به نام خدا

گزارش پروژه بازیابی پیشرفته اطلاعات

# سامانه بازیابی اطلاعات مقالات علمی

گروه ۳۰

پارسا حسینی

درسا مجدی

امیررضا باقری

# چکیده روند پروژه

این پروژه ۴ بخش اصلی دارد. در بخش اول، ابتدا تسک باقی‌مانده از تمرین چهارم انجام شد. در بخش دوم، الگوریتم Rocchio برای گسترش پرس و جو (query expansion) پیاده سازی شد. سپس در ادامه از موتور جست و جو Elastic Search استفاده کردیم تا داده‌های خود را در آن ایندکس کنیم و بتوانیم بر روی آن داده‌ها query بزنیم. در انتها کل بخش‌های پیاده‌سازی شده از تمرینات ۳ و ۴ و ۵، به یک واسط کاربری متصل شدند.

## ۱. تسک دسته‌بندی مقالات علمی

در این بخش، هدف آن است که مقالات را بر اساس محتوای چکیده آنها، به تعدادی دسته با برچسب موضوع علمی (Field of Study) آنها تقسیم کنیم. ابتدا از داده‌های مربوط به مقالات علمی، که از قبل crawl کرده بودیم، استفاده کرده و تعداد هر مقاله موجود برای موضوعات آن را بدست آوردیم. بر این اساس تصمیم گرفته شد که تنها آن موضوعاتی انتخاب بشوند که بیشتر از ۵۰۰ مقاله برای آنها موجود باشد. سپس از هر کدام از موضوعات انتخاب شده، تعداد ۵۰۰ مقاله انتخاب شد تا در مرحله بعدی داده‌های train و test براساس آن تشکیل شود.

```
[ ] th = 500
    selected_categories = [category for category in categories if (df[df["fieldsOfStudy"].apply(lambda x: x == [category])].shape)[0] > th]
    selected_categories

    ['Computer Science',
     'Engineering',
     'Medicine',
     'Psychology',
     'Materials Science']

[ ] data = pd.DataFrame()
    for category in selected_categories:
        data = pd.concat([data, df[df["fieldsOfStudy"].apply(lambda x: x == [category])].sample(n=th)])

    data['fieldsOfStudy'] = data['fieldsOfStudy'].apply(lambda x: x[0])
    data
```

| | title | abstract | fieldsOfStudy |
|---|---|---|---|
| 15438 | PaGAN: Generative Adversarial Network for Pate... | In recent years, Deep Learning methods have be... | Computer Science |
| 16060 | Corpus Viewer: NLP and ML-based Platform for P... | Corpus Viewer is a production service develope... | Computer Science |
| 6430 | Automatic language identification of telephone... | The paper compares the performance of four app... | Computer Science |
| 5655 | Language Model Adaptation for Statistical Mach... | We explore unsupervised language model adaptat... | Computer Science |
| 9328 | The Enterprise Ontology | This is a comprehensive description of the Ent... | Computer Science |
| ... | ... | ... | ... |
| 27463 | Dielectric Properties of Cryogenic Liquids | Electric-breakdown measurements at 60 Hz have ... | Materials Science |
| 22179 | Nanosecond Pulse Transformers | The transmission-line approach to the design o... | Materials Science |
| 28646 | A High-Gain Boost Converter using Voltage-Stac... | This paper suggests anon-isolated high-gain bo... | Materials Science |
| 29244 | A study of characteristic analysis of the thre... | In this paper, the analytic algorithm using eq... | Materials Science |
| 26066 | Influence of Load Resistance on Higher Harmoni... | The higher harmonic voltages observed in the i... | Materials Science |

*Figure 1. انتخاب مقالات بر اساس موضوعات پرتکرار*

در مرحله بعدی با استفاده از متد train_test_split داده‌های جدا شده را بخش‌های train و تست تقسیم و با تعیین stratify، یکنواخت بودن انتخاب مقالات براساس موضوعشان را تضمین کردیم.

