Universidade Federal do Rio de Janeiro (UFRJ)
Departamento de Ciência da Computação (DCC)

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO - UFRJ -

Recuperação da Informação (MAB605)

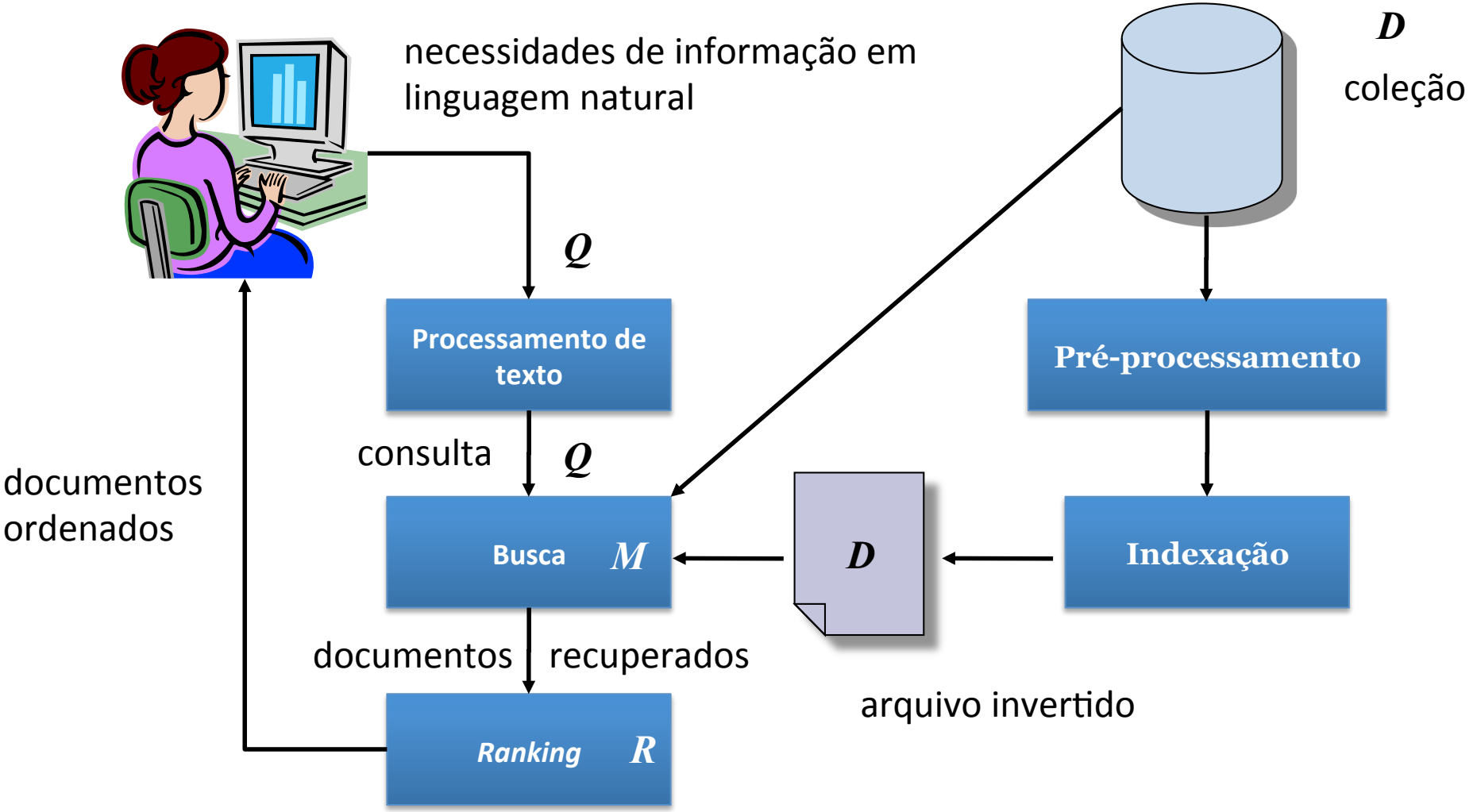
Pré-processamento

Profa. Giseli Rabello Lopes

