هدف و مسئله پژوهش

این پژوهش بر طراحی و پیاده‌سازی یک سیستم پرسش و پاسخ چندوجهی برای زبان فارسی در دامنه دانش شخصیت‌های عمومی ایران تمرکز دارد. با توجه به کمبود منابع داده چندوجهی در زبان فارسی، این پروژه یک راهکار یکپارچه برای حل این مسئله ارائه می‌دهد:

۱. **ساخت پایگاه دانش چندوجهی:** جمع‌آوری و پالایش مجموعه‌ای از بیوگرافی‌های متنی و تصاویر مرتبط با شخصیت‌ها.

۲. **توسعه مدل بازیاب (Retriever) چندوجهی:** آموزش یک مدل Encoder دوگانه که بتواند با دریافت یک پرسش (متنی یا تصویری)، مرتبط‌ترین اسناد متنی و تصویری را از پایگاه دانش بازیابی کند.

۳. **پیاده‌سازی و ارزیابی سیستم یکپارچه:** ادغام مدل بازیاب با یک مدل زبان بزرگ (Generator) برای ساخت یک سیستم سرتاسر که قادر به پاسخگویی دقیق به سوالات است.

این پروژه نه تنها یک راهکار عملی برای پرسش و پاسخ هوشمند ارائه می‌دهد، بلکه با ایجاد زیرساخت داده و مدل، گامی در جهت توسعه پژوهش‌های چندوجهی در زبان فارسی برمی‌دارد.

فصل ۲

روش‌ها و داده

۲-۱ جمع آوری داده

اساس یک سیستم بازیابی اطلاعات کارآمد، مجموعه‌داده‌ای جامع و باکیفیت است. در این پروژه، با هدف ساخت یک سیستم پاسخگویی چندوجهی در حوزه مشاهیر، فاز اول به جمع‌آوری داده‌های متنی و تصویری از دانشنامه آنلاین ویکی‌پدیا فارسی اختصاص یافت. ویکی‌پدیا به دلیل پوشش گسترده اطلاعاتی، ساختار نیمه‌منظم صفحات (infobox) و در دسترس بودن تصاویر مرتبط، به عنوان منبع اصلی انتخاب گردید. فرآیند جمع‌آوری داده شامل چندین مرحله‌ی دقیق از جمله خزش وب، پاک‌سازی و رفع نویز، و نهایتاً ساختارمندسازی اطلاعات بود که در ادامه به تفصیل تشریح می‌گردند.

۲-۱-۱ کرال کردن داده ها

با توجه به ماهیت پویا و ساختار درختی و تودرتوی دسته‌بندی‌های ویکی‌پدیا، پیمایش دستی برای استخراج داده‌ها امری غیرممکن و مستعد خطا بود. از این رو، یک خزشگر (Crawler) خودکار با استفاده از زبان برنامه‌نویسی پایتون پیاده‌سازی شد. این خزشگر با بهره‌گیری از API رسمی مدیاویکی، فرآیند پیمایش را به صورت برنامه‌ریزی‌شده انجام داد.

با الهام از ساختار گرافی ویکی‌پدیا که در آن هر صفحه یک گره (Node) و هر پیوند یک یال (Edge) است، الگوریتم جستجوی سطح به سطح (BFS) برای پیمایش انتخاب شد. این الگوریتم با شروع از یک دسته ریشه (در پروژه ما، دسته‌هایی که با عبارت «سیاست‌مداران اهل ایران» آغاز می‌شوند)، ابتدا تمام زیردسته‌های سطح اول را پیمایش کرده و سپس به سطوح عمیق‌تر می‌رود. این رویکرد تضمین می‌کند که

پیمایش به صورت کنترل‌شده پیش رفته و از افتادن در شاخه‌های بسیار عمیق و نامرتبط جلوگیری شود. برای جلوگیری از پیمایش تکراری و ایجاد حلقه، از یک مجموعه visited برای نگهداری شناسه‌ی صفحات و دسته‌های بازدید شده استفاده گردید.

خروجی نهایی این مرحله، فایلی با فرمت CSV بود که شامل فهرستی از عناوین صفحات (مقالات) مرتبط با حوزه هدف به همراه لینک دسترسی به آن‌ها بود.

این روش مربوط به حوزه مشاهیر سیاسی بود که بطور خاص گروه ما با آن در ارتباط بود.

۲-۱-۲ پاکسازی داده ها پس از کرال

فهرست اولیه عناوین تولید شده توسط خزشگر، حاوی نویز قابل توجهی بود؛ برای مثال، شامل عناوینی می‌شد که به مفاهیم، رویدادها یا لیست‌ها اشاره داشتند و نه به یک شخص حقیقی. بنابراین، یک فرآیند دومرحله‌ای برای رفع نویز و اعتبارسنجی داده‌ها طراحی شد:

۱. **اعتبارسنجی موجودیت انسان :** در این گام، با استفاده از API ویکی‌داده ،(Wikidata) برای هر عنوان صفحه، شناسه موجودیت (Entity ID) متناظر استخراج شد. سپس با بررسی ویژگی P۳۱، مشخص گردید که آیا آن موجودیت از نوع «انسان» Q۵ است یا خیر. عناوینی که این شرط را احراز نمی‌کردند، به عنوان نویز شناسایی و از مجموعه داده حذف شدند. این فرآیند منجر به تولید یک لیست خالص از عناوین صفحات متعلق به اشخاص حقیقی شد.

۲. **فیلتر بر اساس جعبه اطلاعات :(Infobox)** جعبه اطلاعات (Infobox) منبعی غنی از داده‌های ساختاریافته و کلیدی است. در این مرحله، برای هر صفحه معتبر، محتوای HTML آن دریافت و با استفاده از کتابخانه ،BeautifulSoup وجود جدولی با کلاس infobox بررسی شد. صفحاتی که فاقد این جعبه اطلاعاتی بودند، از لیست نهایی کنار گذاشته شدند، زیرا داده‌های ساختاریافته‌ی لازم برای فاز بهبود دادگان را در اختیار نداشتند.

برای رعایت سیاست‌های استفاده از API و جلوگیری از فشار بر سرورهای ویکی‌پدیا و ویکی‌داده، از یک نشست Session HTTP با شناسه کاربری مشخص و همچنین تابع time.sleep برای ایجاد تأخیر بین درخواست‌های متوالی استفاده شد.


سیّد محمود حسابی

زادهٔ	۴ اسفند ۱۲۸۱ ۲۳ فوریهٔ ۱۹۰۳ میدان شاهپور، بازار قوام الدوله، تهران[۱]
درگذشت	۱۲ شهریور ۱۳۷۱ ۳ سپتامبر ۱۹۹۲ (۸۹ سال) بیمارستان قلب دانشگاه ژنو، سوئیس
علت مرگ	بیماری قلبی
آرامگاه	تفرش
ملیت	ایرانی
دیگر نام‌ها	پروفسور حسابی، دکتر حسابی
تحصیلات	دکترای فیزیک لیسانس مهندسی معدن لیسانس ادبیات لیسانس مهندسی راه و ساختمان[۲]
محل تحصیل	دانشگاه تهران دانشگاه سورین دانشگاه آمریکایی بیروت
پیشه(ها)	فیزیک‌دان، سناتور، وزیر فرهنگ
همسر	صدیقه حالری
فرزندان	سید ایرج حسابی انوشه سادات حسابی
والدین	گوهرشاد حسابی سید عباس معزالسلطنه
خویشاوندان	محمد حسابی (برادر)
وبگاه	بنیاد پروفسور حسابی

یادداشت‌ها

[۳][۴][۵]

شکل ۲-۱: *یک نمونه infobox مربوط به صفحه ویکی‌پدیا دکتر حسابی*

۲-۱-۳ فرمت داده ها در JSON

پس از اتمام مراحل پاک‌سازی و اعتبارسنجی، گام حیاتی بعدی، تبدیل داده‌های نیمه‌ساختاریافته‌ی استخراج‌شده به یک فرمت کاملاً ساختاریافته، استاندارد و غنی بود. برای این منظور، یک شِمای (Schema) دقیق در قالب JSON طراحی گردید. هدف از این شِما، فراتر از ذخیره‌سازی ساده‌ی داده‌ها بود؛ این ساختار به عنوان یک مدل مفهومی عمل می‌کند که اطلاعات حیاتی هر شخص را به صورت تفکیک‌شده، سلسله‌مراتبی و قابل فهم برای ماشین، سازماندهی می‌نماید. فرآیند پر کردن این شِما شامل یک مرحله استخراج اطلاعات بود که طی آن، موجودیت‌های کلیدی از متن بیوگرافی و جدول اطلاعات (Infobox) شناسایی و در فیلدهای مربوطه نگاشت داده می‌شدند.

