Web Mining e Recuperação de Informação

Web Mining é a intersecção de várias áreas

- Sistemas de Bancos de Dados
- Ciência da Informação e Bibliotecas Digitais
- Inteligência Artificial
- Processamento de Linguagem natural
- Aprendizagem de Máquina

Sistemas de Bancos de Dados

- Dados *estruturados* armazenados em tabelas relacionais e não em texto livre.
- Eficiente processamento de queries bem formadas em uma query formal (SQL).
- Semântica clara para dados e queries.
- Recentes avanços para dados *semi-estruturados* (XML) torna-o mais próximo de RI (Recuperação de Informação).

Ciência da Informação e Bibliotecas Digitais

- Aspectos humanos de recuperação da informação (interação human-computador, visualização).
- Categorização do conhecimento humano.
- Análise de referências bibliográficas e *bibliometria*
- Trabalhos recentes sobre *bibliotecas digitais* torna-o próximo de RI.

Inteligência Artificial

- Representação de conhecimento, raciocínio, e busca de informação.
- Formalismos para representação de conhecimento e queries:
  - Lógica de Predicados
  - Redes Bayesianas
  - Others …
- Recentes trabalhos sobre ontologias e agentes inteligentes de informação torna-a próxima de RI.

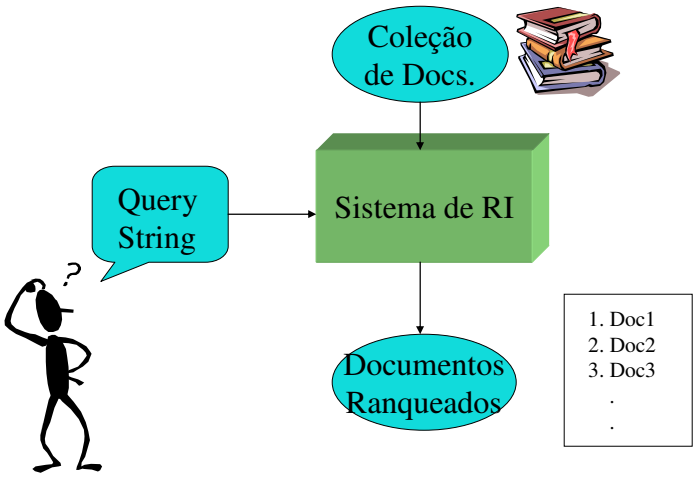
Processamento de Linguagem Natural

- Análise Sintática, semântica e pragmática de texto e discurso.
- Habilidade de analisar sintaxe (estrutura de frases) e semântica pode permitir a recuperação pelo "significado" e não por palavras-chave.
- Resposta a questões, extração de informações, etc.

Aprendizagem de Máquina

- Sistemas capazes de aprender novos conceitos e melhorar com a experiência.
- Aprendizagem supervisionada (classificação ou categorização automática)
- Aprendizagem não-supervisionada (*agrupamento*).

Sistema de RI (Recuperação de Informação)



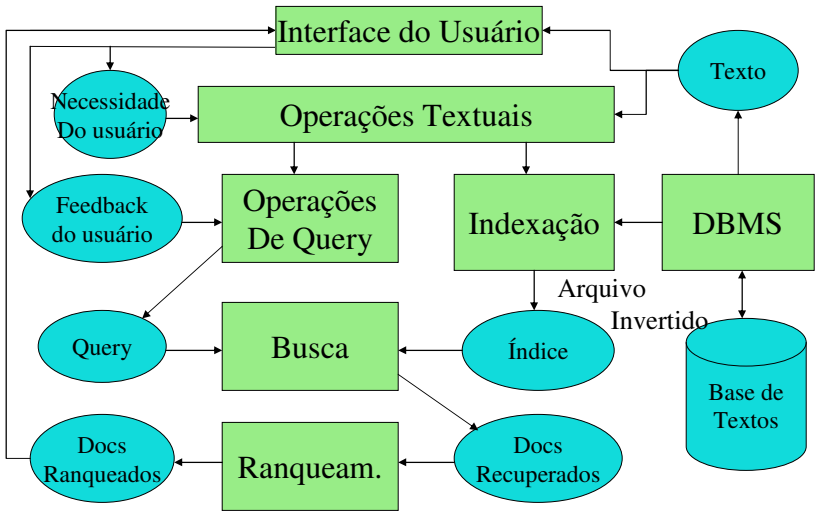
Paradigmas Opostos

- Busca em DBMS
  - Modelos complexos de dados: tabelas, colunas, linhas, tipos de dados
  - Expressiva e poderosa linguagem de query (SQL)
  - Necessário conhecer o esquema para elaborar queries
  - Resposta = conjunto desordenado de tuplas
  - Ranking = critério explícito
- RI (Recuperação de Informação)
  - Dados: conj. de documentos, documento= conj. de termos
  - Termos e Frases presentes ou não
  - Não, esquema geralmente desconhecido e não trivial
  - Resposta = seqüência de documentos
  - Ranking = Ponto central de RI