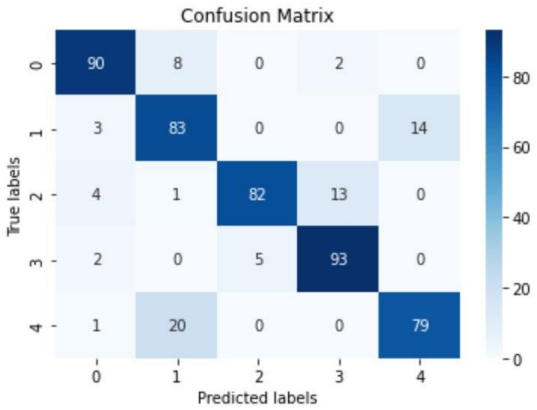
در ادامه با دو روش دسته‌بندی انجام شد. روش اول دسته‌بندی با استفاده از روش سنتی Naïve Bayes است و روش دوم با استفاده از یک مدل بر پایه Transformer پیاده‌سازی شده است.

## ۱.۱ – دسته‌بندی Naïve Bayes

در این بخش از متد TfidfVectorizer کتابخانه sklearn برای تبدیل داده‌های train و تست به ماتریس document term matrix آنها استفاده کرده و با MultinomialNB() دسته‌بندی را انجام دادیم.

در انتها نتایج دسته‌بندی به صورت زیر شد:

```
Accuracy Score : 0.854
Precision Score :  0.8588347281812243
Recall Score :  0.8539999999999999
F1-macro : 0.8545815668979626
-------------------------------------
```




در انتها مدل ذخیره شد تا بتوان بعدا از آن برای پرس‌وجو استفاده کرد. به این صورت که با دریافت یک پرس‌وجو، مدل تشخیص می‌دهد که متن وارد شده متعلق به کدام یک از زمینه‌های علمی است و سپس آن را خروجی می‌دهد. چند نمونه کوئری را در شکل زیر مشاهده می‌کنید:

```
[ ] query = ['voltage power', 'nlp attention', 'emotional human mind', 'materials']
    for q in query:
        nb_classifier.run(q)
        print()
```

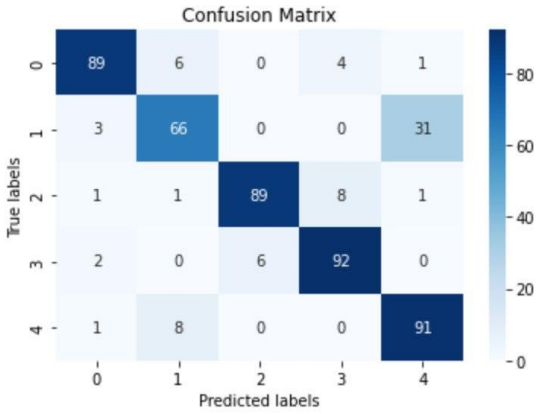
```
Query: voltage power
Predicted Category: ['Engineering'] With Probability: 0.6053348368723953
Execution time: 0.01388406753540039

Query: nlp attention
Predicted Category: ['Computer Science'] With Probability: 0.40300809037074864
Execution time: 0.004285573959350586

Query: emotional human mind
Predicted Category: ['Psychology'] With Probability: 0.4023168755575976
Execution time: 0.003726959228515625

Query: materials
Predicted Category: ['Materials Science'] With Probability: 0.478345723149243
Execution time: 0.013794898986816406
```

## ۱.۲ – دسته بندی با یک مدل بر پایه Transformer

در این بخش ابتدا، توکنایزرها و مدل‌های از قبل آماده شده "distilbert-base-uncased" بارگذاری شدند.