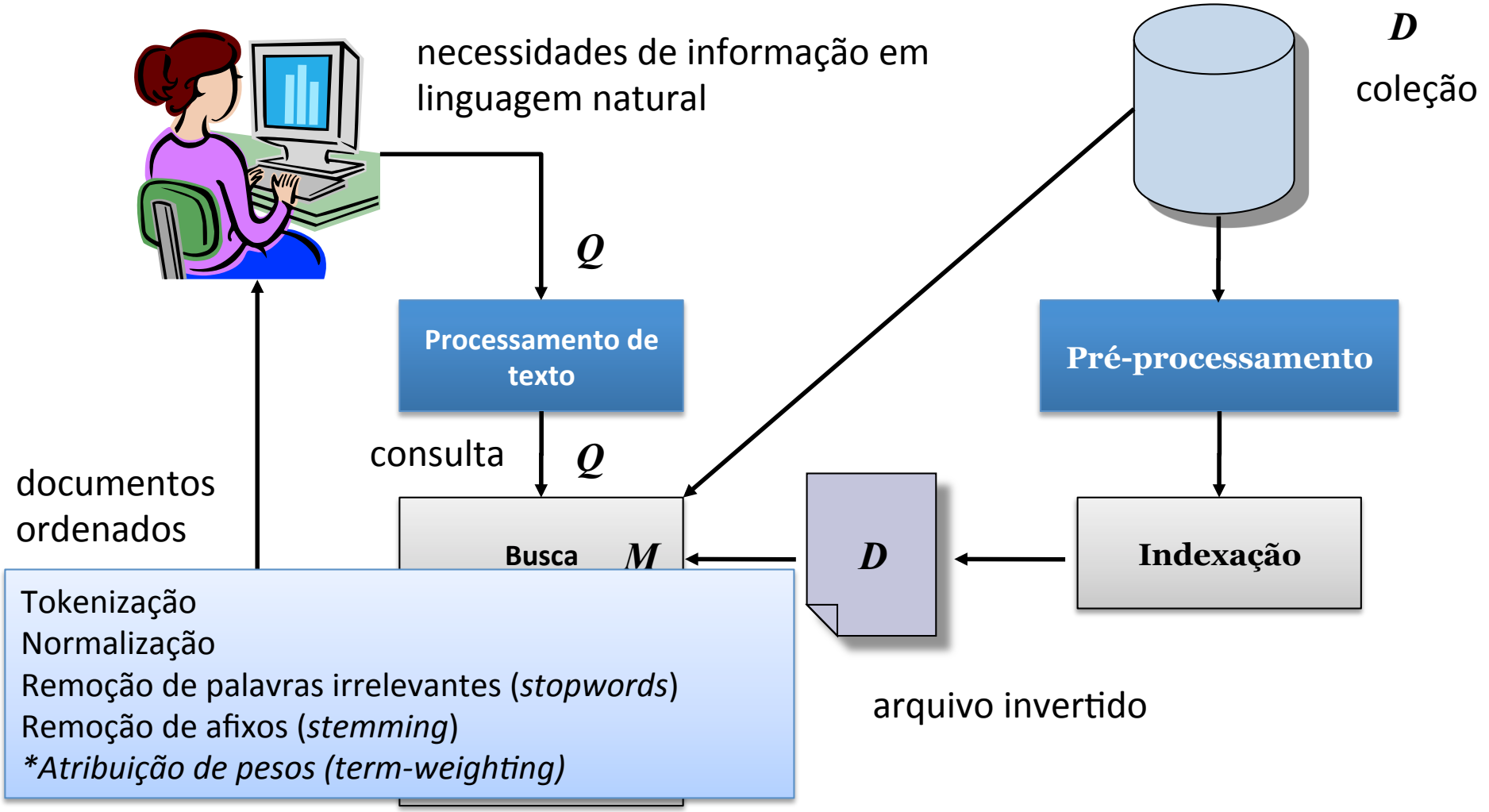
Roteiro

- Introdução
- Pré-processamento de documentos
  - Análise léxica do texto
  - Eliminação de *stopwords*
  - *Stemming*
  - Seleção de palavras-chave
  - Tesauros
- Referências

Componentes de um Sistemas de RI Relembrando...



