Gerenciamento de Dados e Informação

Recuperação de Informação

Fernando Fonseca
Ana Carolina
Robson Fidalgo

Recuperação de Dado X Informação

	Recuperação de Dados	**Recuperação de Informação**
Comparação (*matching*)	Exata	Aproximada
Dados	Estruturados	Não estruturados
Inferência	Dedução	Indução
Modelo	Determinístico	Probabilístico
Ling Consulta	Artificial	Natural
Esp da Consulta	Completa	Incompleta

Recuperação de Informação

- Área de pesquisa e desenvolvimento que investiga métodos e técnicas para a representação, a organização, o armazenamento, a busca e a recuperação de ***itens de informação***
- Objetivo principal
  - Facilitar o acesso a documentos (itens de informação) relevantes à ***necessidade de informação*** do usuário

Histórico

- **1ª Fase**: computadores – cartão perfurado
  - Década de 1950
    - Aplicações: sistemas de recuperação de referências bibliográficas e outros serviços para bibliotecas
    - Técnicas: indexação manual
      - Documentos indexados por termos de um vocabulário restrito montado manualmente (*thesaurus* = dicionário de sinônimos)

Histórico

- **1ª Fase**: computadores – cartão perfurado (Cont.)
  - Década de 1960
    - Aplicações: sistemas de recuperação de documentos *off-line*
    - Sistemas DIALOG e MEDLARS
    - Técnicas: início da indexação automática
      - Título e abstract
    - Algoritmos de busca

Histórico

- **2ª Fase:** Décadas de 1970 e 1980
  - Aumento do poder computacional
  - Aplicações
    - Sistemas de Pergunta-Resposta
      - Técnicas: RI + Processamento de Linguagem Natural
      - Evoluíram para interfaces em Linguagem Natural para BD

Histórico

- **2ª Fase:** Décadas de 1970 e 1980 (Cont.)
  - Aplicações
    - Sistemas de RI on-line
      - Técnicas: Estatística e Probabilidade, Modelo de Espaço Vetorial (Salton 71)
      - SMART: 1º sistema de RI automático para o conteúdo usando Espaço Vetorial
      - Avaliação do desempenho do sistema pelo usuário

Histórico

- **3ª Fase:** Web – Década de 1990 em diante
  - Técnicas tradicionais de RI foram adaptadas ao caso da Web
    - Web: terabytes de dados não estruturados
  - Alguns problemas
    - Escalabilidade das soluções
    - Velocidade de atualização da Web
    - Velocidade de acesso aos documentos armazenados
  - Explosão de serviços + agentes autônomos

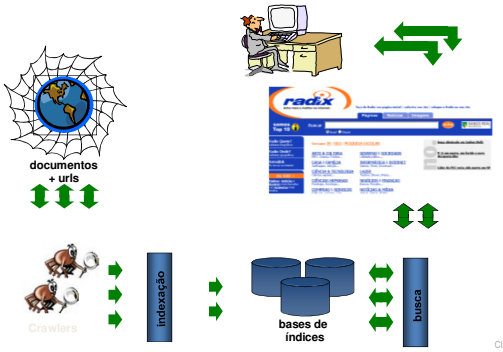
Engenhos de Busca

- A primeira ferramenta usada para consultar a Web
  - Baseados na busca de índices de **palavras** e **frases** que aparecem em documentos
  - Posteriormente foi incluída a exploração da estrutura de ***links*** para aumentar a qualidade das respostas

Engenhos de Busca

- Web & engenhos de busca
  - Facilidade de criação de novos documentos
  - Documentos heterogêneos, semi-estruturados
  - Grande número de informações disponíveis
  - Informação dinâmica
  - Maior número de pessoas interagindo com o sistema
  - Necessidade definida por meio de consulta

Engenhos de Busca



Engenhos de Busca

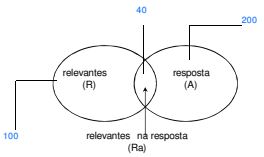
- Interação usuário-engenho de busca
  - Consultas por palavra-chave, linguagem natural
  - Dificuldade em formular consultas adequadas
  - Consultas mal formuladas, resultados de baixa precisão

