بازیابی پیشرفته اطلاعات

دکتر بیگی

گزارش فاز اول پروژه

فاطمه هادی‌زاده ۹۵۱۰۵۹۰۲
فائزه پویامهر ۹۵۱۰۵۴۴۳
فاطمه باقری ۹۵۱۰۵۴۱۹

بخش ۱. پیش‌پردازش اولیه

این بخش در فایل PrepareText قرار دارد.

در این بخش 4 کلاس تعریف شدند.

کلاس read_file:

عملکرد تابع های این کلاس: با ورودی گرفتن آدرس فایل های مورد نیار به خواندن فایل ها و در نهایت خروجی دادن آن ها در قالب لیستی از دیکشنری هایی با دو کلید title و description و value هایی متناسب با مقدار این دو tag به ازای هر مستند در فایل خوانده شده می پردازد.

کلاس GenericPreprocessor:

عملکرد تابع های این کلاس:

تابع preprocess تابع اصلی این کلاس می باشد که وظیفه ی پیش پردازش مستندات و query را بر عهده دارد.

تابع set_stopwords و remove_stopwords در این کلاس وظیفه ی پیدا کردن کلمات پر تکرار و حذف کردن آنها از مستنداتمون را بر عهده دارد.

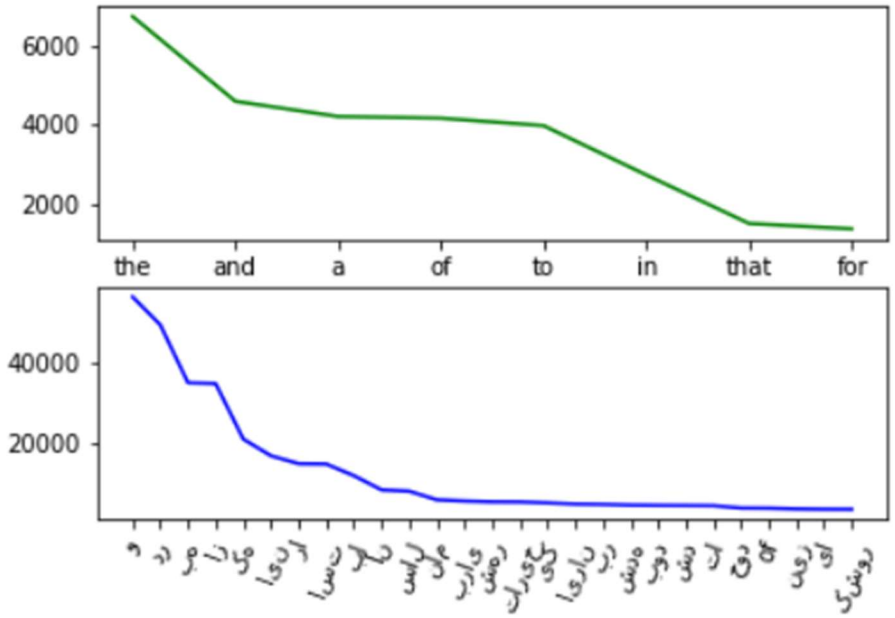
کلاسPersianPreprocessor و کلاس EnglishPreprocessor هم مشابه هم هستند که هرابجکتی از این دو کلاس از جنس کلاس GenericPreprocessor محسوب می شود.هر کدام از این دو کلاس تابع های Normalize و stem و tokenize مخصوص به خود را با کتابخانه های مرتبط به زبانشون (کتابخانه ی NLTK برای پیش پردازش انگلیسی و کتابخانه ی hazm برای پیش پردازش فارسی)پیاده سازی کردند.

