به نام خدا

گزارش فاز سوم پروژه درس بازیابی اطلاعات

مدرس : دکتر سلیمانی

حسین ابراهیمی کندری ـ ۹۵۱۰۵۳۰۲

۱ نحوه بخش‌بندی و کدهای مربوطه

بخش‌های ۱ تا ۵ این فاز در نوت بوک Phase3.ipynb موجود است و رابط کاربری برای کار کردن با این بخش‌ها در سلول آخر قرار دارد. ابتدا باید تمام سلول‌های بالا اجرا شوند و سپس رابط کاربری مورد استفاده قرار گیرد. بخش ۶ در نوت بوک Part 6.ipynb قرار دارد. برای اجرا کردن آن نیاز است تا فایل data در کنار نوت بوک قرار داشته باشد.

۲ نحوه کار با رابط کاربری

با اجرای بخش آخر نوت بوک Phase3.ipynb منوی زیر ظاهر می‌شود:

```
1. Crawl semanticScholar.com (takes about 10 mins ...)
2. Insert data to ElasticSearch
3. Delete data from ElasticSearch
4. Calculate pageRank
5. Search
6. HITS
7. Exit
order of computing: 1 -> 2 -> 3 or 4 or 6 or 5
please enter your command by number:
```

با وارد کردن عدد مربوطه به هر بخش، تابع مورد نظر آن اجرا می‌شود و ورودی‌های مورد نیاز برای آن بخش از کاربر خواسته می‌شود. فقط دقت شود که برای جستجو بر روی نمایه، اگر pageRank قرار باشد که در امتیاز مربوطه دخالت داده شود، ابتدا باید pageRank داک‌ها محاسبه شود.

۳ بخش اول: پیاده‌سازی Crawler

با استفاده از کلاس spider کتابخانه scrapy و overwrite کردن تابع parse مربوط صفحه‌های وب سایت semantic-Scholar.org خزش شده و field های مورد نیاز استخراج می‌شوند و به فرمت json در فایل "papers_index.json" ذخیره می‌شود.
ورودی‌های این شامل لینک‌های اولیه که به صورت پیش فرض قرار دارند و تعداد صفحه‌های وب برای خزش است که از کاربر گرفته می‌شود. خروجی نهایی برای هر داک به صورت زیر است:


```
Out[4]: {'id': '9665247ea3421929f9b6ad721f139f11edb1dbb8',
         'title': 'Learning Longer Memory in Recurrent Neural Networks',
         'authors': ['Tomas Mikolov',
          'Armand Joulin',
          'Sumit Chopra',
          'Micha{\\"e}l Mathieu',
          "Marc'Aurelio Ranzato"],
         'date': '2015',
         'abstract': 'Recurrent neural network is a powerful model that learns temporal patterns in sequential data. For a lo
        ng time, it was believed that recurrent networks are difficult to train using simple optimizers, such as stochastic g
        radient descent, due to the so-called vanishing gradient problem. In this paper, we show that learning longer term pa
        tterns in real data, such as in natural language, is perfectly possible using gradient descent. This is achieved by u
        sing a slight structural modification of the simple recurrent neural network architecture. We encourage some of the h
        idden units to change their state slowly by making part of the recurrent weight matrix close to identity, thus formin
        g kind of a longer term memory. We evaluate our model in language modeling experiments, where we obtain similar perfo
        rmance to the much more complex Long Short Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997).',
         'references': ['f9a1b3850dfd837793743565a8af95973d395a4e',
          '44d2abe2175df8153f465f6c39b68b76a0d40ab9',
          '07ca885cb5cc4328895bfaec9ab752d5801b14cd',
          'd0be39ee052d246ae99c082a565aba25b811be2d',
          '4ef03716945bd3907458efbe1bbf8928dafc1efc',
          'd1275b2a2ab53013310e759e5c6878b96df643d4',
          'ded103d0613e1a8f51f586cc1678aee3ff26e811',
          'f264e8b33c0d49a692a6ce2c4bcb28588aeb7d97',
          'e9fac1091d9a1646314b1b91e58f40dae3a750cd',
          '1fac520eca9767bfe9a28302494c818642a2b2c7']}
```


۴ بخش دوم: Insert & Delete ElasticSearch

در این بخش با استفاده از داده‌ی قسمت قبل که در دیکشنری data ذخیره شده است، داک‌ها در نمایه paper_index در ElasticSearch بارگذاری می‌شوند. ورودی‌های این بخش host و port برای راه‌اندازی ElasticSearch هستند.

