Fabício Jailson Barth

**Recuperação de documentos e pessoas em
ambientes empresariais através de árvores de
decisão**

São Paulo
2009

Fabício Jailson Barth

**Recuperação de documentos e pessoas em
ambientes empresariais através de árvores de
decisão**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Engenharia Elé-
trica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Edson Satoshi Gomi

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com anuência de seu orientador.

São Paulo, de junho de 2009.

Assinatura do autor

Assinatura do orientador

Ficha Catalográfica

Barth, Fabrício Jailson

Recuperação de documentos e pessoas em ambientes empresariais através de árvores de decisão. / F. J. Barth – ed.rev. – São Paulo, 2009.

87 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Recuperação da Informação. 2. Aprendizado Computacional. 3. Gestão da Informação I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. t.

Dedicatória

À Fernanda G. Moreira, minha esposa, com amor, admiração e gratidão por sua compreensão, carinho, presença e incansável apoio ao longo do período de elaboração deste trabalho.

Agradecimentos

Ao Prof. Dr. Edson Satoshi Gomi, pela atenção, orientação e apoio durante a realização deste trabalho.

Aos professores, amigos, colegas e alunos, que, nos anos de convivência, muito me ensinaram, contribuindo para meu crescimento pessoal, científico e intelectual.

Ao Laboratório de Engenharia de Conhecimento (KNOMA), por colocar a disposição os equipamentos para a realização dos experimentos.

Resumo

Este trabalho avalia o desempenho do uso de árvores de decisão como função de ordenação para documentos e pessoas em ambientes empresariais. Para tanto, identificou-se atributos relevantes das entidades a serem recuperadas a partir da análise de: (i) dinâmica de produção e consumo de informações em um ambiente empresarial; (ii) algoritmos existentes na literatura para a recuperação de documentos e pessoas; e (iii) conceitos utilizados em funções de ordenação para domínios genéricos. Montou-se um ambiente de avaliação, utilizando a coleção de referência CERC, para avaliar a aplicabilidade do algoritmo C4.5 na obtenção de funções de ordenação para o domínio empresarial. O uso do algoritmo C4.5 para a construção de funções de ordenação mostrou-se parcialmente efetivo. Para a tarefa de recuperação de documentos não trouxe resultados bons. Porém, constatou-se que é possível controlar a forma de construção da função de ordenação a fim de otimizar a precisão nas primeiras posições do *ranking* ou otimizar a média das precisões (*MAP*). Para a tarefa de recuperação de pessoas o algoritmo C4.5 obteve uma árvore de decisão que consegue resultados melhores que todas as outras funções de ordenação avaliadas. O *MAP* obtido pela árvore de decisão foi 0,83, enquanto que a média do *MAP* das outras funções de ordenação foi de 0,74. Percebeu-se que a árvore de decisão utilizada para representar a função de ordenação contribui para a compreensão da composição dos diversos atributos utilizados na caracterização dos documentos e pessoas. A partir da análise da árvore de decisão utilizada como função de ordenação para pessoas foi possível entender que uma pessoa é considerada especialista em algum tópico se ela aparecer em muitos documentos, aparecer muitas vezes nos documentos e os documentos onde aparece têm uma relevância alta para a consulta.

Palavras-chave: Aprendizagem de Funções de Ordenação. Recuperação da Informação. Aprendizado Computacional. Gestão da Informação.

Abstract

This work evaluates the performance of using decision trees as ranking functions for documents and people in enterprises. It was identified relevant attributes of the entities to be retrieved from the analysis of: (i) the production and consumption of information behavior in an enterprise, (ii) algorithms for documents and people retrieval at literature, and (iii) the concepts used in ranking functions for generic domains. It was set up an evaluation environment, using the CERC collection, to evaluate the applicability of the C4.5 algorithm to obtain a ranking function for the enterprise domain. The use of C4.5 algorithm for the construction of ranking function was proved to be partially effective. In the case of documents retrieval the C4.5 has not found good results. However, it was found that is possible to control the way of building the ranking function in order to optimize the precision in the first positions of the ranking or optimize the mean average precision (*MAP*). For the task of people retrieval the C4.5 algorithm developed a ranking function that obtain better results than all other ranking functions assessed. The value of *MAP* obtained by decision tree was 0,83, while the average *MAP* of other ranking functions was 0,74. The decision tree used to represent the ranking function contributes to understanding the attributes composition used in the characterization of documents and people. Through the analysis of the decision tree used as ranking function for people, we could realise that a person is considered expert in any topic if he/she appear in many documents, appear many times in same documents and documents where he/she appears have a high relevance to the query.

