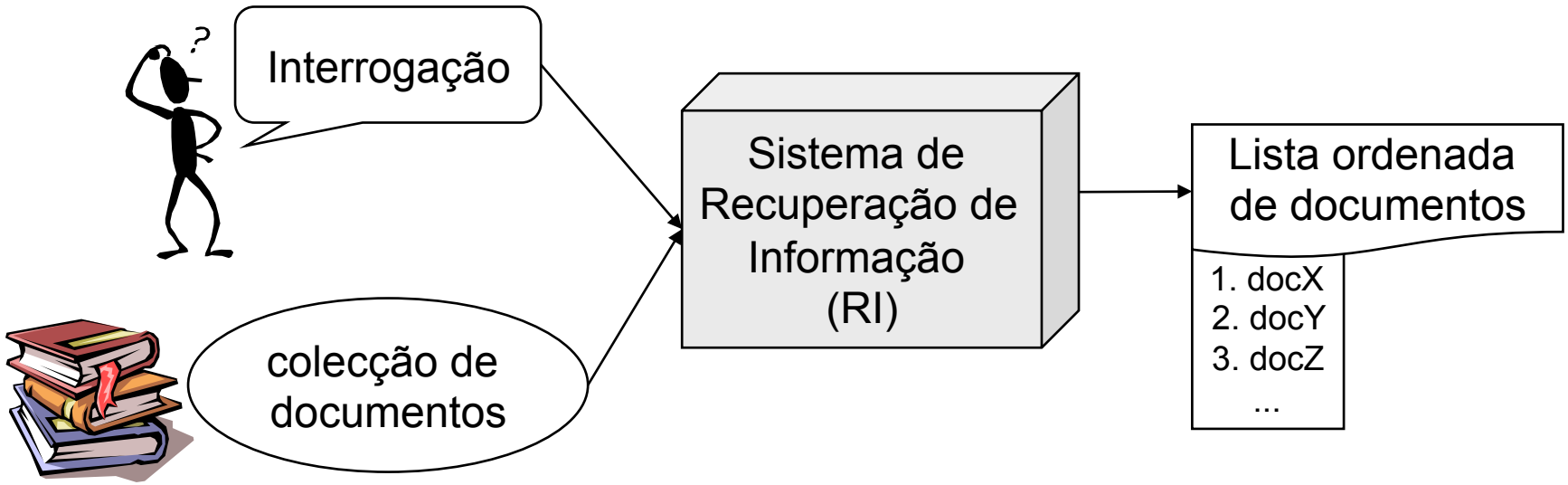
Recuperação de Informação – modelo Booleano

Paulo Trigo Silva

Modelo usual de funcionamento

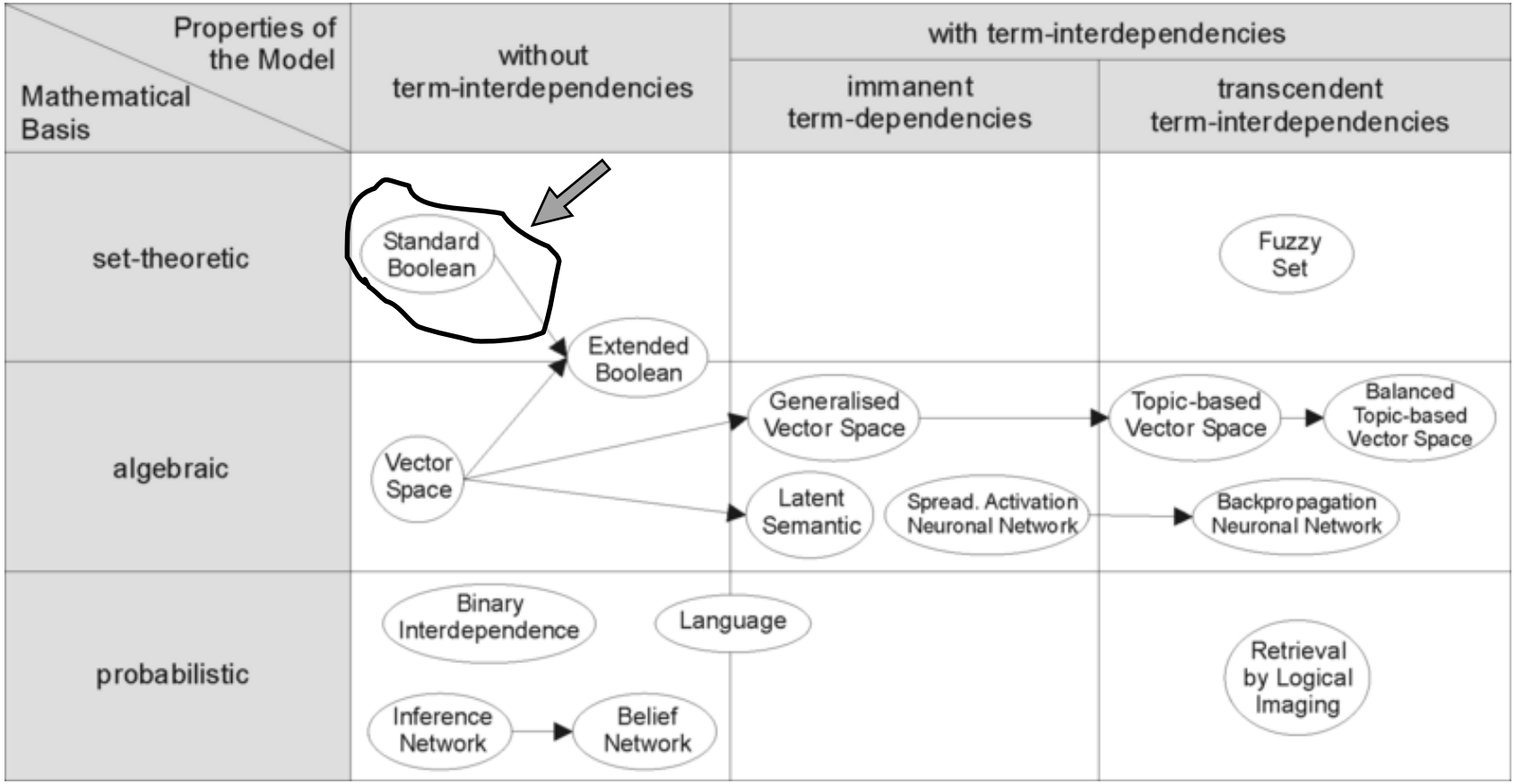
- Entrada:
  - uma colecção de documentos com texto escrito em língua natural
  - uma interrogação do utilizador escrita como sequência de caracteres
- Saída:
  - uma lista ordenada de documentos relevantes para a interrogação



... recuperação de informação em documentos de texto

- Processos
  - indexar documentos
  - extrair os documentos com determinada informação
- Preocupações
  - 1. identificar os documentos relevantes
  - 2. extrair eficazmente grandes quantidades de documentos
- ... existem diferentes modelos para implementar os processos
  - contemplando as preocupações!

