Fabrício Jailson Barth

Uma breve introdução ao tema Recuperação de Informação

São Paulo
2010

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

O tema Recuperação de Informação sempre foi um tema muito explorado na academia e no mercado. A forma com que os eventos acadêmicos são conduzidos demonstra uma maturidade muito grande da área, inclusive com uma ligação muito forte com o mercado.

Inúmeros livros sobre este tema já foram publicados, alguns clássicos são [1] e [3]. No entanto, são poucos os livros publicados em português e são poucos os eventos brasileiros que tratam sobre o assunto.

Durante o meu doutoramento [4], eu tive que realizar uma fundamentação teórica adequada sobre o tema, descrevendo os conceitos e premissas que regem o desenvolvimento da área de Recuperação de Informação, os modelos existentes e os métodos para avaliação dos Sistemas de Recuperação de Informação.

Levando-se em consideração o material escasso sobre o tema em português, eu decidi publicar este material na forma de um relatório. O objetivo deste relatório é fornecer uma introdução ao tema, apresentando os principais conceitos, modelos e métodos existentes.

Este relatório está estruturado da seguinte forma: (i) no capítulo 2 são apresentados as principais definições e conceitos da área; (ii) no capítulo 3 são descritos os principais modelos que regem o desenvolvimento dos Sistemas de Recuperação de Informação, e; (iii) no capítulo 4 são apresentados os métodos para avaliação de Sistemas de Recuperação de Informação.

2 Recuperação de Informação

O termo Recuperação de Informação (RI) foi cunhado por [5] que definiu da seguinte maneira: *"...Recuperação de Informação é o nome do processo onde um possível usuário de informação pode converter a sua necessidade de informação em uma lista real de citações de documentos armazenados que contenham informações úteis a ele..."*.

Para os usuários, o processo de Recuperação de Informação parte de uma necessidade de informação. O usuário fornece a um Sistema de Recuperação de Informação uma consulta formulada a partir da sua necessidade de informação. O sistema então compara a consulta com documentos armazenados. A tarefa do Sistema de Recuperação de Informação é retornar ao usuário os documentos que mais satisfazem a necessidade do usuário [6]. Para um Sistema de Recuperação de Informações, um processo de Recuperação de Informação inicia quando o usuário informa uma consulta ao sistema. Consultas são representações formais das necessidades de informação de um usuário. Em um Sistema de Recuperação de Informação uma consulta não é associada a um único documento em uma coleção. Ao contrário, diversos documentos são retornados através de uma consulta, selecionando-se os documentos que se apresentam como mais relevantes comparando a consulta com as representações dos documentos previamente armazenados [3].

Um Sistema de Recuperação de Informação possui três componentes básicos: aquisição e representação da necessidade de informação; identificação e representação do conteúdo do documento, e; a especificação da função de comparação que seleciona os documentos relevantes baseada nas representações [1].

A consulta em Recuperação de Informação é a forma que o usuário possui para representar a sua necessidade de informação em um Sistema de Recuperação de Informações. A necessidade de informação normalmente não é especificada em linguagem natural e sim através de palavras-chave. Após a elaboração da consulta, o Sistema de Recuperação de Informações tenta localizar objetos que possam ser relevantes para o usuário. Um Sistema de Recuperação de Informações não tem a responsabilidade de responder precisamente a uma consulta mas sim de identificar objetos que permitam

que o usuário satisfaça sua necessidade de informação [1, 3].

Geralmente, o texto não é armazenado por inteiro em um sistema de Recuperação de Informação. Para cada texto são criadas estruturas de dados, com o objetivo de acelerar o seu processo de recuperação [1]. Por exemplo, os textos que devem ser indexados são submetidos a um processo de filtragem de termos relevantes, denominada extração de atributos. Os atributos extraídos a partir deste processo serão utilizados para caracterizar os documentos armazenados [3].

Os Sistemas de Recuperação de Informação devem possuir uma função que compara a consulta fornecida pelo usuário com os textos armazenados no repositório. Esta função deve retornar o grau de relevância dos documentos para a consulta. Os documentos identificados com maior grau de relevância são mostrados primeiro para o usuário [1].

Além dos três componentes descritos acima, a maioria dos Sistemas de Recuperação de Informação incluem um quarto elemento que é chamado de realimentação de relevância. Pode-se pedir ao usuário para selecionar documentos, ou frações de documentos, considerados relevantes a partir de um conjunto recuperado. Este conjunto de documentos relevantes pode ser usado para modificar as representações da necessidade de informação ou a função de comparação para melhorar as respostas a consultas posteriores [7].

3 Modelos de Recuperação de Informação

Segundo [1], um modelo de recuperação de informação é uma quádrupla $\langle D, Q, F, R(q_i, d_j)\rangle$, onde:

- D é um conjunto de representações lógicas dos documentos em uma coleção.
- Q é um conjunto de representações lógicas (consultas) das necessidades de informação dos usuários.
- F é um arcabouço para a modelagem dos documentos, consultas e suas relações.
- $R(q_i, d_j)$ uma função que associa um número real com uma consulta $q_i \in Q$ e uma representação de documento $d_j \in D$. Esta função define uma ordenação entre os documentos com respeito a consulta q_i.

Ao desenvolver um modelo, as formas de representação dos documentos e das necessidades de informação do usuário são as primeiras entidades a serem definidas. Com base nestas entidades, o arcabouço do modelo pode ser definido. Este arcabouço fornece os princípios para a construção da função de ordenação. Por exemplo, os três tipos de modelos para Recuperação de Informação de uso mais difundido são [1]: modelo booleano, modelo vetorial e modelo probabilístico. No modelo booleano, o arcabouço é composto por conjuntos de documentos e operações clássicas da teoria de conjuntos. No modelo vetorial, documentos e consultas são representados como vetores em um espaço n-dimensional e o arcabouço é composto por um espaço n-dimensional e operações de álgebra linear aplicáveis aos vetores. No modelo probabilístico, o arcabouço que define as representações para documentos e consultas é baseado na teoria de probabilidades [1, 3].