نتیجه ی اجرای این بخش:

مشاهده ی کلمات پر تکرار در متون انگلیسی و فارسی داده شده

```
high accurance words in eng docs are:
the 6723
and 4588
a 4206
of 4163
to 3975
in 2734
that 1512
for 1375
_____________________
high accurance words in persion docs are :
56460 و
49496 در
35064 به
34869 از
21033 که
16875 این
14905 را
14816 است
11906 با
8374 آن
8039 سال
5896 نام
5587 برای
5398 شهر
5358 تاریخ
5125 یک
4846 ایران
4755 بر
4576 شده
4510 بود
4479 شد
4438 تا
3846 خود
of 3809
3628 نیز
3558 یا
3537 کشور
```

نماش هستوگرام کلمات پر تکرار در متون فارسی و انگلیسی داده شده



بخش ۲. نمایه‌سازی

کد این بخش در پوشه Index و فایل Indexer.ipynb قرار دارد. توابع کلاس Indexer در ۳ دسته‌ی زیر قرار می‌گیرند:

- اضافه و حذف سند: با استفاده از توابع add_doc و del_doc می‌توان به صورت پویا سند را به نمایه اضافه و از آن حذف کرد. برای این منظور متدهایی برای اضافه کردن ترم‌های هر مستند به ایندکس‌های Positional و Bigram و در صورت لزوم حذف ترم از نمایه‌ها نوشته شده است.
- بارگذاری و ذخیره‌سازی ایندکس و posting list هر یک از ایندکس‌های Positional و Bigram: کلاس را با استفاده از کتابخانه pickle می‌توان در فایل ذخیره و از آن بارگذاری کرد (توابع save_index و load). posting list ایندکس Bigram را می‌توان با توابع save_bigram_posting و load_bigram_posting ذخیره و لود کرد. posting list ایندکس Positional را می‌توان با سه روش بدون فشرده‌سازی، فشرده‌شده با روش Gamma Code و فشرده‌شده با روش Variable Byte ذخیره و بارگذاری کرد (توابع save_posting و load_posting). که فشرده‌سازی با استفاده از کد بخش بعدی انجام می‌شود. با استفاده از اندازه فایل‌های ذخیره‌شده می‌توان میزان حافظه‌ی اشغال شده در هر یک از روش‌های فشرده‌سازی را به دست آورد.
- مجموعه‌ای از توابع get که از خروجی آن‌ها برای پاسخ به پرسش‌های خواسته شده و همچنین در بخش بازیابی و اصلاح پرسمان استفاده خواهد شد.

```
____________________
Enter the section number:
    2
Enter question number:
    1
  -> Indexing is done.

____________________
Enter the section number:
    2
Enter question number:
    2
Enter the term:
    meditation
  -> [94, 327, 373, 801, 828, 863, 951, 963, 1001, 1219, 1384, 1548, 1555, 1605, 1608, 1722, 1872, 2052, 2213, 230
0, 2402, 2519]
```

```
_____________________
Enter the section number:
    2
Enter question number:
    3
Enter the term:
    4
  -> {124: {'title': [0], 'description': [16]}, 486: {'title': [0]}, 537: {'title': [0]}, 550: {'description':
[0]}, 574: {'description': [2]}, 831: {'description': [9]}, 1157: {'description': [20]}, 1213: {'title': [0]}, 124
4: {'title': [0]}, 1354: {'description': [24]}, 1376: {'title': [0]}, 1473: {'title': [0]}, 1484: {'description':
[18]}, 1642: {'title': [0]}, 1647: {'description': [33]}, 1746: {'description': [6]}, 1818: {'description': [48]},
1930: {'title': [0]}, 2052: {'title': [0]}, 2080: {'description': [12]}, 2113: {'title': [0]}, 2298: {'title':
[0]}, 2318: {'title': [0]}, 2337: {'title': [0]}, 2379: {'title': [0]}}

_____________________
Enter the section number:
    2
Enter question number:
    4
Enter the bigram:
    dr
  -> ['drudgeri', 'drug', 'dress', 'dri', 'childdriven', 'kidron', 'dronesey', 'address', 'hundr', 'jawdrop', 'cedr
ic', 'dragonfli', 'dramat', 'hendrik', 'grandchildren', 'communitydriven', 'drive', 'drinkabl', 'drink', 'dragon',
'dandrea', 'papandr', 'drybath', 'drama', 'drummer', 'bedrock', 'dreamt', 'dread', 'bedridden', 'draconian', 'budru
', 'edri', 'bedroom', 'grandchildrenscal', 'adrift', 'drummond', 'firehydr', 'andraka', 'hydrogen', 'drift', 'drier
', 'purposedriven', 'drum', 'syndrom', 'quadrotor', 'drove', 'adrian', 'curiositydriven', 'drumkit', 'drexler', 'da
tadriven', 'andrew', 'backdrop', 'driverless', 'draft', 'cdrom', 'dreamer', 'chandran', 'dr', 'marketdriven', 'eard
rum', 'backyardrocketri', 'drori', 'driver', 'drone', 'augmentedr', 'drew', 'techdriven', 'quadrapleg', 'wondrou',
'dreami', 'audrey', 'alessandra', 'hydrant', 'drinkwat', 'rodriguez', 'driven', 'adriann', 'alejandro', 'drivethru
', 'drivebi', 'android', 'dre', 'children', 'sandra', 'drawback', 'drastic', 'ramachandran', 'fundrais', 'seadragon
', 'andra', 'childreni', 'dropout', 'liquidrepel', 'hadron', 'cathedr', 'overdramat', 'melodrama', 'raindrop', 'ald
rin', 'droid', 'odedra', 'drugresist', 'laundri', 'drywal', 'xdrtb', 'drag', 'drought', 'onehundredth', 'hydrocarbo
n', 'wardrob', 'alessandro', 'selfdriv', 'dreadnoughtu', 'dreamscap', 'daydream', 'schoolchildren', 'andr', 'hindra
nc', 'dream', 'drop', 'andrea', 'dreger', 'drill', 'drunk', 'sandrin', 'draw', 'crowddriven', 'drawn', 'drugrel', '
rodrigo']
```