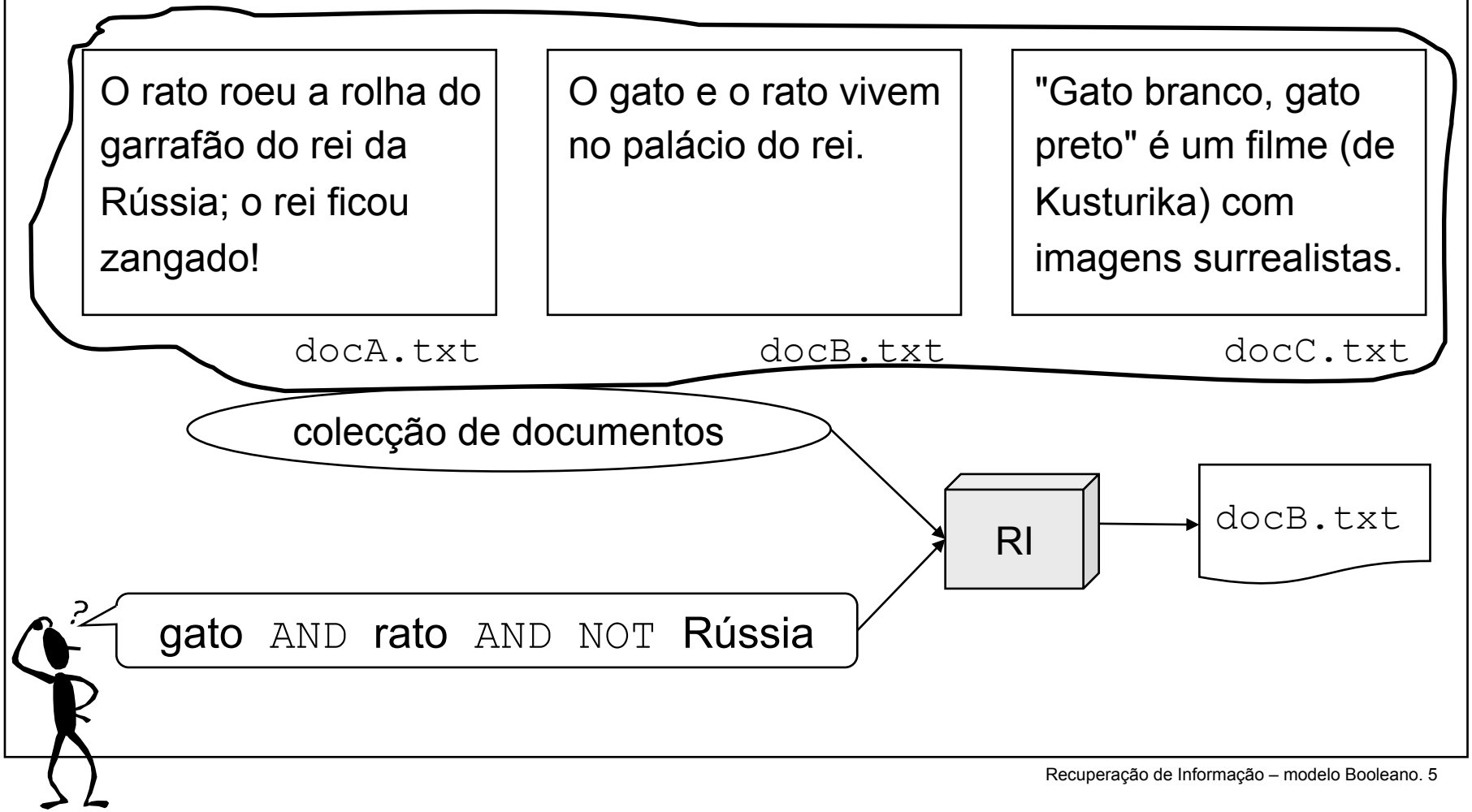
Classificação dos modelos de recuperação de informação



Paulo Trigo Silva

Modelo Booleano – sem interdependência de termos

- A interrogação tem a forma de uma expressão Booleana de termos
  - e.g. gato `AND` rato `AND NOT` Rússia

