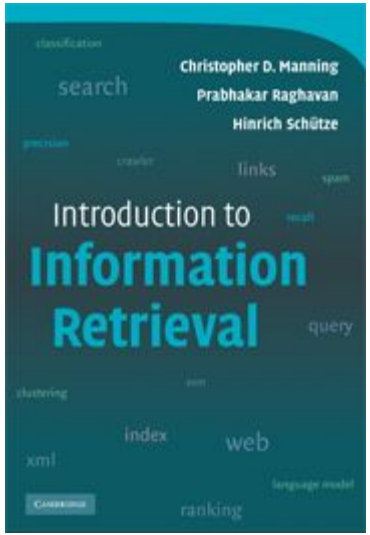
Índices invertidos e algoritmos de busca

Júlio César Batista
https://juliocesarbatista.com/

09/12/2023

BLUMENAU DEV DAY

HACKERSPACE
BLUMENAU



https://nlp.stanford.edu/IR-book/

https://juliocesarbatista.com/categories/recuperação-de-informação/

Assumindo um conjunto de textos

- Caros colegas, a atual estrutura da organização auxilia a preparação e a estruturação das formas de ação
- Nunca é demais insistir que o novo modelo estrutural preconizado contribui para a correta determinação das nossas opções de desenvolvimento futuro
- O incentivo ao avanço tecnológico, assim como o desenvolvimento de formas distintas de atuação nos obriga a análise das opções básicas para o sucesso do programa
- É fundamental ressaltar que a análise dos diversos resultados assume importantes posições na definição dos índices pretendidos

Como resolvemos as seguintes consultas?

- Desenvolvimento
  - Nunca é demais insistir que o novo ... nossas opções de **desenvolvimento** futuro
  - O incentivo ao avanço tecnológico, assim como o **desenvolvimento** de formas ...
- Desenvolvimento AND Análise
  - Nunca é demais insistir .. de **desenvolvimento** futuro
  - O incentivo ao avanço tecnológico, assim como o **desenvolvimento** de ... obriga a **análise** ...
  - É fundamental ressaltar que a **análise** ... pretendidos
- Desenvolvimento OR Análise
  - Nunca é demais insistir que o ... nossas opções de **desenvolvimento** futuro
  - O incentivo ao avanço tecnológico, assim como o **desenvolvimento** de ... a **análise** das ...
  - É fundamental ressaltar que a **análise** dos diversos ...
- Desenvolvimento NOT Formas
  - Caros colegas, a atual estrutura da ... das **formas** de ação
  - Nunca é demais ... determinação das nossas opções de **desenvolvimento** futuro
  - O incentivo ao avanço tecnológico, assim como o **desenvolvimento** de **formas** distintas de ...

Busca com força bruta

N documentos

- Caros colegas, a atual estrutura da organização auxilia a preparação e a estruturação das formas de ação
- Nunca é demais insistir que o novo modelo estrutural preconizado contribui para a correta determinação das nossas opções de desenvolvimento futuro
- O incentivo ao avanço tecnológico, assim como o desenvolvimento de formas distintas de atuação nos obriga a análise das opções básicas para o sucesso do programa
- É fundamental ressaltar que a análise dos diversos resultados assume importantes posições na definição dos índices pretendidos

Média ***M*** palavras por *documento*

Qual o problema da busca com força bruta?

Qual o problema da busca com força bruta?

- O algoritmo tem complexidade **N** * **M**
- Conforme **N** aumenta (páginas na internet, por exemplo), a busca demora mais
- O tamanho dos documentos (**M**) tende a ser finito
- A quantidade de documentos (**N**) tende ao infinito

Matriz de incidência termo x documento

Atributa um **ID** para cada documento

1. Caros colegas, a atual estrutura da organização auxilia a preparação e a estruturação das formas de ação
2. Nunca é demais insistir que o novo modelo estrutural preconizado contribui para a correta determinação das nossas opções de desenvolvimento futuro
3. O incentivo ao avanço tecnológico, assim como o desenvolvimento de formas distintas de atuação nos obriga a análise das opções básicas para o sucesso do programa
4. É fundamental ressaltar que a análise dos diversos resultados assume importantes posições na definição dos índices pretendidos

Matriz de incidência termo x documento

Termo / Documento	Doc 1	Doc 2	Doc 3	Doc 4
Desenvolvimento	0	1	1	0
Análise	0	0	1	1
Formas	1	0	1	0
Futuro	0	1	0	0
Incentivo	0	0	1	0

Matriz de incidência termo x documento

Consulta “Desenvolvimento AND Análise”

- Desenvolvimento: [0, 1, 1, 0]
- Análise: [0, 0, 1, 1]
- Basicamente, um AND binário

- As demais consultas também podem ser implementadas com operações binárias
- Reduz o espaço de busca, considerando apenas as palavras desejadas na busca

Qual o problema da matriz termo x documento?

Qual o problema da matriz termo x documento?

- Tende a ser esparsa: muito mais 0 que 1
- Ocupa muito espaço com 0
- Precisamos armazenar apenas 1

Índice invertido

- Desenvolvimento: [2, 3]
- Análise: [3, 4]
- Formas: [1, 3]
- Futuro: [2]
- Incentivo: [3]
- ...

Termo / Doc	Doc 1	Doc 2	Doc 3	Doc 4
Desenvolvimento	0	1	1	0
Análise	0	0	1	1
Formas	1	0	1	0
Futuro	0	1	0	0
Incentivo	0	0	1	0

Índice invertido

Desenvolvimento: [2, 3]
Análise: [3, 4]
Formas: [1, 3]
Futuro: [2]
Incentivo: [3]
...

- `Dict<str, List<int>>`
- Os IDs de documentos são sempre em ordem crescente
  - A lista de documentos também é conhecida como *postings list*
- Não existem entradas duplicadas na lista de documentos
  - Talvez um `SortedSet<int>` seria uma alternativa para `List<int>`
- Usar `int` no índice, pode ser melhor

Qual o problema do índice invertido?

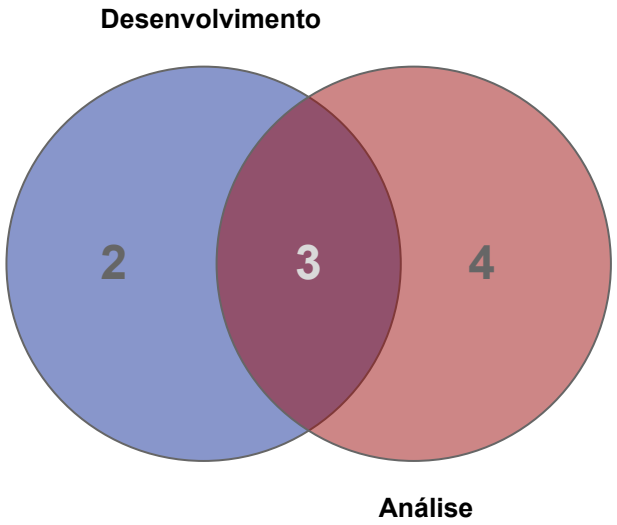
Qual o problema do índice invertido?

- O algoritmo de busca não é apenas uma operação binária
- É preciso implementar os algoritmos para cada caso
  - Em alguns casos, talvez seja melhor, considerando algumas otimizações que são possíveis

Índice Invertido

Consulta “Desenvolvimento AND Análise”

- Desenvolvimento: [2, 3]
- Análise: [3, 4]

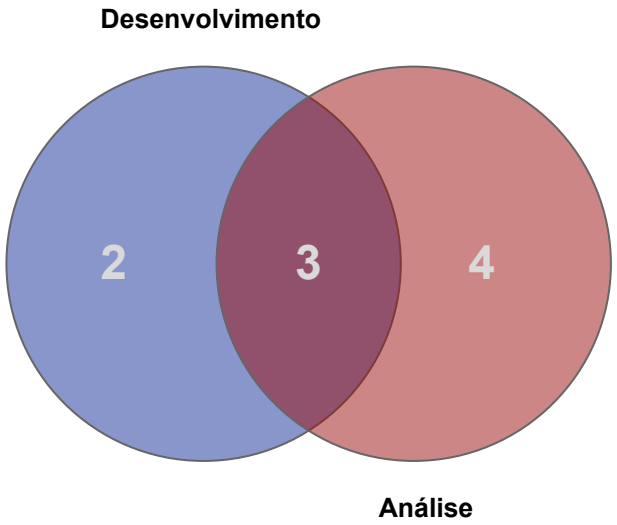



```
def intersect(p1, p2):
  answer = []
  p1 i = 0
  p2 i = 0
  while len(p1) > p1_i and len(p2) > p2_i:
      if p1[p1 i] == p2[p2 i]:
          answer.append(p1[p1_i])
          p1 i += 1
          p2 i += 1
      elif p1[p1_i] < p2[p2_i]:
          p1_i += 1
      else:
          p2 i += 1
  return answer
```

Índice Invertido

Consulta “Desenvolvimento OR Análise”

- Desenvolvimento: [2, 3]
- Análise: [3, 4]

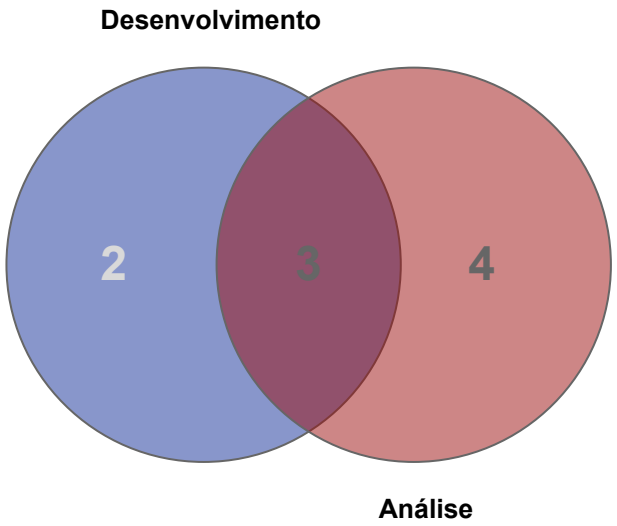



```
def union(p1, p2):
  answer = []
  p1 i = 0
  p2_i = 0
  while len(p1) > p1 i or len(p2) > p2_i:
      if p1 i == len(p1):
          answer.append(p2[p2_i])
          p2 i += 1
      elif len(p2) == p2_i:
          answer.append(p1[p1_i])
          p1 i += 1
      elif p1[p1_i] == p2[p2_i]:
          answer.append(p1[p1_i])
          p1_i += 1
          p2 i += 1
      elif p1[p1 i] < p2[p2 i]:
          answer.append(p1[p1_i])
          p1_i += 1
      else:
          answer.append(p2[p2_i])
          p2 i += 1
  return answer
```

Índice Invertido

Consulta “Desenvolvimento NOT Análise”

- Desenvolvimento: [2, 3]
- Análise: [3, 4]




```
def diff(p1, p2):
  answer = []
  p1 i = 0
  p2 i = 0
  while len(p1) > p1_i:
      if len(p2) == p2 i or p1[p1_i] < p2[p2_i]:
          answer.append(p1[p1_i])
          p1 i += 1
      elif p1[p1 i] == p2[p2_i]:
          p1_i += 1
          p2_i += 1
      else:
          p2 i += 1
  return answer
```

Índice Invertido

Consulta AND otimizada

- `[3, 7, 8, 9, 10] AND [1, 2, 4, 6, 8, 10] AND [1, 2, 3] AND [2, 3, 7, 9]`
- O resultado máximo, terá tamanho `min(len(postings))`
  - No caso acima, o resultado máximo terá 3 documentos, porque um termo aparece em apenas 3 documentos `[1, 2, 3]`
- Fazer a intersecção `[3, 7, 8, 9, 10] AND [1, 2, 4, 6, 8, 10]` resulta em 2 iterações a mais que o necessário em `intersect`
  - Serão feitas 5 iterações ao invés de apenas 3

Índice Invertido

Consulta AND otimizada

- `[3, 7, 8, 9, 10] AND [1, 2, 4, 6, 8, 10] AND [1, 2, 3] AND [2, 3, 7, 9]`
- Solução: ordenar as *postings lists* em order ascendente
- `[1, 2, 3] AND [2, 3, 7, 9] = [2, 3]`
- `[2, 3] AND [3, 7, 8, 9, 10] = [3]`
- `[3] AND [1, 2, 4, 6, 8, 10] = []`

```
def conjunctive intersect(*p):
  postings = sorted(p, key=lambda x: len(x))
  result = postings[0]
  postings = postings[1:]
  while postings and result:
      result = intersect(result, postings[0])
      postings = postings[1:]
  return result
```

Próximos encontros

- Índices invertidos e algoritmos de busca: Parte 2
  - Construção do índice invertido
  - Compressão do índice
  - Otimização das consultas
- Consultas por frases
- Correção ortográfica

09/12/2023

BLUMENAU DEV DAY

HACKERSPACE BLUMENAU

Participe

- Encontros ~mensais
  - Participe dos encontros
  - Apresente sobre algum assunto
  - Leve para a sua empresa

- Hackerspace Blumenau
- GruPy Blumenau
- Elastic Blumenau
- *A sua comunidade aqui*

Bate-papo

Implementação