3.1 Modelo booleano

O modelo booleano considera uma consulta como uma expressão booleana convencional, que liga seus termos através de conectivos lógicos *AND*, *OR* e *NOT* (figura 3.1). No modelo booleano um documento é considerado *relevante* ou *não relevante* a uma consulta, não existe resultado parcial e não há informação que permita a ordenação do resultado da consulta. Desta maneira, o modelo booleano é muito mais utilizado para recuperação de dados do que para recuperação de informação [1, 3].

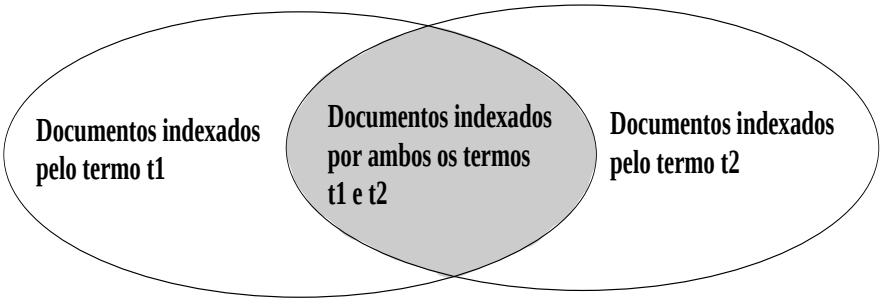

Figura 3.1: Representação do resultado de uma expressão booleana conjuntiva

As principais vantagens do modelo booleano: formalismo claro por trás do modelo e a sua simplicidade - facilmente programável e exato. As principais desvantagens do modelo booleano são: a saída pode ser nula ou possuir muitos documentos e a saída não é ordenada. Apesar das desvantagens serem grandes, o modelo booleano ainda é altamente utilizado em sistemas comerciais [1].

3.2 Modelo vetorial

No modelo vetorial os documentos e consultas são vistos como vetores num espaço vetorial *n*-dimensional, onde a distância vetorial é usada como medida de similaridade. Cada documento é representado como um vetor de termos e cada termo possui um valor associado que indica o grau de importância (peso) deste em um determinado documento [8] apud [1].

Em outras palavras, para o modelo vetorial, o peso $w_{i,j}$ associado com o par (k_i, d_j), palavra k_i e documento d_j, é positivo e não binário. Os termos de uma consulta também são associados a um peso. O vetor para uma consulta $\vec{q}$ é definido como $\vec{q} = (w_{1,q}, w_{2,q}, \cdots, w_{t,q})$, onde $w_{i,q} \geq 0$ e t é o número total de termos no sistema. Um vetor para um documento $\vec{d_j}$ é representado por $\vec{d_j} = (w_{1,j}, w_{2,j}, \cdots, w_{t,j})$ [1].

O modelo vetorial propõem o cálculo do grau de similaridade entre o documento d_j e a consulta q usando a correlação entre os vetores $\vec{d_j}$ e $\vec{q}$. Esta correlação pode ser

quantificada, por exemplo, usando uma medida de similaridade utilizando o cosseno do ângulo de dois vetores (*cosine similarity measure*), por exemplo:

$$sim(q,d_j) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{3.1}$$

Sendo $w_{i,j} \geq 0$ e $w_{i,q} \geq 0$, a similaridade $sim(d_j,q)$ varia de 0 até $+1$. Desta maneira, o modelo vetorial é capaz de ordenar os documentos de acordo com o grau de similaridade de cada documento com a consulta realizada pelo usuário. Um documento pode ser recuperado mesmo se ele satisfaz a consulta apenas parcialmente [1, 3].

Somente os sistemas mais ingênuos contabilizam igualmente todos os termos no vetor. A maioria dos sistemas ignoram palavras comuns como “a” e “de”, conhecidas como *stopwords*. Uma das formas de se calcular o peso de um termo em um documento, dada por [9], tenta balancear o número de ocorrências do termo no documento com o número de documentos onde o termo aparece (equação 3.2).

$$w_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \times \log \frac{N}{n_i} \tag{3.2}$$

onde,

- N = número de documentos da coleção.
- n_i = número de documentos onde a palavra i aparece.
- $\frac{f_{i,j}}{\max_z f_{z,j}}$ = freqüência normalizada da palavra i no documento j.

O peso é proporcional ao número de ocorrências do termo no documento e inversamente proporcional ao número de documentos onde o termo aparece. Trata-se de uma boa forma de contabilizar os termos, designando um peso maior a um termo se ele é um bom discriminante: se aparece numa quantidade pequena de documentos e não em muito deles [7, 3].

As vantagens do modelo vetorial são: ao atribuir pesos aos termos existe uma melhora do desempenho do Sistema de Recuperação de Informação; trata-se de uma estratégia de satisfação parcial da consulta, permitindo encontrar documentos que se aproximam das condições colocadas na consulta, e; os documentos são ordenados de acordo com o seu grau de similaridade [8] apud [1].

Teoricamente, o modelo vetorial assume que os termos encontrados nos documentos são independentes, fato que não é verdade. Entretanto, na prática, modelar as

dependências entre os termos em um documento tem gerado soluções com baixo desempenho em termos de tempo e espaço e não tem encontrado soluções melhores que implementações do modelo vetorial [1].

O modelo vetorial permite o desenvolvimento de soluções simples e rápidas. Por estas razões, o modelo vetorial é amplamente utilizado em soluções para indexação e pesquisa de documentos como, por exemplo, o software *Lucene*[1].

3.3 Modelo probabilístico

O modelo probabilístico, introduzido por [10], é um modelo onde a recuperação é vista como um problema de estimativa da probabilidade de que a representação de um documento corresponda ou satisfaça a representação de uma consulta. A idéia fundamental deste modelo é: dado uma consulta de um usuário, existe um conjunto de documentos que contem exatamente os documentos relevantes e nenhum não relevante [1].

Para determinar este conjunto de documentos relevantes é necessário conhecer algumas características que definam este conjunto. Como estas características não são conhecidas no tempo da consulta, é necessário iniciar um processo para determiná-las [1].

