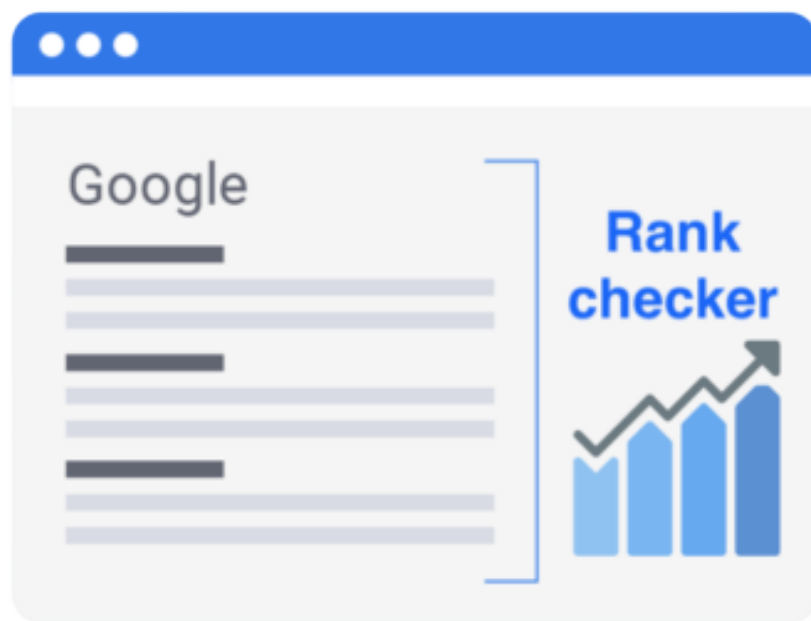


TF-IDF ranking



سید محمد حسین هاشمی ۴۰۲۲۳۶۳۱۴۳

فروردین ۱۴۰۳

فهرست مطالب

۲	۱	کتابخانه‌های مورد نیاز
۲	۲	کلاس Inverted Index
۲	۳	متد <code>-init-</code>
۳	۴	متد <code>add_document</code>
۳	۵	متد <code>rank_document</code>
۴	۶	استفاده

۱ کتابخانه‌های مورد نیاز

```
tf-idf_ranking.py

# Import necessary modules
import re
import sys
import math
from collections import defaultdict
```

کتابخانه‌های مورد نیاز که در پروژه لود شده‌اند.

۲ کلاس Inverted Index

```
tf-idf_ranking.py

# Define a class for the Inverted Index
class InvertedIndex:
```

تمام کدهای مربوطه در این کلاس نوشته می‌شود

۳ متد __init__

```
tf-idf_ranking.py

def __init__(self):
    # Initialize an empty inverted index (a dictionary where each key is a term and the value is a list of
    # document IDs)
    self.index = defaultdict(list)
    # Initialize an empty dictionary to store the IDF (Inverse Document Frequency) values for each term
    self.idf = {}
```

در اینجا یک کالکشن خالی برای ذخیره Inverted Index و همچنین یک دیکشنری برای ذخیره idf هر داکيومنت ایجاد می‌شود.

۴ متد add_document

```
tf-idf_ranking.py

# Method to add a document to the index
def add_document(self, doc_id, terms):
    # Iterate over the unique terms in the document
    for term in set(terms):
        # Add the document ID to the list of document IDs for the term
        self.index[term].append(doc_id)
```

در این متد عملیات اشتراک بین posting list ها انجام می‌شود و نتیجه جست‌وجو برگشت داده می‌شود.

۵ متد rank_document

```
tf-idf_ranking.py

# Method to rank documents based on a query
def rank_documents(self, query_terms):
    # Initialize a dictionary to store the scores for each document
    scores = defaultdict(float)
    # Iterate over each term in the query
    for term in query_terms:
        # Check if the term is in the index (i.e., if it's a valid term)
        if term in self.idf:
            # Iterate over each document ID that contains the term
            for doc_id in self.index[term]:
                # Add the IDF value of the term to the score of the document
                scores[doc_id] += self.idf[term]
    # Return a sorted list of documents by their scores in descending order
    return sorted(scores.items(), key=lambda x: x[1], reverse=True)
```

در این تابع براساس ورودی (query_terms) با فرمول tf-idf امتیازدهی انجام می‌شود و داکيومنت‌ها به‌ترتیب امتیاز برگشت داده می‌شوند.

۶ استفاده

```
tf-idf_ranking.py

# Usage
if __name__ == '__main__':
    # Check if the correct number of command-line arguments are provided
    if len(sys.argv) != 2:
        print("Usage: python tf-idf_ranking.py [file_name]")
        sys.exit(1)

    # Get the filename from the command-line argument
    filename = sys.argv[1]

    # Create an instance of the InvertedIndex class
    index = InvertedIndex()

    # Add documents from the file to the index
    with open(filename, encoding="utf8") as file:
        record_id = 0
        for line in file:
            record_id += 1
            # Split the line into terms using a regular expression and convert to lowercase
            terms = re.split('[^a-zA-z]', line.lower())
            # Add the document to the index
            index.add_document(record_id, terms)

    # Calculate the IDF values for each term
    index.calculate_idf()

    # Prompt the user to enter a query
    query = re.split('[^a-zA-z]', input('Search: ').lower())

    # Rank documents for the query
    ranked_docs = index.rank_documents(query)

    # Print the search results
    print(f"Search results for query '{' '.join(query)}':")
    for doc_id, score in ranked_docs[0: 11]:
        print(f"Document {doc_id}: TF-IDF score = {score:.4f}")
```

برای استفاده از کلاس گفته شده این کد نوشته شده که بعد از فراخوانی کلاس ایجاد شده و پس از خواندن داکيومنت‌ها و ایجاد Inverted Index در آن با استفاده از رنکینگ tf-idf جست‌وجو انجام می‌شود و ۱۰ نتیجه اول به همراه امتیاز خروجی داده می‌شود. برای مثال برای ورودی مانند تصویر زیر:

```
first document
second document
third document|
```

خروجی مانند تصویر زیر تولید می‌شود.

```
Search: third
Search results for query 'third':
Document 3: TF-IDF score = 1.6094
```