بخش ۳. فشرده‌سازی نمایه‌ها

کد این بخش در پوشه Index/Compression و در فایل‌های GammaCode.ipynb و VariableByte.ipynb قرار دارد. هر یک از فایل‌ها شامل دو کلاس Compressor و Decompressor می‌باشد. با فراخوانی تابع write_compress_to_file از کلاس Compressor، با ورودی ایندکس Positional ساخته شده با استفاده از کلاس Indexer، ایندکس را با روش موردنظر فشرده کرده و به صورت بایت به بایت در فایل ذخیره می‌کنیم. با فراخوانی تابع decompress_from_file از کلاس Decompressor، نمایه فشرده‌شده را از فایل خوانده و آن را با توجه به روش فشرده‌سازی decode کرده و از حالت فشرده خارج می‌کنیم و ایندکس Positional اولیه را بازسازی می‌کنیم. برای اندازه‌گیری حافظه اشغال شده قبل و بعد از فشرده‌سازی نمایه را در یک فایل ذخیره کرده و سایز آن فایل را اندازه می‌گیریم. میزان حافظه اشغال‌شده قبل و بعد از هر فشرده سازی به صورت زیر است:

```
_____________________
Enter the section number:
    3
Enter question number:
    1
 -> For English docs Indexer:
       Not compressed index size: 3061159 Bytes
       Variable byte index size: 1003153 Bytes

 -> For Persian docs Indexer:
       Not compressed index size: 23474812 Bytes
       Variable byte index size: 8262065 Bytes


_____________________
Enter the section number:
    3
Enter question number:
    2
 -> For English docs Indexer:
 -> For English docs Indexer:
       Not compressed index size: 3061159 Bytes
       Gamma code index size: 1065569 Bytes

 -> For Persian docs Indexer:
       Not compressed index size: 23474812 Bytes
       Gamma code index size: 9005484 Bytes
```

