

# گزارش تمرین دوم درس پردازش زبان های طبیعی

اعضای تیم: مبینا پولایی، نگین رحیمی یزدی، فاطمه شاه حسینی

## 1- استخراج دیتای متنی و آماده سازی دیتاست آموزش

### 1. پیش پردازش داده

- فایل های فشرده شده ی JSON تاریخی (EVENTS ، Treaties ، WAR\_1 ، WAR\_2 ، filtered\_extracted\_contents) را باز کرده و بارگذاری می کند.
- از کتابخانه های پایتون مانند json ، os ، tqdm ، re و Transformers از HuggingFace استفاده می شود.

### 2. ادغام با مدل های زبانی بزرگ (LLM)

- اتصال به مدل های Gemini و Llama برای تولید متن فارسی.
- رکوردهای ساختاریافته JSON به پاراگراف های طبیعی و روان فارسی تبدیل می شوند، با استفاده از پرامپت های سفارشی و نمونه های few-shot.

### 3. تولید قالب خوانا

- مرور بر رکوردهای تاریخی و تولید خلاصه های فارسی قابل فهم برای انسان.
- خروجی ها به صورت فایل های متنی برای هر دیتاست ذخیره می شوند.

### 4. خوشه بندی و استخراج کلیدواژه ها

- تمام جملات تولید شده بارگذاری می شوند.
- جملات براساس کلیدواژه های تاریخی (مانند اسرائیل، کوروش، رضاشاه، مشروطه، پهلوی، ترکمانچای، مظفرالدین شاه، هسته ای، ایالات متحده، داعش، ساسانیان، عثمانی ) خوشه بندی می شوند.

### 5. تولید پرسش های متنوع

- برای هر خوشه، پرسش هایی متنوع و چندجمله ای تولید می شود که نیاز به استدلال دارند.
- از مدل های زبانی بزرگ برای ایجاد جفت های پرسش و پاسخ فارسی استفاده می شود، به شکلی که قابل تحلیل بیشتر باشند.

```

QUESTION_GEN_PROMPT = """
You are a historical data question generator. Given the following one-line summary of a historical event, generate **5 diverse Persian (Farsi) questions** based on the summary.

- Use various **wh-questions** such as "چه کسی؟ چه چیزی؟ چه زمانی؟ چرا؟ چگونه؟"
- Include different **question types**: factual, reasoning, cause-effect, and inference-based.
- Ensure the questions span multiple fields: title, date, cause, result, impact, key figures, and significance.
- Write questions and answers clearly and concisely in natural, formal Persian (Farsi).
- Do not use emojis or bullet points.
- DO not generate question form references of information.
- Output format must be:
  سؤال: [question]
  پاسخ: [answer]

Here is the historical event summary:
{text}
"""

filename = "/content/drive/MyDrive/NLP_HW2/History/ReadableFormats/QA_records.json"

for sentence in tqdm(all_sentences[:10]):
    response = ask_gemini(QUESTION_GEN_PROMPT.replace('{text}', sentence))

```

1 پرامپت تولید سوال از چانک های ساخته شده

## 6. استخراج و قالب‌بندی پرسش و پاسخ‌ها

- جفت‌های پرسش و پاسخ از خروجی مدل‌ها استخراج و پردازش می‌شوند.
- نتایج به صورت JSON ساختاردهی می‌شوند تا در وظایف بعدی قابل استفاده باشند.

## 7. ذخیره‌سازی نتایج

- جفت‌های تولیدشده پرسش و پاسخ در فایل‌های JSON ذخیره می‌شوند تا بعداً مورد استفاده قرار گیرند.

## 1- آموزش مدل‌ها

ابتدا داده‌های موردنیاز برای آموزش از یک فایل فشرده در گوگل درایو بارگیری و استخراج می‌شوند. این فایل شامل مجموعه‌ای از پرسش و جملات به‌صورت فایل JSON است که جهت آماده‌سازی برای آموزش مدل زبانی مورد استفاده قرار می‌گیرند.

در گام بعدی، داده‌ها از فایل QA\_records\_final.json بارگذاری شده و جملات آن استخراج می‌شوند. به‌منظور یک‌دست‌سازی متن‌ها، کاراکترهای خاصی مثل «\u200c» که مربوط به نیم‌فاصله است، حذف می‌شوند. در نتیجه، یک لیست از جملات متنی تمیز به‌دست می‌آید که می‌توان آن را برای یادگیری مدل زبانی استفاده کرد.

