دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

درس پردازش زبان های طبیعی

عنوان:

سیستم پرسش و پاسخ پزشکی

اعضای گروه:

سید محمدرضا حسینی

فرحان سراوند

وحیدالدین مقیمی

استاد: احسان الدین عسگری

فهرست مطالب

1) **بازیابی متون مرتبط**

2) **تولید پاسخ**

۱. بازیابی متون مرتبط

برای ساخت این سیستم در ابتدا نیاز به پیدا کردن دیتاست مناسب برای انجام این تمرین بود. Medmcqa دیتاست استفاده شده است که یک دیتاست برای پاسخ گویی به سوالات پزشکی میباشد. پس از دانلود دیتاست، برای اینکه بتوانیم جواب مورد نظرمان را بیابیم، میبایست بتوانیم از یک تکنیک مناسب برای بازیابی متون استفاده کنیم. تکنیک استفاده شده بدین صورت است که در ابتدا با استفاده از TF_IDF ، کوئری و سوال فرد و جواب های موجود را تبدیل میکنیم سپس با استفاده از Cosine similiarity میزان شباهت کوئری و جواب ها را میسنجیم و بر اساس بالاترین شباهت جواب ها را مرتب میکنیم

2. تولید پاسخ

هدف از این تسک پیاده سازی یک RAG(Retrieval Augment Generation) است که در آن با استفاده از داده های پزشکی به توان به سوالات مطرح شده پاسخ مناسبی داد.

پس از انتخاب دیتاست و پیاده سازی تکنیک بازیابی متون به سراغ تسک اصلی میرویم. در ابتدا از مدل پایه بدون هیچ فاین تیونیگ استفاده میکنیم.

```
query = "Which vitamin is supplied from only animal source?"

data = load_data(json_file)
filtered_data, texts = preprocess_texts(data)

tfidf_matrix = compute_tfidf(texts, query)
similarities = find_similar_texts(tfidf_matrix)
related_texts = get_related_texts(filtered_data, similarities)

model_name = "t5-small"
model = T5ForConditionalGeneration.from_pretrained(model_name)
tokenizer = T5Tokenizer.from_pretrained(model_name)

input_text = create_input(query, related_texts)
print(f"Input text for the model: {input_text}")

answer = generate_answer(
    model, tokenizer, input_text,
    max_length=150,
    num_beams=10,
    early_stopping=False,
    temperature=0.7,
    top_k=50,
    top_p=0.9,
    repetition_penalty=2.0,
    no_repeat_ngram_size=3,
    length_penalty=3.0,
    do_sample=True
)
print(f"Generated answer: {answer}")
```

پرسش و پاسخ به صورت زیر میباشد:

```
Input text for the model: پـرسش: Which vitamin is supplied from only animal source? زمـینه: Ans.
 (c) Vitamin B12 Ref: Harrison's 19th ed. P 640* Vitamin B12 (Cobalamin) is synthesized solely
 by microorganisms.* In humans, the only source for humans is food of animal origin, e.g.,
meat, fish, and dairy products.* Vegetables, fruits, and other foods of nonanimal origin
doesn't contain Vitamin B12 .* Daily requirements of vitamin Bp is about 1-3 pg. Body stores
are of the order of 2-3 mg, sufficient for 3-4 years if supplies are completely cut off.
Vitamin D is not strictly a vitamin since it can be synthesized in the skin, and under most
conditions, this is the major source of the Vitamin D. Ref : Biochemistry by U. Satyanarayana
3rd edition Pgno : 123 Ans. (a) Animal SourceRef: Harrison / 640Only source of vitamin B12 for
 humans is food of animal origin, e.g., meat, fish, and dairy products.Vegetables, fruits, and
 other foods of non-animal origin are free from cobalamin unless they are contaminated by
bacteria. Cobalamin is synthesized solely by microorganisms.
Generated answer: Biochemistry by U. Satyanarayana 3rd edition Pgno : 123 Ans. (a) Animal
SourceRef: Harrison / 640Only source of vitamin B12 for humans is food of animal origin, e.g.,
 meat, fish, and dairy products.* Vegetables, fruits, and other foods of non-animal origin
doesn't contain Vitamin B12.
```

سپس به سراغ فاین تیون کردن مدل T5 میرویم. برای اینکار تعداد داده های مختلفی آزمایش شد و در نهایت ۲۰۰۰ پرسش و پاسخ اول موجود در دیتاست برای فاین تیون کردن مدل استفاده شد تا بتوان اینکار را در یک زمان مناسب انجام داد. جواب ها در یک لیست و سوال ها در یک لیست دیگر ذخیره میشود. سوال ها به عنوان ورودی و جواب به عنوان خروجی مطلوب که مدل باید به آن دست پیدا کند فرض میشود. برای توکنایز کردن داده ها نیز از توکنایزر مدل T5 استفاده شده تا با آن همخوانی داشته باشد. این مدل ۶ ایپاک فاین تیون شده و Loss های دریافت شده از آموزش و ولیدیشن نشان دهنده موفق بودن فاین تیونیگ میباشد.

```python
json_file = 'train.json'
finetune_data = load_finetune_data(json_file,limit=2000)
inputs, targets = preprocess_finetune_data(finetune_data)

train_inputs, eval_inputs, train_targets, eval_targets = train_test_split(inputs, targets, test_size=0.1)

model_name = "t5-small"
tokenizer = T5Tokenizer.from_pretrained(model_name)
model = T5ForConditionalGeneration.from_pretrained(model_name)

train_encodings = tokenize_data(tokenizer, train_inputs, train_targets)
eval_encodings = tokenize_data(tokenizer, eval_inputs, eval_targets)

train_dataset = Dataset.from_dict(train_encodings)
eval_dataset = Dataset.from_dict(eval_encodings)

finetune_model(model, tokenizer, train_dataset, eval_dataset)

model.save_pretrained("./finetuned_model_6e")
tokenizer.save_pretrained("./finetuned_model_6e")
```

Epoch	Training Loss	Validation Loss
1	5.182100	4.716797
2	4.783800	4.364937
3	4.507000	4.234683
4	4.380300	4.154284
5	4.342100	4.093639
6	4.361300	4.076693

پس از فاین تیون کردن مدل از این مدل برای تولید پاسخ با توجه به دیتا استفاده میکنیم. در ابتدا مطابق روش گفته شده پرسش به همراه جواب ها با استفاده از TF_IDF تبدیل میشوند سپس با استفاده از cosine_similiarity مرتبط ترین جواب ها پیدا میشوند. با دادن مرتبط ترین جواب ها و کوئری به مدل فاین تیون شده، جواب به آن سوال تولید میشود.

برای تولید پاسخ تعدادی پارامتر ست شده است که به صورت زیر میباشد:

max_length = 150 **که حداکثر میزان توکن تولیدی را نشان میدهد**

num_beams = 5 **که تعداد پرتو های استفاده شده در** beam search **را نمایش میدهد. ست کردن درست این پارامتر سبب میشود که تعداد جواب های تولیدی بیشتر شود و بهترین آن ها برای خروجی استفاده شود.**

early_stopping **: ست کردن این پارامتر سبب میشود که در صورتی که به اندازه تعداد پرتو جواب ها را تولید کردیم، عملیات متوقف شود**

temprature=0.7 **با ست کردن این پارامتر میزان قطعی بودن یا تخیلی بودن جواب را تعیین میکنیم. هر چه این عدد کم تر و نزدیک به صفر باشد جواب با توجه به منابع موجود در پایگاه دانش تولید میشود و هرجه بیشتر باشد مدل نیز سعی میکند جواب را طبق دانسته ها تغییر دهد.**

top_k=50 **تعداد کاندیداهای برتر که باید از آن نمونه برداری شود را نشان میدهد.**

top_p = 0.9 **نمونه گیری هسته ای که مجموع احتمالات را تا یک آستانه خاص شامل میشود.**

repetition_penalty=0.2 این پارامتر مقدار جریمه ای که به ازای تکرار کلمات اتفاق میفتد را نشان میدهد. این جریمه از تکرار کردن کلمات توسط مدل جلوگیری میکند.

سپس پس از فاین تیون با استفاده از روش گفته شده و استفاده از TF_IDF و Cosine Similiarity جواب های پیدا شده به مدل داده میشود و مدل جواب نهایی را تولید میکند. جواب RAG پیاده سازی شده با مدل فاین تیون شده به صورت زیر است:

```
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.

Input text for the model: پرسش: Which vitamin is supplied from only animal source? زمینه: Ans. (c) Vitamin B12 Ref: Harrison's 19th ed. P 640* Vitamin B12 (Cobalamin) is synthesized solely by microorganisms.* In humans, the only source for humans is food of animal origin, e.g., meat, fish, and dairy products.* Vegetables, fruits, and other foods of nonanimal origin doesn't contain Vitamin B12 .* Daily requirements of vitamin Bp is about 1-3 pg. Body stores are of the order of 2-3 mg, sufficient for 3-4 years if supplies are completely cut off. Vitamin D is not strictly a vitamin since it can be synthesized in the skin, and under most conditions, this is the major source of the Vitamin D. Ref : Biochemistry by U. Satyanarayana 3rd edition Pgno : 123 Ans. (a) Animal SourceRef: Harrison / 640Only source of vitamin B12 for humans is food of animal origin, e.g., meat, fish, and dairy products.Vegetables, fruits, and other foods of non-animal origin are free from cobalamin unless they are contaminated by bacteria. Cobalamin is synthesized solely by microorganisms.
Generated answer: Biochemistry by U. Satyanarayana 3rd edition Pgno : 123 Ans. (a) Animal SourceRef: Harrison / 640Only source of vitamin B12 for humans is food of animal origin, e.g., meat, fish, and dairy products.* Vegetables, fruits, and other foods of non-animal origin doesn't contain Vitamin B12.
```

برای ارزیابی مدل از بخشی از دیتاست استفاده شده که برای فاین تیون مدل استفاده نشده است تا مدل آن را ندیده باشد و بر اساس آن تست ها انجام شود.

```python
print("Evaluating base model...")
base_references, base_predictions = evaluate_model(base_model, base_tokenizer, eval_data)

print("Evaluating fine-tuned model...")
finetuned_references, finetuned_predictions = evaluate_model(finetuned_model, finetuned_tokenizer, eval_data)

base_bleu_score = bleu.compute(predictions=base_predictions, references=[[ref] for ref in base_references])
# base_rouge_score = rouge.compute(predictions=base_predictions, references=base_references)
# base_meteor_score = meteor.compute(predictions=base_predictions, references=base_references)

finetuned_bleu_score = bleu.compute(predictions=finetuned_predictions, references=[[ref] for ref in finetuned_references])
# finetuned_rouge_score = rouge.compute(predictions=finetuned_predictions, references=finetuned_references)
# finetuned_meteor_score = meteor.compute(predictions=finetuned_predictions, references=finetuned_references)

print("Base Model BLEU Score:", base_bleu_score)
# print("Base Model ROUGE Score:", base_rouge_score)
# print("Base Model METEOR Score:", base_meteor_score)

print("Fine-tuned Model BLEU Score:", finetuned_bleu_score)
# print("Fine-tuned Model ROUGE Score:", finetuned_rouge_score)
# print("Fine-tuned Model METEOR Score:", finetuned_meteor_score)
```

```
Base Model BLEU Score: {'score': 17.57078347680487, 'counts': [1453, 1219, 1070, 944], 'totals': [1890, 1848, 1806, 1764], 'precisions': [76.87830687830687, 65.96320346320347, 59.246954595791806, 53
.51473922902494], 'bp': 0.27747884525674477, 'sys_len': 1890, 'ref_len': 4313}
Fine-tuned Model BLEU Score: {'score': 22.156389962235906, 'counts': [1619, 1102, 855, 693], 'totals': [2958, 2916, 2874, 2832], 'precisions': [54.73292765382015, 37.79149519890261, 29.74947807933194, 24
.470338983050848], 'bp': 0.6324970112819258, 'sys_len': 2958, 'ref_len': 4313}
```

نتایج به دست آمده در بالا نشان میدهد که مدل فاین تیون شده عملکرد بهتری نسبت به مدل ابتدایی دارد و نشان میدهد مدل هنگامی که بر روی داده های پزشکی فاین تیون شده توانسته عملکرد بهتری در پاسخ دادن به سوالات پزشکی داشته باشد.