Keywords: Learning to Rank. Information Retrieval. Machine Learning. Information Management.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	1
1.1	Recuperação de Informação no ambiente empresarial	2
1.2	Sistemas de Recuperação de Informação para o ambiente empresarial	4
1.3	Objetivos	8
1.4	Método da pesquisa	8
1.5	Organização do texto	9
2	Recuperação de Informação	10
2.1	Modelos de Recuperação de Informação	11
2.1.1	Modelo booleano	12
2.1.2	Modelo vetorial	13
2.1.3	Modelo probabilístico	14
2.1.4	Modelo baseado em <i>hyperlinks</i>	17
2.1.5	Breve comparação entre os modelos	20
2.2	Método para avaliação de Sistemas de Recuperação de Informação . .	21
2.2.1	Medidas	23
2.2.2	Coleções de Referência	28
2.2.3	Coleção W3C	30
2.2.4	Coleção CERC	33
2.2.5	Resumo sobre as coleções de referência para o domínio empresarial	34

3	Recuperação de Informações em Empresas	36
3.1	Estratégias para a recuperação de informação em empresas	38
3.1.1	Recuperação de referências de pessoas	39
3.1.2	Recuperação de documentos	44
3.2	Considerações sobre Recuperação de Informações em Empresas . . .	48
4	Proposta e Implementação	51
4.1	Caracterização dos documentos	52
4.2	Caracterização das pessoas	55
4.3	Criação das funções de ordenação	57
4.3.1	Seleção de Atributos	59
4.3.2	Atributos com valores contínuos em algoritmos do tipo TDIDT	59
4.3.3	Poda da árvore	60
4.3.4	Uso de árvores de decisão para ordenação de elementos . . .	61
4.3.5	Configurações do algoritmo C4.5	61
5	Resultados e Discussões	65
5.1	Função de ordenação de documentos	65
5.1.1	Metodologia e ambiente para avaliação	65
5.1.2	Resultados encontrados	66
5.1.3	Análise qualitativa da árvore de decisão	70
5.2	Função de ordenação de pessoas	71
5.2.1	Metodologia e ambiente para avaliação	72
5.2.2	Resultados encontrados	73
5.2.3	Análise qualitativa da árvore de decisão	75
5.3	Detalhes de implementação	77
6	Considerações Finais	79
6.1	Discussão dos resultados	80

6.2	Trabalhos futuros	81
	Referências	83

Lista de Figuras

1.1	Exemplo dos dados utilizados para obtenção de uma função de ordenação	6
2.1	Representação do resultado de uma expressão booleana conjuntiva (AND)	12
2.2	Conjuntos S e T utilizados pelo algoritmo HITS	18
2.3	Precisão e cobertura (BAEZA-YATES; RIBEIRO-NETO, 1999)	23
2.4	Exemplo de curva DCG	28
2.5	Exemplo de curva $NDCG$	29
3.1	Ciclo de acesso à informação	36
4.1	Exemplo de um documento	52
4.2	Texto do exemplo da figura 4.1 após filtro de <i>stop words</i>	52
4.3	Exemplo de árvore de decisão que pode ser construída a partir do conjunto de treinamento da tabela 4.2	58
5.1	Curva NDCG das funções de ordenação para documentos - intervalo da 1ª à 50ª posição	67
5.2	Curva NDCG das funções de ordenação para documentos - intervalo da 1ª à 10ª posição	68
5.3	Curva $P@n$ das funções de ordenação para documentos - intervalo da 1ª à 50ª posição	68
5.4	Curva $P@n$ das funções de ordenação para documentos - intervalo da 1ª à 10ª posição	69
5.5	Extrato da árvore de decisão gerada pelo algoritmo C4.5 para o problema de recuperação de documentos	71
5.6	Curva $P@n$ das funções de ordenação para especialistas	74
5.7	Extrato da árvore de decisão gerada pelo algoritmo C4.5 para o problema de recuperação de especialistas	75