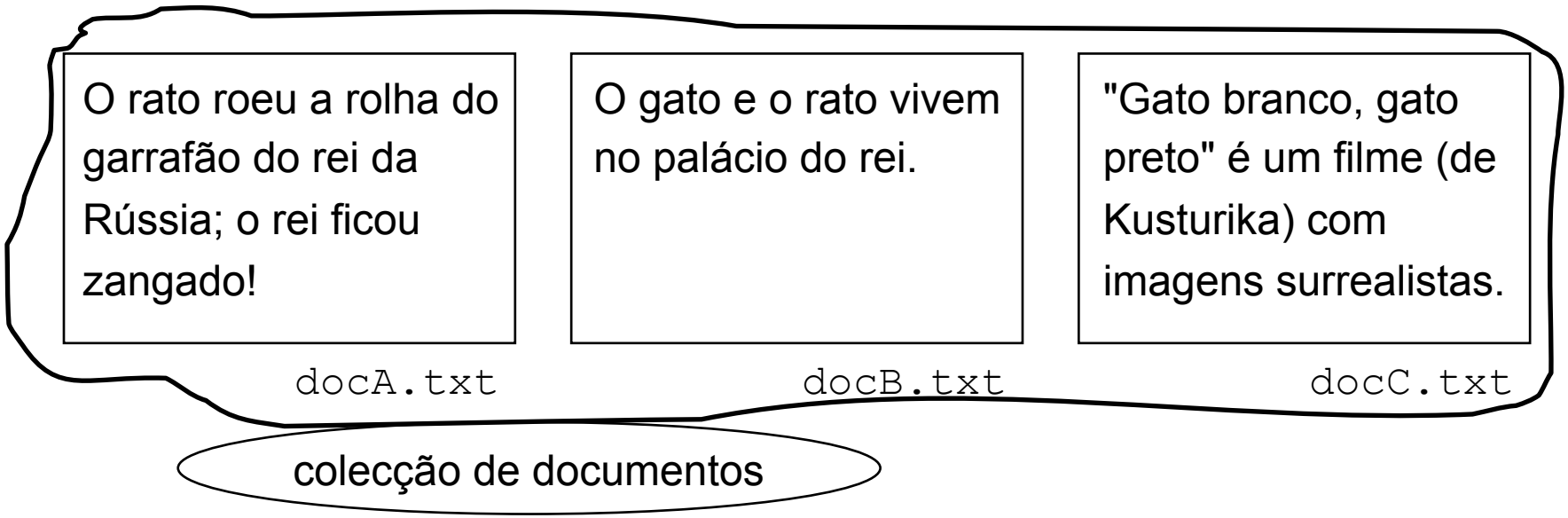


... funcionamento geral do processo de indexação

Considerando uma colecção de documentos:

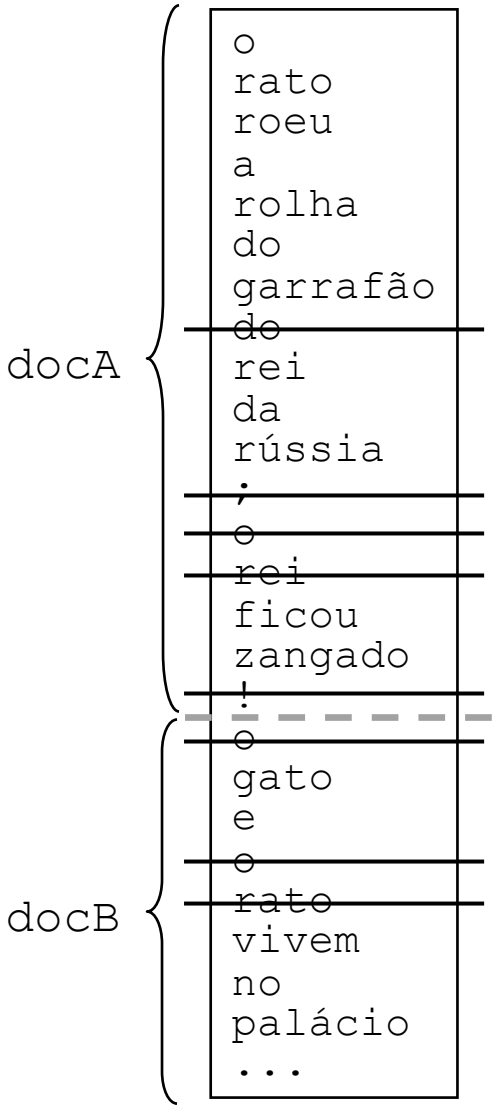
- Percorrer todos os documentos e, para cada documento,
  - construir o conjunto de todos os termos desse documento
- Fazer a união de todos os conjuntos de termos
- Construir matriz de incidência ω
  - ω_{ik} = 1, se o termo i ocorre no documento k
  - ω_{ik} = 0, se o termo i não ocorre no documento k

Exercício – construir matriz de incidência



Paulo Trigo Silva

... exemplo – conjunto de termos e matriz de incidência



documentos

alguns termos

	docA	docB	docC
...	...	...	...
rato	1	1	0
rússia	1	0	0
gato	1	1	1
palácio	0	1	0
filme	0	0	1
...	...	...	...

gato AND rato AND NOT Rússia ?

Como usar a matriz de incidência para obter a resposta?

Paulo Trigo Silva

... recuperação de informação com o modelo Booleano

gato AND rato AND NOT Rússia

	docA	docB	docC
...	...	...	...

```
111 AND 110 AND (NOT 100) =
111 AND 110 AND 011 =
110 AND 011 =
010
```

docB.txt

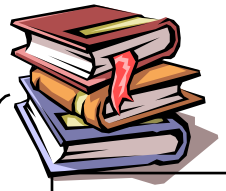
Operadores:

AND, OR e NOT

bit a bit (*bitwise*) sobre cada linha, da matriz de incidência, relativa a cada um dos termos da interrogação.

Paulo Trigo Silva

Exemplo – vamos aumentar a dimensão da colecção



Admita-se uma colecção
com a seguinte dimensão:

Atenção!
aqui estão palavras e não termos

nº de documentos = 1M (≈ 1.000.000)

nº de palavras por documento = 1K (≈ 1.000)

dimensão de cada palavra = 6 byte

Total de informação = 1M x 1K x 6 =

= 6G de informação nos documentos

E a matriz de incidência

- qual a sua dimensão total?
- qual a proporção de 0s e 1s?

```
o
rato
roeu
a
rolha
...
```

Admita-se um conjunto de termos
com a seguinte dimensão:

nº de termos distintos = 500K (≈ 500.000)

Paulo Trigo Silva

Exemplo – matriz de incidência muito esparsa

- Dimensão total da matriz de incidência
  - 500K termos distintos x 1M documentos
  - ≈ meio trilião (½ 10^{12}) de células
  - ≈ ½ terabyte de memória (só para 1M de documentos)!
- Número máximo de 1s na matriz de incidência
  - 1M documentos, cada um com 1K termos, ou seja,
  - ... há no máximo 1M x 1K = 1G (1 bilião, 10^9) de 1s na matriz
  - ... i.e. caso em que todas as palavras de cada documento são termos
- Proporção de 0s e 1s
  - ($10^9 \div$ ½ 10^{12}) x 100 = (2 x 10^{-3}) x 100 = 0.002 x 100 =
  - **no máximo, 0.2% de 1s**, e portanto
  - **no mínimo, 99.8% de 0s**

Demasiado esparso!
Então porque não representar apenas os 1s?

... matriz de incidência muito esparsa (a intuição)

Em geral, cada documento só contém um pequeno subconjunto do conjunto total dos termos!