سپس با استفاده از کتابخانه‌ی **Huggingface Datasets**، این جملات به صورت یک دیتاست ساخت یافته درمی آیند. این مرحله آماده سازی اولیه برای اعمال توکنایزر و سایر پردازش های مربوط به مدل سازی زبانی است.

در مرحله ی بعدی، از توکنایزر مدل **GLot500** استفاده می شود. این مدل، یک مدل زبانی چندزبانه است که از قبل آموزش دیده و توانایی درک زبان های مختلف را دارد. جملات داده شده توسط این توکنایزر به توکن های قابل فهم برای مدل تبدیل می شوند. همچنین در این مرحله، جملات به طول مشخصی (حداکثر 128 توکن) بریده و یا با **Padding** کامل می شوند:

```
1 from transformers import AutoTokenizer
2 tokenizer = AutoTokenizer.from_pretrained('cis-lmu/glot500-base')
3
4 def tokenize_fn(example):
5     return tokenizer(example["text"], truncation=True, padding="max_length", max_length=128)
6
7 tokenized_dataset = dataset.map(tokenize_fn, batched=True)
```

در ادامه، یک کلاسی به نام **DataCollatorForLanguageModeling** از کتابخانه ی **Transformers** مورد استفاده قرار می گیرد تا داده ها را به سبک مدل سازی ماسک شده زبان (MLM) آماده کند. این تکنیک، همان روشی است که در مدل هایی مثل **BERT** استفاده می شود؛ به این صورت که برخی توکن ها به صورت تصادفی ماسک شده و مدل باید آن ها را پیش بینی کند.

پس از آماده سازی داده ها، مدل زبانی **GLot500** با ساختار **AutoModelForMaskedLM** بارگذاری شده و فرآیند آموزش آغاز می شود. از کلاس **Trainer** در ترنسفورمرز برای انجام آموزش استفاده شده است. پارامترهای آموزش مثل تعداد **epoch**، **batch size**، **warmup steps** و **learning rate** نیز تنظیم شده اند.

در پایان، مدل آموزش دیده ذخیره شده و آماده استفاده برای وظایف زبانی دیگر می شود. همچنین خروجی هایی برای بررسی کیفیت یادگیری مدل روی داده های تمرینی نیز در دسترس قرار گرفته اند.

## یادگیری تقابلی

در این بخش، هدف مدل یادگیری این است که نمایش‌های برداری (embedding) پرسش‌ها و جملاتی که با هم مرتبط هستند را به یکدیگر نزدیک کرده و نمایش‌های پرسش و جملات نامرتبط را از هم دور کند.

ابتدا داده‌ها از همان فایل QA\_records\_final.json بارگیری می‌شوند. در این بخش، برای هر ورودی، جمله و پنج پرسش آن استخراج می‌شوند. نمونه‌هایی که دقیقاً شامل پنج پرسش باشند، انتخاب و نگهداری می‌گردند. سپس این زوج‌های (جمله، پرسش) به صورت یک لیست ساختارمند آماده می‌شوند.

برای پیاده‌سازی بخش یادگیری متقابل بین جملات و پرسش‌های مربوطه، کلاس SQDataset به گونه‌ای تعریف گردید که هر نمونه از داده‌ها شامل یک جمله به همراه پنج پرسش مرتبط با آن باشد. این ساختار امکان یادگیری روابط معنایی عمیق‌تری را میان جمله و مجموعه سؤالات مربوط به آن فراهم می‌سازد. به منظور آماده‌سازی داده‌ها برای ورود به مدل، از تابع collate\_fn استفاده شد که در آن، در هر batch، جملات به صورت جداگانه در یک لیست و تمامی سؤالات در لیستی دیگر جمع‌آوری می‌شوند. به این ترتیب، مدل قادر خواهد بود در هر گام آموزشی، شباهت معنایی میان هر جمله و سؤالات متناظر آن را با بهره‌گیری از loss تقابلی به طور مؤثری بیاموزد. این ساختار، مبنای آموزش مدل در بخش یادگیری متقابل بوده و نقش کلیدی در افزایش کیفیت نمایش برداری معنایی ایفا کرده است.

