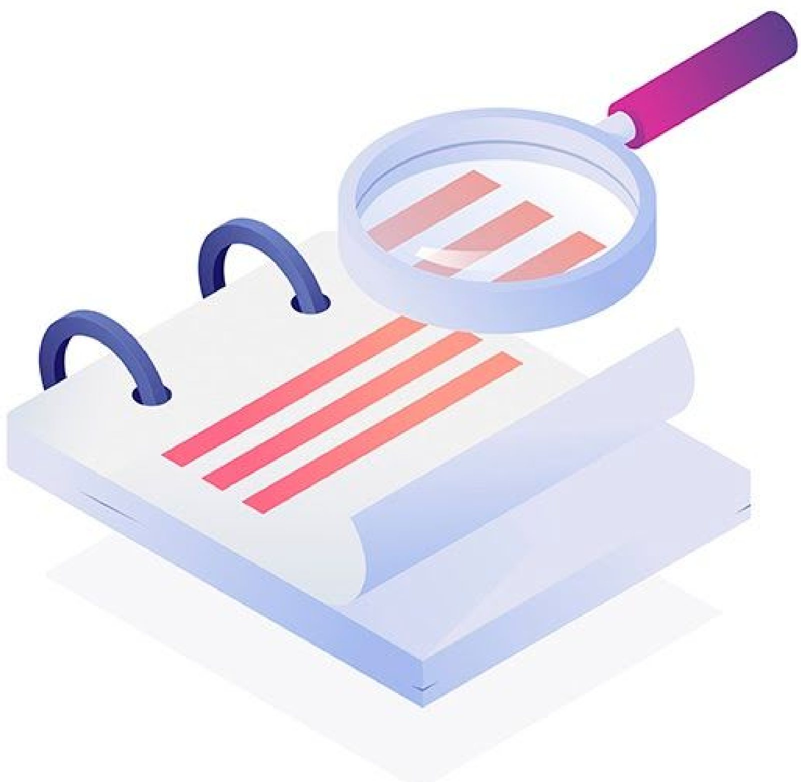
Modern Information Retrieval

Phase #3

Seyed Alireza Fatemi Jahromi

95102062



کد های مربوط در نوت‌بوک TestV0.3.0.ipynb قرار دارند. همچنین رابط کاربری کنسول در main.py در پوشه src قرار دارد، اما نوت‌بوک کامل تر می‌باشد.

کرالر ها در آدرس زیر قرار دارند:

/paper_crawler/paper_crawler/spiders/semanticscholar.py

/paper_crawler/paper_crawler/spiders/researchgate.py

وارد کردن داده‌ها به الاستیک به دو صورت زیر است:

Solution #1

```
[14]: # https://elasticsearch-py.readthedocs.io/en/master/helpers.html#bulk-helpers
# https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-bulk.html


def gendata():
    for idx, item in enumerate(items):
        if item["date"] == "" or not item["date"].isdigit():
            del item["date"]
        yield {
            "_index": "paper-index",
            "_id": item["id"],
            "page_rank": 1.0,
            **item,
        }


bulk(es, gendata())
```

```
[14]: (503, [])
```

Solution #2

```
[ ]: for item in items:
    paper = Paper(meta={"id": item["id"]}, page_rank=1.0, **item)
    paper.save(using=es)
```

نحوه محاسبه pagerank در بخش Calculating page rank نوت‌بوک قرار دارد.