۵ بخش سوم: pageRank

با استفاده از داده‌های ذخیره شده در نمایه و ورودی الگوریتم α که از کاربر خواسته می‌شود، pageRank هر صفحه محاسبه شده و در بخش pageRank هر داک در نمایه ذخیره می‌شود.
اعداد بدست آمده بر اساس ارجاعی است که مقالات crawl شده به هم داده‌اند و گراف بر اساس این تعداد ساخته شده است.
نمونه داک‌های ذخیره شده در ES بعد از اعمال pageRank به شکل زیر است:

```
Out[13]: {'paper': {'id': '9665247ea3421929f9b6ad721f139f11edb1dbb8',
  'title': 'Learning Longer Memory in Recurrent Neural Networks',
  'authors': ['Tomas Mikolov',
   'Armand Joulin',
   'Sumit Chopra',
   'Micha{\\"e}l Mathieu',
   "Marc'Aurelio Ranzato"],
  'date': '2015',
  'abstract': 'Recurrent neural network is a powerful model that learns temporal patterns in sequential data. For a l
ong time, it was believed that recurrent networks are difficult to train using simple optimizers, such as stochastic
gradient descent, due to the so-called vanishing gradient problem. In this paper, we show that learning longer term p
atterns in real data, such as in natural language, is perfectly possible using gradient descent. This is achieved by
using a slight structural modification of the simple recurrent neural network architecture. We encourage some of the
hidden units to change their state slowly by making part of the recurrent weight matrix close to identity, thus formi
ng kind of a longer term memory. We evaluate our model in language modeling experiments, where we obtain similar perf
ormance to the much more complex Long Short Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997).',
  'references': ['f9a1b3850dfd837793743565a8af95973d395a4e',
   '44d2abe2175df8153f465f6c39b68b76a0d40ab9',
   '07ca885cb5cc4328895bfaec9ab752d5801b14cd',
   'd0be39ee052d246ae99c082a565aba25b811be2d',
   '4ef03716945bd3907458efbe1bbf8928dafc1efc',
   'd1275b2a2ab53013310e759e5c6878b96df643d4',
   'ded103d0613e1a8f51f586cc1678aee3ff26e811',
   'f264e8b33c0d49a692a6ce2c4bcb28588aeb7d97',
   'e9fac1091d9a1646314b1b91e58f40dae3a750cd',
   '1fac520eca9767bfe9a28302494c818642a2b2c7'],
  'pageRank': 3.1281574661105475e-08}}
```

۶ بخش چهارم: Search

ورودی‌های این بخش به شکل زیر هستند:

$$search(es,\ title,\ w\_title,\ date,\ w\_date,\ abstract,\ w\_abstract,\ inv\_pageRank = True)$$

که کوئری هر بخش، وزن آن بخش و آیا اینکه pageRank در نتیجه سرچ دخالت داده شود یا نه از کاربر گرفته می‌شود.
با توجه به اینکه مقادیر pageRank های محاسبه شده از 10^{-8} بودند، رابطه‌ی زیر برای دخالت pageRank مورد استفاده قرار گرفت:

$$score^{(d)} = log_{10}(1 + 10^9 \times pageRank^{(d)})$$

تابع امتیازدهی به داک‌ها طوری طراحی شده است که کلمات با ماکسیمم edit distance ۲ می‌توانند به بخش‌های عنوان و متن مقاله match شوند. همچنین در بخش تاریخ، با استفاده از filter مقالاتی که بعد از تاریخ ورودی کاربر، منتشر شده باشند وزن مربوطه به آن‌ها داده می‌شود.

```
In [115]: search(ES, "vision", 3, 2013, 1, "Convolutional Networks", 1, inv_pageRank=True)
```

```
1. title: Rethinking the Inception Architecture for Computer Vision
abstract: Convolutional networks are at the core of most state of-the-art computer vision solutions for a wide variet
y of tasks. Since 2014 very deep convolutional networks started to become mainstream, yielding substantial gains in v
arious benchmarks. Although increased model size and computational cost tend to translate to immediate quality gains
for most tasks (as long as enough labeled data is provided for training), computational efficiency and low parameter
count are still enabling factors for various use cases such as mobile vision and big-data scenarios. Here we are expl
oring ways to scale up networks in ways that aim at utilizing the added computation as efficiently as possible by sui
tably factorized convolutions and aggressive regularization. We benchmark our methods on the ILSVRC 2012 classificati
on challenge validation set demonstrate substantial gains over the state of the art: 21:2% top-1 and 5:6% top-5 error
for single frame evaluation using a network with a computational cost of 5 billion multiply-adds per inference and wi
th using less than 25 million parameters. With an ensemble of 4 models and multi-crop evaluation, we report 3:5% top-
5 error and 17:3% top-1 error on the validation set and 3:6% top-5 error on the official test set.
authors: ['Christian Szegedy', 'Vincent Vanhoucke', 'Sergey Ioffe', 'Jon Shlens', 'Zbigniew Wojna']
date: 2016
10.506389
```

```
2. title: Convolutional networks and applications in vision
abstract: Intelligent tasks, such as visual perception, auditory perception, and language understanding require the c
onstruction of good internal representations of the world (or "features")? which must be invariant to irrelevant vari
ations of the input while, preserving relevant information. A major question for Machine Learning is how to learn suc
h good features automatically. Convolutional Networks (ConvNets) are a biologically-inspired trainable architecture t
hat can learn invariant features. Each stage in a ConvNets is composed of a filter bank, some nonlinearities, and fea
ture pooling layers. With multiple stages, a ConvNet can learn multi-level hierarchies of features. While ConvNets ha
ve been successfully deployed in many commercial applications from OCR to video surveillance, they require large amou
nts of labeled training samples. We describe new unsupervised learning algorithms, and new non-linear stages that all
ow ConvNets to be trained with very few labeled samples. Applications to visual object recognition and vision navigat
ion for off-road mobile robots are described.
authors: ['Yann LeCun', 'Koray Kavukcuoglu', "Cl{\\'e}ment Farabet"]
date: 2010
9.506501
```