در ادامه برای هر جمله و پرسش، embedding جداگانه با استفاده از همان مدل آموزش‌دیده (GLOT500) گرفته می‌شود. توکن [CLS] به عنوان نماینده جمله در نظر گرفته شده و از خروجی مدل استخراج می‌گردد. برای محاسبه ی loss، از تابع InfoNCE استفاده شده است. این تابع بر اساس شباهت کسینوسی بین بردار جمله و پرسش‌هایش، یک مقدار loss تولید می‌کند. هدف این است که embedding جمله به پرسش‌های مرتبط نزدیک شود و از پرسش‌های غیرمرتبط فاصله بگیرد:

```
def info_nce_loss(context_embs, question_embs, temperature=0.05):
    B, D = context_embs.size()
    G = 5
    device = context_embs.device
    context_embs = F.normalize(context_embs, dim=1)          # (B, D)
    question_embs = F.normalize(question_embs, dim=1)        # (B*G, D)

    # Questions → Contexts
    logits_q2c = torch.matmul(question_embs, context_embs.T) / temperature # (B*G, B)
    # Positive mask for q2c: each group of G questions maps to 1 context
    pos_q2c = torch.arange(B, device=device).repeat_interleave(G) # (B*G,)
    loss_q2c = F.cross_entropy(logits_q2c, pos_q2c)

    # Contexts → Questions
    logits_c2q = torch.matmul(context_embs, question_embs.T) / temperature # (B, B*G)
    # Positive mask for c2q: each context matches its G questions
    pos_mask_c2q = torch.zeros(B, B * G, device=device)
    for i in range(B):
        pos_mask_c2q[i, i * G: (i + 1) * G] = 1
    # Log softmax across all candidate questions
    log_probs_c2q = F.log_softmax(logits_c2q, dim=1)
    # Average log-probability across the G positives
    loss_c2q = -(pos_mask_c2q * log_probs_c2q).sum(dim=1).mean()

    loss = (loss_q2c + loss_c2q) / 2
    return loss
```

در انتها، مدل به مدت چند epoch آموزش داده می‌شود و در هر مرحله، loss محاسبه شده گزارش می‌شود تا روند یادگیری قابل مشاهده باشد.

## ارزیابی با مدل‌های مختلف

برای ارزیابی و مقایسه عملکرد مدل fine-tuned شده با روش‌های خواسته شده، ابتدا مجموعه داده ارزیابی که شامل ۵۰ سوال ترکیبی از جملات مجموعه آموزش است، فراهم شد.

## TF-IDF

عملکرد مدل TF IDF برای بازیابی (Retrieve) نزدیک‌ترین جملات مجموعه آموزش به سوالات مطرح شده در داده ارزیابی از طریق شباهت کسینوسی امبدینگ بدست آمده از این روش (بین سوالات و جملات مجموعه آموزش) به صورت زیر بررسی می‌شود:

```

vectorizer = TfidfVectorizer().fit(contexts)
context_vectors = vectorizer.transform(contexts)

def tfidf_answer(question):
    q_vec = vectorizer.transform([question])
    sims = torch.tensor(cosine_similarity(q_vec, context_vectors))
    _, top_indices = torch.topk(sims, k=3)
    return [contexts[i] for i in top_indices[0]]

tfidf_outputs = [tfidf_answer(q) for q in eval_questions]

```

## Fine-tuned و Zero-Shot

برای دو روش بعدی، فرآیند کلی هر دو رویکرد شامل دو مرحله‌ی استخراج embedding و مقایسه شباهت بردارها است. در مرحله اول، از تابع `get_embeddings` برای تولید بردارهای معنایی از جملات متنی استفاده می‌شود. این تابع با استفاده از مدل زبانی مشخص شده و توکنایزر مرتبط، ابتدا داده‌های متنی را توکنایز کرده، آن‌ها را به مدل مورد نظر (Zero-shot و Fine-tuned) می‌دهد و سپس بردار حاصل از توکن [CLS] را استخراج می‌کند. این بردارها با استفاده از نرمال‌سازی به صورت واحد درمی‌آیند تا برای محاسبه شباهت کسینوسی آماده باشند. بدین ترتیب، برای هر جمله، یک embedding استاندارد به دست می‌آید.

در مرحله دوم، تابع `contrastive_answer` وظیفه مقایسه پرسش با جملات را بر عهده دارد. برای هر پرسش، ابتدا embedding آن استخراج و نرمال‌سازی می‌شود. سپس با ضرب داخلی بین این embedding و ماتریس embedding های جملات، شباهت‌های کسینوسی محاسبه می‌شود. در نهایت، سه جمله با بالاترین شباهت به عنوان خروجی بازگردانده می‌شوند.

در رویکرد اول، از مدل زبانی GLOT500 به صورت Zero-Shot استفاده می‌شود. این مدل بدون هیچ‌گونه آموزش اختصاصی روی داده‌های پرسش-جمله، فقط با تکیه بر دانش عمومی خود، برای استخراج embedding و بازیابی پاسخ‌ها به کار گرفته می‌شود. اگرچه این رویکرد بدون هزینه آموزش است، اما معمولاً دقت پایین‌تری دارد.