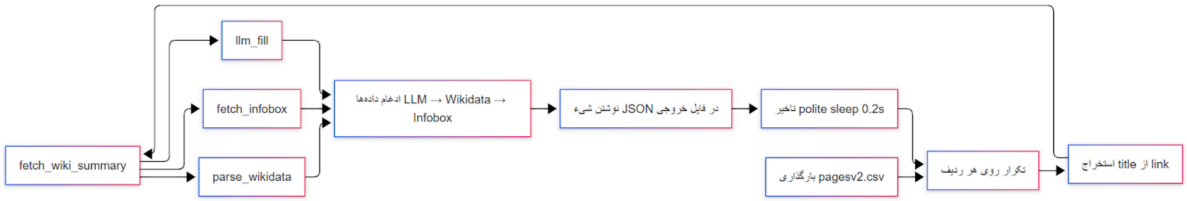
همانطور که در شکل ۲-۲ نشان داده شده است، این ساختار شامل فیلدهای کلیدی مانند `name`، `sex` و `nick-names` برای اطلاعات هویتی پایه است. اطلاعات مربوط به تولد و وفات در اشیاء تودرتوی `birth` و `death` سازماندهی شده‌اند که هرکدام شامل تاریخ، مکان (استان و شهر) و حتی مختصات جغرافیایی (`coordinates`) می‌باشند. فیلدهای مهم دیگری نظیر `era` (دوره تاریخی)، `occupation` (لیست مشاغل)، `works` (آثار) و در نهایت `image` (لیستی از URL تصاویر) این مدل داده را تکمیل می‌کنند.

```
{
  "name": "",
  "sex": "",
  "nick-names": [],
  "birth": {
    "date": "",
    "location": {
      "province": "",
      "city": "",
      "coordinates": {
        "latitude": "",
        "longitude": ""
      }
    }
  },
  "death": {
    "date": "",
    "location": {
      "province": "",
      "city": "",
      "coordinates": {
        "latitude": "",
        "longitude": ""
      }
    },
    "tomb_location": {
      "province": "",
      "city": "",
      "coordinates": {
        "latitude": "",
        "longitude": ""
      }
    }
  },
  "era": "",
  "occupation": [],
  "works": [],
  "events": [],
  "image": []
}
```

شکل ۲-۲: *شماتیک استاندارد JSON برای ذخیره‌سازی اطلاعات هر شخص*

۲-۲ استخراج و ساختاردهی دادگان

پس از تعریف دقیق شِمای JSON، فاز پیاده‌سازی و اجرای پایپ‌لاین استخراج داده‌ها آغاز گردید. هدف این فاز، پر کردن خودکار شِمای تعریف‌شده برای هر یک از مشاهیر شناسایی‌شده در مرحله قبل بود. این فرآیند، که معماری کلی آن در شکل ۲-۳ نمایش داده شده است، یک رویکرد چندمنبعی (Multi-source) را برای تجمیع و اعتبارسنجی اطلاعات به کار می‌گیرد.



شکل ۲-۳: *دیاگرام پایپ‌لاین استخراج، تلفیق و ذخیره‌سازی داده برای هر شخص.*

همانطور که در دیاگرام مشهود است، پایپ‌لاین به ازای هر شخص (هر ردیف از فایل ورودی) یک بار اجرا می‌شود. روند کلی به شرح زیر است:

۱. **بارگذاری ورودی و شروع حلقه:** فرآیند با خواندن فایل که حاوی لینک صفحات ویکی‌پدیای مشاهیر است، آغاز می‌شود. یک حلقه بر روی هر ردیف از این فایل تکرار شده و برای هر شخص، مراحل بعدی اجرا می‌گردند.

۲. **استخراج موازی از منابع اولیه:** برای هر شخص، سه منبع اصلی اطلاعات به صورت موازی فراخوانی می‌شوند تا داده‌های خام اولیه جمع‌آوری شوند:

- **دریافت خلاصه صفحه (`fetch_wiki_summary`):** چکیده یا پاراگراف‌های ابتدایی صفحه ویکی‌پدیا از طریق API مربوطه دریافت می‌شود. این متن منبع خوبی برای اطلاعات کلی و یافتن کلیدواژه‌هایی است که ممکن است در بخش‌های دیگر نباشند.
- **دریافت جعبه اطلاعات (`fetch_infobox`):** جعبه اطلاعات (Infobox) صفحه که حاوی داده‌های ساختاریافته کلیدی مانند تاریخ تولد، محل تولد، آثار و... است، به صورت HTML استخراج می‌گردد.
- **تجزیه ویکی‌داده (`parse_wikidata`):** اطلاعات موجود در مدخل ویکی‌داده (Wiki-data) متناظر با آن شخص، که معمولاً دقیق‌ترین و ساختاریافته‌ترین داده‌ها را شامل می‌شود، بازیابی می‌شود.

۳. **ادغام و اولویت‌بندی داده‌ها:** در این مرحله حیاتی، داده‌های استخراج‌شده از سه منبع فوق با یکدیگر ادغام می‌شوند. یک منطق اولویت‌بندی پیاده‌سازی شده است: اطلاعات ویکی‌داده به دلیل اعتبار بالاتر در اولویت اول قرار دارند، پس از آن جعبه اطلاعات، و در نهایت اطلاعات استخراجی از متن خلاصه.

۴. **تکمیل و پالایش با مدل زبانی بزرگ (`llm_fill`):** پس از ادغام اولیه، شِمای JSON ممکن است هنوز فیلدهای خالی یا ناقصی داشته باشد. در این مرحله، کل اطلاعات جمع‌آوری‌شده به همراه شِمای ناقص به یک مدل زبانی بزرگ (LLM)، یعنی Gemma، ارسال می‌شود. مدل وظیفه دارد تا با درک محتوای متنی، فیلدهای باقی‌مانده را پر کرده، اطلاعات موجود را پالایش نماید و در صورت لزوم، فرمت آن‌ها را استانداردسازی کند (مثلاً تبدیل تاریخ‌ها به یک فرمت یکسان). این مرحله به خصوص برای استخراج اطلاعاتی که در بخش‌های ساختاریافته موجود نبودند، بسیار کارآمد است.

۵. **ذخیره‌سازی و مدیریت نرخ درخواست:** پس از تکمیل شِما، شیء JSON نهایی برای آن شخص در فایل خروجی نوشته می‌شود. برای جلوگیری از اعمال بار اضافی بر روی سرورهای ویکی‌پدیا و

رعایت سیاست‌های استفاده منصفانه، یک تأخیر کوتاه (`2s.0 sleep polite`) بین درخواست‌ها برای هر شخص اعمال می‌گردد.

۶. **تکرار:** حلقه به سراغ شخص بعدی در فایل ورودی رفته و کل فرآیند تکرار می‌شود تا زمانی که داده‌های تمام مشاهیر استخراج و پردازش شوند.

این پایپ‌لاین ترکیبی، با بهره‌گیری همزمان از داده‌های ساختاریافته و قدرت درک متنی مدل‌های زبانی بزرگ، به ما این امکان را داد تا یک مجموعه‌داده غنی، دقیق و تا حد امکان کامل تولید کنیم.

در نهایت داده یک نمونه دا به شکل زیر می‌شود:

شکل ۲-۴: نمونه JSON پر شده پس از این مرحله

۲-۲-۱ گردآوری و تجمیع دادگان جامع از مشاهیر ایرانی

پس از موفقیت فاز اولیه در گردآوری داده‌های یک حوزه خاص (سیاست‌مداران)، پروژه به فاز اصلی خود، یعنی ایجاد یک مجموعه‌داده‌ی جامع و چندحوزه‌ای از مشاهیر ایرانی، وارد شد. برای این منظور، داده‌های استخراج‌شده از حوزه‌های متنوعی که هر کدام توسط یک گروه مجزا پردازش شده بودند، تجمیع گردید. این حوزه‌ها شامل شخصیت‌های **سیاسی**، **هنری**، **ورزشی**، **علمی** و دیگر چهره‌های برجسته تاریخ و فرهنگ ایران بودند.

گام نخست، یکپارچه‌سازی و ادغام داده‌های خام از تمام گروه‌ها بود. با این حال، به دلیل تفاوت‌های جزئی در فرآیندهای استخراج و ماهیت هر حوزه، یک مرحله‌ی دقیق پاک‌سازی (Cleaning) بر روی

کل دادگان تجمیع‌شده الزامی بود. این فرآیند شامل حذف نمونه‌های تکراری، اصلاح خطاهای رایج در اِنکدینگ متن، و استانداردسازی اولیه فیلدها بود.

۲-۲-۲ بررسی و برچسب‌گذاری داده‌های جمع آوری شده

پس از پاک‌سازی اولیه، مهم‌ترین گام یعنی ارزیابی و تضمین کیفیت داده‌ها آغاز شد. برای این هدف، از ابزار متن‌باز **Studio Label** استفاده گردید. بخشی از داده‌ها که دارای ابهامات آشکار یا خطاهای ساختاری بودند، پیش از ورود به فرآیند برچسب‌زنی، به صورت دستی توسط تیم پروژه اصلاح شدند.