5.8	Árvore de decisão com destaque para o ramo que leva ao sub-conjunto com maior número de exemplos positivos	76
-----	---	----

Lista de Tabelas

2.1	Algoritmo para cálculo dos hubs e autoridades (KLEINBERG, 1999) . . .	19
2.2	Exemplo de julgamentos de relevância	22
2.3	Exemplo, com dados retornados por um algoritmo hipotético, utilizado para ilustrar a medida MRR	25
2.4	Detalhes do conjunto de documentos da coleção de referência W3C (CRASWELL; VRIES; SOBOROFF, 2005)	30
2.5	Exemplo de descrição de tópico para o experimento que procura por itens conhecidos	31
2.6	Escala de relevância para os julgamentos do experimento <i>discussion search</i>	32
2.7	Escalas utilizadas nos julgamentos de relevância para os candidatos à especialistas e documentos	32
2.8	Distribuição de <i>hosts</i> por número de páginas	33
2.9	Exemplo de descrição de tópico da coleção CSIRO	34
2.10	Experimentos realizados no <i>TREC Enterprise Track</i>	35
3.1	Melhores resultados da tarefa de recuperação de especialistas do TREC 2005. Os resultados estão ordenados de acordo com o valor do MAP (CRASWELL; VRIES; SOBOROFF, 2005).	41
3.2	Melhores resultados da tarefa de recuperação de especialistas do TREC 2006. Os resultados estão ordenados de acordo com o valor do MAP. Os algoritmos em itálico são algoritmos que tem alguma intervenção humana durante o processo (SOBOROFF; VRIES; CRASWELL, 2006). . .	41
3.3	Melhores resultados da tarefa de recuperação de especialistas do TREC 2007. Os resultados estão ordenados de acordo com o valor do MAP (BAILEY et al., 2007b).	42
3.4	Melhores resultados obtidos na tarefa para recuperação de itens conhecidos (CRASWELL; VRIES; SOBOROFF, 2005)	45

3.5	Melhores resultados na tarefa de recuperação de documentos do TREC 2007 na categoria de propostas automáticas.	47
3.6	Resultados dos experimentos realizados por Shen et al. (2007) na tarefa de recuperação de documentos no TREC 2007	48
4.1	Exemplo de uma coleção processada	53
4.2	Extrato do conjunto de treinamento para o problema onde o objetivo é encontrar uma hipótese que determina se um documento é ou não relevante.	55
4.3	Resumo sobre a composição dos atributos $\{BM25_1, \dots, BM25_4\}$. .	56
4.4	Funcionamento dos algoritmos do tipo TDIDT (RUSSEL; NORVIG, 2003)	58
4.5	Algoritmo iterativo para identificação da hipótese com melhor precisão	63
5.1	Valores de média, desvio padrão, mínimo e máximo dos atributos utilizados no conjunto de dados para a tarefa de recuperação de documentos	66
5.2	Valores de $NDCG@1$ e $NDCG@2$ das funções de ordenação para documentos	67
5.3	Valores de MAP das funções de ordenação para documentos	69
5.4	Valores de $P@1$ até $P@3$ das funções de ordenação para documentos .	69
5.5	Valores de média, desvio padrão, mínimo e máximo dos atributos utilizados no conjunto de dados para a tarefa de recuperação de pessoas .	73
5.6	Valores de $P@1$ até $P@3$ das funções de ordenação para especialistas	74
5.7	Valores de MAP das funções de ordenação de especialistas	74

1 Introdução

Com a explosão da quantidade de fontes de informação digital, aliada à crescente capacidade de acesso à Internet sem limitação de horário e local, descobre-se facilmente que não é raro se chegar a uma situação de impotência e desorganização diante do excesso de informações.

Tal dificuldade também se faz presente em ambientes empresariais. Segundo um relatório do *International Data Group* (IDC) intitulado "*The High Cost of Not Finding Information*" (FELDMAN; SHERMAN, 2003) quantifica os prejuízos causados pela baixa qualidade dos Sistemas de Recuperação de Informações utilizados dentro das organizações. Geralmente, os prejuízos estão associados à perda de oportunidades e produtividade no trabalho (HAWKING, 2004). Feldman e Sherman (2003) estimam que cada colaborador de uma organização gaste em média 30% da sua jornada diária de trabalho procurando informações. Diante deste cenário, uma organização com 1.000 colaboradores, com um salário médio anual de 80.000 dólares, pode ter um prejuízo de 48.000 dólares por semana apenas no que diz respeito ao desperdício de tempo ao procurar por informações.