در رویکرد دوم، همان مدل GLOT500 به کمک یادگیری تقابلی با داده‌های موجود `fine-tune` شده است. در این فرآیند، مدل با استفاده از زوج‌های صحیح پرسش و جمله، آموزش می‌بیند تا بردار پرسش‌ها و

جملات مربوطه را به هم نزدیک و سایر بردارها را از یکدیگر دور کند. در این رویکرد نیز همان ساختار توابع `get_embeddings` و `contrastive_answer` استفاده شده، اما با مدل آموزش دیده.

در مجموع، این دو رویکرد امکان مقایسه مستقیم عملکرد مدل پیش‌آمخته و مدل `fine-tuned` را در مسئله‌ی بازیابی پاسخ فراهم می‌سازند. انتظار می‌رود که مدل `fine-tuned` عملکرد بهتری در تطبیق معنایی پرسش و جملات داشته باشد.

### 3- لیبل گذاری در Label Studio

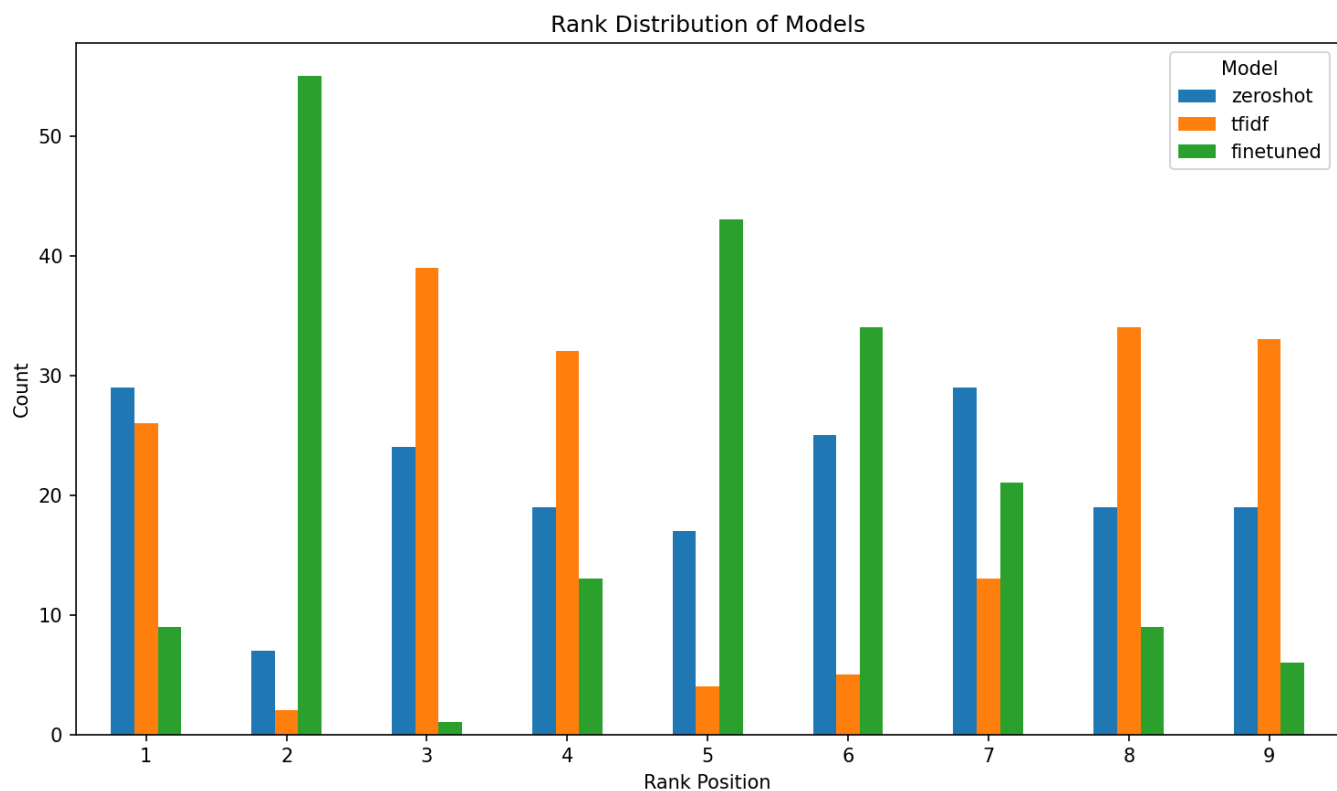
در این بخش، فرایند لیبل گذاری داده‌ها را در Label Studio انجام دادیم. بدین صورت که به هر پاسخ، بر اساس میزان نزدیکی معنایی‌اش به صورت سوال، امتیازی بین ۱ تا ۹ اختصاص دادیم؛ به‌طوری‌که پاسخی که بیشترین هم‌خوانی را با صورت سوال داشتند، امتیاز ۱ و پاسخ‌های بسیار دور از موضوع سوال، امتیاز ۹ دریافت کردند.

