به نام خدا

فاز سوم پروژه بازیابی

Elasticsearch

محمد جواد زندیه ۹۸۳۱۰۳۲

۹ تیر ۱۴۰۱

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

۳-۲ پاسخ‌دهی به پرسمان کاربر

۱. ساخت شاخص در این فاز (با استفاده از Bulk API) را با ساخت شاخص مکانی در فاز یک از نظر زمانی (پیاده سازی و زمان اجرا) مقایسه نمایید.

از لحاظ زمانی ساخت شاخص با استفاده از تکنیک Bulk API بسیار سریع تر از ساخت شاخص با استفاده از تکنیک های پیاده سازی فاز یک می‌باشد. در تصویر زیر زمان ها مشخص شده اند:

فاز سه:

```
start = time.time()
bulk_sync()
end = time.time()
print("Indexing all documents took about {:.2f} seconds".format(end - start))
```

```
Indexing all documents took about 31.43 seconds
```

فاز یک:

```
    docLen = len(stem_stopword_token_normal)
    for pos in range(docLen):
        term = stem_stopword_token_normal[pos]
        if term not in positional_index.keys(): # first visit of this term in all documents
            positional_index[term] = {'tot_freq': 1, 'doc_ids': {docID: {'doc_freq': 1, 'positic
        else:
            positional_index[term]['tot_freq'] += 1
            if docID not in positional_index[term]['doc_ids'].keys(): # first visit of this term
                positional_index[term]['doc_ids'][docID] = {'doc_freq': 1, 'positions': [pos]}
            else: # not first visit of this term in this document
                positional_index[term]['doc_ids'][docID]['doc_freq'] += 1
                positional_index[term]['doc_ids'][docID]['positions'].append(pos)

end = time.time()
print("Indexing all documents took about {:.2f} seconds".format(end - start))
```

```
Indexing all documents took about 120.06 seconds
```

ساخت شاخص در فاز سه ۳۰ ثانیه طول کشید، در حالیکه در فاز یک ۱۲۰ ثانیه طول کشید (حدود ۴ برابر) که قابل مشاهده است که سرعت ساخت شاخص توسط تکنیک Bulk بسیار بیشتر است.

حجم پیاده سازی و سختی کار با positional indexing بسیار بالا است (حجم کد ها در زیر نشان داده شده است):

فاز سه:

```
from elasticsearch.helpers import bulk

def bulk_sync():
    actions = [
        {
            '_index': index_name,
            '_id': doc_id,
            '_source': doc
        } for doc_id,doc in data.items()
    ]
    bulk(es, actions)
```

```
start = time.time()
bulk_sync()
end = time.time()
print("Indexing all documents took about {:.2f} seconds".format(end - start))
```

فاز یک:

```
start = time.time()

positional_index = {}

list_stem_stopword_token_normal = []

# iterating through the json list
for docID in documents:
    # normalize
    normal = my_normalizer.normalize(documents[docID]["content"])
    # tokenize
    token_normal = my_tokenizer.tokenize_words(normal)
    # remove stopwords
    stopword_token_normal = [t for t in token_normal if t not in persian_stopwords]
    # stemming
    stem_stopword_token_normal = [my_stemmer.convert_to_stem(w) for w in stopword_token_normal]
    list_stem_stopword_token_normal.append(stem_stopword_token_normal)

    # --------------------------- positional index ---------------------------
    # creating positional index
    docLen = len(stem_stopword_token_normal)
    for pos in range(docLen):
        term = stem_stopword_token_normal[pos]
        if term not in positional_index.keys(): # first visit of this term in all documents
            positional_index[term] = {'tot_freq': 1, 'doc_ids': {docID: {'doc_freq': 1, 'positic
        else:
            positional_index[term]['tot_freq'] += 1
            if docID not in positional_index[term]['doc_ids'].keys(): # first visit of this tern
                positional_index[term]['doc_ids'][docID] = {'doc_freq': 1, 'positions': [pos]}
            else: # not first visit of this term in this document
                positional_index[term]['doc_ids'][docID]['doc_freq'] += 1
                positional_index[term]['doc_ids'][docID]['positions'].append(pos)

end = time.time()
print("Indexing all documents took about {:.2f} seconds".format(end - start))
```

۲. پرسمان‌های زیر را در نظر بگیرید

الف) یک پرسمان دشوار (مانند "تحریم هسته‌ای" آمریکا ! ایران)

ب) یک پرسمان از کلمات نادر (مانند اورشلیم ! صهیونیست)

برای هر کدام از موارد فوق پرسمان مورد استفاده در فاز یک را تکرار کنید و عملکرد دو موتور بازیابی را از نظر سرعت بازیابی اسناد و کیفیت رتبه‌بندی اسناد مرتبط مقایسه نمایید.