Segundo outro estudo do IDC, intitulado "*European Knowledge Management Fact Book*", em 1999 o custo anual resultante do retrabalho de refazer algo que já está pronto mas que não foi encontrado, realizado por colaboradores de 500 empresas europeias, foi em média 5.000 dólares por trabalhador, com uma estimativa de crescimento para 5.850 dólares no ano de 2003 ((IDC, 2000) *apud* (FELDMAN; SHERMAN, 2003)).

Portanto, percebe-se que a criação de métodos e ferramentas que possibilitem o uso mais eficaz e eficiente de todos esses recursos é uma questão vital para o sucesso dos novos paradigmas de trabalho. Dentro deste contexto, a questão da eficiência e eficácia se traduz na apresentação da informação pertinente e correta, na hora desejada e em qualquer lugar que se esteja.

1.1 Recuperação de Informação no ambiente empresarial

As organizações do século 21 são produtoras e consumidoras de conhecimento, essencial para a permanência da empresa no mercado. A produção de conhecimento dentro de uma organização se inicia a partir de uma pequena fonte de informação. O conhecimento é então distribuído internamente e externamente à organização. Este conhecimento produzido e armazenado é recuperado para satisfazer uma nova necessidade de informação. Aplicar o conhecimento recuperado em uma outra situação cria uma nova informação, e assim, o ciclo reinicia com o incremento de informações disponíveis na organização. Atualmente, os canais de distribuição de conhecimento e informações são os mais variados possíveis, aumentando a vazão da distribuição e aumentando a necessidade de sistemas capazes de monitorar, armazenar e administrar esta informação distribuída. Os Sistemas de Recuperação de Informação e gestão de conteúdo fazem parte deste ciclo de informação e são úteis para a gestão da informação (POLTROCK et al., 2003; HANSEN; JARVELIN, 2005).

Um Sistema de Recuperação de Informação é um software que tem como objetivo facilitar o acesso aos documentos, que estão armazenados neste mesmo software. Exemplos de Sistemas de Recuperação de Informação são: Google¹, Yahoo!², sistemas de busca em bibliotecas virtuais³, sistemas de busca em portais, entre outros. Em um sistema de recuperação de informação o usuário manifesta o seu interesse ou necessidade por um determinado tipo de documento através de uma consulta. Esta consulta é utilizada pelo Sistema de Recuperação de Informação para retornar uma lista de documentos. Idealmente, os documentos recuperados pelo sistema são os documentos que correspondem aos interesses ou necessidades do usuário. O que define se um documento é relevante ou não para uma consulta é um componente do Sistema de Recuperação de Informação chamado Função de Ordenação⁴. Esta função compara a consulta do usuário com os documentos armazenados pelo sistema, retornando o grau de relevância dos documentos para a consulta.

O incremento do conhecimento disponível na organização pode significar uma vantagem competitiva para a própria organização. No entanto, o uso de Sistemas de Recuperação de Informação de baixa qualidade, comum em várias organizações, pode interferir negativamente no processo de geração de conhecimento. Existem estudos que apontam para perdas significativas devido a falta de informações relevantes no

¹<http://www.google.com>

²<http://www.yahoo.com>

³Por exemplo, <http://www.usp.br/sibi>

⁴Do inglês, *Ranking Function*

momento adequado, tais como (FELDMAN; SHERMAN, 2003; HERTZUM; PEJTERSEN, 2000):

- Decisões ruins baseadas em poucas informações;
- Esforço duplicado porque mais de um grupo ou pessoa trabalham na mesma tarefa sem saber que esta tarefa foi parcialmente ou totalmente finalizada;
- Perda de negócios ou vendas porque o potencial cliente não conseguiu encontrar informações sobre o produto ou serviço ou porque o responsável pela área de negócio não soube encontrar as informações certas dentro da própria organização, e;
- Baixa produtividade porque os colaboradores não conseguem encontrar a informação que precisam para executar uma tarefa.