Que obra literária incluirá todo o léxico de uma Língua?

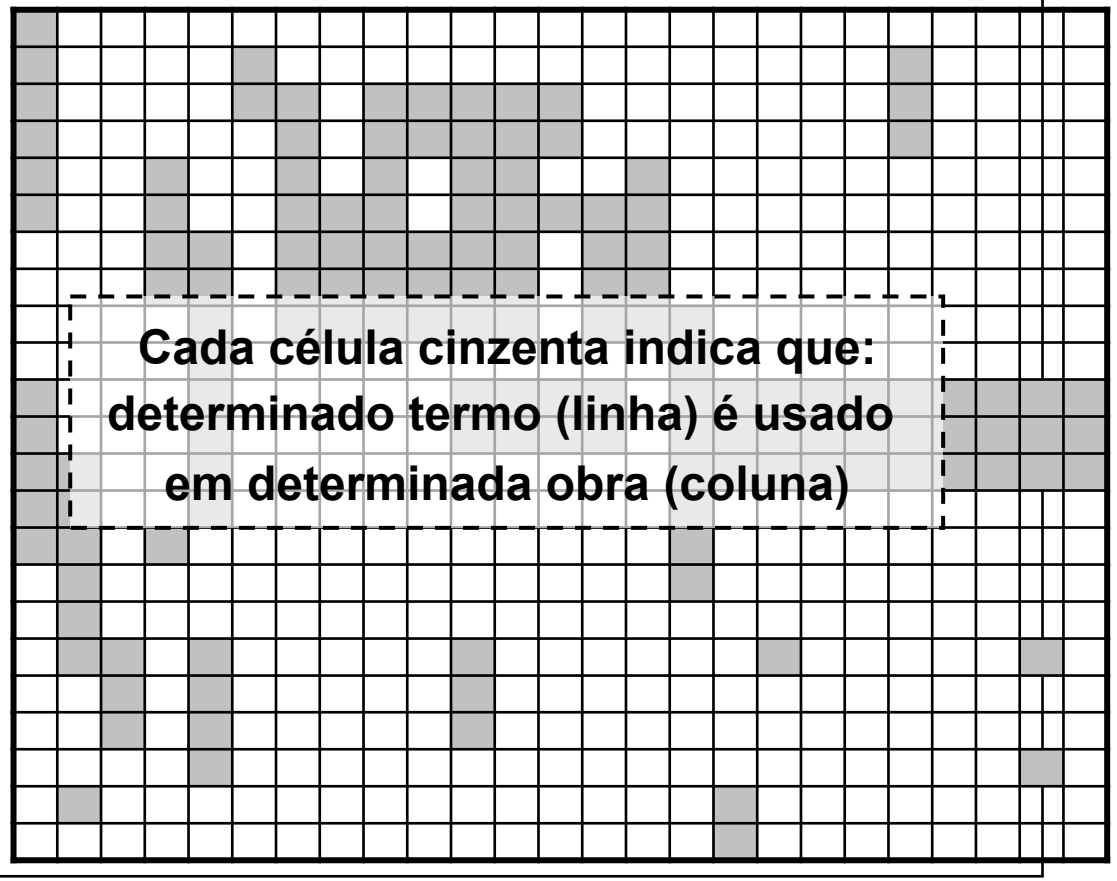
Algumas obras literárias;

e.g., “Os Lusíadas”, “Os Maias”, etc

Lista de termos

e.g., todo o léxico do Português

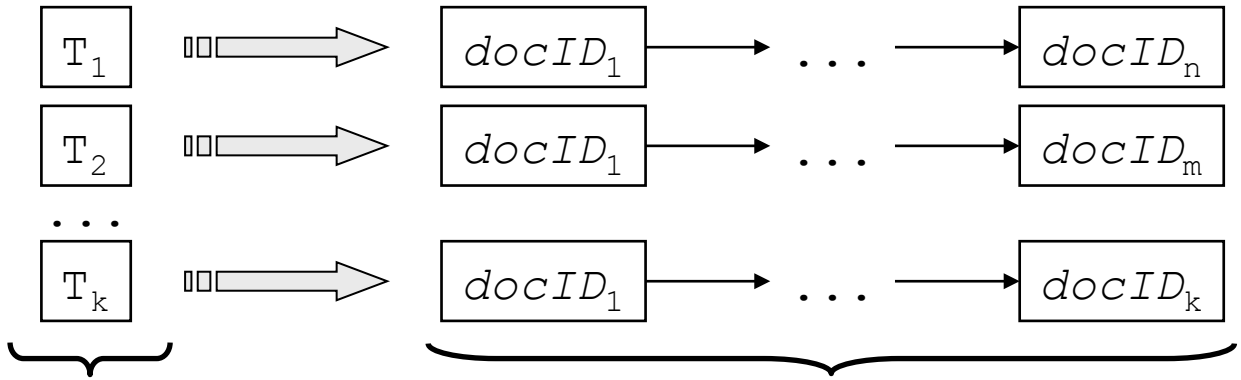
i.e. “o nosso dicionário”