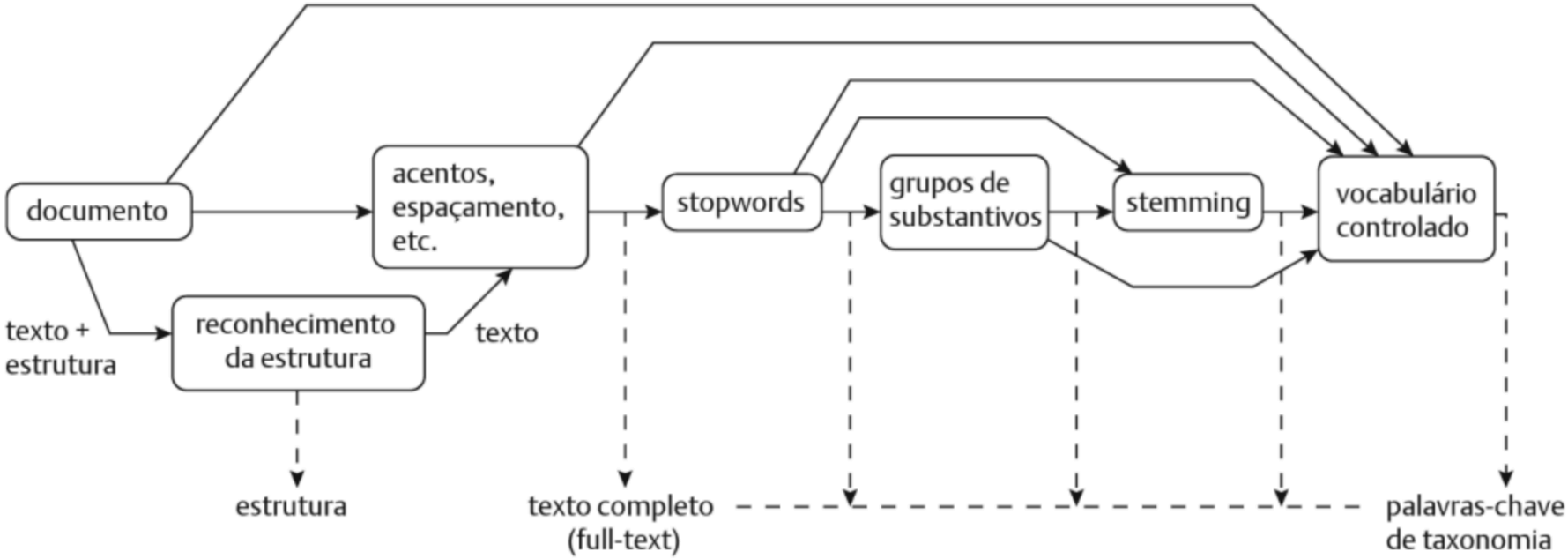
Componentes de um Sistemas de RI Relembrando...



Pré-processamento de documentos
[Baeza-Yates & Ribeiro-Neto, 2013]

- Pode ser dividido em cinco operações (ou transformações) textuais:
  1. Análise léxica do texto
  2. Eliminação de *stopwords*
  3. S*temming* das palavras remanescentes
  4. Seleção de termos de índice ou palavras-chave
  5. Construção de estruturas de categorização de termos (tesauros) - detalhes em aula posterior sobre expansão de consulta

Visão lógica de um documento



Fonte: [Baeza-Yates & Ribeiro-Neto, 2013]

Pré-processamento de documentos
[Baeza-Yates & Ribeiro-Neto, 2013]

- Pode ser dividido em cinco operações (ou transformações) textuais:
  1. Análise léxica do texto
  2. Eliminação de *stopwords*
  3. S*temming* das palavras remanescentes
  4. Seleção de termos de índice ou palavras-chave
  5. Construção de estruturas de categorização de termos (tesauros) - detalhes em aula posterior sobre expansão de consulta