همانطور که در شکل بالا مشخص است هر دوی داک‌ها دارای کلمات vision و Convolutional Networks هستند ولی چون مقاله اول در سالی بعد از ۲۰۱۳ منتشر شده است، امتیاز بالاتری گرفته است.

۷ بخش پنجم: HITS

در این بخش ارجاعات نویسندگان به هم دیگر در بین مقالات crawl شده بررسی شده و ماتریس A ساخته می‌شود به این صورت که اگر نویسنده a_i به نویسنده a_j ارجاع داده باشد، مقدار A_{ij} برابر ۱ خواهد بود. حال از روی این ماتریس بردارهای a و h محاسبه می‌شوند و n نویسنده برتر (از کاربر مقدار n گرفته می‌شود) خروجی داده می‌شود.

۸ بخش ششم: Ranking

ابتدا داده خام داده شده با استفاده از تابع pre_processing و ماتریس X که هر سطر آن ۴۶ ویژگی داک i است و ماتریس Y که درایه اول سطر i میزان ارتباط داک i را با مقدار درایه دوم ماتریس Y که در آن id کوئری نگه داشته می‌شود را مشخص می‌کند.

$$X[i] : 46\ features\ of\ document\ i$$
$$Y[i][0] : r_i$$
$$Y[i][1] : qid$$

سپس با استفاده از تابع transform_pairwise زوج‌های $(d_i - d_j, q, +1\ or\ -1)$ ساخته می‌شوند و آموزش SVM بر روی آن‌ها صورت می‌گیرد.

۱.۸ محاسبه NDCG@5

برای محاسبه این معیار ابتدا داک‌های مربوط به یک کوئری جمع‌آوری شده و برای هر جفت غیر تکراری آن‌ها با استفاده از مدل داده شده پیش‌بینی صورت می‌گیرد. حال برای هر دوتایی از داک‌ها یک رابطه بزرگتر کوچکتری داریم و باید از آن‌ها به یک ترتیب کلی برای کل داک‌های مربوط به آن کوئری برسیم. برای این کار گرافی تشکیل می‌دهیم که گره‌های آن داک‌ها هستند و اگر d_i نسبت به d_j مرتبط‌تر باشد به کوئری، یک یال جهت دار از d_i به d_j وصل می‌کنیم. حال با اجرا کردن topological sort بر روی این گراف، ترتیب کلی داک‌ها بر حسب ارتباط با کوئری بدست می‌آید و می‌توان با استفاده از آن و r_i معیار $NDCG@5$ را محاسبه کرد.

۲.۸ Validation

برای ارزیابی بهترین پارامتر C ابتدا نتیجه را بر روی مقادیر 2^{-10} تا 2^{10} ارزیابی کردیم و بهترین نتایج در بازه‌ی 0.25 تا 1 بدست آمد. سپس در بازه‌ی $[0.25, 1]$ به صورت یکنواخت نمونه‌گیری انجام شد و بهترین نتیجه به ازای $C = 0.456$ بدست آمد که

مقدار $NDCG@5$ بر روی داده val برابر با **0.6905** بود. لازم به ذکر است که برای C های بزرگتر از ۱ مشکل همگرایی رخ می‌داد که با بیشتر کردن مقدار max_iter این مشکل مقداری برطرف شد.

Test ۳.۸

مدل با پارامتر $C = 0.456$ بر روی داده‌ی تست نیز اجرا شد و مقدار $NDCG@5$ برای آن، برابر با مقدار **0.6770** بود.