Paulo Trigo Silva

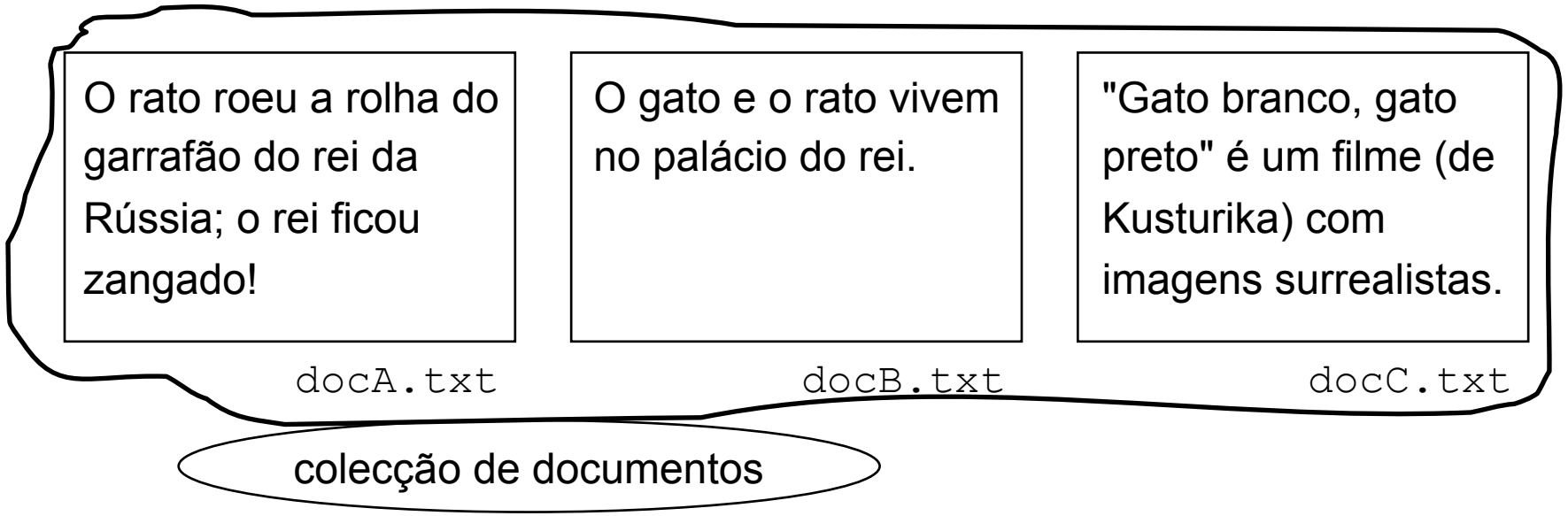
Processo de indexação – usando índices invertidos

- Atribuir a cada documento um valor inteiro único: *docID*
  - e.g. atribuído sequencialmente a cada novo documento analisado
- Para cada termo T
  - armazenar a lista de todos os *docID* que contêm T
- ... usar listas dinâmicas em vez de *arrays* estáticos
  - pois o processo evolui por incrementos sucessivos



Paulo Trigo Silva

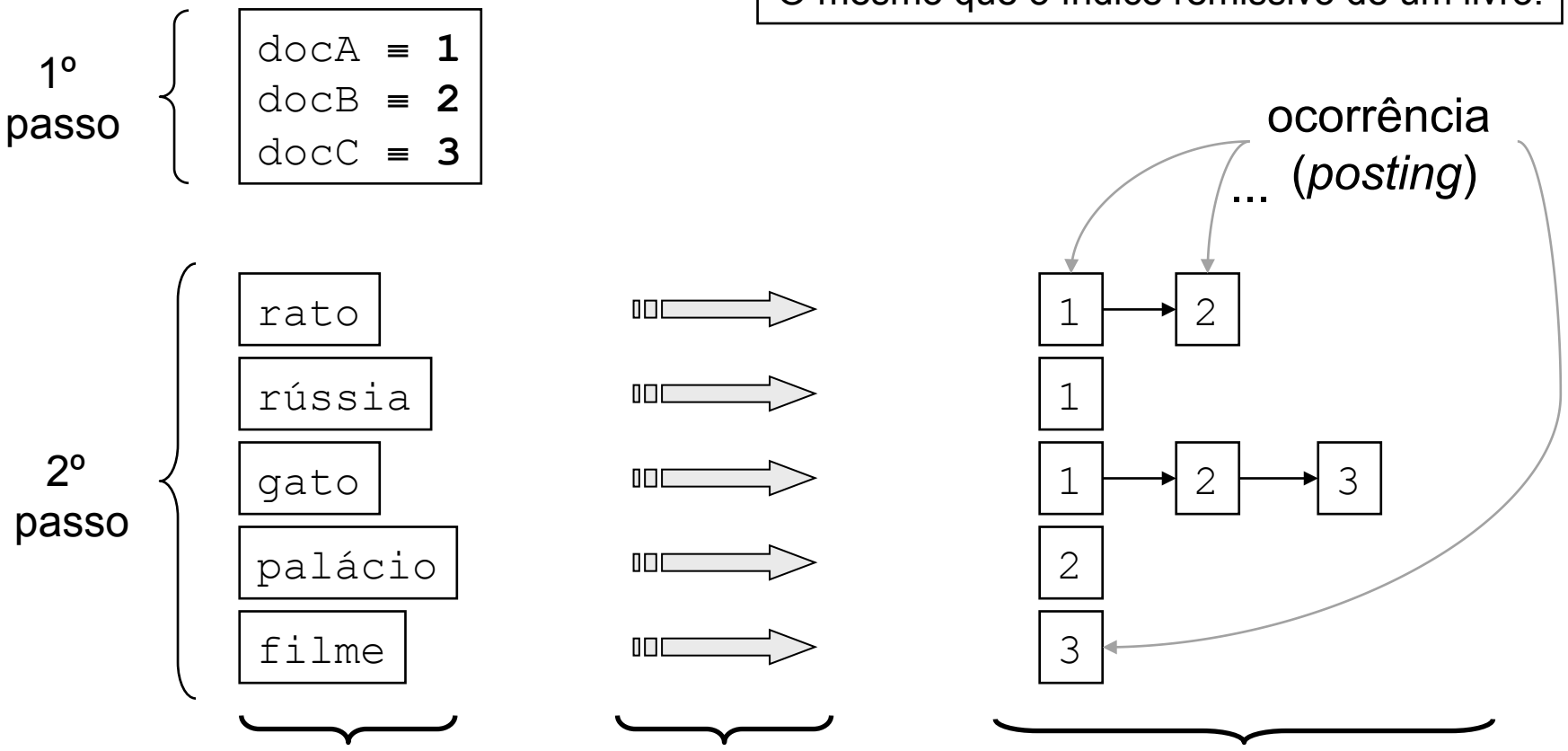
Exercício – indexar a colecção de documentos



Paulo Trigo Silva

... exemplo – índices invertidos

O mesmo que o índice remissivo de um livro!



Paulo Trigo Silva

Construção de índices invertidos – processo geral

documentos a indexar

Amigos, Romanos, e guerreiro!

...

