pesquisa_do_documento

June 11, 2024

1 Busca de Índice e Consulta de documento

Autor: Davi J. Leite Santos
Versão: 0.0.3
Data: 25 de Abril de 2024
Localização: Ribeirão das Neves, Minas Gerais - Brasil

1.1 Contato

- **Endereço:** Ribeirão das Neves, Minas Gerais - Brasil
- **Email:** davi.jls@outlook.com
- **LinkedIn:** davi-j-leite-santos
- **Website:** davijls.com.br

1.2 Principais Competências

- **Cibersegurança**
- **Segurança da Informação**
- **Operações de TI**

2 Sobre o codigo

Este código Python aborda a indexação e pesquisa de documentos com foco na eficiência e relevância. Ele utiliza ferramentas de processamento de linguagem natural (NLP) para preparar os dados e realiza pesquisas rápidas e precisas no índice, identificando e exibindo os domínios mais relevantes com pontuações calculadas com base na frequência e tempo de busca.

2.0.1 Componentes Principais do Código

1. Configurações Iniciais

- O código começa com a importação de módulos essenciais, como `json`, `re`, `nltk`, `os`, `time` e funcionalidades de `collections.Counter` e `urllib.parse.urlparse`.
- Downloads de componentes do NLTK necessários para tokenização, remoção de stopwords e stemming em português.

2. Funções de Limpeza e Processamento de Texto

- **clean_text**: Limpa o texto removendo caracteres especiais e stopwords, convertendo tudo para minúsculas.

- **apply_stemming**: Aplica o stemming às palavras tokenizadas para reduzir a sua forma base, ajudando na indexação e na busca.
- **preprocess_query**: Encapsula o processo de limpeza e stemming em uma função destinada a preparar consultas de pesquisa.

3. Funções de Carregamento e Salvamento de Dados

- **load_json** e **save_json**: Carregam e salvam dados em formato JSON, respectivamente, com a manipulação correta da codificação para preservar caracteres não ASCII.
- **load_data**: Carrega dados de vocabulário e índice geral previamente processados.

4. Função de Busca

- **search_query**: Processa termos de consulta, recuperando e listando documentos relevantes com base nos termos pesquisados. Registra o tempo de busca, possibilitando avaliações de performance.

5. Análise e Apresentação de Resultados

- **extract_domain**: Extrai o domínio de nomes de arquivo formatados, assumindo que os nomes contêm informações de domínio.
- **calculate_score**: Calcula uma pontuação para cada domínio com base na frequência de ocorrência e no tempo de busca, ponderando esses fatores para enfatizar mais a frequência.
- **display_results_with_scores**: Organiza e exibe os resultados da busca, mostrando os domínios mais relevantes juntamente com suas frequências e pontuações calculadas.

2.0.2 Uso e Benefícios

- Este script é altamente eficiente para sistemas de recuperação de informação onde a precisão e a velocidade são cruciais.
- As funções de limpeza e stemming são particularmente úteis para tratar dados em português, tornando-o relevante para aplicações em idiomas além do inglês.
- A funcionalidade de pontuação e classificação de domínios é inovadora, pois combina métricas de frequência e performance de busca para destacar os resultados mais relevantes.

Este sistema é uma solução completa para busca e análise de documentos indexados, ideal para ser aplicado em contextos onde a velocidade e a relevância da informação são prioritárias, como em serviços de busca na web ou sistemas internos de gestão de conhecimento.

3 ———————————————————————————-

4 Imoportações

[ ]:
```
import json
import os
import re
import nltk
import time
from nltk.corpus import stopwords
```

```
from nltk.stem import RSLPStemmer
from collections import Counter
from urllib.parse import urlparse

nltk.download("rslp")
nltk.download("punkt")
nltk.download("stopwords")

# Inicializar o Stemmer para português
stemmer = RSLPStemmer()
stop_words = set(stopwords.words("portuguese"))
```

```
[nltk_data] Downloading package rslp to
[nltk_data]     C:\Users\davim\AppData\Roaming\nltk_data…
[nltk_data]   Package rslp is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\davim\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\davim\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

5 Funções de Limpeza e Processamento de Texto

[ ]:
```
def load_json(file_path):
    with open(file_path, "r", encoding="utf-8") as f:
        return json.load(f)


def save_json(data, file_path):
    with open(file_path, "w", encoding="utf-8") as f:
        json.dump(data, f, ensure_ascii=False, indent=4)


def clean_text(text):
    text = re.sub(r"[^\w\s]", " ", text)  # Remove special characters
    text = text.lower()  # Convert to lowercase
    tokens = text.split()
    tokens = [word for word in tokens if word not in stop_words]  # Remove␣
 ↪stopwords
    return " ".join(tokens)


def apply_stemming(text):
    tokens = nltk.word_tokenize(text, language="portuguese")
    stemmed_tokens = [stemmer.stem(token) for token in tokens]
    return " ".join(stemmed_tokens)
```