در رتبه‌بندی، معیارهای مختلفی را نیز در نظر گرفتیم. برای مثال، اگر سوال مربوط به یک شخص بود، پاسخی که هرچند نادرست اما همچنان به افراد اشاره داشتند، امتیاز بهتری نسبت به پاسخی که درباره‌ی رویدادها بودند، دریافت می‌کردند.

همچنین برای جلوگیری از هرگونه سوگیری، پیش از آغاز فرایند لیبل گذاری، ترتیب پاسخ‌ها را به‌صورت تصادفی تغییر دادیم تا فرد لیبل‌گذار از ترتیب و امتیازدهی اولیه اطلاعی نداشته باشد.

### 4- تحلیل و بررسی نتایج

در این گزارش، عملکرد سه مدل بازیابی پاسخ شامل **TF-IDF**، **Zero-shot** و **Fine-tuned** بر اساس داده‌های انسانی و معیارهای آماری مقایسه شده است.



2 توزیع خروجی های سه مدل روی رتک ها

## ۱. تحلیل آماری

- آزمون فریدمن: مقدار  $\chi^2$  برابر ۵.۱۶۸۷ و مقدار  $p$  برابر ۰.۰۷۵۴ به دست آمد که نشان می دهد تفاوت کلی بین مدل ها از نظر رتبه بندی نزدیک به معنی دار بودن است ولی کاملاً قطعی نیست.
- آزمون ویلکاکسون (جفت به جفت):

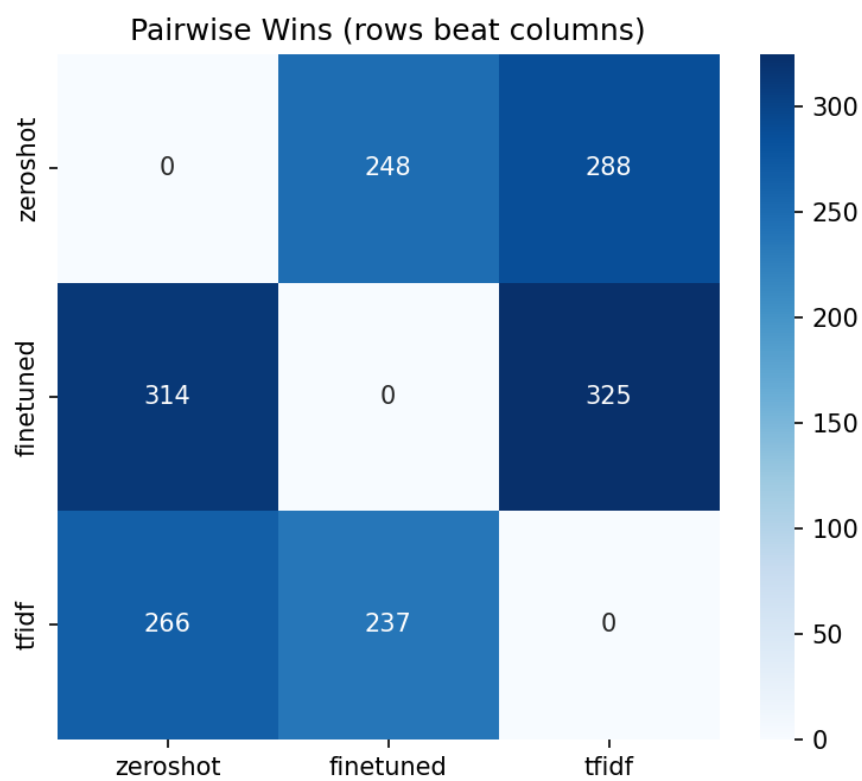
- TF-IDF در برابر Zero-shot:  $p=0.۴۷۸۶$  (تفاوت معنادار نیست)
- TF-IDF در برابر Fine-tuned:  $p=0.۰۰۷۰$  (تفاوت معنادار و به نفع Fine-tuned)
- Zero-shot در برابر Fine-tuned:  $p=0.۰۴۸۲$  (تفاوت معنادار و به نفع Fine-tuned)

این نتایج نشان می دهد که مدل Fine-tuned نسبت به دو مدل دیگر به طور معناداری عملکرد بهتری دارد.



