Chỉ mục tìm kiếm

FAISS và Elaticsearch cho phép tìm kiếm các ví dụ trong tập dữ liệu. Điều này có thể hữu ích khi bạn muốn truy xuất các ví dụ cụ thể từ tập dữ liệu có liên quan đến nhiệm vụ NLP của mình. Ví dụ: nếu bạn đang thực hiện nhiệm vụ Trả lời câu hỏi trên miền mở, bạn có thể muốn chỉ trả lại các ví dụ có liên quan đến việc trả lời câu hỏi của bạn.

Hướng dẫn này sẽ chỉ cho bạn cách tạo chỉ mục cho tập dữ liệu để cho phép bạn tìm kiếm nó.

THẤT BẠI

FAISS truy xuất tài liệu dựa trên sự giống nhau trong cách biểu diễn vectơ của chúng. Trong này Ví dụ: bạn sẽ tạo biểu diễn vectơ bằng mô hình DPR.

1. Tải xuống mô hình DPR từ Transformers:

```
>>> from transformers import DPRContextEncoder, DPRContextEncoderTokenizer
>>> import torch
>>> torch.set_grad_enabled(False)
>>> ctx_encoder = DPRContextEncoder.from_pretrained("facebook/dpr-ctx_encoder-single-nq-base")
>>> ctx_tokenizer = DPRContextEncoderTokenizer.from_pretrained("facebook/dpr-ctx_encoder-single-nq
```

2. Tải tập dữ liệu của bạn và tính toán các biểu diễn vectơ:

```
>>> from datasets import load_dataset
>>> ds = load_dataset('crime_and_punish', split='train[:100]')
>>> ds_with_embeddings = ds.map(lambda example: {'embeddings': ctx_encoder(**ctx_tokenizer(example
```

3. Create the index with Dataset.add_faiss_index():

```
>>> ds_with_embeddings.add_faiss_index(column='embeddings')
```

4. Bây giờ bạn có thể truy vấn tập dữ liệu của mình bằng chỉ mục nhúng. Tải câu hỏi DPR Encoder, and search for a question with Dataset.get_nearest_examples():

```
>>> from transformers import DPRQuestionEncoder, DPRQuestionEncoderTokenizer
>>> q_encoder = DPRQuestionEncoder.from_pretrained("facebook/dpr-question_encoder-single-nq-base")
>>> q_tokenizer = DPRQuestionEncoderTokenizer.from_pretrained("facebook/dpr-question_encoder-singl

>>> question = "Is it serious ?"
>>> question_embedding = q_encoder(**q_tokenizer(question, return_tensors="pt"))[0][0].numpy()
>>> scores, retrieved_examples = ds_with_embeddings.get_nearest_examples('embeddings', question_em
>>> retrieved_examples["line"][0]
'_điều đó_ nghiêm trọng à? Nó không nghiêm trọng chút nào. Nó chỉ đơn giản là một sự tưởng tượng để g
```

5. You can access the index with Dataset.get_index() and use it for special operations, e.g.
   truy vấn nó bằng cách sử dụng range_search :

```
>>> faiss_index = ds_with_embeddings.get_index('embeddings').faiss_index
>>> limits, distances, indices = faiss_index.range_search(x=question_embedding.reshape(1, -1), thr
```

6. When you are done querying, save the index on disk with Dataset.save_faiss_index():

```
>>> ds_with_embeddings.save_faiss_index('embeddings', 'my_index.faiss')
```

7. Reload it at a later time with Dataset.load_faiss_index():

```
>>> ds = load_dataset('crime_and_punish', split='train[:100]')
>>> ds.load_faiss_index('embeddings', 'my_index.faiss')
```

Elaticsearch

Không giống như FAISS, Elaticsearch truy xuất tài liệu dựa trên kết quả khớp chính xác.

Khởi động Elaticsearch trên máy của bạn hoặc xem hướng dẫn cài đặt Elaticsearch nếu bạn không đã cài đặt rồi.

1. Tải tập dữ liệu bạn muốn lập chỉ mục:

```
>>> from datasets import load_dataset
>>> squad = load_dataset('rajpurkar/squad', split='validation')
```

2. Build the index with Dataset.add_elasticsearch_index():

```
>>> squad.add_elasticsearch_index("context", host="localhost", port="9200")
```

3. Then you can query the context index with Dataset.get_nearest_examples():

```
>>> query = "machine"
>>> scores, retrieved_examples = squad.get_nearest_examples("context", query, k=10)
>>> retrieved_examples["title"][0]
'Lý thuyết tính toán_phức tạp'
```

4. Nếu bạn muốn sử dụng lại chỉ mục, hãy xác định tham số es_index_name khi bạn xây dựng chỉ số:

```
>>> from datasets import load_dataset
>>> squad = load_dataset('rajpurkar/squad', split='validation')
>>> squad.add_elasticsearch_index("context", host="localhost", port="9200", es_index_name="hf_squa
>>> squad.get_index("context").es_index_name
hf_squad_val_context
```

5. Reload it later with the index name when you call Dataset.load_elasticsearch_index():

```
>>> from datasets import load_dataset
>>> squad = load_dataset('rajpurkar/squad', split='validation')
>>> squad.load_elasticsearch_index("context", host="localhost", port="9200", es_index_name="hf_squ
>>> query = "machine"
>>> scores, retrieved_examples = squad.get_nearest_examples("context", query, k=10)
```

Để sử dụng Elaticsearch nâng cao hơn, bạn có thể chỉ định cấu hình của riêng mình bằng tùy chỉnh cài đặt:

```
>>> import elasticsearch as es
>>> import elasticsearch.helpers
>>> from elasticsearch import Elasticsearch
>>> es_client = Elasticsearch([{"host": "localhost", "port": "9200"}])# default client
>>> es_config = {
..."settings": {
..."số_of_shard": 1,
..."analysis": {"analyzer": {"stop_standard": {"type": "standard", " stopwords": "_englis
...},
..."mappings": {"properties": {"text": {"type": "text", "analyzer": "standard", "similarity":
... }# default config
>>> es_index_name = "hf_squad_context"# name of the index in Elasticsearch
>>> squad.add_elasticsearch_index("context", es_client=es_client, es_config=es_config, es_index_na
```