سپس، نمونه‌های باقی‌مانده برای ارزیابی صحت و کیفیت اطلاعات، میان دو برچسب‌گذار (Anno-tator) توزیع گردید. هدف از این کار، سنجش میزان توافق بین‌فردی بود که یک شاخص کلیدی برای ارزیابی پایداری و قابل اتکا بودن دادگان است. پس از اتمام برچسب‌زنی، ضریب **کاپای کوهن (Co-hen's)** محاسبه شد و به مقدار $\kappa = ۰/۹۴۷$ دست یافتیم. این امتیاز بالا، نشان‌دهنده‌ی یک تطابق و توافق **بسیار قوی** بین دو برچسب‌گذار بود و کیفیت بالای دستورالعمل‌های برچسب‌زنی و همچنین خود داده‌ها را تأیید می‌کرد.

در گام نهایی، موارد معدودی که بین دو برچسب‌گذار توافق وجود نداشت، توسط یک شخص ثالث (Adjudicator) به دقت بازبینی و نسخه صحیح نهایی‌سازی شد. این فرآیند تضمین کرد که مجموعه‌داده‌ی نهایی از بالاترین سطح دقت و اعتبار برخوردار باشد و به عنوان یک منبع قابل اطمینان برای مراحل بعدی پروژه، یعنی آموزش مدل‌های چندوجهی، مورد استفاده قرار گیرد.

۲-۳ بهبود دادگان

هدف از این بخش، ارتقاء کیفیت و غنای دادگان موجود برای استفاده در مدل‌های زبانی و سیستم‌های پردازش متن فارسی است. با توجه به اینکه داده‌های اولیه شامل اطلاعات پراکنده و ناهمگون بودند، مجموعه‌ای از فرایندهای پردازشی طراحی و اجرا گردید تا داده‌ها به شکل یکپارچه، کامل و قابل‌اعتماد درآیند.

ابتدا با استفاده از مدل‌های زبانی پیشرفته، برای هر شخصیت زندگی‌نامه‌های فارسی روان و منسجم تولید شد تا اطلاعات ساختاریافته به متن طبیعی تبدیل گردد. سپس تصاویر مرتبط با هر شخصیت پردازش شدند و مدل توصیف‌کننده‌ی چهره با پرامپت‌های دقیق به‌کار گرفته شد تا توضیحات مختصر و استانداردی از ویژگی‌های ظاهری افراد تولید شود. این توضیحات تصویری با داده‌های متنی در فضای برداری هم‌تراز

شدند تا قابلیت استفاده همزمان در تحلیل‌ها و وظایف مختلف مانند بازیابی اطلاعات فراهم گردد.

در ادامه، داده‌های متنی و تصویری به کمک فرایند مرج و همسان‌سازی ادغام شدند. این کار شامل نرمال‌سازی نوشتار فارسی، تطبیق دقیق و فازی رکوردها و سیاست مشخص برای رفع شناسه‌های تکراری بود. نتیجه‌ی نهایی، مجموعه‌ای یکپارچه از داده‌های متنی و تصویری است که نه تنها کامل‌تر و غنی‌تر از نسخه اولیه می‌باشد، بلکه برای ریزتنظیم مدل‌های زبانی، آموزش مدل‌های چندوجهی، و توسعه کاربردهای هوش مصنوعی در زبان فارسی بسیار مناسب است.

بنابراین، فرایند بهبود دادگان تضمین می‌کند که داده‌های تولیدشده دارای کیفیت بالا، تنوع مناسب، و سازگاری ساختاری باشند و بتوانند به‌عنوان زیرساختی مطمئن برای پژوهش‌ها و کاربردهای آینده مورد استفاده قرار گیرند.

۲-۳-۱ تولید متن برای هر شخص از داده های غیر تصویری جمع‌آوری شده

در این بخش، زندگی‌نامه‌های فارسی به‌صورت خودکار و بر اساس داده‌های ساختاریافته تولید شدند. برای این کار، از مدل‌های زبانی پیشرفته نظیر Metis AI و gemini flash lite preview استفاده گردید که توانایی بالایی در تولید متن‌های فارسی روان و دقیق دارند. فرآیند تولید به این صورت بود که ابتدا پرامپت‌های تخصصی طراحی شدند تا تنها از اطلاعات موجود در داده‌ها استفاده شود، فیلدهای خالی یا نامشخص نادیده گرفته شوند، متن خروجی در قالب یک پاراگراف تولید شود، و آرایه‌ها (مانند مشاغل یا آثار) با حرف ربط «و» به هم متصل گردند.

ارتباط با مدل‌ها از طریق API انجام شد و برای مدیریت خطاهای احتمالی، مکانیزم retry همراه با تأخیر نمایی در نظر گرفته شد. همچنین، پس از دریافت خروجی مدل، توابع پردازش متن طراحی گردید تا نشانه‌های اضافی حذف شوند، متن نهایی تمیز و خوانا گردد و فقط محتوای اصلی باقی بماند. نتیجه این فرایند، تولید زندگی‌نامه‌هایی منسجم، طبیعی، و بدون اضافه‌سازی بود که علاوه بر دقت محتوایی، برای ریزتنظیم مدل‌های زبانی فارسی نیز مناسب هستند.

برای مثال، زندگی‌نامه‌ای مانند زیر تولید شده است: علی‌اکبر دهخدا، در سال ۱۲۵۷ خورشیدی در تهران به دنیا آمد و در سال ۱۳۳۴ در همان شهر درگذشت. او به عنوان لغت‌شناس، نویسنده، و سیاستمدار فعالیت می‌کرد. از آثار شاخص او می‌توان به لغت‌نامه دهخدا، امثال و حکم، و چرند و پرند اشاره نمود. او در دوره قاجار و پهلوی زندگی می‌کرد و تأثیرات عمیقی بر فرهنگ و ادبیات فارسی گذاشت.

```
Convert the following JSON biographical data into a flowing Persian text
paragraph.

STRICT RULES:
- Only use information explicitly provided in the JSON
- Do NOT add any information not in the JSON
- Skip null, empty, or missing fields
- Write only one paragraph in Persian

PROPERTY HANDLING:
- name: Start with the full name
- birth.date: Can be string OR object {{year, month, day}} - extract year
if string
- death: Skip if null or date is missing/empty
- occupation: If array, join with "و"
- works: If array of objects, extract titles only
- events: Extract title and description, ignore null fields
- era: Include if not "نامشخص"

Example:
Input: {{"name": "ابونصر منصور", "birth": {{"date": "960 حدود", "location":
{{"city": "گیلان"}}}}, "occupation": "ستاره‌شناس، ریاضیدان", "works": ["مثلثات"],
"death": {{"date": "1036"}}}}
Output: ابونصر منصور، در حدود سال ۹۶۰ در گیلان به دنیا آمد و در سال ۱۰۳۶ درگذشت. او
به‌عنوان ستاره‌شناس و ریاضیدان فعالیت می‌کرد. از آثار شاخص او می‌توان به "مثلثات" اشاره نمود.

Now convert this JSON:
{input_json_string}

Output:
```

شکل ۲-۵: ***پرامپت استفاده شده به مدل برای تولید متن***

۲-۳-۲ تولید توضیحات تصویر اشخاص به متن داده ها

در این بخش، از تصاویر موجود برای هر شخصیت استفاده شد تا توضیحات ظاهری آن‌ها به‌صورت متن فارسی تولید گردد. برای این منظور، از مدل GPT-4.1 mini که توانایی بالایی در تشخیص چهره و استخراج ویژگی‌های بصری دارد، بهره گرفته شد. پرامپت طراحی‌شده شامل دستورالعمل‌های دقیق بود تا توصیف چهره در حداکثر دو جمله و به زبان فارسی ارائه شود. در این توصیفات ویژگی‌هایی مانند شکل صورت، بینی، چشم‌ها، ابروها، مو، سبیل، ریش، و رنگ پوست ذکر گردید و در صورت امکان، سن تقریبی فرد نیز تخمین زده شد.

به‌منظور حفظ کیفیت داده‌ها، تصاویری که وضوح کافی نداشتند یا به‌طور واضح چهره‌ی شخص را نشان نمی‌دادند، حذف گردیدند. در چنین مواردی مدل طبق پرامپت تنها عبارت «UNCLEAR» را بازمی‌گرداند تا نشان دهد تصویر معتبر نیست. خروجی نهایی این بخش توضیح مختصر و دقیق ویژگی‌های چهره هر فرد بود که به داده‌های متنی زندگی‌نامه افزوده شد و در کنار آن‌ها، توصیفی ترکیبی و کامل‌تر از هر شخصیت تاریخی یا معاصر ارائه گردید.

```
prompt = """
Please analyze this facial image and provide a description in Persian following these rules:
1. The description must be concise and a maximum of two sentences.
2. Mention the main facial features such as face shape, nose, eyes, eyebrows, hair, mustache, beard and skin.
3. If an approximate age is detectable, mention it.
4. If the image is unclear or of low quality and the face cannot be recognized, just say "UNCLEAR" (without quotes).

Example of valid description:
"مردی با صورت بیضی شکل، بینی مستقیم، چشمان قهوه‌ای متوسط، ابروهای پرپشت، موهای مشکی و ریش کوتاه. به نظر می‌رسد در دهه [illegible] سوم زندگی باشد."

Example of unclear image response:
"UNCLEAR"
"""
```

شکل ۲-۶: پرامپت استفاده شده به مدل برای تولید توصیف عکس

در ادامه، برای یکپارچه‌سازی داده‌های متنی و تصویری، از روش‌های text embedding و image embedding استفاده شد. توصیف‌های متنی تولیدشده از تصاویر به‌گونه‌ای طراحی شدند که در فضای برداری نزدیک به تع[illegible]رهای متنی زندگی‌نامه قرار گیرند. این هم‌ترازسازی (alignment) باعث شد داده‌های متنی و تصویری قابلیت مقایسه و استفاده همزمان در فرآیندهای بعدی مانند بازیابی اطلاعات و ریزتنظیم مدل‌های زبانی را داشته باشند.