Recuperação de Informação

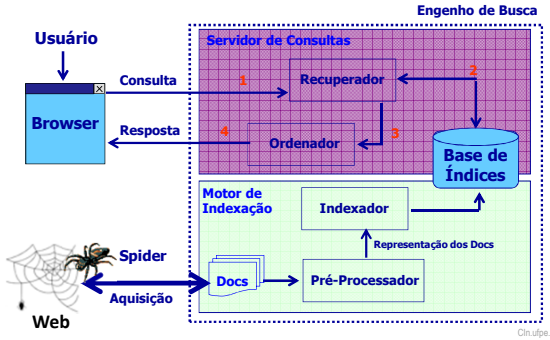
- Sistemas de Recuperação de Informação (SRI)
  - Um *sistema para RI automático* pode ser visto como a parte do sistema de informação responsável pelo armazenamento ordenado dos documentos em um BD, e sua posterior recuperação, para responder a consultas de usuários

Recuperação de Informação

- Avaliando SRI
  - Relevância (R)
  - Relevância na resposta (Ra)
  - Precisão (Ra/R)
  - Cobertura (Ra/A)



Sistemas de RI na Web Arquitetura



Sistemas de Recuperação de Informação

- Etapas principais
  - Aquisição (seleção) dos documentos
  - Representação (preparação) dos documentos
  - Indexação dos documentos
  - Busca (casamento com a consulta)
  - Recuperação

Etapa 1: Aquisição (seleção) de Documentos

- Manual ⇒ Para sistemas gerais de RI
  - E.g.: Sistemas de bibliotecas
- Automática ⇒ Para sistemas na Web
  - Uso de *crawlers* (*spiders*)
    - Programas que navegam pela Web e fazem *download* das páginas para um servidor
    - Partem de um conjunto inicial de *links*
    - Executam busca em largura ou em profundidade

Etapa 1: Aquisição (seleção) de Documentos

- Automática (cont)
  - *Crawler* do Google
    - Executa em várias máquinas em paralelo
    - Indexa milhões de páginas por dia

Etapa 2: Pré-Processamento dos Documentos

- Objetivo
  - Criar uma representação computacional do documento seguindo algum modelo
- Fases
  - Operações sobre o texto
  - Criação da representação

Etapa 2: Pré-Processamento dos Documentos

Pré-Processamento: Operações sobre o texto

- Análise léxica
  - Converte uma cadeia de caracteres em uma cadeia de palavras/termos
- Eliminação de *stopwords*
  - Palavras consideradas irrelevantes
    - Ex.: artigos, pronomes, alguns verbos, "WWW"...

Pré-Processamento: Operações sobre o texto

- *Stemming*
  - Redução de uma palavra ao seu radical
    - Geralmente, apenas eliminação de sufixos
  - Possibilita casamento entre variações de uma mesma palavra

Pré-Processamento: Operações sobre o texto

- *Stemming*

Termo	Stem
engineering	engineer
engineered	engineer
engineer	engineer

Regras de redução:

ing -> 0

ed -> 0

Pré-Processamento: Representação do Documento

- Texto Completo
  - Difícil (caro) de manipular computacionalmente
- Dado um documento, identificar os conceitos que melhor descrevem o seu conteúdo
- Representar o documento como um Centróide
  - Lista de termos **com pesos associados ou não**
  - Problema: perda da semântica

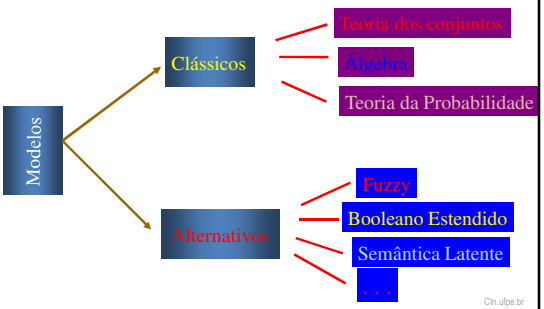
Pré-Processamento: Representação do Documento