بخش ۴. اصلاح پرسمان

در این بخش یک کلاس edit_query تعریف شده. تابع is_english زبان پرسمان را مشخص می کند. تابع jac_card() با استفاده از تابع bi_gram() فاصله‌ی جاکارد دو واژه را به دست می آورد. همچنین تابع edit_distance() فاصله‌ی ویرایشی دو واژه را به دست می آورد. در تابع edit_query() در آغاز با استفاده از تابع bi_gram() و توابع تعریف شده در بخش های پیشین، کلمات نزدیک از نظر bigramها به هر token در پرسمان را پیدا می کنیم. سپس فاصله‌ی جاکارد این کلمات با واژه‌ی اصلی را حساب می کنیم. 10 واژه که نزدیک ترین فاصله‌ی جاکارد را دارند انتخاب می کنیم. از بین این واژگان، واژه‌ای که کمترین edit distance را با واژه‌ی اصلی دارد را به عنوان پاسخ برمی‌گردانیم.

```
_____________________
Enter the section number:
    4
Enter question number:
    1
Enter the query:
    porpose
  -> purpos


_____________________
```

```
_____________________
Enter the section number:
    4
Enter question number:
    2
Enter first word:
    forest
Enter second word:
    force
  -> 0.3

_____________________
Enter the section number:
    4
Enter question number:
    3
Enter first word:
    forest
Enter second word:
    force
  -> 3

_____________________
```

بخش ۵. جست‌وجو و بازیابی اسناد

یک کلاس Searcher داریم که توابع جست‌وجو در آن نوشته شده‌اند. تابع search() و proximity_search() به ترتیب جست‌وجو به روش های عادی و proximity را انجام می‌دهند. تابع query_weight() وزن یک پرسمان را برمی‌گرداند. تابع doc_weight() با دریافت یک پرسمان و لیستی از documentهای مرتبت، با محاسبه‌ی وزن هر مستند، بهترین مستندات را برمی‌گرداند. تابع search() ابتدا مستندات شامل کلمات پرسمان را پیدا می کند و با استفاده از query_weight() و doc_weight() مرتبت‌ترین مستندات را پیدا می‌کند. تابع proximity_search() با استفاده از ساختمان‌داده‌ی heap، مستنداتی که واژگان پرسمان در آن‌ها با فاصله‌ی K آمده‌باشند را پیدا می‌کند و سپس مانند تابع search() با استفاده از query_weight() و doc_weight() مرتبت‌ترین مستندات را برمی‌گرداند.

```
_____________________
Enter the section number:
    5
Enter question number:
    1
Enter the query:
    daniel show
Enter search field (title/description) or press Enter to skip:

  Scored tf-idf Search result for "daniel show" is:
  -> [2335, 2432, 1026, 1159, 1191, 1561, 1947, 1025, 3, 1028]
  Enter a doc_id to load and show the doc's content(press Enter to skip):
        1026
{'title': 'A flirtatious aria', 'description': 'Can opera be ever-so-slightly sexy? The glorious soprano Danielle d
e Niese shows how, singing the flirty "Meine Lippen, sie küssen so heiss." Which, translated, means, as you might g
uess: "I kiss so hot." From Giuditta by Frans Lehár; accompanist: Ingrid Surgenor.'}
------
     doc_id:


_____________________
Enter the section number:
    5
Enter question number:
    2
Enter the query:
    daniel show
Enter the window size:
    4
  Proximity search with K = 4 Search result for 'daniel show' is:
  -> [1561, 1026, 1159]
  Enter a doc_id to load and show the doc's content(press Enter to skip):
        1561
{'title': "For argument's sake", 'description': 'Why do we argue? To out-reason our opponents, prove them wrong, an
d, most of all, to win! Right? Philosopher Daniel H. Cohen shows how our most common form of argument -- a war in w
hich one person must win and the other must lose -- misses out on the real benefits of engaging in active disagreem
ent.'}
------
     `   ``

_____________________
Enter the section number:
    5
Enter question number:
    1
Enter the query:
    رشته
Enter search field (title/description) or press Enter to skip:

  Scored tf-idf Search result for "رشته" is:
  -> [238, 346, 1536, 1, 1538, 1027, 1028, 1541, 519, 1543]
  Enter a doc_id to load and show the doc's content(press Enter to skip):
        238
{'title': 'رشته مروارید', 'description': '[[پرونده:Rshtay-mrwari.jpg|200px|بندانگشتی|روجلد رشته ی مروارید چاپ ۱۳۸
رشته مروارید\'\'\' (به کردی سورانی: \'\'\'رشتەی مرواری\'\'\') نام کتابی کردی است نوشتهٔ [[علاءالدین سجاد\'\'\'\n\n]]۸
الدین سجادی , انتشارات\u200cرشته ی مروارید , علا*\n==منابع==.های کوتاه است\u200cی]]. که شامل ضرب المثلها، جوک و قصه
رشته ی مروارید , علاالدین سجادی *\n زاده , ۲۰۰ صفحه - وزیری (شومیز) - چاپ 1 سال ۱۳۷۳- ۲۰۰۰ نسخه - ۲۰۰۰ ﷼\u200cت افخم
رشته ی مروارید , علاالدین سجادی , انتشارات "نامعلوم " , ۸۷۲ صفحه , ۱۳۸۸- کامل *\n, انتشارات محمدی , سقز ۱۳۷٤ - ٤ جلد
در ۱ جلد\n\n{{کتاب-خرد}}\n\n[[ده:کتاب\u200cهای کردی]]'}
------
     doc_id:


_____________________
Enter the section number:
    5
Enter question number:
    1
Enter the query:
    طلوع خورشید
Enter search field (title/description) or press Enter to skip:

  Scored tf-idf Search result for "طلوع خورشید" is:
  -> [985, 255, 689, 1507, 17, 766, 1311, 1500, 438, 1202]
  Enter a doc_id to load and show the doc's content(press Enter to skip):
        985
{'title': 'امپرسیون، طلوع آفتاب', 'description': '[[پرونده:Claude Monet, Impression, soleil levant.jpg|30بندانگشتی|
امپرسیو\'\'\'\n\n[[امپرسیون، طلوع خورشید\'\'\' اثر نقاش فرانسوی \'\'\'کلود مونه\'\'\' (۱۸۴۰ - ۱۹۲۶ میلادی)\'\'\'|0px
نام اثری است از نقاش سرشناس [[فرانسه|فرانسوی]]، [[کلود {{Impression soleil levant|به فرانسوی}} \'\'\'ن، طلوع خورشید
مونه]] ([[۱۸۴۰ (میلادی)|۱۸۴۰]] - [[۱۹۲۶ (میلادی)|۱۹۲۶]] میلادی) که جنبش هنری [[امپرسیونیسم]] نام خود را از آن گرفته\u2
با اینکه تحقیقات جدید فاش کرد که تاریخ پایان ) است\u200cاین اثر که به سال [[۱۸۷۲ (میلادی)|۱۸۷۲]] امضا شده\n\n.است00
```