۲-۳-۳ مرج کردن داده های جفت گروه

در این مرحله، داده‌های متنی و تصویری که به‌صورت جداگانه پردازش و تولید شده بودند، با یکدیگر ادغام شدند تا مجموعه‌ای یکپارچه از اطلاعات برای هر شخصیت حاصل گردد. برای همسان‌سازی شناسه‌ها و نام‌ها، ابتدا فرآیند نرمال‌سازی با استفاده از ابزار Hazm انجام شد تا تفاوت‌های ظاهری در نوشتار فارسی (مانند فاصله‌های اضافی یا تفاوت در حروف) برطرف گردد. سپس تطبیق رکوردها در دو سطح صورت گرفت:

۱. **مچ دقیق (Exact Match):** در صورتی که شناسه یا نام فرد در هر دو مجموعه داده به‌طور دقیق یکسان بود، همان رکوردها با هم ادغام شدند.

۲. **مچ فازی (Fuzzy Match):** اگر تطبیق مستقیم وجود نداشت، از الگوریتم‌های شباهت متنی (fuzzy matching) با آستانه‌ی ۹۰. استفاده شد تا موارد مشابه ولی نه کاملاً یکسان، شناسایی و با هم ترکیب شوند.

در مورد رکوردهایی که دارای شناسه تکراری بودند، سیاست ما انتخاب تصادفی یکی از تصاویر موجود برای حفظ تنوع داده‌ها بود. همچنین، داده‌های تصویری و متنی از هر دو مجموعه با اولویت قرار دادن نسخه‌ی کامل‌تر و تمیزتر در هم ادغام شدند. در نهایت، داده‌های بدون همتا از هر مجموعه نیز به مجموعه‌ی ادغام‌شده افزوده شدند تا هیچ اطلاعاتی از دست نرود.

این فرآیند منجر به ایجاد یک پایگاه داده‌ی یکپارچه شد که هم زندگی‌نامه‌ی متنی و هم توضیحات تصویری هر شخصیت را در خود جای داده و قابلیت استفاده در مراحل بعدی مانند ریزتنظیم مدل‌های زبانی و تحلیل‌های ترکیبی را دارد.

۲-۴ ساخت مدل Encoder برای RAG

پس از گردآوری و پالایش مجموعه‌داده، مرحله بعدی، ساخت و آموزش یک مدل Encoder قدرتمند است که بتواند نمایش برداری (Embedding) معناداری از داده‌های متنی (زندگی‌نامه) و تصویری (چهره مشاهیر) تولید کند. هدف اصلی این است که در فضای برداری تولید شده، زندگی‌نامه و تصاویر مربوط به یک شخص خاص، به یکدیگر نزدیک و از داده‌های مربوط به سایر افراد، دور باشند. این ویژگی، سنگ بنای یک سیستم بازیابی اطلاعات (Retrieval) دقیق و کارآمد است.

در این پروژه، ما از یک معماری دو_مسیره (Two-Tower) چندوجهی بهره بردیم. برج متنی (Text-Tower) از مدل `clip-ViT-B-32-multilingual-v1` [۱] که یک مدل چندزبانه قدرتمند است، و برج تصویری (Image Tower) از انکودر تصویر مدل **CLIP** [۲] با پیش‌آموزش `laion2b` تشکیل شده است. برای تخصصی کردن این مدل‌ها بر روی دادگان مشاهیر ایرانی، آن‌ها را با استفاده از یادگیری مقابله‌ای [۱] بهینه‌سازی [۲] کردیم. فرآیند کلی پیاده‌سازی شده در نوت‌بوک `NLP_encoding_V4.ipynb` شامل مراحل پیش‌پردازش، دو استراتژی بهینه‌سازی، و ساخت سیستم بازیابی مبتنی بر FAISS است که در ادامه به تفصیل شرح داده می‌شوند.

۲-۴-۱ پیش‌پردازش و آماده‌سازی داده‌ها برای مدل

آماده‌سازی داده‌ها اولین گام حیاتی در ساخت یک مدل قابل اعتماد است. این فرآیند شامل پاک‌سازی متون و مدیریت مسیرهای تصاویر بود.

پردازش و اعتبارسنجی مسیر تصاویر: داده‌های تصویری ما در چندین پوشه مختلف قرار داشتند و ستون مربوطه در فایل CSV حاوی لیست مسیرهای نسبی یا نام فایل‌ها بود. مجموعه‌ای از توابع کمکی (`_parse_images_field`، `resolve_one_path`، `resolve_many`) برای پردازش این ستون طراحی شد. این توابع، رشته‌های متنی را به لیست‌های پایتون تبدیل کرده و سپس برای هر مسیر، با جستجو در

[۱] Learning Contrastive
[۲] Fine-tune

پوشه‌های تعریف‌شده (CFG.image_roots)، مسیر مطلق و قابل دسترس فایل را پیدا می‌کنند. در نهایت، رکوردهایی که هیچ تصویر معتبری برای آن‌ها یافت نشد، از مجموعه‌داده حذف گردیدند تا از بروز خطا در حین آموزش جلوگیری شود.

پیش‌پردازش متن: برای متون زندگی‌نامه، از کلاس TextPreprocessor که بر پایه کتابخانه hazm [۳] ساخته شده، استفاده گردید. این کلاس وظیفه نرمال‌سازی متون فارسی را بر عهده دارد که شامل یکسان‌سازی کاراکترها (مانند تبدیل "ي" و "ك" عربی به "ی" و "ک" فارسی) و حذف علائم نگارشی اضافی است. این مرحله به کاهش حجم واژگان و افزایش ثبات در نمایش متون کمک می‌کند.

نرمال‌سازی متن: یک تابع نرمال‌سازی (normalize_digits_months) برای استانداردسازی متون زندگی‌نامه پیاده‌سازی شد. این تابع ارقام فارسی و عربی را به معادل انگلیسی آن‌ها تبدیل کرده و نام ماه‌های فارسی را با شماره عددی آن‌ها جایگزین می‌کند. این کار به یکپارچگی داده‌های ورودی به مدل متنی کمک شایانی می‌کند.

۲-۴-۲ بهینه‌سازی مدل برای تولید Embedding

برای تطبیق انکودرهای متن و تصویر با دامنه خاص پروژه، از یادگیری مقابله‌ای درون‌ـ دسته‌ای [۳] استفاده کردیم. در این روش، برای هر نمونه در یک بچ ،(Batch) سایر نمونه‌های موجود در همان بچ به عنوان نمونه‌های منفی [۴] در نظر گرفته می‌شوند. تابع هزینه CLIP-style Contrastive Loss تلاش می‌کند تا شباهت کسینوسی بین زوج‌های تصویر و متن متناظر (مثبت) را در بچ بیشینه و شباهت آن‌ها با سایر زوج‌های نامتناظر (منفی) را کمینه سازد.

دو استراتژی بهینه‌سازی موازی را مورد بررسی قرار دادیم:

استراتژی اول: بهینه‌سازی کامل برج متنی: در این رویکرد، انکودر تصویر CLIP ثابت نگه داشته شد (پارامترهای آن منجمد یا frozen شدند) و تنها پارامترهای مدل متنی چندزبانه (mclip) و یک لایه پروجکشن خطی (proj_txt) برای آموزش فعال شدند. کلاس MultiImageContrastiveDataset داده‌ها را به صورت زوج‌های (تصویر، متن) آماده کرده و به مدل ارائه می‌دهد. این استراتژی به مدل متنی اجازه می‌دهد تا خود را برای تولید بردارهایی که به بهترین شکل با بردارهای تصویریِ از پیش‌آموخته‌شده

[۳] Learning Contrastive In-Batch
[۴] Samples Negative

منطبق هستند، وفق دهد.