A Recuperação de Informação em ambientes empresariais difere de várias maneiras da Recuperação de Informação em outros contextos. Por exemplo, a noção de resposta adequada para uma consulta realizada no ambiente empresarial é razoavelmente diferente da noção de resposta adequada para uma consulta realizada na Internet. Na Internet, existem muitos documentos que podem ser considerados relevantes para uma consulta. Em uma organização, a noção de resposta adequada é definida como a resposta correta. Normalmente, um grande número de consultas realizadas em uma organização têm um conjunto pequeno de documentos considerados relevantes. A resposta correta para uma consulta em uma organização não necessariamente inclui os documentos mais populares - conceito normalmente utilizado para definir a resposta de uma consulta na Internet. Encontrar a resposta correta geralmente é mais difícil do que encontrar a melhor resposta (MUKHERJEE; MAO, 2004).

Outra característica muito peculiar do ambiente empresarial é o fato de que as fontes de informação mais comum são documentos e pessoas (POLTROCK et al., 2003). Ou seja, os Sistemas de Recuperação de Informações Empresariais devem recuperar, além da infinidade de diferentes tipos de documentos, referências de pessoas que possam satisfazer a necessidade de informação de um determinado usuário.

O conteúdo da Internet e o conteúdo dos documentos de uma organização são criados de formas e com objetivos diferentes (FAGIN et al., 2003). O conteúdo existente na Internet reflete a opinião de muitos autores, que são livres para publicar qualquer coisa. O conteúdo existente na Intranet de uma organização reflete a visão de uma organização. O conteúdo de uma Intranet é criado para disseminar informação, mais do que atrair e capturar a atenção de um grupo específico de usuários, por exemplo.

Nem todos os colaboradores de uma organização podem publicar conteúdo livremente na Intranet de uma organização (MUKHERJEE; MAO, 2004; BERTINO; RAMAMRITHAM, 2004).

Muitas das consultas, entre 60% e 80%, feitas dentro de um ambiente empresarial têm como objetivo recuperar algo que o usuário já viu. O usuário lembra de algum atributo do documento e utiliza este atributo como termo em uma consulta (MUKHERJEE; MAO, 2004). Tipicamente, em uma organização um documento não referencia outro documento através de *hyperlinks*. Conseqüentemente, algoritmos normalmente utilizados na Web, como *PageRank* (BRIN; PAGE, 1998) e *HITS* (KLEINBERG, 1999), não têm o mesmo desempenho em ambientes empresariais. Desta maneira, outros modelos e algoritmos devem ser explorados para melhorar o desempenho de Sistemas de Recuperação de Informações utilizados em ambientes empresariais (MUKHERJEE; MAO, 2004; HAWKING, 2004).

1.2 Sistemas de Recuperação de Informação para o ambiente empresarial

Num Sistema de Recuperação de Informação Empresarial a meta é encontrar todos os documentos e pessoas que forneçam a informação necessária para alguém realizar uma tarefa. Um Sistema de Recuperação de Informação Empresarial deve apresentar primeiro aqueles itens que são mais úteis ao usuário. Esta noção de utilidade depende muito da natureza da organização, da identidade do usuário e das características da tarefa que o usuário está executando (HAWKING, 2004). Um desenvolvedor de software quando informa ao Sistema de Recuperação de Informação a consulta “*xml parser*” espera por um resultado diferente do resultado que um analista de sistemas espera, por exemplo.

A área de pesquisa em Sistemas de Recuperação de Informação Empresarial tem como objetivos entender as necessidades de informação dos usuários e entender as características das fontes de informação de uma organização para permitir o desenvolvimento de modelos e algoritmos apropriados para a recuperação de informação nesta organização (CRASWELL; VRIES; SOBOROFF, 2005).

O principal problema com o desenvolvimento de Sistemas de Recuperação de Informação para o ambiente empresarial é a definição de funções de ordenação eficientes, isto é, funções de ordenação capazes de retornar apenas documentos e referências de pessoas relevantes, classificados de acordo com o grau de relevância do documento ou da pessoa para uma determinada necessidade de informação (MUKHERJEE; MAO,

2004). Ou seja, acredita-se que a baixa qualidade dos atuais sistemas de recuperação de informação para o ambiente empresarial é consequência da não definição adequada de funções de ordenação eficientes.