Paralelos

- SQL -> XML search
- Árvores, links de referência
- Links rotulados (relacionamentos)
- Nós podem conter:
  - dados estruturados;
  - campos de texto
- Query envolve nós de dados e rótulos dos links
  - Conhecimento parcial do esquema
- Resposta = conjunto de caminhos
- Web search <- RI
- Docs são nós de um grafo
- Hiperlinks têm importância mas não uma semântica
  - Ex.: Google
- Dados textuais
  - Dados são documentos
- Linguagem de consulta permanece primitiva
  - Sem tipos de dados
  - Sem árvores de tags
- Resposta = lista de urls

Arquitetura de um sistema de RI



Relevância dos Resultados

- Relevância é um julgamento subjetivo e pode incluir:
  - Relação com o assunto da query.
  - Característica temporal (informação recente).
  - Ser confiável (de origem confiável).
  - Satisfazer as metas do usuário (uso da informação).

Componentes de um Sistema de RI

- Operações Textuais: formar palavras para índices.
  - Standardização (caps …)
  - Remoção de Stopword
  - Stemming
- Indexação: constrói um *índice invertido* de palavras para ponteiros de documentos.
- Busca: recupera documentos que contém um termo da query usuando o índice invertido.
- Ranqueamento: ranqueia todos os documentos usando uma métrica relevante.

Componentes de um Sistema de RI

- Interface do usuário: gerencia a interação com o usuário:
  - Entrada de dados e saíde de documentos
  - Feedback de relevância.
  - Visualização de resultados.
- Operações de Query: modifica a query para melhorar a recuperação:
  - Expansão de query usando um thesaurus.
  - Transformação de query usando feedback de relevância.

RI e a Web

- A criação da Web fez crescer muito o interesse sobre RI:
  - Repositório universal de conhecimento;
  - Livre acesso;
  - Fonte distribuída de informação;
  - RI parece ser a solução para se encontrar informação na Web

RI e a Web

Em 2002: 5 Exabytes de novas informações;

- Meio magnético (HD's): 92%;
- Filmes: 7%;

5 Exabytes = 5 milhões de Terabytes = 5.000.000.000.000.000.000 bytes;

2 vezes a quantidade de 1999, dando um percentual de 30% de aumento por ano.

RI e a Web

Email:

- 31 bilhões de e-mails / ano = 400.000 Tbytes de nova informação;

A Internet (*Web*):

- 170 Tbytes de informação = 17 vezes o conteúdo impresso da *US Library of Congress*.

Fonte: http://www.itfacts.biz/

IR e a Web

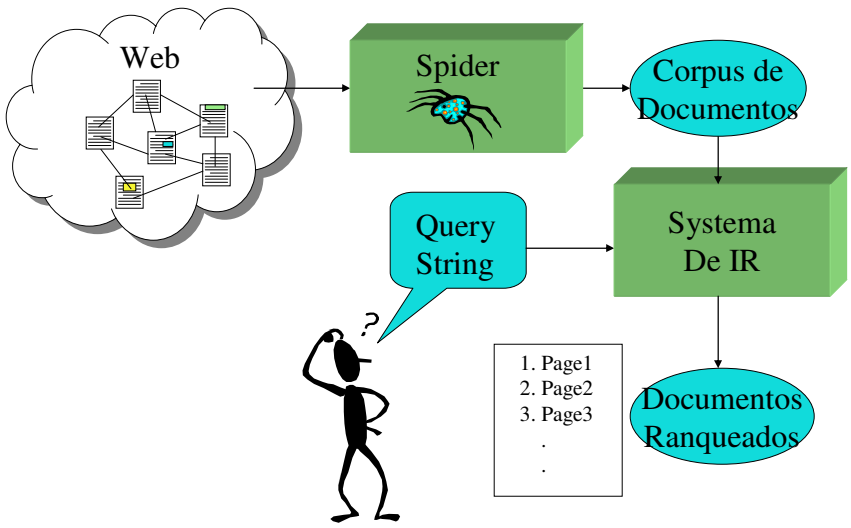
Sites de Busca:

- "Yahoo", "Google", etc. = a 1ª opção de acesso para usuários;
- Um usuário típico: 11 h 20 m / mês;
- Acesso à informação desejada = 1 / 3 do tempo;

Busca na Web

- Aplicação de RI à documentos HTML da Web.
- Diferenças:
  - Corpus de documentos é gerado com spiders.
  - Pode explorar o layout estrutural da página HTML (XML).
  - Documentos são modificados constantemente.
  - Pode explorar a estrutura de links da web.