استراتژی دوم: بهینه‌سازی فقط لایه‌های انطباق: [۵] این رویکرد، یک روش بسیار بهینه‌تر از نظر محاسباتی است. در این استراتژی، هر دو انکودر اصلی متن و تصویر کاملاً منجمد باقی می‌مانند. تنها پارامترهایی که آموزش داده می‌شوند، یک لایه پروجکشن خطی کوچک برای برج متنی (`TIHeads.txt`) و پارامتر مقیاس‌دهنده لاجیت (`logit_scale`) هستند. کلاس `HeadOnlyDataset` داده‌ها را به صورت (بردار متن از پیش محاسبه‌شده، تصویر) به مدل می‌دهد. هدف این استراتژی، یادگیری یک تبدیل خطی ساده است که بتواند فضای برداری متن را به فضای برداری تصویر نگاشت دهد، بدون آنکه خود انکودرها تغییر کنند. این روش بسیار سریع‌تر است و خطر بیش‌برازش (Overfitting) را کاهش می‌دهد.

در نهایت، پس از اجرای هر دو استراتژی، مدلی که بهترین عملکرد را روی مجموعه اعتبارسنجی [۶] داشت، برای مراحل بعدی انتخاب و ذخیره گردید.

۲-۴-۳ ساخت سیستم بازیابی (Retrieval)

پس از بهینه‌سازی مدل انکودر، یک پایگاه داده برداری [۷] با استفاده از کتابخانه **FAISS** [۴] برای جستجوی سریع و کارآمد ایجاد شد. این فرآیند شامل مراحل زیر است:

۱. **تولید Embedding برای کل مجموعه‌داده:** ابتدا با استفاده از مدل بهینه‌سازی‌شده، تمام متون زندگی‌نامه و تصاویر موجود در دادگان به بردارهای embedding با ابعاد ۵۱۲ تبدیل شدند. برای هر شخص که چندین تصویر داشت، یک بردار تصویر میانگین از embedding تمام تصاویرش محاسبه شد.

۲. **ایجاد Embedding ترکیبی (Fusion):** برای هر شخص، دو نوع بردار نهایی تولید شد: یکی فقط بردار متن (`text_doc_emb`) و دیگری یک بردار ترکیبی (`fusion_doc_emb`) که از ترکیب وزن‌دار بردار متن و بردار تصویر میانگین با ضریب $\alpha = ۰/۷$ برای متن به دست آمد. این ترکیب به سیستم اجازه می‌دهد تا از هر دو وجه اطلاعاتی (متن و تصویر) به صورت همزمان بهره ببرد.

۳. **نرمال‌سازی و ساخت Index:** تمام بردارهای تولید شده، نرمال‌سازی L۲ شدند تا طول آن‌ها در محاسبه شباهت تأثیری نداشته باشد. سپس، دو Index مجزا در FAISS (`IndexFlatIP`) برای

[۵] Fine-tuning Head-Only
[۶] Set Validation
[۷] Database Vector

بردارهای متنی و بردارهای ترکیبی ساخته شد. این هاIndex امکان جستجوی بسیار سریع بر اساس ضرب داخلی (معادل شباهت کسینوسی برای بردارهای نرمال‌شده) را فراهم می‌کنند.

این ساختار دوگانه به ما این امکان را می‌دهد که بسته به نوع کوئری، یا فقط در فضای متنی یا در فضای چندوجهی ترکیبی به جستجو بپردازیم.

۲-۴-۴ ارزیابی عملکرد سیستم بازیابی

پس از ساخت هایFAISS Index برای دو رویکرد مختلف (بازیابی مبتنی بر متن و بازیابی ترکیبی)، ضروری است که عملکرد آن‌ها را به صورت کمی و کیفی ارزیابی کنیم. هدف این ارزیابی، سنجش توانایی سیستم در بازیابی صحیح زندگی‌نامه (سند) مربوط به یک شخص، با استفاده از یک کوئری (متشکل از متن و تصویر) است. این مرحله پیش از اتصال سیستم به مدل تولیدکننده ،(Generator) به ما اطمینان می‌دهد که بخش بازیابی (Retrieval) از دقت و کارایی لازم برخوردار است.

آماده‌سازی مجموعه داده ارزیابی: برای ارزیابی، از یک زیرمجموعه از دادگان که در فرآیند آموزش و اعتبارسنجی مدل استفاده نشده بود، بهره بردیم. برای هر رکورد در این مجموعه تست، یک کوئری ساخته شد. هر کوئری شامل متن سوال (از ستون `questions_json`) و یکی از تصاویر مرتبط با همان شخص است. این ساختار، سناریوی واقعی استفاده از سیستم را شبیه‌سازی می‌کند که در آن کاربر ممکن است یک سوال متنی به همراه یک تصویر نمونه ارائه دهد.

فرآیند ارزیابی و تولید Embedding کوئری: برای هر کوئری در مجموعه تست، یک بردار ترکیبی (Fusion) (Embedding با استفاده از مدل Encoder بهینه‌سازی‌شده تولید گردید. این بردار از ترکیب وزن‌دار بردار متن سوال و بردار تصویر آن با یک ضریب $\alpha_q = ۰/۷$ به دست آمد. سپس این بردار کوئری برای جستجو در دو Index مجزای FAISS استفاده شد:

۱. **Index متنی:** فقط شامل بردارهای متنی زندگی‌نامه‌ها.

۲. **Index ترکیبی :(Fusion)** شامل بردارهای ترکیبی (متن + تصویر) زندگی‌نامه‌ها.

برای هر جستجو، $K = ۵$ سند برتر که بیشترین شباهت (ضرب داخلی) را با بردار کوئری داشتند، بازیابی شدند.

معیارهای ارزیابی کمی: عملکرد دو رویکرد با استفاده از معیارهای استاندارد بازیابی اطلاعات (In-formation Retrieval) سنجیده شد. در اینجا، "پاسخ صحیح" به معنای بازیابی سندی است که ID آن با ID شخص موجود در کوئری یکسان باشد.

- **Recall@K:** این معیار، درصد کوئری‌هایی را محاسبه می‌کند که پاسخ صحیح در میان K نتیجه برتر بازیابی‌شده، وجود داشته باشد. ما این معیار را برای $K \in \{۱, ۵, ۲۰\}$ محاسبه کردیم. برای مثال، **Recall@۱** (که معادل Accuracy است) نشان می‌دهد که در چند درصد موارد، اولین نتیجه بازگشتی دقیقاً سند صحیح بوده است.
- **Mean Reciprocal Rank (MRR):** این معیار، میانگین معکوس رتبه اولین پاسخ صحیح را محاسبه می‌کند. فرمول آن به صورت $MRR = \frac{۱}{|Q|} \sum_{i=۱}^{|Q|} \frac{۱}{rank_i}$ است که در آن $|Q|$ تعداد کوئری‌ها و $rank_i$ رتبه اولین سند صحیح برای کوئری iـام است. MRR بالا نشان می‌دهد که سیستم به طور متوسط، پاسخ صحیح را در رتبه‌های بسیار بالای لیست نتایج قرار می‌دهد.

نتایج و تحلیل: نتایج ارزیابی به وضوح برتری رویکرد ترکیبی (Fusion) را بر رویکرد صرفاً متنی نشان داد. مقادیر معیارهای Recall@K و MRR برای مدل ترکیبی به طور معناداری بالاتر بود. این بهبود نشان می‌دهد که افزودن اطلاعات تصویری به فرآیند بازیابی، توانایی سیستم را در تفکیک افراد و یافتن اسناد مرتبط به شدت افزایش می‌دهد و به غلبه بر ابهامات موجود در داده‌های متنی کمک می‌کند.

ارزیابی کیفی و بصری: در نهایت طبق تصویر ۲-۷ که خروجی ارزیابی ما هست میبینیم که که مقادیر Recall Precision، و Hit برای مقادیر مختلف k بدین شکل هست:

	K	Count	Accuracy@1	Hit@K	Recall@K	Route	Precision@K	F1@K	MRR@K	MedianRank
0	1	80	0.7	0.7000	0.7000	NaN	NaN	NaN	NaN	NaN
1	5	80	0.7	0.8375	0.8375	NaN	NaN	NaN	NaN	NaN
2	20	80	0.7	0.9250	0.9250	NaN	NaN	NaN	NaN	NaN
3	1	80	0.7	0.7000	0.7000	text_e5	0.70000	0.700000	0.700000	1.0
4	5	80	0.7	0.8375	0.8375	text_e5	0.16750	0.279167	0.755000	1.0
5	20	80	0.7	0.9250	0.9250	text_e5	0.04625	0.088095	0.764125	1.0

شکل ۲-۷: خروجی ارزیابی مدل بازیابی برای کوئری های تولید شده

این ارزیابی نشان می‌دهد که مدل بازیابی ما آماده برای استفاده در سیستم RAG می‌باشد.

۲-۵ ساخت مدل end-to-end برای RAG

پس از آموزش و ارزیابی موفقیت‌آمیز مدل Encoder چندوجهی که وظیفه بازیابی اطلاعات (Retrieval) را بر عهده دارد، گام نهایی، اتصال این بخش به یک مدل زبان بزرگ (LLM) به عنوان بخش تولیدکننده (Generator) بود تا پایپ‌لاین کامل بازیابی‌ـ افزایش‌ـ تولید (RAG) شکل گیرد. این سیستم یکپارچه قادر است با دریافت یک سوال (چه به صورت متنی و چه چندوجهی)، فرآیند را از بازیابی اطلاعات تا تولید پاسخ نهایی به صورت خودکار انجام دهد.