Dado um consulta q e um documento d_j em uma coleção, o modelo probabilístico tenta estimar a probabilidade de um usuário considerar relevante o documento d_j. Este modelo assume que esta probabilidade de relevância depende apenas da representação da consulta e da representação do documento [11].

O modelo assume que existe um subconjunto de documentos que o usuário prefere como resposta para a consulta q. Este conjunto de documentos ideais é representado por R e devem maximizar a probabilidade de relevância para o usuário. Documentos no conjunto R são rotulados como relevantes para a consulta q. Documentos que não estão neste conjunto são considerados não relevantes para q e são rotulados como $\overline{R}$, o complemento de R [11].

Os termos que ocorrem nos documentos em R podem ser utilizados para encontrar outros documentos relevantes. Este princípio é chamado de Princípio da Ordenação Probabilística é assume que a distribuição de termos na coleção seja capaz de informar a relevância provável de um documento para uma consulta [1].

O peso dos termos em um modelo probabilístico são todos binários, por exem-

[1] http://lucene.apache.org/java/2_3_0/scoring.html

plo, $w_{i,j} \in \{0,1\}$[2] e $w_{i,q} \in \{0,1\}$[3]. Uma consulta q é um subconjunto do índice de termos. Dado R como o conjunto de documentos relevantes ou, inicialmente, considerados como relevantes. Dado $\overline{R}$ como o complemento de R, ou seja, o conjunto de documentos não relevantes. Dado $P(R|\vec{d}_j)$ como sendo a probabilidade do documento d_j ser relevante para a consulta q e $P(\overline{R}|\vec{d}_j)$ como sendo a probabilidade do documento d_j não ser relevante para a consulta q. A similaridade $sim(d_j,q)$ de um documento d_j para uma consulta q é definida como:

$$sim(d_j,q) = \frac{P(R|\vec{d}_j)}{P(\overline{R}|\vec{d}_j)} \tag{3.3}$$

Aplicando a regra de Bayes [12] tem-se:

$$sim(d_j,q) = \frac{P(\vec{d}_j|R) \times P(R)}{P(\vec{d}_j|\overline{R}) \times P(\overline{R})} \tag{3.4}$$

onde,

- $P(\vec{d}_j|R)$, probabilidade de selecionar aleatóriamente o documento d_j do conjunto R de documentos relevantes.
- $P(R)$, probabilidade de um documento selecionado randomicamente da coleção ser relevante.
- $P(\vec{d}_j|\overline{R})$ e $P(\overline{R})$ são análogas e complementares as probabilidades apresentadas acima.

Sendo $P(R)$ e $P(\overline{R})$ os mesmos valores para todos os documentos da coleção, pode-se:

$$sim(d_j,q) \sim \frac{P(\vec{d}_j|R)}{P(\vec{d}_j|\overline{R})} \tag{3.5}$$

Assumindo a independência entre termos, tem-se:

$$sim(d_j,q) \sim \prod_{i=1}^{M} \frac{P(d_{j,i}|R)}{P(d_{j,i}|\overline{R})} \tag{3.6}$$

onde M é igual ao número de palavras distintas que ocorrem na coleção de documentos.

[2] $w_{i,j}$ representa o peso do termo i no documento j.
[3] $w_{i,q}$ representa o peso do termo i na consulta q.

A principal vantagem do modelo probabilístico é, teoricamente, o fato dos documentos serem ordenados de forma decrescente por suas probabilidades de serem relevantes. Algumas evidências parecem indicar que este modelo tem um desempenho melhor do que o modelo vetorial [13, 1].

As principais desvantagens deste modelo são: (i) separação aleatória da coleção em dois subconjuntos: documentos relevantes e não relevantes; (ii) o modelo não faz uso da freqüência dos termos no documento, e; (iii) assume a independência entre termos. De qualquer maneira, como discutido na seção sobre o modelo vetorial, ainda não está claro se a premissa de independência entre termos é ruim em situações práticas [13, 1].

Um exemplo concreto, e de sucesso, da aplicação do modelo probabilístico é a função de ordenação BM25 [1, 3]. Esta função de ordenação faz uso da freqüência dos termos no documento e também leva em consideração o tamanho dos documentos (equação 3.7).

$$sim(q,d_j) = \sum_{t \in q} \log \frac{N}{n_i} \times \frac{(k_1+1) \times f_{t,d}}{k_1((1-b)+b \times (T_d/T_{med})) + f_{t,d}} \tag{3.7}$$

onde:

- $f_{t,d}$ significa a frequência do termo t no documento d.
- T_d significa o tamanho do documento d.
- T_{med} significa o tamanho médio dos documentos na coleção.
- k_1 é um parâmetro de valor positivo que calibra a escala do $f_{t,d}$. Se k_1 igual a 0 então o retorno de $sim(q,d_j)$ é similar ao resultado do modelo booleano.
- b é um parâmetro ($0 \leq b \leq 1$) que determina a influência do tamanho dos documentos no cálculo da $sim(q,d_j)$. Quando $b = 0$ o valor de $sim(q,d_j)$ não é normalizado considerando T_d e T_{med}.

3.4 Modelo baseado em *hyperlinks*

Em algoritmos baseados em *hyperlinks*, o cálculo da relevância de um documento leva em consideração os *hyperlinks* entre os documentos de uma coleção [3, 2, 14]. Um *hyperlink* é uma estrutura comum em documentos no formato HTML. Esta estrutura possui duas propriedades importantes [3]:

- Um *hyperlink* é um sinal de qualidade: se existe um *hyperlink* no documento d_1 apontando para o documento d_2 então isto significa que o autor do documento d_1 percebeu que existe alguma relevância no documento d_2.
- O texto do *hyperlink* descreve o objeto do documento: o conteúdo que está entre as *tags* do *hyperlink* descrevem de forma sumarizada o conteúdo do documento referenciado, no caso, o documento d_2.