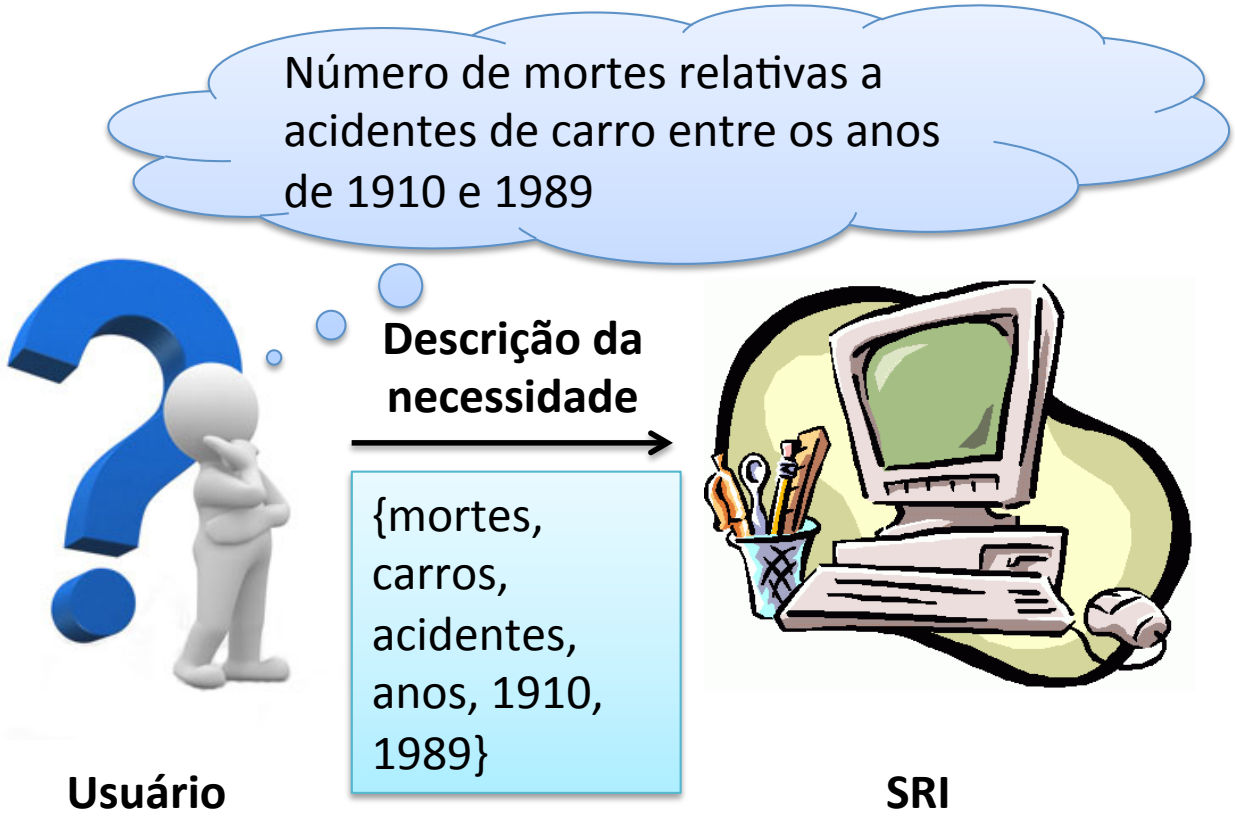
Análise léxica do texto

[Baeza-Yates & Ribeiro-Neto, 2013]

- Processo de converter uma sequência de caracteres em uma sequência de palavras
- Principal objetivo: identificar palavras no texto
- Separadores de palavras:
  - **Espaço:** separador mais comum
  - **Números:** inerentemente vagos, precisam de contexto para desambiguação
  - **Hífens:** quebrar palavras com hífen
  - **Marcas de pontuação:** permitem distinguir **x.id** de **xid** em um programa
  - **"Caixa" das letras:** permite distinguir **Banco** de **banco**

Análise léxica do texto
[Baeza-Yates & Ribeiro-Neto, 2013]