کنسول.

کد کنسول در فایل console.ipynb قرار دارد. با فراخوانی تابع console، سامانه بازیابی شروع به کار می‌کند. ابتدا متغیرهای مورد نیاز مقداردهی اولیه می‌شوند. و مستندات فارسی و انگلیسی پیش‌پردازش می‌شوند. مستندات پیش‌پردازش شده را در دو متغیر eng_docs و per_docs نگه‌می‌داریم. سپس هر بار از ورودی شماره بخش و شماره سوال آن بخش را می‌گیرد و متناسب با سوال ورودی‌ها لازم را گرفته و خروجی موردنظر را تولید می‌کند.

```
console()
```

```
Sections:
 1: Preprcess
  Questions:
    1: Get text input and display its words after pre-processing
    2: Show high occurrence words for English and Persian docs

 2: Indexing
  Questions:
    1: Index English and Persian docs
    2: Get term input and display its posting list
    3: Get term input and display its posting list with positions
    4: Get bigram input and display its posting list

 3: Compressing
  Questions:
    1: Compress Index with variable byte and show occupied memory before and after compression
    2: Compress Index with gamma code and show occupied memory before and after compression
    3: Store indexes in file and load them from file
    2: Show high occurrence words for English and Persian docs

 4: Edit query
  Questions:
    1: Show edited query
    2: Calculate the jaccard distance of two words
    3: Calculate the edit distance of two words

 1: Search
  Questions:
    1: Scored search in vector space with lnc-ltc method
    1: Proximity search with enterd K in vector space with lnc-ltc method

Enter 'exit' to terminate.

_____
```