در مرحله بعدی، داده‌های train و test به دیتاست تبدیل شده و tokenizer بر روی آنها اعمال شد تا آنکه داده‌ها آماده ورودی دادن به سیستم شوند. پارامترهای train تنظیم شده و سپس مدل بر روی داده‌های یادگیری fit شد. در انتها نتایج دسته‌بندی به صورت زیر شد:

```
Accuracy Score : 0.854
Precision Score :  0.8594453211537253
Recall Score :  0.8539999999999999
F1 Score : 0.8529452660792325
---------------------------------------
```




در انتها مدل ذخیره شد تا بتوان بعدا از آن برای پرس‌وجو استفاده کرد. به این صورت که با دریافت یک پرس‌وجو، مدل تشخیص می‌دهد که متن وارد شده متعلق به کدام یک از زمینه‌های علمی است و سپس آن را خروجی می‌دهد. چند نمونه کوئری را در شکل زیر مشاهده می‌کنید:

```
[ ]  query = ['voltage power', 'nlp attention', 'emotional human mind', 'materials']
     for q in query:
         tf_classifier.run(q)
         print()
```

```
Query: voltage power
Predicted Category: ['Engineering'] With Probability: 0.4179365634918213
Execution time: 0.19341468811035156

Query: nlp attention
Predicted Category: ['Computer Science'] With Probability: 0.682012140750885
Execution time: 0.18227171897888184

Query: emotional human mind
Predicted Category: ['Psychology'] With Probability: 0.44307082891464233
Execution time: 0.1897449493408203

Query: materials
Predicted Category: ['Materials Science'] With Probability: 0.37241172790527344
Execution time: 0.1737501621246338
```

## ۲. گسترش پرس‌وجو (Query Expansion)

برای گسترش پرس‌وجو از الگوریتم rocchio استفاده کردیم. در این روش ابتدا نتایجی که بیشترین ارتباط را با کوئری داشتند و همچنین نتایجی را که کمترین ارتباط را داشتند، بدست می‌آوریم. سپس بردارهای تعبیه مربوط به نتایج مرتبط و غیر مرتبط را در نظر گرفته و میانگین آنها را به عنوان امتیاز مربوط به آنها محاسبه می‌کنیم. نهایتا بردار تعبیه کوئری را با استفاده از مدل، محاسبه کرده و آن را با امتیاز نتایج مرتبط (rel_score) جمع و امتیاز نتایج غیر مرتبط (irrel_score) را از آن کم می‌کنیم. نتیجه حاصل به عنوان کوئری اصلاح شده بازگردانده می‌شود.

13 lines (12 sloc) | 604 Bytes Raw Blame

```python
import numpy as np

# Query exapnsion for fasttext and transformers
def Rocchio(model, query, k=10):
    indices = model.most_similar(query, False, 2*k)[0]
    relevant_ind = indices[:k]
    irrelevant = indices[k:]
    embedding_rel = [model.doc_embedding['embedding'][i] for i in relevant_ind]
    embedding_irrel = [model.doc_embedding['embedding'][i] for i in irrelevant]
    rel_score = np.mean(embedding_rel, axis=0)
    irrel_score = np.mean(embedding_irrel, axis=0)
    modified_query = model.embed(model.preprocessor.run(query)).reshape(1, -1) + rel_score - irrel_score
    return modified_query
```

نتایج حاصل از کوئری‌های تمرین سوم، پس از گسترش پرس‌وجو در لینک زیر موجود است:

https://docs.google.com/spreadsheets/d/1nZNZJqwV0jO3qidjMFgmN6hK4uoi0eDV0DsqQv3VKjI/edit?usp=sharing

## ۳. موتور جست‌وجو Elastic Search

در این بخش ابتدا یک سرور local برای elastic search ایجاد شده و سپس داده‌های موجود در دیتافریم در آن ایندکس می‌شوند. از elasticsearch نسخه 7 استفاده کردیم زیرا نسخه ۸ آن مشکلاتی ایجاد می‌کرد.

هنگام کوئری نیز، با استفاده از روش match، تعداد k داده مرتبط نمایش داده می‌شوند. خروجی‌های مربوط به کوئری با استفاده از این موتور جست‌وجو در لینک زیر موجود است:

https://docs.google.com/spreadsheets/d/1nZNZJqwV0jO3qidjMFgmN6hK4uoi0eDV0DsqQv3VKjI/edit?usp=sharing

## ۴. رابط کاربری

در نهایت تمامی بخش‌های پروژه به یک رابط کاربری متصل شدند. توضیحات مربوط به راه‌اندازی و استفاده از این رابط کاربری، در بخش README ریپو گیت پروژه توضیح داده شده است.