- Representação do documento como um Centróide

"Se o desonesto soubesse a vantagem de ser honesto, ele seria honesto ao menos por desonestidade."
Sócrates

Centróide	
honesto	2
desonesto	1
soubesse	1
vantagem	1
seria	1
menos	1
desonestidade	1
socrates	1

Modelos de Representação de Documentos



Modelo Booleano

- Baseado na Teoria dos Conjuntos
- Documentos e consultas são representados como conjuntos de termos de índices
- Centróide sem pesos associados
- A representação indica apenas se o termo está ou não presente no documento

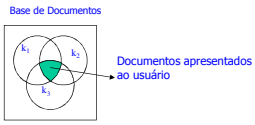
Modelo Booleano: sem pesos associados

- Simples de implementar e usar, porém de baixo desempenho
- Documentos e consultas representados como vetores binários de tamanho *n*
  - Cada posição corresponde a um termo usado na indexação dos documentos sendo considerados
  - Consulta: termos conectados por *AND*, *OR* e *NOT*

Modelo Booleano: sem pesos associados

- Relevância "binária"
  - O documento é considerado relevante *sse* seu "casamento" com a consulta é verdadeiro
  - Não é possível ordenar os documentos recuperados

Consulta: $k_1 \wedge k_2 \wedge k_3$



Modelo Espaço Vetorial

- Modelo Algébrico
- Centróide com pesos associados

Modelo Espaço Vetorial:
com pesos associados

- Consultas (*q*) e Documentos (*d*) são representados como vetores em um espaço *n*-dimensional
  - Onde *n* é o número total de termos usados para indexar os documentos sendo considerados
- Relevância: cosseno do ângulo entre *q* e *d*
  - Quanto maior o cosseno, maior é a relevância de *d* para *q*

Modelo Espaço Vetorial:
com pesos associados

- Ordenação: dada pelo cosseno do ângulo entre *q* e *d*

Consulta **q** : Brasil Olimpíadas Sidney

Documento **d** :

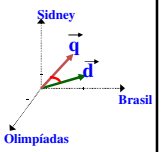
Brasil em Sidney 2000
O Brasil não foi bem no quadro das medalhas da Olimpíada de Sidney 2000 ...

Representação de **q**

Brasil	0.4
Olimpíadas	0.3
Sidney	0.3

Representação de **d**

Brasil	0.5
Olimpíadas	0.3
Sidney	0.2



Representação do Documento com Pesos

- Centróide
  - Pesos associados aos termos como indicação de relevância
    - Frequência de ocorrência do termo no documento
    - TF-IDF = *Term Frequency x Inverse Document Frequency*

Representação do Documento com Pesos

- TF-IDF também considera palavras com baixa ocorrência na base de documentos como melhores discriminantes

$$TFIDF(\omega) = TF(\omega) \cdot \log\left(\frac{|D|}{DF(\omega)}\right)$$

TF(w): frequência da palavra w no doc
D = total de documentos
DF(w): frequência de w em D

Representação do Documento com Pesos

- Centróide
  - Limitar tamanho do centróide em 50 mantendo apenas termos com maior peso
    - Aumenta a eficiência do sistema
    - Estudos mostram que isso não altera muito o poder de representação do centróide

Representação do Documento com Pesos

- Enriquecendo a representação
  - Considerar formatação do texto como indicação da importância dos termos
    - Título, início, negrito,...
  - Adicionar informação sobre a localização do termo no documento

Representação de documentos usada pelo Google

Doc :xxx
word : z - hit hit hit hit
word : y - hit hit hit ...
word : w - hit

hit: 1bit capitalization; 3bit font size; 12 bit position

Etapa 3: Indexação dos Documentos

- Objetivo
  - Facilitar busca dos documentos no repositório digital
- Opção imediata: varrer o texto completo
  - Busca sequencial *on-line*
  - Textos pequenos ou muito voláteis