Construção de termos (*tokenizer*)

a detalhar posteriormente

conjunto de "tokens"

Amigos	Romanos	e	guerreiro

Análise linguística

"token" na forma canónica ("steaming"); "stop word" removida

amigo	romano	guerreiro

Indexação

amigo	2 → 4 → ...	
romano	1 → 2 → ...	
guerreiro	13 → 16	

índices invertidos

Paulo Trigo Silva

Indexação – passo 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

julius.txt

docID = 1

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

brutus.txt

docID = 2

construir sequência de pares <termo, docID>

termo	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Paulo Trigo Silva

Indexação – passo 2

termo	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

ordenar por

<termo>

termo	docID
ambitious	2
be	2
brutus	1
brutus	2
caesar	1
caesar	2
caesar	2
capitol	1
did	1
enact	1
hath	2
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
was	1
was	2
with	2
you	2

Indexação – passo 3

termo	docID
ambitious	2
be	2
brutus	1
brutus	2
caesar	1
caesar	2
caesar	2
capitol	1
did	1
enact	1
hath	2
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
was	1
was	2
with	2
you	2

- agregar múltiplos termos de um mesmo documento;
- registar frequência (número de ocorrências) nesse documento

termo	docID	frequência
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
caesar	1	1
caesar	2	2
capitol	1	1
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
was	1	1
was	2	1
with	2	1
you	2	1

Paulo Trigo Silva

Indexação – passo 4

separar

termo	docID	frequência
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
caesar	1	1
caesar	2	2
capitol	1	1
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
was	1	1
was	2	1
with	2	1
you	2	1

dicionário

novo!

termo	quantidade de documentos	frequência na colecção
ambitious	1	1
be	1	1
brutus	2	2
caesar	2	3
capitol	1	1
did	1	1
enact	1	1
hath	1	1
I	1	2
i'	1	1
it	1	1
julius	1	1
killed	1	2
let	1	1
me	1	1
noble	1	1
so	1	1
the	2	2
told	1	1
was	2	2
with	1	1
you	1	1

ocorrências (*postings*)

docID	frequência
2	1
2	1
1	1
2	1
1	1
2	2
1	1
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
1	1
2	1
2	1

apontadores

Como processar uma interrogação?

Brutus `AND` Caesar

1. localizar **Brutus** no dicionário
2. obter as sua lista ocorrências
3. localizar **Caesar** no dicionário
4. obter as sua lista ocorrências
5. **intersectar** as duas listas de ocorrências

vamos admitir esta saída

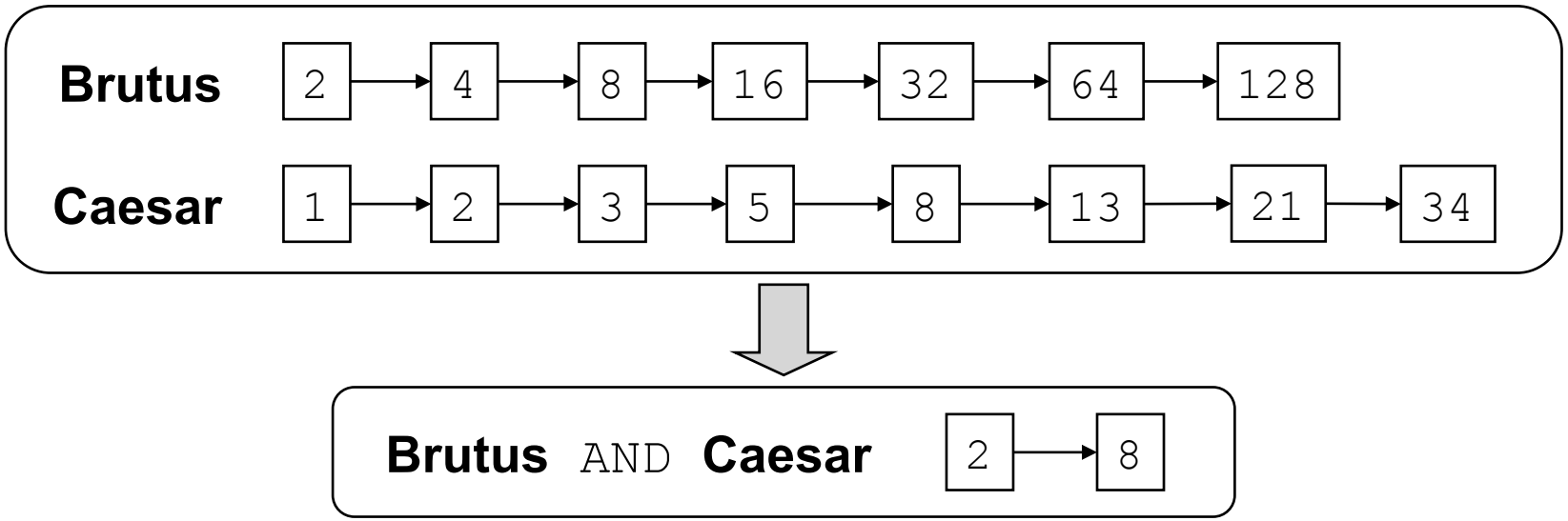
Brutus → 2 → 4 → 8 → 16 → 32 → 64 → 128

Caesar → 1 → 2 → 3 → 5 → 8 → 13 → 21 → 34

Como calcular a intersecção das duas listas?

Como calcular a intersecção das duas listas?

Percorrendo as duas listas simultaneamente, pois estão ordenadas por `docID`



Qual a relação entre o tempo para realizar a operação de intersecção e a dimensão das listas de ocorrências?

Paulo Trigo Silva

Operador de conjunção (AND) – propriedades

Tempo aumenta **linearmente** com dimensão das listas.
Pressuposto essencial: ordenação, por docID, das listas.

i.e. se as listas tiverem dimensão N e K, a intersecção tem O(N+K) operações.