```
def preprocess_query(query):
    cleaned_query = clean_text(query)
    stemmed_query = apply_stemming(cleaned_query)
    return stemmed_query
```

6 Funções para Carregar Dados

[ ]:
```
def load_data(vocab_geral_path, index_geral_path):
    vocab_geral = load_json(vocab_geral_path)
    index_geral = load_json(index_geral_path)
    return vocab_geral, index_geral
```

[ ]:
```
# Defina os caminhos dos arquivos
vocab_geral_file = "cleaned_vocab_geral.json"
index_geral_file = "cleaned_index_geral.json"
```

[ ]:
```
# Carregar os dados apenas uma vez
vocab_geral, index_geral = load_data(vocab_geral_file, index_geral_file)
```

7 Função de Busca

[ ]:
```
def search_query(query, vocab_geral, index_geral):
    # Registrar o tempo de início da busca
    start_time = time.time()

    # Processar a query
    query_terms = preprocess_query(query).split()
    query_terms = list(set(query_terms))

    # Obter IDs das palavras da query
    word_ids = []
    for term in query_terms:
        if term in vocab_geral:
            word_ids.append(vocab_geral[term])

    # Recuperar documentos correspondentes
    result_files = set()
    for word_id in word_ids:
        word_id = str(word_id)
        if word_id in index_geral:
            result_files.update(index_geral[word_id]["file_names"])

    # Registrar o tempo de término da busca
    end_time = time.time()
```

```
    search_time = end_time - start_time

    return list(result_files), search_time
```

[ ]:
```
# Exemplo de uso
query = "buceta cu vagina anal sexo toy pussy pinto dick desgraça tnc fuder␣
 ↪sefuder fdp"
result_files, search_time = search_query(query, vocab_geral, index_geral)
```

8 Extraindo os melhores sites

[ ]:
```
def extract_domain(file_name):
    parts = file_name.split("]")
    if len(parts) > 0:
        domain = parts[0].replace("_", ".")
        return domain
    return None


def calculate_score(frequency, search_time):
    # Defina os pesos para frequência e tempo de busca
    frequency_weight = 0.7
    time_weight = 0.3

    # Calcule o score ponderado
    score = (frequency * frequency_weight) - (search_time * time_weight)
    return score


def display_results_with_scores(result_files, search_time):
    domain_counter = Counter()
    for file in result_files:
        domain = extract_domain(file)
        if domain:
            domain_counter[domain] += 1

    sorted_domains = domain_counter.most_common()

    print(f"{'Site':<47}{'Frequency':<13}{'Score(ms acomulado)':<10}")
    print("=" * 80)
    for domain, frequency in sorted_domains:
        score = calculate_score(frequency, search_time)
        print(f"{domain:<47}{frequency:<13}{score:<10.2f}")
```

[ ]:
```
# Exibir resultados organizados por site com score e tempo de busca
print(f"Query que foi feita: {query}\n")
```

```
display_results_with_scores(result_files, search_time)