Etapa 3: Indexação dos Documentos

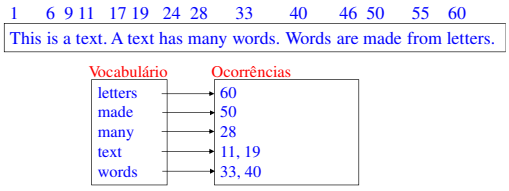
- Para bases maiores: indexar os documentos
  - Índices invertidos
  - Vetores e árvores de sufixos
  - Arquivos de assinatura

Índices Invertidos: Estrutura

- Composição: vocabulário (em ordem alfabética) e posição de ocorrência no texto
- Possibilita consulta/busca por proximidade e por termo composto
- Espaço requerido: pequeno para vocabulário, porém grande parte para ocorrências

Índices Invertidos: Estrutura

- Exemplo

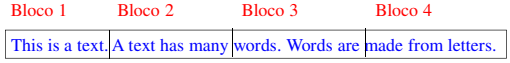


Índices Invertidos: Técnicas para redução de espaço

- *Stemming* (reduz vocabulário)
- Ocorrências dos termos endereçadas por
  - Blocos no texto
  - Endereço do documento inteiro
    - Engenhos de busca na Web
  - Poucos ponteiros, ponteiros menores e menos ocorrências

Índices Invertidos: Técnicas para redução de espaço

- Google: endereçamento "hierárquico" de blocos
  - Bloco + posição relativa da palavra dentro do bloco

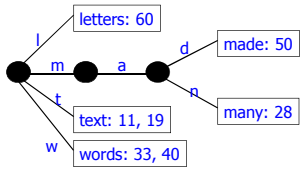


Busca em Índices Invertidos

- Algoritmos seguem 3 etapas
  - Busca as palavras da consulta no vocabulário: Hashing, tries, B-trees
  - Recupera as ocorrências de todas as palavras da consulta encontradas no vocabulário
  - Combina as ocorrências recuperadas de acordo com a consulta
    - Termos compostos
    - Proximidade
    - Operações booleanas

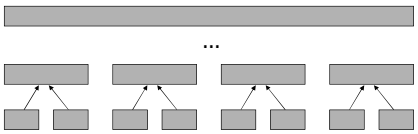
Índices Invertidos: Construção

- Baixo custo de busca
  - *O(número de caracteres)*
- Palavras inseridas em uma árvore do tipo *Trie*



Índices Invertidos: Construção

- *Tries:* muito espaço requerido
  - Para bases grandes, usa-se paginação
    - Índices parciais persistentes
    - *Merge* dos índices

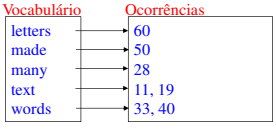


Índices Invertidos: Construção

- Ao final do processo, tem-se
  - Um arquivo de ocorrências dos termos
  - Outro arquivo do vocabulário com ponteiro para ocorrências
    - Pode ser mantido na memória

Índices Invertidos: Construção

- Relembrando o exemplo dado anteriormente:



Etapa 4: Recuperação

- Obtenção dos documentos que satisfazem uma consulta *(query)*
- Índices Invertidos
  - Procurar termos da consulta no vocabulário
  - Custo de busca e armazenamento é sublinear
    - O ($n^{0.85}$)

Etapa 4: Recuperação

- Tabelas *hash*, *tries*, ...
  - O(tamanho da palavra)
- Lista em ordem alfabética
  - O(log (tamanho do texto))
  - Mais barato em termo de espaço

Etapa 4: Recuperação

- Consultas simples
  - Recupera documentos nos quais a palavra ocorre pelo menos uma vez
- Consultas compostas (booleanas)
  - Recupera documentos nos quais cada palavra da consulta ocorre pelo menos uma vez
  - *Merge* de listas
    - Combina as listas de documentos recuperados de acordo com o operador booleano da consulta

String Matching

- Método usado em vários sistemas
  - Busca por palavras, comparação entre arquivos
- Encontrar todas as ocorrências de uma determinada *string* (padrão) em um texto
- Várias soluções existentes

String Matching Aproximado

- Dado um padrão P de tamanho m, um texto T de tamanho n, onde m,n>0, um inteiro k>0 e uma função de distância d, encontrar todas as substrings S de T tal que d(P,S)<=k
  - d - número de operações necessárias para transformar S em P
  - "Um texto qualquer"
  - "Eu testo.."
    "texto" (d=1)

String Matching Aproximado

- Várias Soluções existentes
  - Força bruta (BF)
  - Landau-Vishken
  - Boyer-Moore (BM)
  - Shift-Or (SO)
  - ...