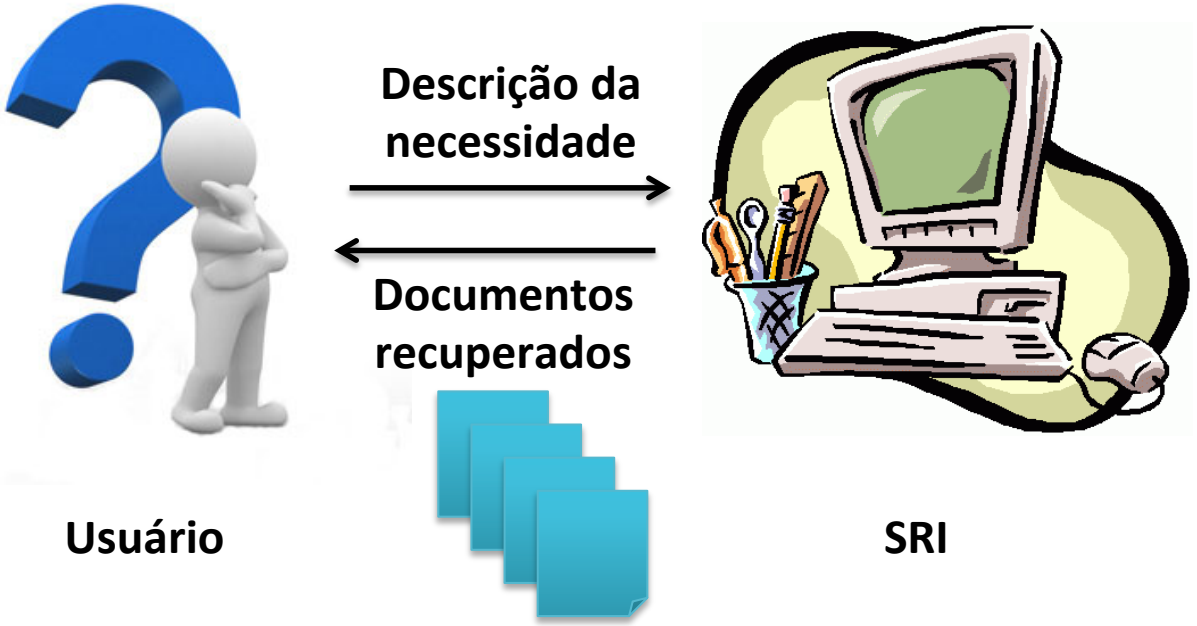
- Outro exemplo:



Análise léxica do texto

[Baeza-Yates & Ribeiro-Neto, 2013]

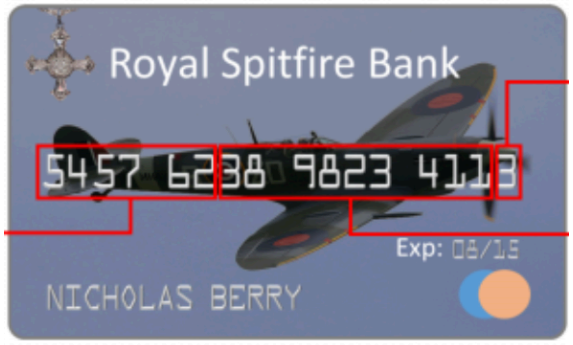
Variedade de documentos que não se referem a nenhum desses anos; Números por si só são muito vagos; Em geral desconsiderados como termos de indexação!



Análise léxica do texto

[Baeza-Yates & Ribeiro-Neto, 2013]

- Outros exemplos:
  - 510A.C.
  - 5457613898234113
  - 20-10-15; 20/10/2015; 20 de outubro de 2015
    - Análise léxica avançada → normalizar datas e números e unificar os formatos (versões numéricas e por extenso de datas ou números)
  - estado-da-arte → estado da arte
  - guarda-chuva



Não há solução clara!
Algumas máquinas de busca na Web estão optando por evitar completamente as operações textuais.