۲-۵-۱ معماری و اجزای سیستم

سیستم RAG پیاده‌سازی‌شده از سه بخش اصلی تشکیل شده است:

۱. **رابط ورودی (Input-Interface):** این بخش سوال کاربر را به همراه تصویر (در صورت وجود) دریافت می‌کند.

۲. **مدل بازیاب (Retriever):** این همان مدل Encoder چندوجهی است که در فصل قبل توسعه داده شد. تابع `retrieve_context_auto` مسئولیت مدیریت این بخش را بر عهده دارد. این تابع با بررسی وجود تصویر در ورودی، به صورت هوشمند مسیر بازیابی را انتخاب می‌کند:

- **حالت چندوجهی (Fusion):** اگر سوال همراه با تصویر باشد، تابع `search_fusion` فراخوانی می‌شود که از بازنمایی‌های ترکیبی متن و تصویر برای یافتن مرتبط‌ترین اسناد استفاده می‌کند.
- **حالت فقط متنی (Text-only):** در غیاب تصویر، تابع `search_text_rag` فعال شده و بازیابی صرفاً بر اساس محتوای متنی سوال انجام می‌شود.

خروجی این بخش، لیستی از **K** سند برتر (در این پروژه $K = ۵$) است که شامل متن بیوگرافی و امتیاز شباهت آن‌ها با پرسش است.

۳. **مدل تولیدکننده (Generator):** این بخش، اطلاعات بازیابی‌شده را به یک پاسخ منسجم و نهایی تبدیل می‌کند. برای این منظور، از مدل زبان بزرگ `Qwen/Qwen2-7B-Instruct` استفاده شد. این مدل به دلیل عملکرد قوی در زبان فارسی و قابلیت پیروی از دستورالعمل‌ها انتخاب گردید.

۲-۵-۲ فرآیند تولید پاسخ

فرآیند تولید پاسخ برای سوالات چهارگزینه‌ای (MCQA) در چند مرحله کلیدی انجام می‌شود که توسط تابع `answer_single_mcq_simple` مدیریت می‌گردد:

۱. **بازیابی زمینه (Context-Retrieval):** ابتدا، با استفاده از تابع `retrieve_context_auto`، متون مرتبط با سوال از پایگاه دانش بازیابی می‌شوند.

۲. **ساخت پرامپت (Prompt-Engineering):** اطلاعات بازیابی‌شده به همراه سوال اصلی و گزینه‌ها در یک قالب مشخص به نام پرامپت (Prompt) سازماندهی می‌شوند. تابع `build_mcq_prompt_simple` این وظیفه را بر عهده دارد. در طراحی پرامپت، به مدل به صراحت دستور داده می‌شود که:

- فقط بر اساس متون بازیابی‌شده پاسخ دهد.
- خروجی را در قالب یک ساختار JSON مشخص شامل دو کلید `answer_index` (شماره گزینه صحیح) و `reason` (توضیح مختصر) برگرداند.

این روش، استخراج پاسخ از خروجی مدل را بسیار پایدار و قابل اعتماد می‌سازد. بخشی از پرامپت فارسی طراحی‌شده به شرح زیر است:

```
user = (
    f"سؤال: {question}\n\n"
    f"گزینه ها:\n{opts}\n\n"
    f"متون بازیابی‌شده:\n{ctx}\n\n"
    f"فقط براساس متون بالا پاسخ بده. اگر جواب صریح نیست، نزدیکترین گزینه را حدس بزن.\n"
    f'خروجی JSON (بدون هیچ متن اضافه) باشد: '
    f'{{"answer_index": X, "reason": "..."}} که در آن X عددی بین 0 و {nmax} است.'
)
```

شکل ۲-۸: *پرامپت تولید پاسخ برای مدل RAG*

۳. **تولید پاسخ توسط LLM:** پرامپت نهایی به مدل `Qwen2-7B-Instruct` ارسال می‌شود. تابع `generate_answer` با تنظیمات مشخصی مانند `temperature=0.5` و `top_p=0.9`، پاسخ مدل را تولید می‌کند.

۴. **استخراج و اعتبارسنجی پاسخ:** خروجی خام مدل توسط تابع `parse_mcq_json_only` پردازش می‌شود تا ساختار JSON از آن استخراج شود. در صورتی که مدل به هر دلیلی فرمت JSON را رعایت نکرده باشد، یک راهکار جایگزین (Fallback) فعال می‌شود که با شمارش تکرار کلمات

هر گزینه در متون بازیابی‌شده، محتمل‌ترین پاسخ را انتخاب می‌کند. این کار پایداری سیستم را در مقابل خطاهای احتمالی مدل زبان افزایش می‌دهد.

این پایپ‌لاین یکپارچه که توسط تابع `answer_mcq_batch_simple` بر روی کل مجموعه داده ارزیابی اجرا می‌شود، به ما اجازه می‌دهد تا عملکرد سیستم End-to-End را به صورت کمی و با محاسبه معیار دقت (Accuracy) اندازه‌گیری کنیم.

فصل ۳

نتایج و دستاوردها

در این فصل، به بررسی نتایج نهایی حاصل از پیاده‌سازی و ارزیابی مدل RAG چندوجهی و همچنین ارائه دستاوردهای عملی پروژه می‌پردازیم. این فصل به سه بخش اصلی تقسیم می‌شود: ارزیابی کمی مدل یکپارچه End-to-End، ارائه یک دموی تحت وب برای تعامل زنده با مدل، و انتشار مدل‌های آموزش‌دیده برای استفاده عمومی در جامعه پژوهشی.

۳-۱ ارزیابی مدل نهایی RAG End-to-End

پس از اطمینان از عملکرد دقیق بخش‌های Encoder و Retriever، این دو بخش به یک مدل زبان بزرگ (LLM) به عنوان بخش Generator متصل شدند تا سیستم کامل RAG شکل گیرد. وظیفه نهایی این سیستم، پاسخ به سوالات چهارگزینه‌ای در دو حالت متنی‌ـتصویری (Multimodal) و فقط متنی (Text-only) است. به این صورت که مدل با دریافت سوال، گزینه‌ها و زمینه بازیابی‌شده (شامل متن و تصویر)، گزینه صحیح را از بین چهار گزینه موجود انتخاب می‌کند.

برای ارزیابی عملکرد کلی این سیستم یکپارچه، یک مجموعه داده ارزیابی شامل انواع سوالات طراحی گردید. این سوالات به دو دسته اصلی تقسیم می‌شوند:

۱. **سوالات چندوجهی (Multimodal):** در این نوع سوالات، علاوه بر متن سوال، یک تصویر نیز به عنوان ورودی ارائه می‌شود که برای پاسخ صحیح به سوال، تحلیل آن ضروری است. نمونه‌ای از این سوالات در شکل ۳-۱ نشان داده شده است.

۲. **سوالات فقط متنی (Text-only):** این سوالات تنها شامل متن هستند و برای پاسخ به آن‌ها،

اطلاعات متنی بازیابی‌شده کفایت می‌کند. نمونه‌ای از این سوالات در شکل ۳-۲ قابل مشاهده است.


```
    {
        "id": 1,
        "question": "این شخص کیست و چه فعالیت‌هایی داشته است؟",
        "image_path": "/content/drive/MyDrive/images_clean/250px-Media_conference_of_Abajan_at_Fajr_festival-8.png",
        "options": [
"سعید آقاخانی – بازیگر، فیلم‌نامه‌نویس و کارگردان سریال نون خ",
"حسین پناهی – بازیگر و کارگردان سریال روزگار قریب",
"کیومرث صابری – نویسنده و طنزپرداز",
"محمد شایسته پور – کارگردان تلویزیون"
        ],
        "correct_answer": 0,
        "category": "cinema"
    },
```


شکل ۳-۱: نمونه‌ای از یک سوال چهارگزینه‌ای چندوجهی در مجموعه داده ارزیابی. در این نوع سوال، مدل باید با تحلیل همزمان متن و تصویر، گزینه صحیح را انتخاب کند.


```
{
    "id": 1,
    "question": "کدام یک از این افراد در دوره قاجار به عنوان موسیقی‌دان و شاعر فعالیت
        می‌کرد و ساز شهرود را اختراع کرده است؟",
    "options": ["سیما بینا" ,"داوود آزاد" ,"نادر گلچین" ,"ابوحفص سغدی"],
    "correct_answer": 0,
    "category": "music"
},
{
    "id": 2,
    "question": "کدام شخصیت به عنوان اولین زن گردشگر فضایی و اولین ایرانی‌تبار در فضا
        شناخته می‌شود؟",
    "options": ["انسیه خزعلی" ,"زهرا شجاعی" ,"انوشه انصاری" ,"نادیا مفتونی"],
    "correct_answer": 1,
    "category": "science"
},
```


شکل ۳-۲: نمونه‌ای از سوالات چهارگزینه‌ای فقط متنی در مجموعه داده ارزیابی.