String Matching Aproximado

- No Radix
  - Implementação e adaptação dos algoritmos (BF,BM,SO)
  - Número de erros permitidos baseia-se no tamanho do termo a ser comparado
    - "Mp3" – 1 erro é permitido
    - "Download" – 3 erros são permitidos

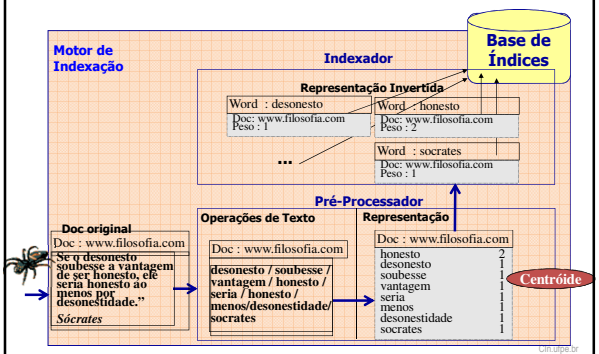
Etapa 5: Ordenação

- Ordenar os documentos recuperados de acordo com sua relevância em relação à consulta
- Relevância: difícil de medir
  - Mede-se a similaridade entre cada documento e a consulta
- Modelo "Espaço Vetorial"
  - Similaridade é proporcional ao cosseno do ângulo entre o vetor que representa o documento e o vetor da consulta
  - Tende a retornar documentos pequenos

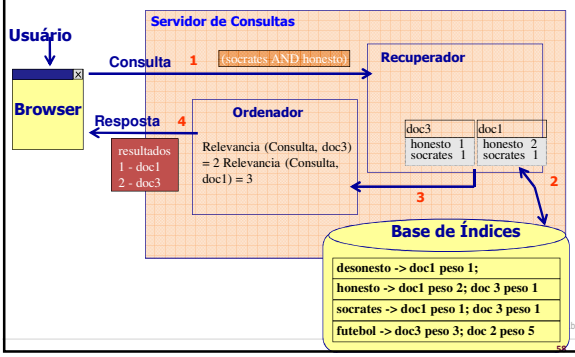
Etapa 5: Ordenação

- Google
  - Proximidade das palavras da consulta no documento
  - Tamanho da fonte, texto de *links*, ...
  - *PageRank*

Motor de Indexação



Servidor de consultas



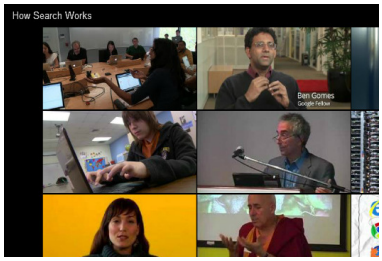
Dados do Google (2013)

- 100 bilhões de buscas na Web por mês
- 30 trilhões de páginas individuais (aumentou 30 vezes em 5 anos)
- Índice correspondente: 1000 Terabytes
- Busca feito em aproximadamente 1/8 de segundos
- Para verificar a qualidade dos resultados - buscadores humanos: 40 mil vezes por ano
- Spam: 40.000 a 60.000 notificações por mês aos donos dos sites

Dados do Google (2013)

- Ranking das respostas considera
  - Atualidade dos resultados
  - Qualidade do website
  - Confiança e adequação do conteúdo
  - Contexto do usuário: localização, buscas anteriores, conexões no Google+
- http://www.google.com/insidesearch/howsearchworks/thestory/

Como a busca do google funciona?



http://www.youtube.com/watch?v=BNHR6IQJGZs