Pré-processamento de documentos
[Baeza-Yates & Ribeiro-Neto, 2013]

- Pode ser dividido em cinco operações (ou transformações) textuais:
  1. Análise léxica do texto
  2. Eliminação de *stopwords*
  3. *Stemming* das palavras remanescentes
  4. Seleção de termos de índice ou palavras-chave
  5. Construção de estruturas de categorização de termos (tesauros) - detalhes em aula posterior sobre expansão de consulta

Eliminação de *stopwords*
[Baeza-Yates & Ribeiro-Neto, 2013]

- *Stopwords*
  - Palavras que aparecem muito frequentemente
  - Geralmente, não são boas como discriminantes
  - Normalmente, removidas dos termos de índice em potencial
  - Candidatas naturais: artigos, preposições, conjunções
- Eliminação de *stopwords*
  - Reduz o tamanho do índice em 40% ou mais
  - À custa de reduzir *recall*: incapaz de recuperar documentos que contenham **"to be or not to be"**

Adoção de índice textual completo (sem remoção de *stopwords*) é feita por algumas máquinas de busca na Web.

Recursos

- "O objetivo da Linguateca, um **centro de recursos -- distribuído -- para o processamento computacional da língua portuguesa**, é servir a comunidade que se dedica ao processamento da nossa língua"
  - http://www.linguateca.pt
  - Listas de *stopwords* na língua portuguesa: http://www.linguateca.pt/chave/stopwords/

Recursos

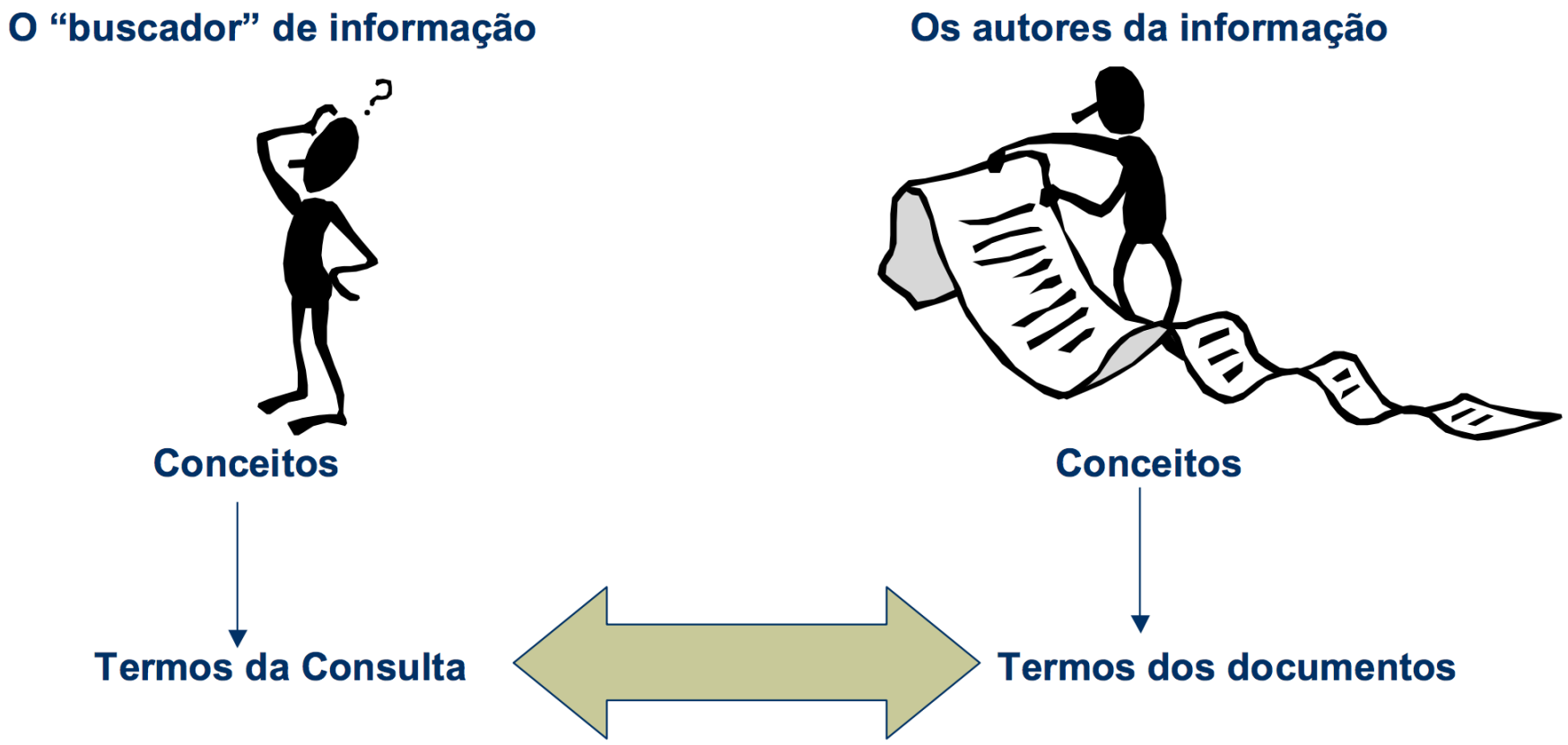
- "The Lemur Project develops search engines, browser toolbars, text analysis tools, and data resources that support research and development of information retrieval and text mining software."
  - http://www.lemurproject.org

