Sistema de busca:
Críticas de filmes

Leonardo Alves (las3)
Leonardo Schettini (ljsa)
Marcos Barreto (msb5)

Recuperação de Informação - 2018.2

Indexador

Normalização dos atributos

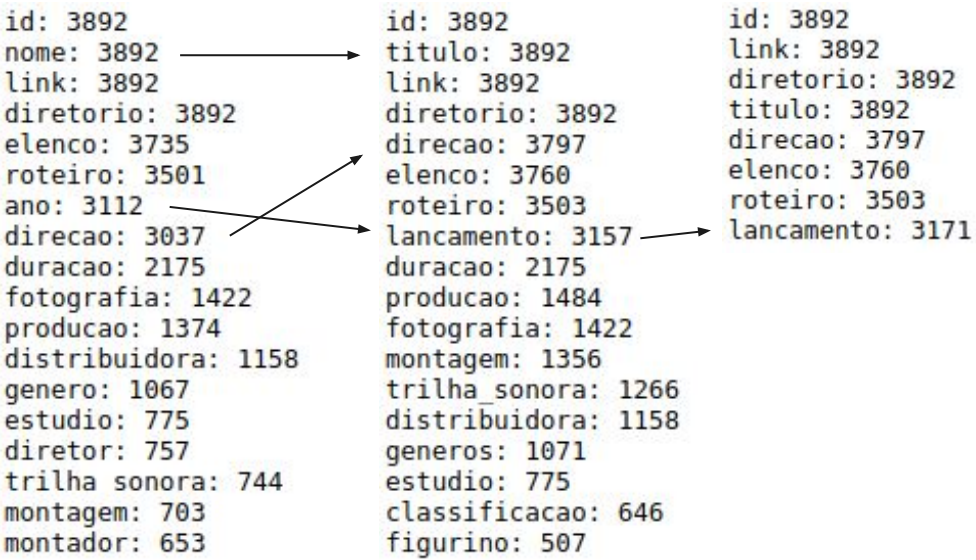
- Atributos não extraídos
  - “id”, “link” e “diretorio”
- Limpeza das strings
  - “direcao”, “ direcao”
- Remoção de informações adicionais
  - “elenco”, “elenco (vozes)”
- Lematização
  - “diretor”, “direcao”

Lematização

```
synonyms = {
    'direcao': ['diretor'],
    'montagem': ['montador'],
    'trilha_sonora': ['musica'],
    'titulo': ['titulo_original', 'nome'],
    'lancamento': ['data_lancamento', 'ano'],
    'generos': ['genero'],
    'arte': ['arte_final'],
    'elenco': ['editora_brasil', 'editora_original', 'vozes', 'elenco_vozes', 'vozes_originais'],
    'producao': ['design_producao'],
    'nacionalidade': ['pais', 'nacionalidades']
}

synonyms_index = {}
for lemma, synonyms in synonyms.items():
    synonyms_index[lemma] = lemma
    for synonym in synonyms:
        synonyms_index[synonym] = lemma
```

Resultado


id: 3892
nome: 3892
link: 3892
diretorio: 3892
elenco: 3735
roteiro: 3501
ano: 3112
direcao: 3037
duracao: 2175
fotografia: 1422
producao: 1374
distribuidora: 1158
genero: 1067
estudio: 775
diretor: 757
trilha sonora: 744
montagem: 703
montador: 653
id: 3892
titulo: 3892
link: 3892
diretorio: 3892
direcao: 3797
elenco: 3760
roteiro: 3503
lancamento: 3157
duracao: 2175
producao: 1484
fotografia: 1422
montagem: 1356
trilha_sonora: 1266
distribuidora: 1158
generos: 1071
estudio: 775
classificacao: 646
figurino: 507
id: 3892
link: 3892
diretorio: 3892
titulo: 3892
direcao: 3797
elenco: 3760
roteiro: 3503
lancamento: 3171

Pré-processamento dos dados

- Atributos selecionados
  - Título, Direção, Elenco, Roteiro e Lançamento
- Tokenização
- Discretização de atributos numéricos
  - Lançamento: _1950, 1950_1959, 1960_1969, 1970_1979, 1980_1989, 1990_1999, 2000_2004, 2005_2009, 2010_2012, 2013_2015, 2016_2017, 2017_2018