بخش جستجو به همراه نتایج با تاثیر و بدون تاثیر page rank

کویری:

```python
return es.search(
    index="paper-index",
    body={
        "query": {
            # https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-function-score-query.html#function-weight
            "function_score": {
                "functions": [
                    {
                        "filter": {"match": {"title": title_search}},
                        "weight": title_weight,
                    },
                    {
                        "filter": {"match": {"abstract": abstract_search}},
                        "weight": abstract_weight,
                    },
                    # https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-range-query.html
                    {
                        "filter": {"range": {"date": {"gte": year_search}}},
                        "weight": year_weight,
                    },
                    # https://www.elastic.co/guide/en/elasticsearch/reference/current/static-scoring-signals.html
                    # https://www.elastic.co/guide/en/elasticsearch/reference/7.x/query-dsl-rank-feature-query.html#rank-feature-query-saturation
                    {
                        "script_score": {
                            "script": {
#                                  "source": "_score * saturation(doc['page_rank'].value, 10)"
                                "source": "_score * doc['page_rank'].value"
                            }
                        }
                    },
                ]
            }
        }
    },
)
```

```python
title_search = "lottery"
abstract_search = "language"
year_search = 2018
title_weight = 20
abstract_weight = 10
year_weight = 5
```

```
[54]: res = search(title_search, abstract_search, year_search, apply_page_rank=True)
      for hit in res["hits"]["hits"]:
          print(f'{hit["_source"]["title"]}: {hit["_score"]}')

Visual Referring Expression Recognition: What Do Systems Actually Learn?: 0.29866457
Memory Architectures in Recurrent Neural Network Language Models: 0.27257547
Towards a Unified Natural Language Inference Framework to Evaluate Sentence Representations: 0.19155078
Dissecting Contextual Word Embeddings: Architecture and Representation: 0.15634124
The Lottery Ticket Hypothesis: Training Pruned Neural Networks: 0.13396016
Natural Language Inference over Interaction Space: 0.12812993
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: 0.10748777
Attention-Based Convolutional Neural Network for Machine Comprehension: 0.097398795
A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference: 0.09511792
Annotation Artifacts in Natural Language Inference Data: 0.09371895

[53]: res = search(title_search, abstract_search, year_search, apply_page_rank=False)
      for hit in res["hits"]["hits"]:
          print(f'{hit["_source"]["title"]}: {hit["_score"]}')

The Lottery Ticket Hypothesis: Training Pruned Neural Networks: 100.0
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: 50.0
Character-Level Language Modeling with Deeper Self-Attention: 50.0
U-Net: Machine Reading Comprehension with Unanswerable Questions: 50.0
GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding: 50.0
Memory Architectures in Recurrent Neural Network Language Models: 50.0
Annotation Artifacts in Natural Language Inference Data: 50.0
Transforming Question Answering Datasets Into Natural Language Inference Datasets: 50.0
Visual Referring Expression Recognition: What Do Systems Actually Learn?: 50.0
Visual Dialog: 50.0
```

```
title_search = "deep learning"
abstract_search = "intelligent"
year_search = 2015
title_weight = 10
abstract_weight = 10
year_weight = 5
```

```
[56]: res = search(title_search, abstract_search, year_search, apply_page_rank=True)
      for hit in res["hits"]["hits"]:
          print(f'{hit["_source"]["title"]}: {hit["_score"]}')

Fixed point optimization of deep convolutional neural networks for object recognition: 0.5498063
To go deep or wide in learning?: 0.42167333
Neural Machine Translation by Jointly Learning to Align and Translate: 0.38621685
A Deep Neural Network Compression Pipeline: Pruning, Quantization, Huffman Encoding: 0.3734357
A Deep Reinforced Model for Abstractive Summarization: 0.2914382
Deep Residual Learning for Image Recognition: 0.27725756
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: 0.21497554
Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures: 0.20561633
Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories: 0.20554593
Structured Pruning of Deep Convolutional Neural Networks: 0.191215

[57]: res = search(title_search, abstract_search, year_search, apply_page_rank=False)
      for hit in res["hits"]["hits"]:
          print(f'{hit["_source"]["title"]}: {hit["_score"]}')

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: 100.0
Data-free Parameter Pruning for Deep Neural Networks: 100.0
A Deep Neural Network Compression Pipeline: Pruning, Quantization, Huffman Encoding: 100.0
Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding: 100.0
ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression: 100.0
Learning both Weights and Connections for Efficient Neural Network: 100.0
Deep Fried Convnets: 100.0
Very Deep Convolutional Networks for Natural Language Processing: 100.0
Learning to Skim Text: 100.0
Deep Residual Learning for Image Recognition: 100.0
```

```
title_search = "classification"
abstract_search = "neural"
year_search = 2000
title_weight = 50
abstract_weight = 40
year_weight = 5
```

```
[64]: res = search(title_search, abstract_search, year_search, apply_page_rank=True)
      for hit in res["hits"]["hits"]:
          print(f'{hit["_source"]["title"]}: {hit["_score"]}')

Large-Margin Classification in Infinite Neural Networks: 5.6436925
ImageNet Classification with Deep Convolutional Neural Networks: 2.4965262
High-Performance Neural Networks for Visual Object Classification: 1.8588557
Training CNNs with Low-Rank Filters for Efficient Image Classification: 0.8912874
Character-level Convolutional Networks for Text Classification: 0.7963235
Some Improvements on Deep Convolutional Neural Network Based Image Classification: 0.782767
Convolutional Neural Networks for Sentence Classification: 0.7670854
SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition: 0.7627265
Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification: 0.75370294
Visualizing and Understanding Neural Models in NLP: 0.52961636

[65]: res = search(title_search, abstract_search, year_search, apply_page_rank=False)
      for hit in res["hits"]["hits"]:
          print(f'{hit["_source"]["title"]}: {hit["_score"]}')

Training CNNs with Low-Rank Filters for Efficient Image Classification: 1000.0
Character-level Convolutional Networks for Text Classification: 1000.0
Large-Margin Classification in Infinite Neural Networks: 1000.0
Convolutional Neural Networks for Sentence Classification: 1000.0
Some Improvements on Deep Convolutional Neural Network Based Image Classification: 1000.0
Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification: 1000.0
High-Performance Neural Networks for Visual Object Classification: 1000.0
ImageNet Classification with Deep Convolutional Neural Networks: 1000.0
Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields: 100.0
Part-based statistical models for object classification and detection: 100.0
```

نحوه محاسبه HITS در بخش HITS نوت‌بوک قرار دارد.

بخش ششم

در این بخش پس از خواندن داده‌ها و تشکیل بردار های دسته مثبت و منفی برای هر کویری، دسته‌بند SVM را با آموزش می‌دهیم.

در هنگام تست با استفاده از الگوریتم quicksort و دسته‌بند آموزش دیده، داک ها را مرتب سازی می کنیم.

نتایج مربوطه در بخش Ranking SVM نوت‌بوک قرار دارد.