Inúmeras funções de ordenação para o ambiente empresarial foram propostas nos últimos anos. Muitas das funções propostas foram baseadas em modelos de recuperação de informação tradicionais (ver capítulo 2) e outras a partir da análise do conteúdo disponível em empresas (ver capítulo 3). Infelizmente, a aplicação de funções de ordenação oriundas dos modelos tradicionais e de funções criadas a partir da análise do conteúdo disponível em empresas não têm tido um desempenho adequado. Especula-se que:

- A diversidade das informações existentes no ambiente empresarial dificulta a identificação de padrões que possam ser utilizados para a construção manual de funções de ordenação, e;
- As peculiaridades das informações contidas no ambiente empresarial impossibilita o emprego direto de funções de ordenação dos modelos clássicos de recuperação de informação.

Definir uma função de ordenação é um problema central para muitas aplicações da área de Recuperação de Informação. Dependendo da aplicação, a função de ordenação pode ser utilizada para recuperar, recomendar ou filtrar itens. Dado um conjunto de itens, uma função de ordenação deve calcular um valor para cada um dos objetos da coleção e ordená-los de acordo com este valor. O valor atribuído a cada um dos objetos pode representar o grau de relevância, preferência ou importância, dependendo da aplicação (LIU et al., 2007).

Na maioria dos casos, as funções de ordenação são definidas manualmente. Ao definir uma função de ordenação manualmente é necessário realizar vários ajustes nos parâmetros, pois na maioria dos casos as funções de ordenação padrão possuem algumas variáveis que precisam ser ajustadas. Em outros casos, pode ser necessário a composição de várias funções de ordenação padrão ou a criação de uma nova função. Isto muitas vezes é difícil de ser elaborado e de ser testado (JOACHIMS et al., 2007).

Tradicionalmente, o ajuste dos parâmetros de uma função de ordenação é realizado com base em um pequeno conjunto de itens rotulados como relevantes ou não para algumas necessidades de informação. Isto pode gerar uma situação onde o desempenho da função de ordenação para o conjunto de itens é alto, mas para uma situação real o desempenho é baixo. Isto é conhecido como sobre-ajuste da função de ordenação ao conjunto de itens (LIU et al., 2007; JOACHIMS et al., 2007).

Tais conjuntos de itens rotulados podem ser obtidos através de coleções de referência ou a partir do registro de utilização (*log*) de Sistemas de Recuperação de Informação. Uma coleção de referência é formada por um conjunto de documentos, um conjunto de consultas e julgamentos de relevância. Os julgamentos de relevância são descrições da relevância de um documento para uma consulta. Um exemplo de julgamentos de relevância é apresentado na tabela do lado esquerdo da figura 1.1.

As coleções de referência têm sido amplamente utilizadas no teste de novos Sistemas de Recuperação de Informação. O registro de utilização é um componente, presente na maioria dos modernos Sistemas de Recuperação de Informação, que tem como objetivo monitorar a interação do usuário com o sistema e gravar esta interação em um arquivo de *log*. As informações contidas neste arquivo podem ser úteis na identificação de julgamentos de relevância de usuários reais de um Sistema de Recuperação de Informação. Estes registros de utilização podem ser transformados em pares de itens e consultas com um julgamento de relevância associado: relevante ou não relevante, nos mesmos formatos dos julgamentos de relevância de uma coleção de referência (tabela do lado esquerdo da figura 1.1).

Com o aumento dos registros de utilização dos Sistemas de Recuperação de Informação e das coleções de referência é possível pensar em outras abordagens capazes de manipular muitos parâmetros com base em um conjunto grande de itens rotulados (QIN et al., 2008; CRASWELL; VRIES; SOBOROFF, 2005; SOBOROFF; VRIES; CRASWELL, 2006).

Ao invés da função de ordenação ser criada de maneira manual, pode-se utilizar algoritmos que percorram a tabela de julgamentos de relevância para encontrar uma especificação de documento relevante. Esta especificação pode ser utilizada como uma função de ordenação em um Sistema de Recuperação de Informação. No entanto, para que isto aconteça, é necessário que cada item dos julgamentos de relevância $\langle \text{consulta}, \text{item}, \text{relevância} \rangle$ seja caracterizado por um conjunto de atributos $\langle a_1, a_2, \dots, a_n \rangle$, formando assim, um conjunto de dados para a obtenção da função de ordenação como apresentado na figura 1.1.