Sistema de busca na Web



Outras tarefas relacionadas a RI na Web

- Categorização automática de documentos
- Filtragem de Informação (spams)
- Roteamento de Informação
- Agrupamento automatizado de documentos
- Sistemas de recuperação de produtos
- Extração de Informação
- Integração de Informação
- Resposta a queries

Query

- Qual peça de Shakespeare contém as palavras ***Brutus*** *AND* ***Caesar*** mas *NOT* ***Calpurnia***?
- Para responder esta questão pode-se listar todas as peças de Shakespeare contendo ***Brutus*** e ***Caesar*** e então eliminar aquelas que não contém ***Calpurnia***?
  - Isso é muito lento (para grandes bases)
  - *NOT* é difícil de fazer
  - Outras operações

Modelo Booleano

- Baseado na álgebra booleana
  - Documentos são conjuntos de termos
  - Queries são expressões booleanas
- Historicamente o modelo mais comum
  - Muitos mecanismos de busca também o utilizam

Exemplo

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

1 se peça contém palavra, 0 caso contrário

Modelo Booleano

- Query: “Quais peças de Shakespeare contém as palavras ***Brutus*** *AND* ***Caesar*** mas *NOT* ***Calpurnia***?
- Assim nós temos um vetor 0/1 de termos
- Para responder a query: selecione os vetores para “***Brutus, Caesar*** e not(***Calpurnia)***” (operação binária *AND)*
  - 110100 *AND* 110111 *AND* 101111 = 100100
  - Resultado: “Antony and Cleopatra” e “Hamlet”

Grandes bases de documentos

- Considere *n* = 1M de documentos, cada um com 1K (1000) termos.
- Na média 6 bytes/term incluindo pontuação, espaçamento
  - 6GB de dados.
- Existem aproximadamente *m* = 500K (500.000) termos distintos nesses dados.

Modelo Booleano

- Vantagens:
  - Fácil de entender.
  - Formalismo claro.
- Podem ser extendidos para incluir ranqueamento.
- Modelo eficiente.

Modelos Booleanos - Desvantagens

- Decisão binária vs resposta parcial
- Por default, nenhum ranqueamento é fornecido
- Difícil de representar queries complexas por parte dos usuários
- Geralmente retorna muitos documentos ou pouco documentos
- Não consegue capturar o feedback de relevância do usuário

É impossível construir essa matriz

- Uma matriz 500K x 1M tem meio trilhão de 0's e 1's
- Mas ele tem não mais do que um bilhão de 1's
  - Matriz é extremamente esparsa
- Qual é a melhor representação?

Índices invertidos

- Documentos são "parseados" para a extração de termos e estes são salvos com o ID do documento

Doc 1

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

Doc 2

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

Term	Doc #
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

- Depois que todos os documentos forem analisados os arquivos de índices são ordenados em função dos termos

Term	Doc #
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

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
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
you	2
was	1
was	2
with	2

- Múltiplas entradas para um termo são fusionadas e as frequências somadas

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
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
you	2
was	1
was	2
with	2

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
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
you	2	1
was	1	1
was	2	1
with	2	1

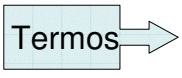
- O arquivo é normalmente dividido em um dicionário e em um arquivo *indexado*

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
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
you	2	1
was	1	1
was	2	1
with	2	1

Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
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
you	1	1
was	2	2
with	1	1

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
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
2	1
1	1
2	1
2	1

- Qual o problema com o armazenamento?



Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
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
you	1	1
was	2	2
with	1	1

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
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
2	1
1	1
2	1
2	1

Duas forças conflitantes

- Um termo como ***Calpurnia*** aparecerá em um documento dentre 1 milhão de docs – o armazenamento do ponteiro para este termos representará um espaço de $\log_2$ 1M ~ 20 bits.
- Um termo como ***the*** ocorre em praticamente todos os documentos, assim 20 bits/ponteiro é muito caro.
  - Prefira o vetor de 0/1 neste caso.