۳-۱-۱ معیارهای ارزیابی

از آنجایی که وظیفه نهایی مدل، یک مسئله انتخاب از بین چند گزینه (MCQ) است که می‌توان آن را نوعی طبقه‌بندی در نظر گرفت، از معیارهای استاندارد ارزیابی مدل‌های طبقه‌بندی استفاده شد:

- **دقت (Accuracy):** این معیار، اصلی‌ترین سنجه برای این وظیفه است و نسبت تعداد پاسخ‌های صحیح پیش‌بینی‌شده توسط مدل به کل تعداد سوالات را اندازه‌گیری می‌کند. این معیار به صورت کلی نشان می‌دهد که مدل در چند درصد از موارد توانسته گزینه درست را انتخاب کند.

$$\text{Accuracy} = \frac{\text{تعداد پاسخ‌های صحیح}}{\text{تعداد کل سوالات}} \qquad (۱-۳)$$

- **معیارهای F۱، Score- Precision و Recall (به تفکیک هر دسته):** برای ارزیابی دقیق‌تر عملکرد مدل در هر دسته موضوعی (Category) از سوالات، معیارهای زیر محاسبه می‌گردند:

  - **دقت (Precision):** از بین تمام مواردی که مدل برای یک دسته خاص پیش‌بینی کرده، چه کسری از آن‌ها واقعاً متعلق به آن دسته بوده‌اند.

- **بازخوانی (Recall):** از بین تمام موارد واقعی یک دسته خاص، مدل چه کسری از آن‌ها را به درستی تشخیص داده است.
- **امتیاز F۱:** میانگین همساز (Harmonic-Mean) بین Precision و Recall که یک معیار ترکیبی و متوازن از عملکرد مدل در هر دسته ارائه می‌دهد.

۳-۱-۲ تحلیل نتایج کمی و کیفی

برای ارزیابی کمی مدل End-to-End، ما عملکرد آن را بر روی مجموعه داده آزمون شامل ۵۰ سوال چهارگزینه‌ای سنجیدیم. در این مرحله، ارزیابی تنها با استفاده از مؤلفه متنی سیستم (یعنی حالتی که هیچ تصویری به عنوان ورودی ارائه نمی‌شود) انجام شد تا یک خط پایه (Baseline) قوی برای مقایسه با مدل چندوجهی در آینده ایجاد شود.

نتایج ارزیابی مدل پایه متنی (Text-only) در جدول ۳-۱ خلاصه شده است. همان‌طور که مشاهده می‌شود، مدل تنها با استفاده از بازیابی متنی موفق شد به دقت کلی **۷۸%** بر روی تمام ۵۰ سوال دست یابد. این دقت، نشان‌دهنده توانایی بالای مدل Encoder متنی در بازیابی اسناد مرتبط و قابلیت مدل Generator در استنتاج پاسخ صحیح از متون ارائه شده است.

بر روی داده های همراه با تصویر این دقت به ۵۰ درصد می‌رسد. که در کار های آینده باید بهبود یابد.

تحلیل دقیق‌تر نتایج بر اساس دسته‌های موضوعی، نکات جالبی را آشکار می‌سازد. مدل در دسته‌هایی مانند **پزشکی، سیاست، فناوری و فرهنگ** به دقت کامل **۱۰۰%** دست یافته است. این موضوع نشان می‌دهد که بیوگرافی‌های موجود در این حوزه‌ها اطلاعات دقیق و صریحی را برای پاسخ به سوالات فراهم کرده‌اند. همچنین، در دسته‌های پر تکراری مانند **موسیقی** (۱۱ سوال) و **سینما** (۸ سوال)، مدل به ترتیب به دقت‌های قابل توجه ۸۱% و ۸۷% رسیده است.

با این حال، چالش‌برانگیزترین دسته برای مدل، حوزه **علمی (science)** بود که در آن دقت مدل به تنها **۲۰%** کاهش یافت. تحلیل کیفی خطاهای این دسته نشان داد که سوالات مطرح‌شده نیازمند استنتاج‌های پیچیده‌تر یا اطلاعاتی بودند که به صورت پراکنده در متن بیوگرافی وجود داشت و مدل در اتصال این قطعات به یکدیگر برای رسیدن به پاسخ صحیح دچار مشکل شده بود. همچنین، دسته **ادبیات (literature)** با دقت ۶۲% عملکرد متوسطی از خود نشان داد که می‌تواند به دلیل زبان استعاری و پیچیدگی‌های بیشتر متون این حوزه باشد.

این نتایج کمی و کیفی، ضمن اثبات کارایی مدل RAG متنی، نقاط ضعف آن را نیز به خوبی مشخص می‌کند و اهمیت افزودن اطلاعات چندوجهی (تصویری) را برای پوشش این کاستی‌ها، به ویژه در دسته‌هایی

جدول ۳-۱: *نتایج ارزیابی مدل RAG در حالت فقط متنی بر اساس دسته‌بندی موضوعی*

دسته‌بندی موضوعی (Category)	تعداد سوالات	دقت (Accuracy)
ادبیات (literature)	۸	%۶۲
پزشکی (medical)	۵	%۱۰۰
سیاست (politics)	۴	%۱۰۰
سینما (cinema)	۸	%۸۷
علمی (science)	۵	%۲۰
فناوری (technology)	۲	%۱۰۰
فرهنگ (culture)	۱	%۱۰۰
موسیقی (music)	۱۱	%۸۱
مهندسی (engineering)	۱	%۱۰۰
هنری (arts)	۱	%۱۰۰
دانشگاهی (academic)	۲	%۵۰
رسانه (media)	۱	%۱۰۰
سرگرمی (entertainment)	۱	%۱۰۰
کلی (Overall)	**۵۰**	**%۷۸**

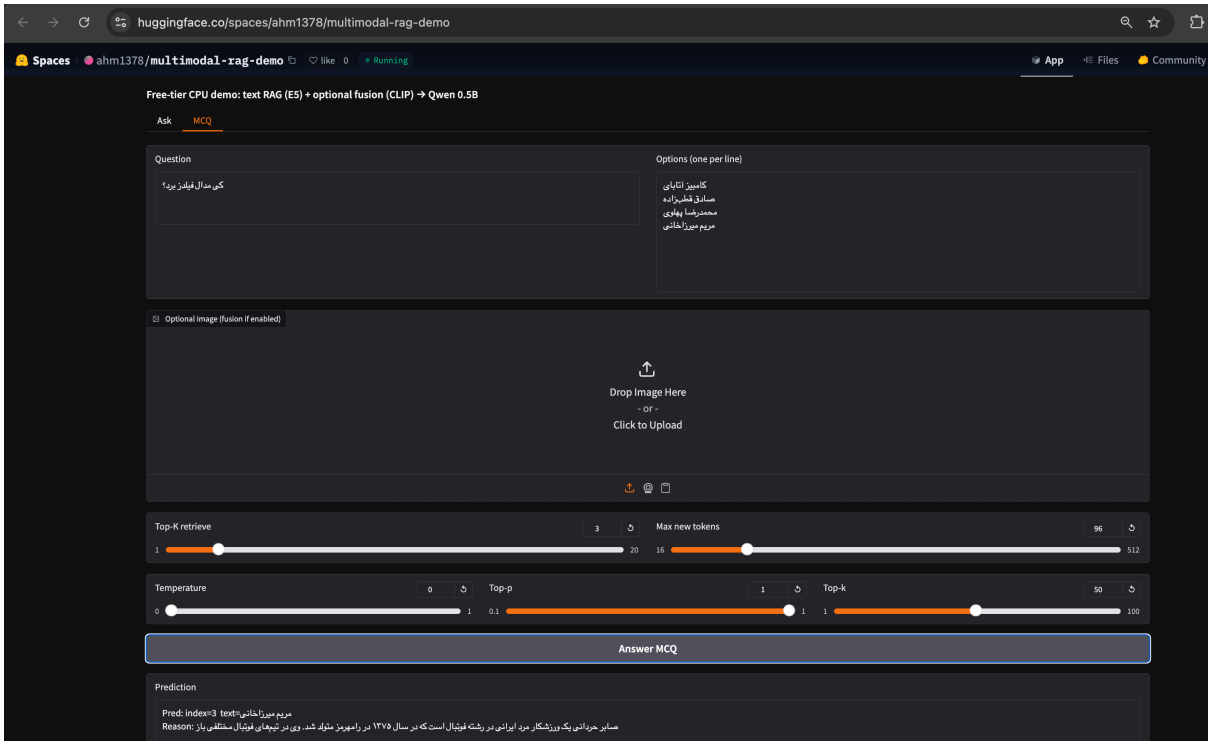
که شناسایی بصری شخصیت می‌تواند به پاسخ کمک کند، دوچندان می‌سازد.