۲. نتایج مقایسه زوجی (Pairwise Wins)

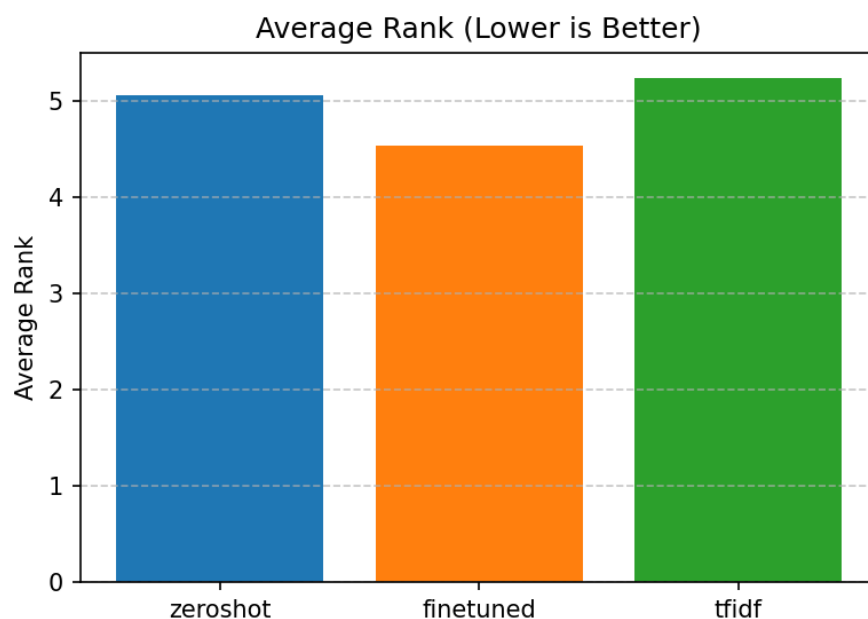
مدل اول	مدل دوم	دفعات برد مدل اول
Zero-shot	Fine-tuned	۲۴۸
Zero-shot	TF-IDF	۲۸۸
Fine-tuned	TF-IDF	۳۲۵
Fine-tuned	Zero-shot	۳۱۴
TF-IDF	Fine-tuned	۲۳۷
TF-IDF	Zero-shot	۲۶۶



### ۳. خلاصه کلی

- میانگین رتبه مدل‌ها:

- Fine-tuned: ۴.۵۳۴۰
- Zero-shot: ۵.۰۶۳۸
- TF-IDF: ۵.۲۳۹۳



۳ میانگین رتبه مدل‌ها

- امتیاز نرمال شده:

- Zero-shot: ۱۰۷۳۳۶
- Fine-tuned: ۱۰۹۱۵۰
- TF-IDF: ۱۰۷۳۰۳

### ۴. تحلیل عملکرد مدل‌ها

- مدل **Fine-tuned** با توجه به امتیازات آماری و رتبه‌بندی‌ها عملکرد بهتری نسبت به مدل‌های پایه دارد و توانسته در پاسخ به انواع سوالات پیچیده‌تر، کیفیت پاسخ بهتری ارائه کند.

- مدل **Zero-shot** به دلیل توانایی در پاسخدهی عمومی و بدون نیاز به آموزش اختصاصی، در برخی موارد نزدیک به مدل **Fine-tuned** عمل کرده است.
- مدل **TF-IDF** که بر مبنای تشابه برداری ساده کار می‌کند، عملکرد کمتری داشته و در پاسخ به سوالات پیچیده یا نیازمند درک عمیق‌تر، ضعیف‌تر عمل کرده است.

## ۵. تحلیل رفتاری در انواع سوالات

- مدل **Fine-tuned** در سوالات تخصصی و دارای ساختار پیچیده عملکرد بهتری نشان داده است. برای مثال در مورد زیر از رجال سیاسی دوران پهلوی که در دوران قاجار هم فعالیت سیاسی داشته اند پرسیده شده است. ممکن است مدل پایه به خوبی بتواند از رجال سیاسی قاجار یا پهلوی به طور جداگانه افرادی را پیدا کند، ولی تجمیع این دو احتمالا شرایط ریترو را سخت‌تر می‌کند و به دانش تخصصی مدل آموزش دیده روی داده تاریخی نیاز است.

```
{
  "question": "د. کدام یک از رجال سیاسی دوران پهلوی، در دوران قاجار نیز فعالیت سیاسی داشته و در چه سمت‌هایی مشغول به کار بوده است؟",
  "finetuned_8": "مت‌میکرد، ضمن اینکه با امیرکبیر نیز دوستی دیرینه داشت. برای کسب اطلاعات بیشتر در مورد این شخصیت، می‌توان به کتاب: \"ای این رویداد یافت\" امتیازنامه رویترا\" در سال ۱۸۷۲، قراردادی میان پاول بولیوس رویترا و دولت قاجار منعقد شد که به: \"zeroshot_6\" می‌در تهران متولد شد. او در پایتخت چشم به جهان گشود و در عرصه سیاست آن دوران، نامی آشنا بود. این رخداد در کتاب: \"zeroshot_4\" پهلوی، در سال ۱۲۵۳ هجری شمسی در کرمان دیده به جهان گشود. او فرزند شاهرخ بود. برای کسب اطلاعات بیشتر، میتوان به کتاب: \"tfidf_3\" را تصویب کرده‌اند که نشان‌دهنده اهمیت جهانی این پیمان است. اطلاعات بیشتر در این مورد در وبسایت ویکیپدیا موجود است. \"finetuned_9\" از معکازان سید ضیاء‌الدین در کمیته زرگنده (آهن) بود. اطلاعات بیشتر در مورد زندگی و فعالیت‌های وی را میتوان در کتاب: \"finetuned_7\" می‌در ساری چشم به جهان گشود. او فرزند میرزا علیاکبر حافظ‌الصحه بود. این رویداد مهم در تاریخ معاصر ایران، در کتاب: \"zeroshot_5\" نغمی در بوشهر چشم به جهان گشود. پدر ایشان میرزا حسین کازرونی بود. اطلاعات بیشتری در خصوص این شخصیت و زادگاهش در کتاب: \"tfidf_2\"، بر اساس داده‌های موجود، ایشان در تهران به دنیا آمدند. برای کسب اطلاعات بیشتر در این زمینه، میتوان به کتابی با عنوان: \"tfidf_1\"
}
```

یا این مورد که احتمالا محل تولد و سوابق شغلی این دو رجل سیاسی در اطلاعات ضمنی مدل zero shot نیامده است و مدل فاین تیون شده به طور مشخصی عملکرد بهتری داشته است.

```
{
  "question": "محل تولد و سوابق شغلی محمود پولادین و محمود جم، دو شخصیت برجسته دوران پهلوی را ارائه دهید.",
  "finetuned_7": "در کمیته زرگنده (آهن) بود. اطلاعات بیشتر در مورد زندگی و فعالیت‌های وی را میتوان در کتاب: \"finetuned_8\" در کمیته زرگنده (آهن) بود. اطلاعات بیشتر در مورد زندگی و فعالیت‌های وی را میتوان در کتاب: \"tfidf_1\" و تأثیرات این شورش، اطلاعاتی در دسترس نیست. با این حال، این واقعه در ویکی‌پدیا ذکر شده است. \"zeroshot_4\" برای زندگی او در پایتخت به شمار میرود. اطلاعات بیشتری در مورد زندگی و فعالیت‌های وی در کتاب: \"finetuned_9\" اهمیت تاریخی خود را در نشان دادن تلاش‌های ایران برای حفظ سلطه در این منطقه حساس نشان میدهد. \"zeroshot_6\" ۱۴۰۰ وارد مرحله‌های جدید شده است و نشان از اهمیت متقابل این دو کشور در عرصه بین‌المللی دارد. \"zeroshot_5\" به سمت معاونت حکومت لاهیجان منصوب شد. اطلاعات بیشتری درباره زندگی و فعالیت‌های وی در کتاب: \"tfidf_3\" به نابودی پتخانه و کشتار گسترده انجامید \"سومناث\" در سال ۴۱۷ هجری قمری، رویدادی با عنوان: \"tfidf_2\" برای زندگی او در پایتخت به شمار میرود. اطلاعات بیشتری در مورد زندگی و فعالیت‌های وی در کتاب: \"tfidf_2\"
}
```