Questões sobre índices

- Como processar uma entrada?
- Quais termos indexar em um doc.?
  - Todas as palavras ou somente as "mais importantes"?
- Lista de Stopwords: termos que são comuns devem ser ignorados para a indexação
  - *ex.*, ***the, a, an, of, to*** …

Questões sobre o que indexar

Cooper's concordance of Wordsworth was published in 1911. The applications of full-text retrieval are legion: they include résumé scanning, litigation support and searching published journals on-line.

- ***Cooper's*** vs. ***Cooper*** vs. ***Coopers***.
- ***Full-text*** vs. ***full text*** vs. {***full, text***} vs. ***fulltext.***
- Acentuação: ***résumé*** vs. ***resume***.

Pontuação

- ***Ne'er***: é uma terminologia específica, é necessário uma normalização "manual".
- ***State-of-the-art***: quebre a seqüência com hífen.
- ***U.S.A.*** vs. ***USA*** – use valor default.
- ***a.out***

Números

- 3/12/91
- Mar. 12, 1991
- 55 B.C.
- B-52
- 100.2.86.144
  - Geralmente não indexada como texto
  - Criação de datas para documentos

Case Sensitive

- Reduza todas as letras para minúsculo
  - Exceção: letras maiúsculas dentro de sentenças
    - *ex.,* ***General Motors***
    - ***Fed*** vs. ***fed***
    - ***SAIL*** vs***. sail***

Thesaurus

- Manipular sinônimos e homônimos
  - Classes de equivalências construídas manualmente
    - ex., ***car*** = ***automobile***
    - ***your vs you're***
- Idexar tais equivalências, ou expandir query?

Correção ortográfica

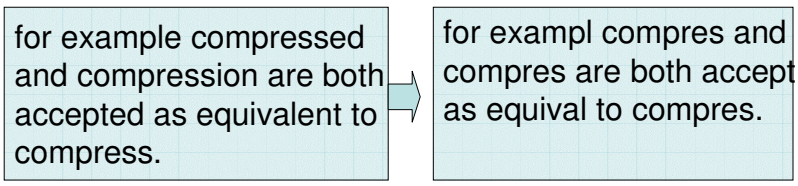
- Procure todos os termos que têm até 3 diferenças 3 (Inserção/Deleteção/substituição de caracteres) em uma query
  - *ex.,* ***Alanis Morisette***
- Correção ortográfica é cara e torna lenta a query (em até 100 vezes)
  - Utilise esta estratégia apenas quando o índice retorna zero matches.
  - Oq fazer se o documentos contém erros?

Lematização

- Reduz formas variantes para a forma básica
- Ex.,
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
- *the boy's cars are different colors* → *the boy car be different color*

Stemming

- Reduz os termos para suas "raízes" antes de indexar
  - Dependente de idioma
  - ex., ***automate(s), automatic, automation*** são reduzidos a ***automat***.



Algoritmo de Porter

- Algoritmo mais comum de stemming para English
- Convenções + 5 fases de reduções
  - Fases aplicadas sequencialmente
  - Cada fase consiste de um conjunto de comandos
  - Convenção simples: das regras em um comando composto, selecione aquela que se aplica ao maior sufixo.

Regras típicas de Porter

- *sses* $\rightarrow$ *ss*
- *ies* $\rightarrow$ *i*
- *ational* $\rightarrow$ *ate*
- *tional* $\rightarrow$ *tion*

Outros stemmers

- Outros stemmers existem, ex., stemmer de Lovins
  http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm
- Único passo, remoção do maior sufixo (em torno de 250 regras)
- Motivado pela Lingüistica e IR
- Análise morfológica completa - modestos benefícios para recuperação

Busca além de termos

- Como manipular frases?
- Proximidade: Encontre ***Gates*** *NEAR* ***Microsoft***.
  - Precisa indexar para capturar a posição da informação nos docs.
- Zonas em documentos: Encontre documentos com (*author* = ***Ullman***) *AND* (texto contém ***automata***).

Proximidade
Dentro do texto

Acúmulo de evidência

- Ocorrência do termo em 1 vs. 0
  - Ocorrência 2 vs. 1
  - Ocorrência 3 vs. 2, etc.
- Precisa da freqüência do termo nos docs

Ranqueando resultados de buscas

- Queries booleanas representam inclusão ou exclusão dos documentos.
- É necessário medir a proximidade da query para cada doc.
- Decidir se os docs representados ao usuário cobrem diferentes aspectos da query.