Índice Invertido

- Único índice para atributos extraídos e palavras comuns
  - “atributo.valor” e “palavra”
- Frequência por documento
  - Estruturado com dicionário

```
In [70]: inverted_index['titulo.pele']
Out[70]: (5, {1: 1, 231: 1, 841: 1, 1138: 1, 2085: 1})
```

  - Estruturado com lista e ordenado por frequência

```
In [68]: inverted_index_list['titulo.pele']
Out[68]: (5, [(1, 1), (231, 1), (841, 1), (1138, 1), (2085, 1)])
```

  - Estruturado com lista, ordenado por frequência e com compressão

```
In [69]: inverted_index_list_compressed['titulo.pele']
Out[69]: (5, [(1, 1), (230, 1), (610, 1), (297, 1), (947, 1)])
```

Tamanho dos Índices

	JSON	Binário
Dicionário	63.874.599 Bytes	14.780.893 Bytes
Lista e ordenado por frequência	179.085.831 Bytes	19.790.761 Bytes
Lista, ordenado por frequência e com compressão	173.896.785 Bytes	17.815.244 Bytes

Tamanho dos Índices

- Lista de itens contém mais objetos
  - Lista externa
  - Uma tupla por par chave-valor
- Mais detalhes:
  - https://stackoverflow.com/questions/53376510/disk-space-python-dictionary-vs-list
- OrderedDict
  - Dicionário que mantém a ordem dos elementos adicionados
  - Benefícios de listas e dicionários.

Ranqueamento

Ranqueamento

- Modelo de espaço de vetores (cosseno)
  - Com e sem tf-idf
- Term-at-a-time
- Entrada: consulta do usuário
- Saída: lista de documentos ranqueados pelo cosseno

Processamento

- Inicialização da base
  - Leitura do índice invertido
  - Construção dos vetores base
    - Docs
    - Lengths
    - Scores

Processamento

- Pré-processamento da consulta
  - Consulta geral ou estruturada
  - Peso do termo na consulta: TF
  - Interseção de termos da consulta na base
- Inicialização do vetor de score para a consulta
  - Cópia do vetor score base

Processamento

- Para cada termo, os scores são atualizados
  - Term-at-a-time
- Peso dos termos
  - Na consulta: TF
  - No documento: TF ou TF-IDF
- Retorna uma lista ordenada pelo score
  - Ordem decrescente
  - Par (docID, score)

Comparação de rankings

- Correlação de Spearman
- 5 consultas comparando o efeito do tf-idf
- 5 consultas mistas comparando:
  - Similaridade da consulta
  - Efeito do tf-idf

Comparando consultas em relação ao `idf`

```
spearman_correlation(qp.rank('comedia romantica'), qp.rank('comedia romantica', idf=True))
```

```
0.9993746084832799
```

```
spearman_correlation(qp.rank('ficcao cientifica'), qp.rank('ficcao cientifica', idf=True))
```

```
0.9999959152369184
```

```
spearman_correlation(qp.rank('batalha epica'), qp.rank('batalha epica', idf=True))
```

```
0.9999836279732061
```

```
spearman_correlation(qp.rank('historia depressiva'), qp.rank('historia depressiva', idf=True))
```

```
0.9974768885787155
```

```
spearman_correlation(qp.rank('personagem encantador'), qp.rank('personagem encantador', idf=True))
```

```
0.9942959416140704
```

Consultas iguais, variando idf

Comparando consultas com a correlação de spearman

```
spearman_correlation(qp.rank('o misterio do relogio na parede'), qp.rank('misterio relogio parede'))
```

```
0.6629680913242246
```

```
spearman_correlation(qp.rank('o misterio do relogio na parede', attr="titulo"), qp.rank('misterio relogio
 parede', attr="titulo"))
```

```
0.9684932416669699
```

```
spearman_correlation(qp.rank('o misterio do relogio na parede', idf=True), qp.rank('misterio relogio pared
e', idf=True))
```

```
0.694384598335422
```

```
spearman_correlation(qp.rank('comedia romantica'), qp.rank('comedia romance'))
```

```
0.9010293568973288
```

```
spearman_correlation(qp.rank('comedia romantica', idf=True), qp.rank('comedia romance', idf=True))
```

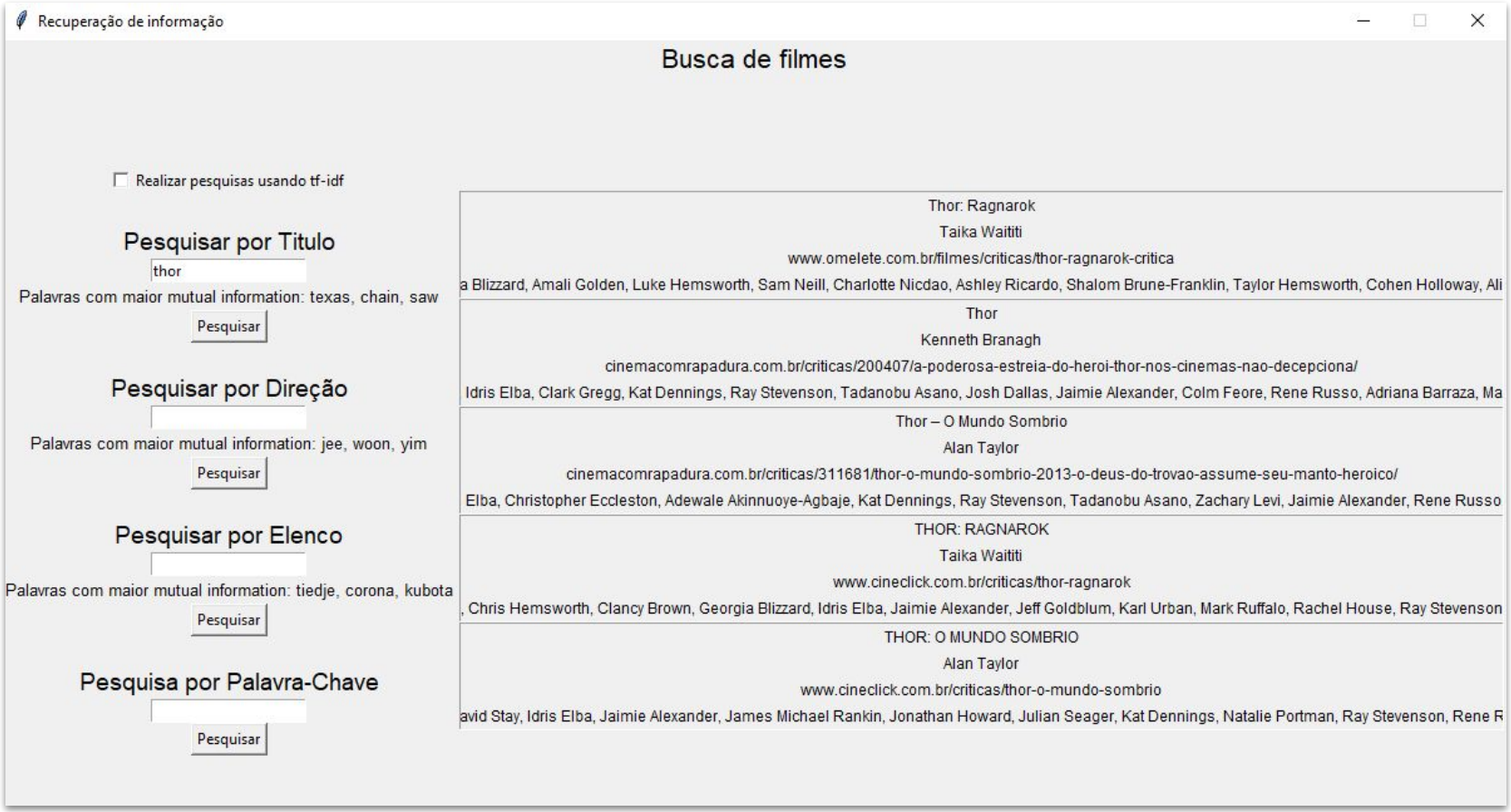
```
0.8820985895661346
```

Consultas mistas, comparando similaridade da consulta e idf

Interface

Mutual Information

- Título (3742)
  - texas, chain, saw
- Direção (2243)
  - jee, woon, yim
- Elenco (18104)
  - tiedje, corona, kubota



Interface Utilizando Tkinter

Obrigado!

Github do Projeto