- مدل Zero-shot در سوالات عمومی و بدون نیاز به داده‌های آموزش اختصاصی، پاسخ‌های قابل قبولی داشته است. در مثال پایین صرفاً دو شخصیت سیاسی زمان پهلوی مورد سوال قرار گرفته‌اند که با دانش ضمنی مدل هم قابل پاسخگویی است.

```
{
  "question": "دو شخصیت سیاسی که در دوران پهلوی فعالیت داشتند را نام ببرید و محل تولد هر یک را مشخص کنید",
  "zeroshot_6": "بازنامه رویتر" در سال ۱۸۷۲، قراردادی میان پاول پولیوس رویتر و دولت قاجار منعقد شد که به-
  "finetuned_9": "باجی به مبلغ ۳۰۰۰ طلای \پیمان صلح پنجاه ساله\" در سال ۵۶۲ میلادی، پیماننامه‌ای با عنوان "
  "tfidf_2": "گشود. پدر ایشان میرزا حسین کازرونی بود. اطلاعات بیشتری در خصوص این شخصیت و زادگاهش در کتاب "
  "zeroshot_5": "نگاه جامع تاریخ، تولد محمدعلی بامداد در آذربایجان غربی رخ داده است. اطلاعات بیشتر در کتاب "
  "finetuned_8": "نگاه جامع تاریخ است. متأسفانه، به دلیل کمبود داده‌ها، قادر به ارائه توضیحات بیشتری نیستم "
  "finetuned_7": "بود و زندگی خود را در پایتخت آغاز کرد. اطلاعات بیشتر در مورد این شخصیت و تولدش در کتاب "
  "tfidf_3": "به جهان گشود. او فرزند حاج ابوالقاسم بود. این شخصیت برجسته، اطلاعات بیشتری درباره‌اش در کتاب "
  "zeroshot_4": "اما سالهای شروع و پایان فعالیت ایشان نامشخص است. برای کسب اطلاعات بیشتر، میتوان به منبع "
  "tfidf_1": "در دهه ۱۹۶۰ افزایش یافته بود، به تصویب رسید. این کنوانسیون، که توسط کمیسیون حقوق بینالملل "
}
```

- مدل TF-IDF بیشتر در سوالات ساده‌تر و با واژگان مشخص موفق بوده است اما در سوالات نیازمند درک معنایی عمیق‌تر ضعف داشته است. برای مثال در این مورد به علت وجود واژه مشخص عهد نامه ترکمانچای در سوال ساده مطرح شده، مدل tfidf عملکرد بهتری داشته است.

```
{
  "question": "و همچنین نشستهای بعدی مرتبط با این دریا داشت؟ ترکمانچای عهدنامه",
  "tfidf_2": "بتوانید در منابع مختلف از جمله ویکیپدیا بیابید، ترکمانچای عهدنامه",
  "tfidf_1": "بتوانید در منابع مختلف از جمله ویکیپدیا بیابید، ترکمانچای عهدنامه",
  "finetuned_7": "دریا پرداختند. این نشست، که در پی رویدادهای مهمی چون عهدنامه",
  "finetuned_8": "نوست، دوره‌ای پر از تنش و درگیری بود که نهایتاً با امضای عهدنامه",
  "zeroshot_4": "انجام شد. برای اطلاعات بیشتر، میتوان به صفحه ویکیپدیا با عنوان "
  "finetuned_9": "درنسازی ارتش و تقویت روابط خارجی ایران در آن دوران تاریخی است",
  "zeroshot_6": "عات بیشتر در این زمینه را میتوان در منابعی مانند ویکیپدیا یافت",
  "zeroshot_5": "میتوان در منابعی همچون گزارشهای سرگور اوزلی و جیمز موریه یافت",
  "tfidf_3": "ن پیمان، گامی مهم در جهت حفاظت از محیط زیست دریای خزر محسوب میشود",
}
```

## ۶. جمع‌بندی و پیشنهادات

- تحلیل آماری نشان می‌دهد که آموزش اختصاصی (Fine-tuning) تأثیر مثبت و معناداری روی کیفیت پاسخ‌ها دارد.
- ترکیب روش‌های آماری با ارزیابی انسانی، دید جامع‌تری از نقاط قوت و ضعف مدل‌ها ارائه می‌دهد.

- برای بهبود بیشتر، توصیه می‌شود مدل Fine-tuned با داده‌های متنوع‌تر و باکیفیت‌تر آموزش داده شود و روش‌های Zero-shot به عنوان مکمل در شرایطی که داده آموزش محدود است، استفاده شوند.
- تحلیل دقیق‌تر رفتار مدل‌ها در انواع مختلف سوالات می‌تواند به بهینه‌سازی استراتژی‌های بازیابی کمک کند.