O(x) ≡ tem ordem de grandeza x

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

pressuposto da ordenação

Paulo Trigo Silva

Optimizar processamento das interrogações

- Como optimizar o processamento de uma interrogação?
  - escolhendo a sequência de operações que origina o menor trabalho!
- Heurística para optimizar o processamento
  - "processar termos por ordem crescente de documentos na coleção"
- ... se começar por intersectar as duas listas de menor dimensão,
  - então os resultados intermédios serão sempre inferiores à menor lista!
  - ... e assim faremos o menor trabalho (comparações, afectações, etc)

... é essencial ter, no dicionário e por termo, o número de documentos

A escolha é feita sem acesso a disco (dicionário está em memória)

Exemplo – optimizar processamento

Brutus AND Caesar AND Calpurnia

Brutus 2 → 4 → 8 → 16 → 32 → 64 → 70

Caesar 1 → 2 → 3 → 5 → 8 → 16 → 32 → 64

Calpurnia 3 → 8 → 70

menores listas ≡ Butus; Calpurnia

Brutus AND Calpurnia 8 → 70

Caesar 1 → 2 → 3 → 5 → 8 → 16 → 32 → 64

(Brutus AND Calpurnia) AND Caesar 8

Paulo Trigo Silva

... expressão de interrogações conjuntivas – algoritmo

t_1 AND t_2 AND t_3 AND ... AND t_n

```
INTERSECT(⟨t_1, ..., t_n⟩)
1  terms ← SORTBYINCREASINGFREQUENCY(⟨t_1, ..., t_n⟩)
2  result ← POSTINGS(FIRST(terms))
3  terms ← REST(terms)
4  while terms ≠ NIL and result ≠ NIL
5  do result ← INTERSECT(result, POSTINGS(FIRST(terms)))
6     terms ← REST(terms)
7  return result
```

... ordenar

... algoritmo anterior

Paulo Trigo Silva

Outras interrogações Booleanas

... uma curiosidade:
Calpurnia Pisonis (nasceu por volta de 77 a.C.), filha de Lucius Calpurnius Piso Caesoninus, foi uma nobre romana, a terceira e última esposa de Júlio César,

(Brutus `OR` Caesar) `AND` (Calpurnia `OR` Lucius)

1. obter por termo o número de documentos em que ocorre
2. calcular soma das frequências de cada disjunção (`OR`)
3. processar por ordem crescente da dimensão de cada `OR`

perspectiva "conservadora" da estimativa da dimensão resultante

Paulo Trigo Silva

Operador de disjunção (OR)

... substituir, no algoritmo da conjunção (AND), por

```
INTERSECT(p1, p2)
1  answer ← ⟨ ⟩
2  while p1 ≠ NIL and p2 ≠ NIL
3  do if docID(p1) = docID(p2)
4        then ADD(answer, docID(p1))
5             p1 ← next(p1)
6             p2 ← next(p2)
7        else if docID(p1) < docID(p2)
8               then p1 ← next(p1)
9               else p2 ← next(p2)
10 return answer
```

```
Union(p1, p2)
...
Add(answer, docID(p2))
do if docID(p1) ≠ docID(p2)
...
```

o resto do algoritmo não se altera

Importante:
admite-se que `Add` só insere caso o último elemento inserido seja diferente do que se pretende inserir.

Paulo Trigo Silva

Operador de subtracção (composição de AND e NOT)

Brutus AND NOT Caesar

... substituir, no algoritmo da conjunção (AND), por

```
INTERSECT(p1, p2)
1  answer ← ⟨ ⟩
2  while p1 ≠ NIL and p2 ≠ NIL
3  do if docID(p1) = docID(p2)
4        then ADD(answer, docID(p1))
5             p1 ← next(p1)
6             p2 ← next(p2)
7        else if docID(p1) < docID(p2)
8               then p1 ← next(p1)
9               else p2 ← next(p2)
10 return answer
```

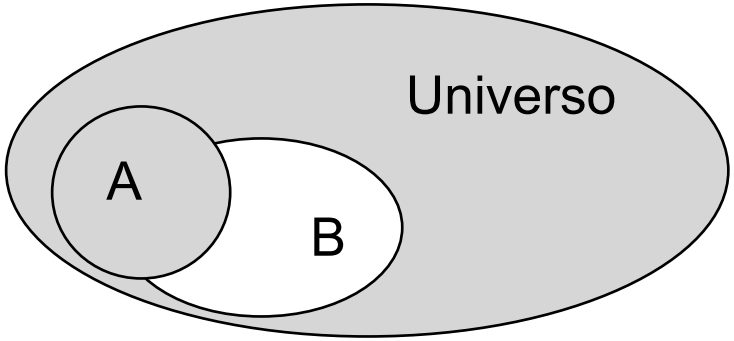
```
Minus(p1, p2)
...
do if docID(p1) ≠ docID(p2)
...
```

o resto do algoritmo não se altera

Paulo Trigo Silva

Operador para composição de OR e NOT

Brutus OR NOT Caesar



A OR NOT B
=
Universo - (B - A)

resolver invocando as funções já definidas
ou
construir novo algoritmo para optimizar esta operação...

Paulo Trigo Silva

Exercício (ordem de processamento)

(tangerina OR árvore) AND
(goiabada OR nuvem) AND
(caleidoscópio OR olho)

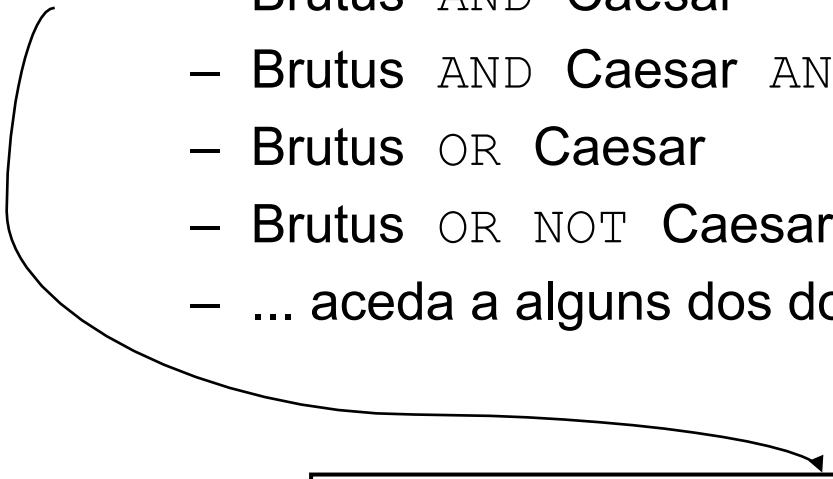
termo	frequência na colecção
olho	213312
caleidoscópio	97009
goiabada	107913
nuvem	271658
tangerina	46653
árvore	316812

Recomende uma ordem de processamento (de operadores) para executar a interrogação

Paulo Trigo Silva

Exercício (criticar motor de busca)

- Utilize o motor de busca sobre textos de Shakespeare, em
  - `http://www.rhymezone.com/shakespeare/`
- Experimente a pesquisa por "keyword" para encontrar:
  - Brutus `AND` Caesar
  - Brutus `AND` Caesar `AND NOT` Calpurnia
  - Brutus `OR` Caesar
  - Brutus `OR NOT` Caesar
  - ... aceda a alguns dos documentos encontrados.