print(f"\nSearch completed in {search_time:.2f} seconds")
```

```
Query que foi feita: buceta cu vagina anal anal sexo toy pussy

Site                                                    Frequency    Score(ms acomulado)
================================================================================
www.terra.com.br                                        304          212.77
imirante.com                                            278          194.57
emerj.tjrj.jus.br                                       241          168.67
www.metropoles.com                                      179          125.27
www.poder360.com.br                                     154          107.77
oglobo.globo.com                                        142          99.37
blogdaboitempo.com.br                                   111          77.67
mundoeducacao.uol.com.br                                111          77.67
jovempan.com.br                                         76           53.17
glamour.globo.com                                       49           34.27
gq.globo.com                                            47           32.87
nao-palavra.blogspot.com                                46           32.17
static.poder360.com.br                                  45           31.47
blogbvps.com                                            39           27.27
escolakids.uol.com.br                                   38           26.57
exercicios.mundoeducacao.uol.com.br                     32           22.37
extra.globo.com                                         29           20.27
revistamarieclaire.globo.com                            24           16.77
revistaquem.globo.com                                   24           16.77
brasilescola.uol.com.br                                 22           15.37
www.techtudo.com.br                                     21           14.67
revistacrescer.globo.com                                20           13.97
vestibular.mundoeducacao.uol.com.br                     17           11.87
vogue.globo.com                                         14           9.77
static.mundoeducacao.uol.com.br                         10           6.97
anchor.fm                                               10           6.97
agenciabrasil.ebc.com.br                                9            6.27
monografias.brasilescola.uol.com.br                     9            6.27
www.gov.br                                              9            6.27
blog.sensorion.com.br                                   7            4.87
www.jovempan.com.br                                     7            4.87
revistamonet.globo.com                                  6            4.17
www.blogsoestado.com                                    6            4.17
cbn.globo.com                                           5            3.47
psd.org.br                                              5            3.47
pox.globo.com                                           5            3.47
500palavrasdotcom.wordpress.com                         5            3.47
jornal.usp.br                                           5            3.47
hotm.art                                                4            2.77
www.omnycontent.com                                     4            2.77
```

epocanegocios.globo.com	4	2.77
www.embrapa.br	4	2.77
valorinveste.globo.com	3	2.07
www12.senado.leg.br	3	2.07
ri.multiplan.com.br	3	2.07
globorural.globo.com	3	2.07
vestibular.brasilescola.uol.com.br	3	2.07
educador.brasilescola.uol.com.br	3	2.07
bhfm.globo.com	3	2.07
blogbvps.wordpress.com	2	1.37
servicos.terra.com.br	2	1.37
vidadebicho.globo.com	2	1.37
www25.senado.leg.br	2	1.37
revistagalileu.globo.com	2	1.37
preply.com	2	1.37
www1.folha.uol.com.br	2	1.37
bit.ly	2	1.37
www.ofpalavra.com.br	2	1.37
s1.trrsf.com	2	1.37
open.spotify.com	2	1.37
ofpalavra.kpages.online	2	1.37
www.tjdft.jus.br	2	1.37
wp.me	2	1.37
www.oabsp.org.br	2	1.37
elpais.com	2	1.37
www.ebc.com.br	1	0.67
api.mziq.com	1	0.67
www.lanacion.com.ar	1	0.67
justica.sp.gov.br	1	0.67
psd.org.br.txt	1	0.67
files.metropoles.com	1	0.67
exercicios.brasilescola.uol.com.br	1	0.67
www.barrashopping.com.br	1	0.67
ictdf.org.br	1	0.67
veja.abril.com.br	1	0.67
www.camara.leg.br	1	0.67
imagem.camara.gov.br	1	0.67
extra.globo.com.txt	1	0.67
gq.globo.com.txt	1	0.67
revistamarieclaire.globo.com.txt	1	0.67
www.ibge.gov.br	1	0.67
www2.camara.leg.br	1	0.67
wiredfestival.globo.com.txt	1	0.67
bhfm.globo.com.txt	1	0.67
revistaquem.globo.com.txt	1	0.67
bvps.fiocruz.br	1	0.67
www.cnj.jus.br	1	0.67
sobreuol.noticias.uol.com.br	1	0.67

radioglobo.globo.com.txt	1	0.67
vejario.abril.com.br	1	0.67
www.escoladapalavra.art.br	1	0.67
www.cnnbrasil.com.br	1	0.67
feriadellibro.com.txt	1	0.67
umsoplaneta.globo.com	1	0.67
noticias.uol.com.br	1	0.67
pipelinevalor.globo.com	1	0.67
revistamonet.globo.com.txt	1	0.67
www.whatsapp.com	1	0.67
www.metropoles.com.txt	1	0.67
anpocs.com	1	0.67
www.parkshoppingcanoas.com.br	1	0.67
www.boitempoeditorial.com.br	1	0.67
www.mirantefm.com.txt	1	0.67
www.policiacivil.go.gov.br	1	0.67
www.parkjacarepagua.com.br	1	0.67
amb.org.br	1	0.67
imirante.com.txt	1	0.67
policies.google.com	1	0.67
www.rubensricupero.com	1	0.67
vidadebicho.globo.com.txt	1	0.67
radioglobo.globo.com	1	0.67
compromissosocial.org.br	1	0.67
fundacaocasa.sp.gov.br	1	0.67
www.ofpalavra.com.br.txt	1	0.67
pipelinevalor.globo.com.txt	1	0.67
www.redeomnia.com	1	0.67
pt.wikipedia.org	1	0.67
www.gumgum.com	1	0.67
www.heraldsun.com.au	1	0.67
www.nytimes.com	1	0.67
pox.globo.com.txt	1	0.67
valorinveste.globo.com.txt	1	0.67
webstories.metropoles.com	1	0.67
doi.org	1	0.67
www.miranteam.com.txt	1	0.67
www.blogger.com	1	0.67
sistemas.vestibular.uerj.br	1	0.67
s3.glbimg.com	1	0.67
cetesb.sp.gov.br	1	0.67
web.archive.org.txt	1	0.67
www.codevasf.gov.br	1	0.67
emerj.tjrj.jus.br.txt	1	0.67
tvbrasil.ebc.com.br	1	0.67
ofpalavra.kpages.online.txt	1	0.67
mpc.tc.df.gov.br	1	0.67
eventos.congresse.me	1	0.67

```
glamour.globo.com.txt                    1          0.67
agenciabrasil.ebc.com.br.txt             1          0.67
abrir.link                               1          0.67

Search completed in 0.09 seconds
```

[ ]:

[ ]: