مبانی هوش محاسباتی (پروژه پنجم)

کیارش آستان‌بوس ۴۰۰۱۲۶۲۵۷۰
محمدرضا نادری ۴۰۰۱۲۶۲۳۸۶

مهر ۱۴۰۳

فهرست مطالب

۱) فاز ۱

۱).۱ متد های استفاده شده

متد های استفاده شده اکثرا متد های رایج در پردازش زبان هستند و صرفا چند متد با توجه به موضوع دیتاست اضافه شده.

Methods

- Normalizing : n’t -> not & ’re -> are
- پاک کردن لینک ها چون محتوا مفیدی ندارند
- پاک کردن کد ها
- پاک کردن اعداد. اعداد شاید در بعضی جاها مثل ورژن پایتون مفید باشند اما با پاک کردن آنها خروجی بهتر بود.
- پاک کردن تگ های xml
- بردن تمام کلمات به حروف کوچک
- استفاده از کلمات اصلی به جای نسخه خلاصه شده مثل integer & int که در نهایت این مورد استفاده نشد ولی در نمایش سه بعدی این کلمات نزدیک یکدیگر بودند
- توکنایز کردن
- lemmetize
- پاک کردن stopword ها
- در ابتدا از تابعی برای اصلاح غلط املایی استفاده شده بود که با توجه به وجود اصطلاح های خاص دیتاست و زمانبر بودن استفاده نشد

۲) فاز ۲

۱.(۲ Word Vectors

دیتای پیش پردازش شده را از فاز یک گرفته و دو ستون title و body رو با یکدیگر combine کردیم. بعد از tokenize کردن این دیتای combine شده، آن را به word۲vec دادیم و خروجی مناسبی رو دریافت کردیم. همچنین هایپر پارامتر ها به صورت زیر انتخاب شده اند

- **Vector Size : ۱۵۰** سایز های دیگر هم مثل ۱۰۰ و ۵۰ و ۲۰۰ امتحان شد ولی یا خیلی بزرگ و پیچیده بودند برای دیتاست یا خیلی هم ساده بودند که در نهایت ۱۵۰ بهترین سایز بود
- **Window : ۵** ویندو ۳ و ۷ هم امتحان شد اما مقدار رایج که نتیجه بهتری هم به ما میداد ویندو ۵ بود. چون نه خیلی به صورت لوکال پردازش میکند نه خیلی کلمات دور را استفاده میکند.
- پارامتر های دیگر نظیر ایپاک و sg نیز بررسی شده اند.

Testing Vectors

```
word2vec_model.wv.most_similar('gmail')[:10]
```

```
[('email', 0.6617672443389893),
 ('satheezkumar', 0.6415207386016846),
 ('rediff', 0.6091378331184387),
 ('mailgun', 0.6057800650596619),
 ('wex', 0.6016517281532288),
 ('gsuite', 0.6000697612762451),
 ('rohanfile', 0.5981631278991699),
 ('abcdf', 0.5897029638290405),
 ('mail', 0.5784346461296082),
 ('seleniumtest', 0.5765087604522705)]
```

```
word2vec_model.wv.most_similar('python')[:10]
```

```
[('pyenv', 0.5758589506149292),
 ('guppy', 0.5722858905792236),
 ('pyparsing', 0.57035893201828),
 ('numpy', 0.5672104954719543),
 ('pypdf', 0.5635924935340881),
 ('komodo', 0.547966480255127),
 ('mkl', 0.5477471351623535),
 ('multiplataform', 0.5427808165550232),
 ('argparser', 0.541533350944519),
 ('pyth', 0.5366126298904419)]
```

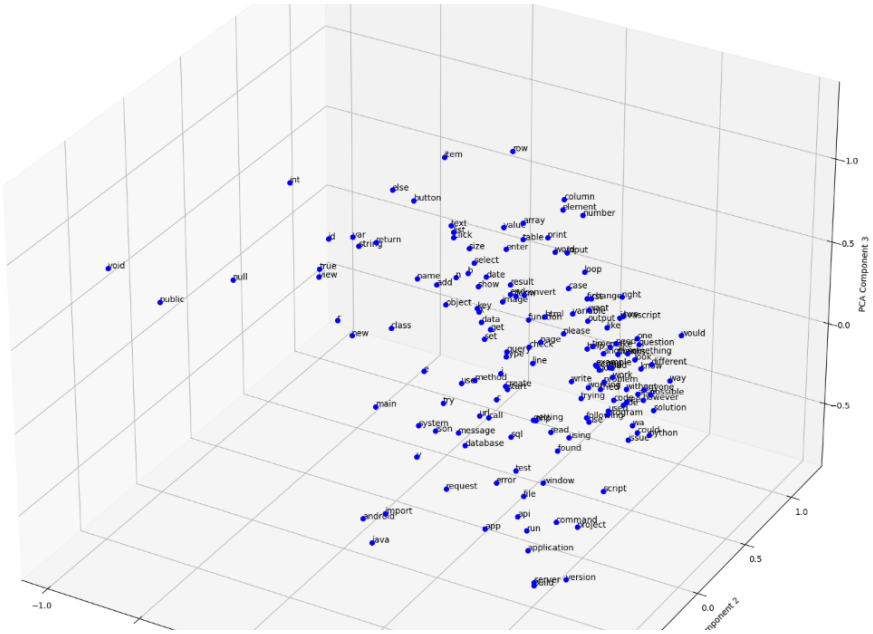
```
word2vec_model.wv.most_similar('google')[:10]
```

```
[('appscript', 0.6758095026016235),
 ('appindexing', 0.638527512550354),
 ('hangout', 0.6181865930557251),
 ('ndb', 0.6133236289024353),
 ('aww', 0.5981616377830505),
 ('shiv', 0.5977445244789124),
 ('xjr', 0.5939909815788269),
 ('appinvite', 0.587768018245697),
 ('zza', 0.5862111449241638),
 ('cameraposition', 0.5849073529243469)]
```

شکل ۱:

همان طور که در عکس بالا مشخص است کلمات مرتبط به هم، بردار های نزدیک به یکدیگر دارند مثل python،pyenv یا کلمات مرتبط به google یا gmail که همگی نزدیک به یک دیگرند.

Plot ۳D ۱.۱.(۲



شکل ۲:

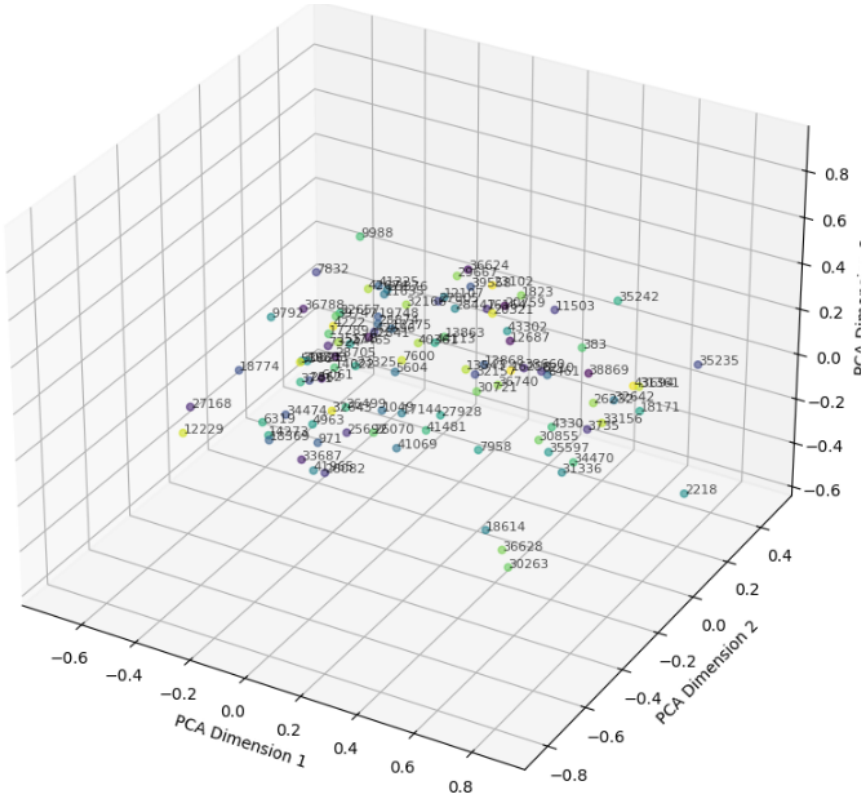
عکس بالا دارای ۱۰۰ کلمه است و در نمودار با کلمات بیشتر دسته بندی کلمات واضح تر است. به عنوان مثال کلماتی مثل string ، int که مربوط به نوع متغیر هستند نزدیک به یک دیگرند. یا کلماتی مثل private ، public که در کلاس ها کاربرد دارند نزدیک یک دیگرند. اما در قسمتی از نمودار تراکم داده ها بالاست که دلایلش میتواند موارد زیر باشد:

- بالانس نبودن دیتاست
- بهینه نبودن پیش پردازش یا پارامتر های مدل
- چون بعد وکتور ها بالاست، با کاهش ابعاد روی یکدیگر افتاده اند ولی در بعد های بالاتر به خوبی جدا شده اند

Document Vectors ۲.(۲

برای ساخت بردار داکیومنت ها، میانگین بردار های کلمات داکیومنت را محاسبه میکنیم. استفاده از idf tf هم امتحان شد که وزن کلمات مهم تر بیشتر باشد اما با مشکل memory برخوردیم.

Plot ۳D ۱.۲.(۲



شکل ۳:

به عنوان مثال داکیومنت های ۱۸۶۱۴ ، ۳۶۶۲۸ نزدیک یکدیگرند و وقتی به محتوای آنها نگاه میکنیم، هر دو در مورد جاوا هستند. قابل انتظار است که مشکلات نمودار بخش قبل رو در این قسمت هم مشاهده میکنیم.

Query ۳.(۲

در این بخش بردار داکیومنت ها هم با استفاده از میانگین گیری محاسبه شده هم با استفاده از tf. idf

```
Query: pythonic way kill thread period timer

Most similar questions:
1. would accurate solution multiple consecutive timer java android word best option delay next piece code current timer ha finished background information question u
nclear practice coding would like create free app w ad help practice freediving basic idea several timer run example minute breath hold breath minute minute breath h
old breath minute second minute breath hold breath minute second second usually round breath holding lead consecutive timer first thought use timer class know pause
code timer finished timer ran simultaneously read countdownlatch handler enough experience judge would best purpose python version used sleep worked fine longer time
r wa inaccurate second problem another option believe use regular timer run loop displayed text know infinite loop would crash application interfere timer accuracy h
a negative side effect help thought would appreciated thanks (Similarity: 0.80)
2. want call function certain time irrespective code timer select (Similarity: 0.78)
3. working game one main component countdown timer however timer delayed able deduce would like decrement per second however seems decrementing every second timer se
t code print screen code code help greatly appreciated (Similarity: 0.77)
4. want launch method separated thread periodically every minute wa using realize timer cause memory leak need remove timer use task first idea launch task following
way taskwork task factory startnew dowork dowork method private void dowork true stuff thread sleep way launch task avoiding approach thanks (Similarity: 0.77)
5. seen many stack google answer delay function suggestion timer way delay function using timer way event event eg onclick blur done check event end call function la
zy manner without using timer (Similarity: 0.76)
```

شکل ۴: Average

برای مقایسه بهتر، متن بعد از پیش پردازش پرینت شده و میتوان به وضوح دید نتایج همگی مرتبط به کوئری هستند. کوئری مربوط به تایمر، پایتون و ترد هست که خروجی ها همگی در مورد تایمر و ترد هستند.

```
Query: pythonic way kill thread period timer

Most similar questions:
1. would accurate solution multiple consecutive timer java android word best option delay next piece code current timer ha finished background information question u
nclear practice coding would like create free app w ad help practice freediving basic idea several timer run example minute breath hold breath minute minute breath h
old breath minute second minute breath hold breath minute second second usually round breath holding lead consecutive timer first thought use timer class know pause
code timer finished timer ran simultaneously read countdownlatch handler enough experience judge would best purpose python version used sleep worked fine longer time
r wa inaccurate second problem another option believe use regular timer run loop displayed text know infinite loop would crash application interfere timer accuracy h
a negative side effect help thought would appreciated thanks (Similarity: 0.76)
2. working game one main component countdown timer however timer delayed able deduce would like decrement per second however seems decrementing every second timer se
t code print screen code code help greatly appreciated (Similarity: 0.75)
3. wa wondering would possible create timer isnt bound single form basically im making quiz want user minute answer question minute current form close endscreen appe
ar time must also visible time help would great (Similarity: 0.75)
4. first intention cheating one people tend see flaw everywhere day job interview may required online test javascript timer countdown enough time google answer way s
ee correct wrong timer decreasing variable reach say stop test seems many case possible look source code find name variable force increase using console time world i
nterested always possible manipulate timer variable using console way prevent (Similarity: 0.75)
5. want call function certain time irrespective code timer select (Similarity: 0.75)
```

شکل ۵: tf idf

نتیجه این روش هم مثل روش قبل کاملا مرتبط به کوئری هست و حتی اشتراک هم با یکدیگر دارند. انتظار میرود در تعداد خروجی بالا، این روش عملکرد بهتری داشته باشد چون وزن کلمات مهم بالاتر است. همچین میتوان تاثیر استفاده از tf idf را نتیجه دید که جایگاه خروجی قسمت قبل در این قسمت عوض شده.

۳) فاز ۳

در این فاز ابتدا بر روی ۱۰ درصد از دیتای validation همان پیش پردازش فاز اول انجام شده است. بعد از آموزش مدل، ۵ تا از نزدیک ترین همسایه هر داکیومنت استخراج شد. سپس فاصله هر تگ با بردار داکیومنت مقایسه شد و ۵ تا از نزدیک ترین تگ ها انتخاب شدند و در نهایت مدل ارزیابی شد.

accuracy : ۶۲

این درصد میتواند خوب باشد اما دلایل متفاوتی میتواند وجود داشته باشد که اگر آن ها حل شود یا بهتر شود باعث بهبود کیفیت میشود.

یکی از عوامل میتواند وجود تگ های غیر مرتبط در دیتاست باشد که با متن در تناقض است. یکی از عوامل دیگر توزیع نامتعادل تگ و یا کمبود داده برای ترین میتواند باشد.

ه جای استفاده از KNN میتوان از مدل های پیشرفته تر شبکه عصبی یا مدل های ensemble برای بهبود کیفیت استفاده کرد که به دلیل پیچیدگی در پروژه استفاده نشده است . همچنین مدل نتوانسته همبستگی بین تگ ها را به خوبی یاد بگیرد. یکی دیگر از راه حل ها برای بهبود کیفیت استفاده از مدل پیشرفته تر مثل BERT است.

```
Correct prediction (933):
1. True : ['javascript'], Predicted : ['java', 'javascript', 'sorting', 'substr']
2. True : ['javascript', 'jquery', 'html', 'ajax', 'twitter-bootstrap'], Predicted : ['javascript', 'jquery', 'html', 'mobile']
3. True : ['php', 'html'], Predicted : ['php', 'simpledateformat', 'date', 'java', 'android']

Faild prediction (567):
1. True : ['sql'], Predicted : ['performance', 'subquery', 'database', 'tsql', 'oracle']
2. True : ['c#'], Predicted : ['wpf', 'linq', 'namespaces', 'multithreading']
3. True : ['youtube', 'schema', 'google-search', 'structured-data', 'google-search-console'], Predicted : ['javascript', 'android', 'html', 'cordova', 'json']
```

شکل ۶:

در مورد پیشبینی های درست میتوان دید که تگ ها خیلی خوب پیشبینی شده اند و حتی بیشتر از یک اشتراک داشته اند. اما پیشبینی های نادرست به دو دسته تقسیم شده اند. یا مرتبط به یکدیگرند اما تگ یکسان ندارند مثل مورد اول که در مورد دیتابیس است. و یا هم اصلا به یکدیگر مرتبط نیستند که میتوان نتیجه گرفت مشکل از بردار ها است. در کل این مورد را با افزایش تگ به ازای هر داده میتوان به خوبی بهبود داد.