  - Lista de *stopwords* da língua inglesa (*Lemur*)
    - http://www.search-engines-book.com/data/stopwords

Pré-processamento de documentos
[Baeza-Yates & Ribeiro-Neto, 2013]

- Pode ser dividido em cinco operações (ou transformações) textuais:
  1. Análise léxica do texto
  2. Eliminação de *stopwords*
  3. S*temming* das palavras remanescentes
  4. Seleção de termos de índice ou palavras-chave
  5. Construção de estruturas de categorização de termos (tesauros) - detalhes em aula posterior sobre expansão de consulta

Problema central em RI



Será que esses termos representam os mesmos conceitos?

Stemming

[Baeza-Yates & Ribeiro-Neto, 2013]

- Usuário especifica palavra em uma consulta mas apenas uma variante dessa está presente em um documento relevante
  - Plurais, gerúndios e sufixos que indicam passado (variações sintáticas)
  - Parcialmente resolvido pela adoção de *stems*
- ***Stem***
  - Porção de uma palavra que resta após a remoção de afixos (prefixos/sufixos)
  - Ex.: **connect** – stem de *connected, connecting, connection, connections*

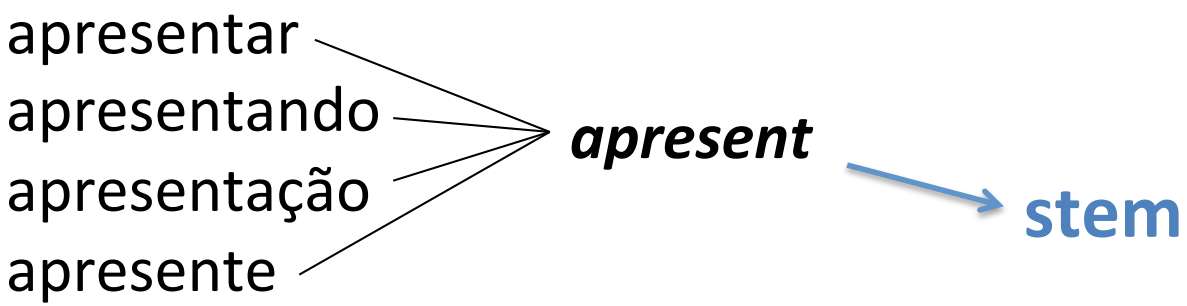
Stemming

[Baeza-Yates & Ribeiro-Neto, 2013]

- *Stemming* reduz o tamanho do índice
- Existem controvérsias sobre os benefícios de *stemming* para recuperação
- Muitas máquinas de busca não adotam algoritmo algum de *stemming*

Stemming

- Exemplo:
  - Consulta: “como apresentar artigos científicos”
  - Doc 1: … apresentando um artigo científico …
  - Doc 2: … apresentação de um artigo científico …
  - Doc 3: … apresente artigos científicos …



Stemming

[Baeza-Yates & Ribeiro-Neto, 2013]

- Tipos de estratégias de *stemming*:
  - Busca em tabela: busca do *stem* de uma palavra em uma tabela
  - Remoção de afixos: na qual a parte mais importante é a remoção de sufixos
  - Variedade de sucessores: determina os limites dos morfemas e utiliza conhecimento da linguística estrutural
  - N-gramas: identifica digramas e trigramas (*clustering* de termos)