Dados Estruturados vs Não-estruturados

- Dados estruturados são armazenados normalmente em tabelas

Empregado	Gerente	Salário
Smith	Jones	50000
Chang	Smith	60000
Ivy	Smith	50000

Permite queries para intervalos de valores e matching exatos para texto,

Salário < 60000 AND Manager = Smith.

Dados Não-estruturados

- Normalmente está em texto livre
- Permite
  - Queries usando palavras chaves e operadores
  - Queries conceituais podem ser efetuadas,
    - Encontre todas páginas web relacionadas a "abuso de drogas"
- Modelo clássico de busca para documentos textuais

Dados Semi-estruturados

- De fato, quase nenhum dado é “não-estruturado”
- Ex., este slide tem zonas distintas como título e itens
- Isso facilita a busca em dados “semi-estruturados” como
  - *Título* contém dados AND *Itens* contém busca

Mecanismos avançados de busca semi-estruturada

- *Title* relacionado a Programação Orientada a Objetos AND *Autor* algo parecido com stro*rup
- onde * é o operador coringa
- Problemas:
  - Como processar “relacionado a”?
  - Como ranquear documentos?
- Falaremos mais adiante sobre XML

Agrupamento e classificação

- Dado um conjunto de docs, agrupe-os em clusters baseado em seu conteúdo.
- Dado um conjunto de tópicos mais um novo documento *D*, decida dentro de qual(is) tópicos colocar o documento *D*.

A web e seus desafios

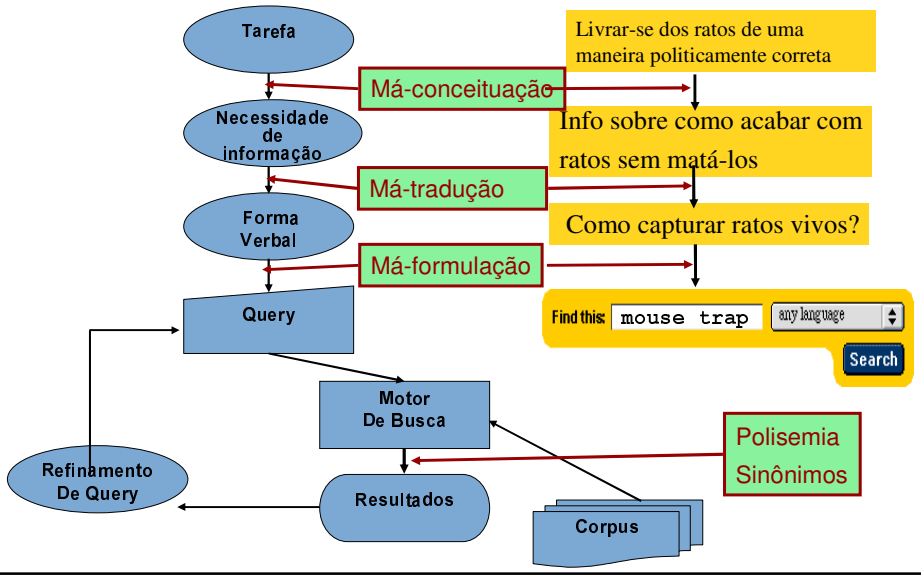
- Documentos com características particulares e diversas
- Satisfazer a necessidade de diferentes usuários, tipos de queries e necessidades de informação
- Além de termos, explora idéias de redes sociais
  - Análise de links, ranking de documentos ...

Avaliação de um sistema de Recuperação de Informação – Parte I

- Como podemos avaliar a qualidade de um sistema de RI?
  - Velocidade de indexação
  - Quantidade de índices, tamanho do corpus
  - Velocidade de processameno da query
  - **"Relevância" dos resultados**

- Nota: **necessidade de informação** é traduzida por uma **query**
- Relevância está relacionada à **necessidade de informação** *não* à **query**

O modelo de busca clássico



Problemas

- Má-conceituação = informação que vc acha necessária não é necessária para realizar a tarefa
- Má-tradução = verbalização não reflete a informação necessária
- Má-formulação = query atual não reflete a informação necessária
- Polisemia = uma palavra com múltiplos significados
- Sinônimos = mesmo conceito expresso de várias formas

Recursos para a aula de hoje

- *Managing Gigabytes*, Capítulo 3.
- *Modern Information Retrieval,* Capítulo 7.2
- Stemmer de Porter:
  - http://www.tartarus.org/~martin/PorterStemmer/
- http://trec.nist.gov