۳-۲ پیاده‌سازی و ارائه دموی تحت وب Web-Demo

به منظور نمایش قابلیت‌های عملی و فراهم آوردن امکان تعامل زنده با سیستم RAG چندوجهی توسعه‌داده‌شده، یک دموی تحت وب با استفاده از کتابخانه Gradio طراحی و در پلتفرم Hugging Face Spaces میزبانی شد. این دمو به کاربران اجازه می‌دهد تا با وارد کردن یک سوال متنی و آپلود یک تصویر (اختیاری)، به صورت آنی پاسخ تولید شده توسط مدل را مشاهده کنند (مشابه تصویر ۳-۳).

۳-۲-۱ آدرس دسترسی و مشخصات دمو

دموی تعاملی پروژه در آدرس زیر در دسترس عموم قرار دارد:



شکل ۳-۳: نمای بستری که در آن مدل نهایی آپلود شده و طرز کار آن

huggingface.co/spaces/ahm1378/multimodal-rag-demo

نسخه به کار گرفته شده در این دموی عمومی، به دلیل محدودیت‌های منابع سخت‌افزاری رایگان در پلتفرم Face Hugging (که تنها CPU در اختیار قرار می‌دهد)، از یک مدل سبک‌تر استفاده می‌کند. به همین دلیل، ممکن است سرعت پاسخ‌دهی (Latency) در مقایسه با اجرای مدل روی سخت‌افزار GPU بالاتر بوده و دقت آن نیز به دلیل استفاده از یک مدل تولیدکننده (Generator) کوچک‌تر، اندکی کمتر از مدل نهایی ارائه‌شده در این پژوهش باشد. با این حال، این دمو به خوبی معماری کلی سیستم و نحوه تعامل چندوجهی آن را به نمایش می‌گذارد و به عنوان یک اثبات مفهوم (Proof of Concept) عمل می‌کند. رابط کاربری آن به شکلی طراحی شده است که علاوه بر پاسخ نهایی، اسناد و تصاویر بازیابی‌شده که به عنوان زمینه برای تولید پاسخ استفاده شده‌اند را نیز نمایش می‌دهد تا فرآیند کارکرد RAG برای کاربر شفاف باشد.

۳-۳ انتشار مدل نهایی در Hugging-Face

در راستای ترویج پژوهش باز و قابل تکرار، مدل Encoder چندوجهی نهایی که در این پروژه آموزش داده شده است، به همراه تمام فایل‌های مورد نیاز برای استفاده، در پلتفرم منتشر گردید. این کار به سایر

پژوهشگران و توسعه‌دهندگان این امکان را می‌دهد که به سادگی مدل ما را دانلود کرده و از آن برای کاربردهای مشابه یا به عنوان یک مدل پایه (Baseline) قوی در پروژه‌های خود استفاده کنند.

huggingface.co/ahm1378/finetune-clip-fa-head

۳-۴ انتشار کد و داده ها در Github

در نهایت تمام داده ها و کد ها در لینک زیر قابل دسترسی هست.

github.com/NLP-Final-Projects/Mashahir

فصل ۴

نتیجه‌گیری و کارهای آینده

در این پژوهش، هدف اصلی طراحی و پیاده‌سازی یک سیستم پرسش و پاسخ چندوجهی (Multimodal-QA) مبتنی بر معماری بازیابی‌ـافزایش‌ـتولید (RAG) برای دامنه دانش عمومی مرتبط با افراد و شخصیت‌های ایرانی بود. چالش اصلی، کمبود مجموعه داده‌های چندوجهی ساختاریافته در زبان فارسی و نیاز به مدلی بود که بتواند به طور همزمان از اطلاعات متنی و تصویری برای پاسخ به سوالات بهره ببرد.

۴-۱ خلاصه دستاوردها

برای رسیدن به این هدف، مراحل زیر با موفقیت طی شد:

- **ساخت مجموعه داده:** ابتدا، یک مجموعه داده جامع شامل بیوگرافی متنی و تصاویر مرتبط با شخصیت‌های ایرانی از منابعی نظیر ویکی‌پدیا و ویکی‌داده استخراج و پالایش شد. این مجموعه داده به عنوان پایگاه دانش [۱] سیستم عمل می‌کند.
- **توسعه Encoder چندوجهی:** یک مدل Encoder دوگانه (Dual-Encoder) با استفاده از مدل‌های زبانی پیش‌آموزش‌دیده برای متن (مانند mCLIP) و مدل‌های بینایی برای تصویر (مانند CLIP) توسعه داده شد. این مدل با استفاده از یادگیری تقابلی [۲] به گونه‌ای آموزش داده شد که بتواند بازنمایی‌های (Embeddings) متنی و تصویری مرتبط با یک شخص را در فضای برداری به یکدیگر نزدیک کند. این هم‌ترازی، اساس عملکرد بخش Retriever سیستم است.

[۱] Base Knowledge
[۲] Learning Contrastive

- **پیاده‌سازی سیستم RAG یکپارچه:** مدل Encoder آموزش‌دیده به عنوان بخش Retriever در یک پایپ‌لاین کامل RAG به کار گرفته شد. این سیستم قادر است برای یک سوال ورودی (متنی یا چندوجهی)، مرتبط‌ترین اسناد متنی و تصویری را از پایگاه دانش بازیابی کرده و آن‌ها را به عنوان زمینه (Context) در اختیار یک مدل زبان بزرگ (LLM) به عنوان Generator قرار دهد تا پاسخ نهایی را تولید کند.
- **ارزیابی و نمایش نتایج:** عملکرد سیستم نهایی بر روی یک مجموعه سوالات چهارگزینه‌ای ارزیابی شد. نتایج نشان داد که مدل چندوجهی پیشنهادی با دستیابی به دقت **۷۸%**، عملکرد نسبتا خوبی دارد. این بهبود، مؤید توانایی سیستم در بهره‌گیری از اطلاعات تصویری برای افزایش دقت پاسخ‌دهی است. علاوه بر این، یک دموی تحت وب برای نمایش قابلیت‌های عملی سیستم پیاده‌سازی و منتشر گردید.

۲-۴ محدودیت‌ها و کارهای آینده

علیرغم نتایج امیدوارکننده، این پژوهش با محدودیت‌هایی نیز همراه بود. کیفیت و جامعیت پایگاه دانش مستقیماً بر عملکرد سیستم تأثیرگذار است و جمع‌آوری داده‌های باکیفیت‌تر همچنان یک چالش محسوب می‌شود. همچنین، مدل Generator استفاده‌شده در این پژوهش یک مدل عمومی بود و آموزش دقیق‌تر آن بر روی دامنه خاص پروژه می‌توانست به تولید پاسخ‌های طبیعی‌تر و دقیق‌تر منجر شود.

برای کارهای آینده، مسیرهای زیر پیشنهاد می‌گردد:

- **گسترش پایگاه دانش:** افزودن اطلاعات بیشتر و متنوع‌تر، از جمله ویدیو و صوت، برای ساخت یک سیستم RAG با قابلیت‌های چندوجهی غنی‌تر.
- **بهینه‌سازی مدل :Generator** آموزش دقیق (Fine-tuning) مدل زبان بزرگ بر روی وظیفه پرسش و پاسخ مبتنی بر زمینه بازیابی‌شده، به منظور بهبود کیفیت و سبک پاسخ‌ها.
- **ارزیابی انسانی:** انجام ارزیابی توسط کاربران انسانی برای سنجش معیارهایی نظیر میزان رضایت، طبیعی بودن پاسخ‌ها و مفید بودن اطلاعات بازیابی‌شده که توسط معیارهای خودکار قابل اندازه‌گیری نیستند.
- **بررسی معماری‌های پیشرفته‌تر:** تحقیق بر روی معماری‌های جدیدتر RAG که اطلاعات متنی و تصویری را به شکل عمیق‌تری با یکدیگر ادغام می‌کنند (Fusion-in-Decoder) به جای ارائه آن‌ها به عنوان زمینه صرف.

در مجموع، این پروژه با موفقیت نشان داد که استفاده از معماری RAG چندوجهی راهکاری مؤثر برای ساخت سیستم‌های پرسش و پاسخ هوشمند در زبان فارسی است و زیربنای مناسبی برای تحقیقات آتی در این حوزه فراهم می‌آورد.

Bibliography

[1] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 11 2019. Association for Computational Linguistics.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. CLIP Encoder .

[3] A. Nourian et al. Hazm: Python library for digesting persian text. https://github.com/sobhe/hazm, 2017. Hazm .

[4] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. In *IEEE Transactions on Big Data*, volume 7, pages 535–547. IEEE, 2019. FAISS .

[5] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85, 2014. (Wikidata) .

[6] O. Medelyan, D. Milne, C. Legg, and I. H. Witten. Mining meaning from wikipedia. In *Putting semantic web in practice*, pages 115–151. Springer, 2009. .

[7] Google. Gemma: Open Models by Google, 2024. Gemma .

[8] M. Tkachenko, M. Malykh, A. Kovalev, N. Vakhrushev, A. Holmanyuk, and A. Shelmanov. Label Studio: A versatile data annotation tool. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 1–8, 2021. Label Studio

.

[9] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960. (Cohen's Kappa)

.

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2021. LoRA -

.