الف) خروجی فاز سه:

```
171 results in 3.417 s:
```

https://www.farsnews.ir/news/14001214000789/نماینده-کلیمیان-در-مجلس-دهم--کارتل‌ها-بر-تصمیم-ات-هیأت-حاکمه-آمریکا

https://www.farsnews.ir/news/14001217000665/خباثت‌های-آمریکا-در-برجام-روسیه-و-چین-را-هم-به-این-کشور-بدبین-کرد-

https://www.farsnews.ir/news/14001213000089/آمریکا-با-بحران آفرینی-به-حیات-خود-ادامه-می-ده-د

https://www.farsnews.ir/news/14001211000321/آمریکا-دولت‌های-متحد-خود-را-در-زمان-اضطرار-تنه-ا-می‌گذارد

https://www.farsnews.ir/news/14001211000898/سود-مافیای-اسلحه‌سازی-آمریکا-در-نا امن-بودن-جها-ن-است

https://www.farsnews.ir/news/14001214000825/آمریکا-رژیمی-مافیایی-است-و-مردم-در-تصمیمات-حا-کمان-آن-جایگاهی-ندارند

https://www.farsnews.ir/news/14001212000725/آمریکا-برای-فروش-تسلیحات-خود-به-ایجاد نا امنی-و-بحران آفرینی-نیاز-دارد

https://www.farsnews.ir/news/14001204000444/آمریکایی-در-جام-یاشاردوغو-جردن-با-روز-هم-می-۲۸-آید

https://www.farsnews.ir/news/14001213000328/کارتل‌های-اقتصادی-رؤسای-جمهور-آمریکا-را-تعیین-می‌کنند

https://www.farsnews.ir/news/14001211000220/نماینده-کلیمیان-در-مجلس-منافع-رژیم-مافیایی-آم-ریکا-در-ایجاد-نا امنی-است

زمانی که طول میکشد بازیابی صورت گیرد:

```
start = time.time()
res = es.search(index=index_name, body=query_json, _source= ["url"])
res = dict(res)
end = time.time()
print("searching documents took about {:.2f} seconds".format(end - start))
```

```
searching documents took about 6.31 seconds
```

خروجی فاز یک:

خروجی ای نشان داده نمی‌شود زیرا در مدل بولین، یک داکیومنت یا با کوئری شباهت دارد و یا ندارد و میزان شباهت و رنکینگ درنظر گرفته نمی‌شود و از آنجایی که هیچ یک از اسناد نیست که هم واژه آمریکا را داشته باشد، همه واژه ایران را نداشته باشد و هم اینکه "تحریم هسته‌ای" در آن آمده باشد پس هیچ خروجی ای را نشان نمی‌دهد (در اسناد بازیابی شده توسط فاز سه هم موارد گفته شده قابل مشاهده است، به عنوان مثال در اولین

سند بازگردانده شده "تحریم هسته‌ای" نیست اما چون رتبه بندی هم داریم با توجه به score ای که کلمه آمریکا گرفته است، به عنوان سند اول بازگردانده شده است).

```
Query:  "تحریم هسته ای" آمریکا ! ایران"
--------------------------------------------------
no result
```

زمانی که طول می‌کشد بازیابی صورت گیرد:

```
Query:  "تحریم هسته ای" آمریکا ! ایران"
--------------------------------------------------
searching documents took about 1.76 seconds
no result
```

استفاده از elasticsearch نتایج بهتری را نشان میدهد اما زمان بازیابی اسناد در با استفاده از positional indexing خیلی کمتر است.

ب) خروجی فاز سه:

هیچ سندی بازیابی نمی‌شود.

```
0 results in 2.144 s:
```

زمان بازیابی در فاز سه:

```
searching documents took about 0.75 seconds
```

خروجی فاز یک:

هیچ سندی بازیابی نمی‌شود.

```
Query:  اورشلیم ! صهیونیست
--------------------------------------------------
no result
```

هیچ سندی نیست که کلمه اورشلیم را داشته باشد اما صهیونیست را نداشته باشد.

```
Query:  اورشلیم ! صهیونیست
--------------------------------------------------
searching documents took about 0.58 seconds
no result
```

زمان بازیابی در فاز یک کمتر است.

۳. با ذکر علت بیان کنید شما به عنوان کاربر استفاده از کدام مدل را ترجیح می‌دهید.

توجه: برای بررسی دقت رتبه‌بندی، بررسی سه سند اول کافیست.

پیاده سازی فاز سوم و استفاده از elasticsearch و تکنیک Bulk API. درست است که در فاز اول، سرعت بازیابی اسناد بیشتر بود، اما کیفیت اسناد بازگردانده شده در فاز سوم بهتر است و بعلاوه اینکه رتبه بندی نیز شده است، در نتیجه با توجه به معیار happiness و مقدار relevance بودن، استفاده از bulk api بهتر است.

۳-۳ تصحیح املایی