Estas propriedades, principalmente a primeira, vem sendo exploradas ao longo de alguns anos na tentativa de aumentar a eficiência de Sistemas de Recuperação de Informação, especialmente, os sistemas voltados para a WEB. As implementações deste modelo com maior destaque são: HITS [2] e PAGERANK [14].

3.4.1 HITS

[2] propôs um modelo baseado em *hyperlinks* que permite inferir a autoridade de um documento dentro de um conjunto de documentos. Esse modelo é baseado na relação entre páginas que são autoridades sobre um tópico e páginas que interligam essas autoridades (*hubs*). O algoritmo HITS (*Hyperlink Induced Topic Search*) assume duas premissas:

- Se o documento d_1 possui um *hyperlink* para o documento d_2 então o autor do documento d_1 considera que o documento d_2 contêm informações valiosas.
- Se o documento d_1 está apontando para um grande número de documentos de qualidade então a opinião do documento d_1 torna-se mais valiosa e o fato do documento d_1 apontar para o documento d_2 pode sugerir que o documento d_2 também é um bom documento.

O funcionamento do algoritmo HITS inicia a partir de um conjunto S de documentos, retornados por uma função de ordenação qualquer que leva em consideração apenas os termos da consulta fornecida pelo usuário. Este conjunto inicial S é denominado de conjunto raiz. Este conjunto é expandido para um conjunto raiz maior, denominado T constituído pela adição de qualquer documento que referencia ou é referenciado por qualquer documento do conjunto S. Uma ilustração dos conjuntos S e T é apresentada na figura 3.2.

Depois de formados os conjuntos S e T, o algoritmo HITS então associa para cada documento d_x um peso de hub $h(d_x)$ e um peso de autoridade $a(d_x)$. O algoritmo

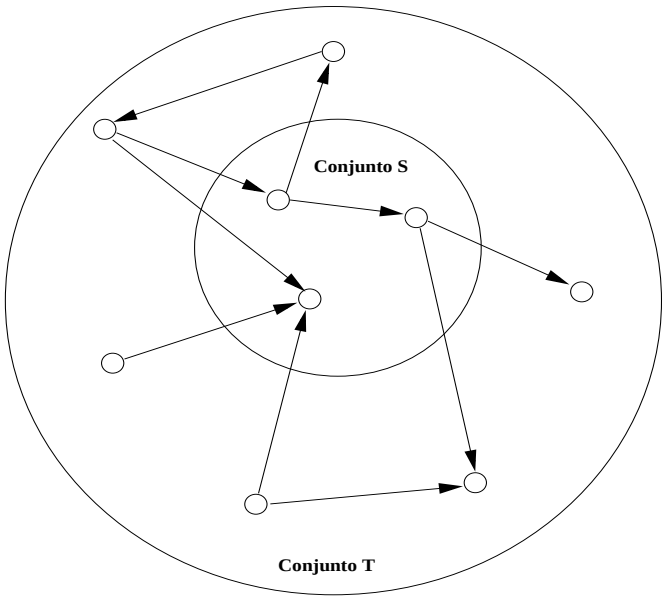

Figura 3.2: Conjuntos S e T utilizados pelo algoritmo HITS

atualiza de maneira iterativa o peso de hub e autoridade de cada documento, utilizando as equações abaixo:

$$a(d_i) = \sum_{d_j \to d_i} h(d_j) \tag{3.8}$$

$$h(d_i) = \sum_{d_i \to d_j} a(d_j) \tag{3.9}$$

onde, a representação $d_i \to d_j$ denota que o documento d_i possui um *hyperlink* para o documento d_j.

O algoritmo iterativo para cálculo dos valores de hub e autoridade para todos os documentos é apresentado na tabela 3.1. Este algoritmo pode ser utilizado para filtrar os n documentos com maior autoridade ou os n documentos com maior hub.

3.4.2 PAGERANK

O algoritmo PAGERANK [14] calcula a importância de um documento dentro de uma coleção através da análise das citações entre os documentos da coleção. Estas citações podem ser referências bibliográficas de trabalhos científicos ou *hyperlinks* em documentos HTML, por exemplo.

A fórmula para o cálculo do PAGERANK de um documento é dada pela equação abaixo:

Iterate(G,k)
G: coleção de n documentos
k: um número natural
$z \leftarrow$ um vetor $(1,1,\cdots,1) \in \Re^n$
$a_0 \leftarrow z$
$h_0 \leftarrow z$
for $i = 1,2,\cdots,k$ **do**
  Aplica a equação 3.8 em (a_{i-1}, h_{i-1}) para obter os valores de a'_i
  Aplica a equação 3.9 em (a'_i, h_{i-1}) para obter os valores de h'_i
  Normaliza os valores de a'_i, obtendo a_i
  Normaliza os valores de h'_i, obtendo h_i
end for
Return (a_k, h_k)

Tabela 3.1: Algoritmo para cálculo dos hubs e autoridades [2]

$$PR(A) = (1-d) + d(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}) \tag{3.10}$$

onde:

- A é um documento.
- $T_{1,2,\cdots,n}$ são documentos que contêm links para A.
- $PR(A)$ é o PAGERANK do documento A.
- $C(T)$ é o número de links de T para outros documentos.
- d é a probabilidade do navegador (pessoa) sair da página. Segundo [14], o valor normalmente adotado para d é 0.85.

Os valores de $PR(A)$ para todos os documentos da coleção podem ser calculados usando um algoritmo iterativo simples [14], parecido com o apresentado na figura 3.1.

Diferente do HITS, o algoritmo PAGERANK calcula o $PR(A)$ para todos os documentos da coleção antes que qualquer consulta aconteça. Quando um usuário realiza uma consulta, o Sistema de Recuperação de Informação pode fazer uso de qualquer função de ordenação para recuperar um conjunto de documentos que depois será reordenado de acordo com o valor do PAGERANK de cada documento.

3.5 Breve comparação entre os modelos

Em geral, o modelo booleano é considerado o modelo clássico mais fraco. O principal problema do modelo booleano é a incapacidade de reconhecer relevâncias parciais,

documentos que satisfazem a necessidade do usuário de forma parcial. Isto acaba fazendo com que os sistemas que implementam o modelo booleano tenham um baixo desempenho [1].