Stemming – Busca em tabela

- Utiliza uma tabela com todos os termos de indexação e seus respectivos *stems*

Termo	*Stem*
abandona	*abandon*
abandonar	*abandon*
abandonado	*abandon*

- O processo de radicalização consiste em consulta a esta tabela
- Problemas:
  - Não existem tabelas de stems prontas e sua construção é cara
  - *Overhead* do armazenamento da tabela

Stemming – Remoção de afixos

- Removem prefixos e/ou sufixos das palavras deixando o *stem*
- A maioria dos *stemmer*s em uso atualmente enquadram-se nesta categoria
- Ex.:
  - Porter – "An algorithm for suffix stripping" Program 14(3), 130-137. 1980.
    - Algoritmo de remoção de sufixos para a língua inglesa

Stemming – Porter Stemmer

- Efetua a remoção de sufixos
- Baseia-se em regras que são aplicadas caso determinadas condições forem satisfeitas
  - Condição sobre o stem
    - Medida do número de sequências de vogais-consoantes
    - Conter um determinado caracter
  - Condições sobre o sufixo
  - Condições sobre a regra
- Regras são agrupadas em passos (5 passos)
  - Apenas uma regra pode ser aplicada em cada passo
  - Dentro de um passo, o sufixo mais longo é testado primeiro

Stemming – Porter Stemmer

- Ex.: Regras de redução de plural (*Step* 1.a)

– sses → ss	stresses → stress
– ies → i	ponies → poni
– ss → ss	caress → caress
– s → Ø	cats → cat

Stemming – Porter Stemmer

- Regras usando conceito da "medida" de uma palavra
  - Verificar o número de sílabas para ver se uma palavra é longa o suficiente para que seja razoável considerá-la parte correspondente de uma regra como um **sufixo** ao invés de parte do *stem* de uma palavra

Stemming – Porter Stemmer

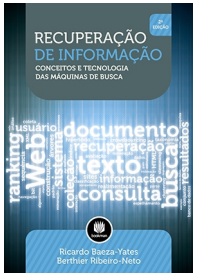
- [C](VC){m}[V]
  - onde: V = vogal; C = consoante; (VC){m} = VC repetida m vezes; [ ] opcional
  - Ex.:
    - m=0 tr, ee, tree, y, by
    - m=1 tro**ub**le, o**at**s, tre**es**, **iv**y
    - m=2 tro**ub**l**es**, pr**iv**at**e**, o**at**e**n**, **or**r**er**y

- (m > 1) ement → Ø r**ep**l**ac**ement → replac

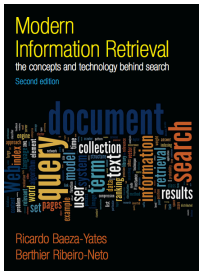
Stemming – Porter Stemmer

- Recursos
  - Site (informações e implementações em diversas linguagens de programação)
    - http://tartarus.org/~martin/PorterStemmer/
  - Texto do artigo original
    - http://tartarus.org/~martin/PorterStemmer/def.txt
  - *Javascript* online
    - http://9ol.es/porter_js_demo.html

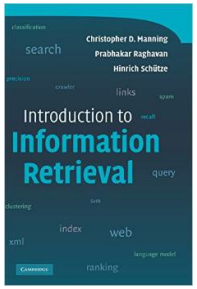
Referências



- Baeza-Yates, R.; Ribeiro-Neto, B. Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca. 2 ed. Bookman, 2013.



- Baeza-Yates, R.; Ribeiro-Neto, B. Modern Information Retrieval. Wokingham, UK: Addison-Wesley, 2 ed., 2011.



- Manning, C. D.; Raghavan, P.; Schütze, H. Introduction to Information Retrieval. Cambridge University Press, 2008.

  Online edition 2009: http://nlp.stanford.edu/IR-book/

Universidade Federal do Rio de Janeiro (UFRJ)
Departamento de Ciência da Computação (DCC)

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO - UFRJ -

Recuperação da Informação (MAB605)

Dúvidas?

Profa. Giseli Rabello Lopes
giseli@dcc.ufrj.br
CCMN - DCC - Sala E-2012