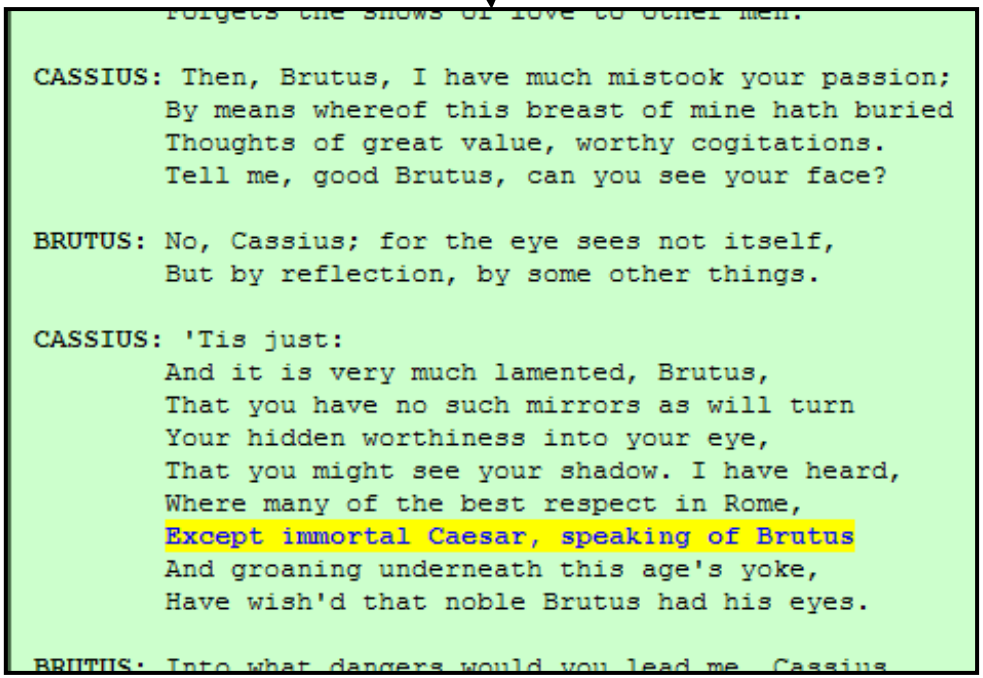
Proponha 5 aspectos a melhorar neste motor de pesquisa.

Paulo Trigo Silva

... pesquisar textos de Shakespeare

... um dos textos encontrados

Except immortal ***caesar***, speaking of ***brutus*** → Julius Caesar: I, ii



forgets the shows of love to other men.

CASSIUS: Then, Brutus, I have much mistook your passion;
By means whereof this breast of mine hath buried
Thoughts of great value, worthy cogitations.
Tell me, good Brutus, can you see your face?

BRUTUS: No, Cassius; for the eye sees not itself,
But by reflection, by some other things.

CASSIUS: 'Tis just:
And it is very much lamented, Brutus,
That you have no such mirrors as will turn
Your hidden worthiness into your eye,
That you might see your shadow. I have heard,
Where many of the best respect in Rome,
Except immortal Caesar, speaking of Brutus
And groaning underneath this age's yoke,
Have wish'd that noble Brutus had his eyes.

BRUTUS: Into what dangers would you lead me, Cassius

Paulo Trigo Silva

Outras pesquisas (para além dos termos)

Pesquisar:

- o rato roeu a rolha
- Instituto Superior de Engenharia de Lisboa

→ frases

- Torvalds *PERTO-DE* Linux
- Gates *PERTO-DE* Microsoft

→ proximidade (capturar posição no documento)

- autor = Ullman `AND` (texto *CONTÉM* autómato)

→ zonas do documento

Paulo Trigo Silva

Como apresentar o resultado da pesquisa?

O modelo Booleano responde a uma interrogação com o conjunto dos documentos que satisfazem a condição de pesquisa.

- Mas é preciso oferecer primeiro aquilo que é mais relevante!
  - ... implica impor relação de ordem ao conjunto obtido (passar a lista)
  - ... implica agrupar documentos que cobrem vários aspectos da interrogação
- Para definir uma relação de ordem
  - é necessário "medir a proximidade" da interrogação a cada documento
- Para agrupar documentos
  - é necessário identificar os documentos que "melhor cobrem" a interrogação

Paulo Trigo Silva

Outros problemas – agrupamento e classificação

O problema do agrupamento ("clustering"):

Dado um conjunto de documentos caracterizar subconjuntos (i.e., fazer grupos ou "clusters") com conteúdo semelhante.

... algoritmo muito conhecido: **K-means**

constrói K grupos optimizando critério de partição (semelhança).

O problema da classificação ("classification"):

Dado um conjunto de tópicos (classes) e um novo documento D (objecto), decidir a que tópico (classe) pertence esse documento.

... algoritmo muito conhecido: **"K Nearest Neighbors" (KNN)**

atribui o objecto à classe mais comum nos seus K vizinhos.

... existem diversos outros algoritmos para atacar estes problemas!

Paulo Trigo Silva

Sistemas de RI – código fonte aberto (“open source”)

- Lucene (“The Apache Software Foundation”)
  - `http://lucene.apache.org/`
- Lemur (“CMU, University of Massachusetts”)
  - `http://www.lemurproject.org/`
- Zettair (“RMIT, Australia”)
  - `http://www.seg.rmit.edu.au/zettair/`
- MG (“RMIT & Melbourne, Australia; Waikato, New Zealand”)
  - `http://www.ncsi.iisc.ernet.in/raja/netlis/wise/mg/mainmg.html#Features`
- ... e outros!

Exercício

Identifique sistemas de recuperação de informação.

Faça uma análise comparativa de 3 sistema de recuperação de informação.

Paulo Trigo Silva