Existe diversos trabalhos na literatura comparando os modelos vetorial e probabilístico [15, 9]. Alguns destes trabalhos sugerem que o modelo probabilístico supera o modelo vetorial [15] e outros sugerem que o modelo vetorial supera o modelo probabilístico em coleções de documentos genéricos [9]. Aparentemente, o modelo vetorial é mais popular em implementações de Sistemas de Recuperação de Informações. No entanto, não existe nenhum trabalho que comprove que implementações utilizando o modelo vetorial tenham um desempenho melhor que implementações que usem o modelo probabilístico [3].

Os modelos booleano, vetorial e probabilístico estão, quase que por inteiro, no nível de palavras [7]. Como conseqüência, diversas pesquisas tentaram responder à seguinte pergunta: a Recuperação de Informação não melhoraria se fossem utilizadas técnicas mais sofisticas para representar os documentos e as necessidades de informação dos usuários? Muitos têm trilhado esta via, porém segundo [16]: "surpreendentemente nenhuma das tentativas apresentou uma melhora significativa numa faixa ampla de tarefas relacionadas com a Recuperação de Informação".

Os algoritmos do modelo baseado em *hyperlinks* saem do nível de palavras ao fazer uso de *hyperlinks* e citações. Os algoritmos deste modelo, principalmente o PAGERANK, conseguem um desempenho melhor na WEB porque manipulam um conjunto adicional de informações relevantes para o ambiente, no caso: a conexão entre documentos através de *links*.

Todos os modelos apresentados nesta seção foram construídos de maneira experimental, através do uso de coleções de referência. Alguns destes algoritmos, quando aplicados em ambientes reais, têm a sua eficiência avaliada e comprovada ao longo do tempo através de avaliações subjetivas dos usuários.

4 Método para avaliação de Sistemas de Recuperação de Informação

As medidas mais comuns para avaliar o desempenho de um sistema computacional são tempo e espaço. Quanto menor o tempo de resposta de um sistema e quanto menor o espaço em memória utilizado, melhor o sistema é considerado. No entanto, para sistemas onde o objetivo é recuperar informações outras métricas devem ser utilizadas [1].

Para a consulta realizada pelo usuário não existe uma resposta exata. Os documentos recuperados são ordenados de acordo com a sua relevância em relação a consulta. As métricas utilizadas para avaliar um Sistema de Recuperação de Informação devem medir quão relevante é o conjunto de documentos, recuperados pelo sistema, para o usuário [1].

Para comparar a efetividade de um Sistema de Recuperação de Informação são usadas coleções de teste padronizadas. Uma coleção deste tipo consiste dos seguintes elementos [7, 1, 3]:

- *Um conjunto de documentos*, que pode conter somente alguns dados como título, autor e resumo ou então o texto completo. Podem ser utilizadas informações adicionais, tais como um conjunto de termos usado como vocabulário de controle, descritores designados por autor e informação sobre citações.

- *Um conjunto de consultas*, exemplos de requisição de informações, constituídas por consultas reais submetidas por usuários, seja usando linguagem natural ou alguma linguagem formal de consulta. Ocasionalmente consultas artificialmente construídas são utilizadas: consultas construídas para recuperar documentos conhecidos ou o texto de um documento usado como amostra, por exemplo.

- *Um conjunto de julgamentos de relevância*: para cada consulta existente no conjunto de consultas são fornecidos documentos, pertencentes a coleção de docu-

mentos, considerados relevantes pelos usuários que submetem a consulta ou por especialistas do domínio. Para coleções pequenas, isto pode ser obtido revisando todos os documentos. Para coleções grandes, geralmente são combinados os resultados de diferentes representações da consulta construída por diferentes usuários. Via de regra, são preferidas as consultas e julgamentos obtidos diretamente dos usuários. Na tabela 4.1 é possível visualizar um exemplo de julgamentos de relevância.

Tabela 4.1: Exemplo de julgamentos de relevância

Identificador do tópico	Documento	Relevância
1	CSIRO135-03599247	2
1	CSIRO141-07897607	1
...	...	...
50	CSIRO265-01044334	0
50	CSIRO265-01351359	1
50	CSIRO266-04184084	0

O conceito de relevância está associado a necessidade de informação, não está associada a consulta. Um documento deve ser considerado relevante se e somente se suprir a necessidade de informação. Um documento não pode ser considerado relevante se todas as palavras que aparecem na consulta aparecem no documento também, por exemplo [3].

Dado um algoritmo de recuperação de informação, as medidas de avaliação devem quantificar a similaridade entre o conjunto de documentos recuperados e o conjunto de documentos considerados relevantes pelos especialistas. Isto fornece uma estimativa da qualidade do algoritmo de recuperação de informação avaliado. As medidas utilizadas para a avaliação dos algoritmos são uma forma de quantificar algo inerentemente subjetivo [7, 1, 3].

4.1 Medidas

Existem diversas medidas que podem ser utilizadas para avaliar o desempenho de um Sistema de Recuperação de Informação. As medidas mais utilizadas são precisão[1] e cobertura[2] [1].

Precisão é a proporção dos documentos recuperados que são relevantes para uma dada consulta em relação ao total de documentos recuperados. Cobertura é a razão

[1] Do inglês, *precision*
[2] Do inglês, *recall*

entre o número de documentos recuperados que são relevantes para uma consulta e o total dos documentos na coleção que são relevantes para a consulta. Supondo que o conjunto de documentos relevantes para uma consulta (relevantes) e o conjunto de documentos recuperados por uma consulta (recuperados) são conhecidos, pode-se definir estas medidas como [1]:

$$precisao = \frac{|relevantes \cap recuperados|}{|recuperados|} \tag{4.1}$$

$$cobertura = \frac{|relevantes \cap recuperados|}{|relevantes|} \tag{4.2}$$

A figura 4.1 ilustra estes conceitos.

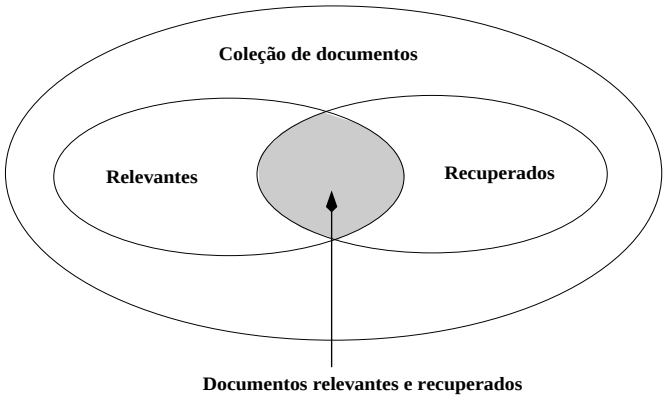

Figura 4.1: Precisão e cobertura [1]

Um Sistema de Recuperação de Informação pode equilibrar precisão e cobertura. No caso extremo, um sistema que retorna todos os documentos da coleção de documentos como seu conjunto de resultados tem a garantia de uma cobertura igual a 100%, mas terá baixa precisão. Por outro lado, um sistema pode retornar um único documento e ter um baixo índice de cobertura, mas teria uma chance razoável de 100% de precisão [16]. Uma forma de definir uma média harmônica entre precisão e cobertura é através da medida F[3] (equação 4.3) [3].

$$F = \frac{2 \times (precisao \times cobertura)}{precisao + cobertura} \tag{4.3}$$

As medidas de cobertura e precisão foram definidas quando as pesquisas de Recuperação de Informação eram feitas principalmente por bibliotecários interessados em resultados completos. Hoje, a maioria das consultas (centenas de milhões por dia) são feitas por usuários que estão menos interessados em perfeição e mais interessados em encontrar uma resposta imediata, uma resposta que apareça no topo da lista de

[3]Do inglês, *F-measure*

resultados [16].

Precisão, cobertura e medida F são medidas baseadas em conjuntos, computadas usando um conjunto de documentos não ordenados. Estas medidas não são suficientes para medir o desempenho da maioria dos atuais Sistemas de Recuperação de Informações, que fornecem um resultado ordenado segundo algum critério [3].

As medidas de avaliação utilizadas para medir o desempenho de Sistemas de Recuperação de Informações, que fornecem um resultado ordenado, são: precisão em n (*P@n*), *Mean Average Precision* (MAP) e *Normalized discount cumulative gain* (NDCG). As definições e a forma de cálculo destas medidas são apresentadas nas próximas seções.

4.1.1 Precisão em n (*P@n*)

A *P@n* mede a relevância dos *n* primeiros documentos de uma lista ordenada:

$$P@n = \frac{r}{n} \tag{4.4}$$

onde, *n* é o número de documentos retornados e *r* é o número de documentos considerados relevantes e retornados até a posição *n* da lista ordenada. Por exemplo, se os 10 primeiros documentos retornados por uma consulta são {*relevante, irrelevante, irrelevante, relevante, relevante, relevante, irrelevante, irrelevante, relevante, relevante*} então os valores de *P*@1 até *P*@10 são {*1, 1/2, 1/3, 2/4, 3/5, 4/6, 4/7, 4/8, 5/9, 6/10*}, respectivamente. Para um conjunto de consultas deve-se calcular a média de *P@n*.

4.1.2 Mean Average Precision (MAP)

MAP é uma medida que tenta sumarizar todos os valores *P@n*. *Average Precision (AP)* é definido como uma média para todos os valores de *P@n* para todos os documentos relevantes:

$$AP = \frac{\sum_{n=1}^{N} P@n \times rel(n)}{r_q} \tag{4.5}$$

onde: r_q é o número total de documentos considerados relevantes para a consulta; N é o número de documentos recuperados, e; $rel(n)$ é uma função binária sobre a relevância do n^{th} documento:

$$rel(n) = \begin{cases} 1 & \text{se o } n^{th} \text{ documento é relevante} \\ 0 & \text{caso contrário} \end{cases} \tag{4.6}$$

Assim sendo, o valor de MAP é a média dos valores AP para todas as consultas.

4.1.3 Mean Reciprocal Rank (MRR)

MRR é uma medida utilizada somente nos casos onde existe uma única resposta correta. O valor de MRR é definido da seguinte maneira:

$$MRR = \frac{\sum_{i=1}^{N} \frac{1}{p_i}}{N} \tag{4.7}$$

onde, N é o número de consultas e p_i é a posição onde o documento correto, da consulta i, é encontrado na lista ordenada, retornada pelo algoritmo avaliado. Por exemplo, o valor de MRR para os dados da tabela 4.2 é igual a:

$$MRR = \frac{\frac{1}{2} + 1 + \frac{1}{2}}{3} = \frac{2}{3} = 0.66$$

Tabela 4.2: Exemplo utilizado para ilustrar a medida MRR

Consultas	Resultados	Resposta correta	*Rank*	*Reciprocal Rank*
q_1	doc_1, doc_{10}	doc_{10}	2	$1/2$
q_2	doc_3, doc_4	doc_3	1	1
q_3	doc_6, doc_7, doc_3	doc_7	2	$1/2$

4.1.4 Normalized discount cumulative gain (NDCG)

O NDCG foi desenvolvido para manipular múltiplos níveis de relevância. Neste caso, os julgamentos de relevância sobre os documentos são fornecidos através de uma escala, por exemplo: 3 (muito relevante), 2 (relevante), 1 (pouco relevante) e 0 (não relevante).

Nos atuais ambientes para recuperação de informação, a quantidade de documentos considerados relevantes para uma consulta pode exceder a capacidade do usuário em tratar esta informação. Por isso, é desejável que o Sistema de Recuperação de Informação retorne os documentos com maior relevância, entre os documentos relevantes.

As medidas *P@n* e MAP apenas conseguem tratar situações onde o nível de re-

levância é binário (relevante ou não relevante). Usando as medidas *P@n* e MAP as diferenças entre métodos para recuperação de informação ótimos e bons pode não aparecer. O objetivo do NDCG é fazer com que esta diferença se torne visível na avaliação dos métodos.

O ganho acumulado (CG) na posição i é calculando somando-se a relevância dos documentos encontrados a partir da posição 1 até a posição i na lista ordenada. Formalmente, o CG na posição i é definido recursivamente da seguinte maneira:

$$CG[i] = \begin{cases} G[i] & \text{se } i = 1 \\ CG[i-1] + G[i] & \text{caso contrário} \end{cases} \tag{4.8}$$

onde, $G[i]$ é a relevância do documento encontrado na posição i da lista ordenada. Por exemplo, considere a seguinte lista de relevância dos documentos retornados $G' = \{3,2,3,0,0,1,2,2,3,0,\cdots\}$.

Para a lista G' é retornado a seguinte lista de $CG' = \{3,5,8,8,8,9,11,13,16,16,\cdots\}$. O ganho acumulado de qualquer posição na lista de ordenação pode ser lido diretamente: $CG[7] = 11$, por exemplo.

A idéia por trás da medida *DCG* é baseada em duas regras:

- Documentos extremamente relevantes são mais importantes (valiosos) que documentos com relevância marginal.
- Quanto mais baixa a posição do documento na lista ordenada (*ranking*), menor o valor deste documento para o usuário.

Esta idéia é implementada utilizando a equação abaixo [17]:

$$DCG[i] = \begin{cases} G[i] & \text{se } i = 1 \\ DCG[i-1] + \frac{G[i]}{\log i} & \text{caso contrário} \end{cases} \tag{4.9}$$

ou seja, quanto maior o valor de i maior será o desconto dado. Para a lista G' é retornado a seguinte lista $DCG' = \{3,5,6.89,6.89,6.89,7.28,7.99,8.66,9.61,9.61,\cdots\}$.

É possível ter uma idéia da qualidade do resultado encontrado em *DCG* comparando o resultado encontrado com uma lista ordenada de melhor resultado teórico possível (*MR*). A lista *MR* é definida de acordo com os índices de relevância utilizados na coleção de referência. Se na coleção de referência são utilizados os valores 3, 2, 1 e 0 para indicar a relevância de cada documento para cada consulta então uma definição possível para a lista ordenada de melhor resultado teórico possível é [18]:

$$MR[i] = \begin{cases} 3 & \text{se } i \leq m \\ 2 & \text{se } m < i \leq m+l \\ 1 & \text{se } m+l < i \leq m+l+k \\ 0 & \text{caso contrário} \end{cases} \tag{4.10}$$

onde, m, l e k representam o número de documentos encontrados na coleção de referência com índice de relevância 3, 2 e 1, respectivamente. Um exemplo de lista ordenada do melhor resultado teórico possível é $MR' = \{3,3,3,2,2,2,2,1,1,0,\cdots\}$.

Os valores para m, k e l são extraídos a partir dos dados da coleção de referência. As listas G', I', CG' e DCG' são definidas para cada consulta. Para comparar algoritmos são calculadas as médias de G', I', CG' e DCG' para todas as consultas.

Na figura 4.2 é possível visualizar uma curva com os valores apresentados em DCG' e a sua relação com o que seria o melhor resultado teórico possível (MR).

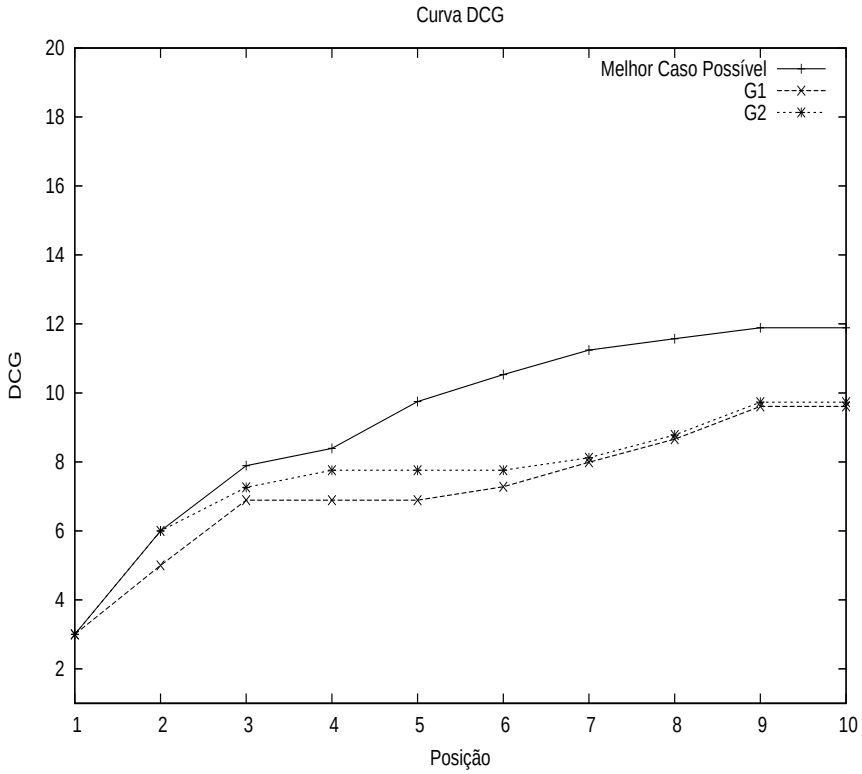

Figura 4.2: Exemplo de curva *DCG*

Os valores de *DCG* para cada função de ordenação podem ser normalizados dividindo-se os valores de *DCG* pelo valor correspondente no vetor *DCG* ideal. Desta forma, para qualquer posição em *DCG*, o valor normalizado 1 representa a situação ótima.

Dado um vetor *DCG* ($V = \{v_1, v_2, \cdots, v_k\}$) de uma função de ordenação qualquer e um vetor *DCG* com desempenho ideal ($I = \{i_1, i_2, \cdots, i_k\}$), o vetor *NDCG* de V é obtido através da seguinte equação [18]:

$$NDCG(V) = \{v_1/i_1, v_2/i_2, \cdots, v_k/i_k\} \tag{4.11}$$

Na figura 4.3 é possível visualizar um exemplo de curva *NDCG* onde são utilizados os mesmos dados que foram apresentados na figura 4.2.

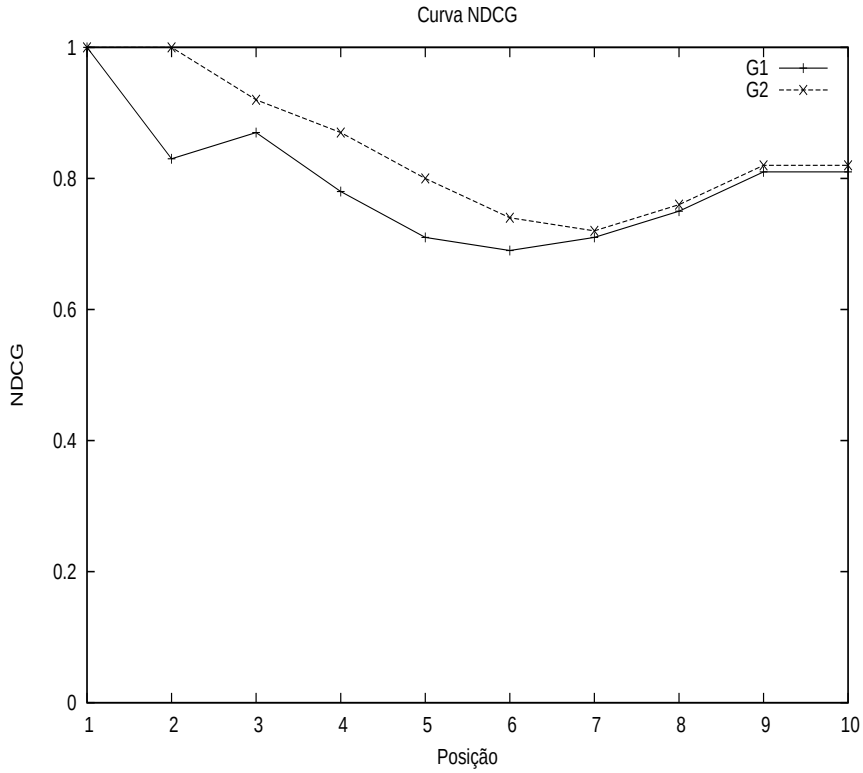

Figura 4.3: Exemplo de curva *NDCG*

4.2 Coleções de Referência

Coleções de referência são caras pois requerem julgamentos de relevância para sua criação. Tais julgamentos consomem muita mão-de-obra para um número significativo de consultas. Estudos deste tipo começaram em meados da década de 50 com a coleção *Cranfield* e continuaram nos anos 90 com a coleção TREC[4] (de *Text Retrieval Conference*, cuja primeira versão é de 1992) [1, 3].

A coleção *Cranfield* foi a primeira coleção que permitiu quantificar de maneira precisa o desempenho de um algoritmo de recuperação de informação. A criação desta coleção iniciou em 1950 e contêm 1398 resumos de artigos sobre Aerodinâmica, um conjunto de 225 consultas e um conjunto de julgamentos de relevância para todas as consultas. Atualmente, esta coleção é considerada muito pequena para qualquer experimento [3].

[4] http://trec.nist.gov/

Atualmente, o NIST (*The U.S. National Institute of Standards and Technology*) é o responsável por manter um ambiente para testes, incluindo diversas seções (*tracks*). A principal seção é o TREC *Ad Hoc track*, utilizada durante os primeiros 8 anos de existência do TREC, 1992 até 1999. Durante os anos de existência da conferência TREC foram criadas inúmeras seções: blog (*Blog Track*), filtragem de informações (*Filtering Track*) e spam (*Spam Track*), por exemplo. Cada seção tem o seu processo de criação dos julgamentos de relevância e da coleção de documentos [3].

Referências Bibliográficas

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern Information Retrieval. ACM Press, 1999.

[2] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5):604–632, 1999.

[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. An Introduction to Information Retrieval. Cambridge University Press, 2008.

[4] Fabrício J. Barth. Recuperação de documentos e pessoas em ambientes empresariais através de árvores de decisão. PhD thesis, Escola Politécnica da Universidade de São Paulo, 2009.

[5] C. N. Mooers. Zatoeodmg applied to mechanical organization of knowledge. American Documentation, 2:20–32, 1951.

[6] Steven Poltrock, Jonathan Grudin, Susan Dumais, Raya Fidel, Harry Bruce, and Annelise Mark Pejtersen. Information seeking and sharing in design teams. In GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, pages 239–247, New York, NY, USA, 2003. ACM Press.

[7] Gustavo Alberto Giménez Lugo. Um Modelo de Sistemas Multiagentes para Partilha de Conhecimento utilizando Redes Sociais Comunitárias. PhD thesis, Escola Politécnica da Universidade de São Paulo, Abril 2004.

[8] Gerard Salton. The SMART Retrieval System - Experiments in Automatic Document Processing. Prentice-Hall, 1971.

[9] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Cornell University, 1987.

[10] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. Journal of the American Society for Information Science and Technology, 27(3):129–146, 1976.

[11] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. Information Processing and Management, 36(6):779–808, 2000.

[12] Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.

[13] William S. Cooper. The formalism of probability theory in ir: a foundation or an encumbrance? In SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 242–247, New York, NY, USA, 1994. Springer-Verlag.

[14] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30(1-7):107–117, 1998.

[15] W. Bruce Croft and John Lafferty, editors. Language Modeling for Information Retrieval. Springer-Verlag, 2003.

[16] Stuart J. Russel and Peter Norvig. Artificial intelligence: a modern approach. Prentice-Hall, 2 edition, 2003.

[17] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 41–48, New York, NY, USA, 2000. ACM Press.

[18] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems, 20(4):422–446, 2002.