Figura 1.1: Exemplo dos dados utilizados para obtenção de uma função de ordenação

Neste caso, o problema de construção da função de ordenação deve ser pensado como um problema de aprendizagem de máquina, onde o objetivo é encontrar um padrão entre os atributos e a relevância de cada julgamento existente no conjunto de dados. Este padrão é utilizado para descrever o conceito de item relevante e para construir a função de ordenação (JOACHIMS et al., 2007; LIU et al., 2007).

A tarefa de aprendizagem da função de ordenação de um Sistema de Recuperação de Informação é um processo que recebe como entrada um conjunto de dados D , formado por um conjunto de instâncias (denominadas exemplos) no formato $\langle q, i, r \rangle$, onde q é uma consulta (representada por uma lista de termos $\{t_1, t_2, \dots, t_n\}$), i é um item (representado por uma lista de atributos $\{a_1, a_2, \dots, a_m\}$) e r é a relevância de i para q . A relevância é representada por um conjunto de valores categóricos (i.e., 0, 1 e 2). A saída da tarefa de aprendizagem é uma hipótese que descreve o conceito de item relevante. Esta hipótese pode ser utilizada no lugar da função de ordenação de um Sistema de Recuperação de Informação (VELOSO et al., 2008; TROTMAN, 2005).

O uso de algoritmos de aprendizagem de máquina pode contribuir na construção de funções de ordenação eliminando a necessidade de ajustes manuais dos parâmetros, combinando múltiplos atributos encontrados nos itens analisados, aumentando a expressividade da função de ordenação e evitando sobre-ajuste das funções de ordenação sobre o conjunto de treinamento (LIU et al., 2007). Além disso, se o algoritmo de aprendizagem de máquina utilizado permitir a criação de estruturas simbólicas, que sejam compreensíveis por pessoas, é possível entender como os atributos utilizados para representar os itens podem formar uma função de ordenação e é possível ter uma explicação do porquê determinado documento ou pessoa foi considerado relevante ou não pelo sistema.

Aparentemente, o uso de algoritmos de aprendizagem de máquina simbólicos na criação de funções de ordenação fornece inúmeros benefícios para o problema de recuperação de informação no ambiente empresarial. No entanto, existem diversos algoritmos de aprendizagem de máquina que geram representações simbólicas. Algumas das representações adotadas são mais expressivas que outras: lógica de primeira ordem é mais expressiva que lógica proposicional, por exemplo. Quanto mais expressiva a linguagem adotada para a representação de uma hipótese, maior é o custo computacional que o algoritmo de aprendizagem de máquina tem para gerar esta hipótese. Por isso, que neste trabalho optou-se pela utilização de árvores de decisão na representação de funções de ordenação. Árvores de decisão são representadas utilizando sentenças proposicionais, que consegue representar uma decisão complexa a partir de diversas decisões elementares e o custo computacional que um algoritmo indutor de árvores de decisão tem para gerar uma árvore de decisão é baixo.

1.3 Objetivos

O objetivo principal deste trabalho é:

- i. Avaliar o desempenho do uso de árvores de decisão como função de ordenação para documentos e pessoas em ambientes empresariais.

Os objetivos secundários, necessário para alcançar o objetivo principal, são:

- i. Identificar atributos que caracterizam as fontes de informação, documentos e pessoas, de tal forma que seja possível construir uma árvore de decisão para a ordenação de documentos e pessoas;
- ii. Avaliar, utilizando medidas convencionais da área de Recuperação de Informação, se a função de ordenação, na forma de árvore de decisão, criada obtém melhores resultados que outras funções de ordenação existentes, e;
- iii. Avaliar se a árvore de decisão utilizada para representar a função de ordenação contribui para: a compreensão da razão da utilização de determinados atributos na composição da função de ordenação, e explicar porque determinado documento ou pessoa foi considerado relevante ou não.

1.4 Método da pesquisa

Para alcançar os objetivos enumerados acima foram realizadas as seguintes atividades:

- i. Através da revisão da literatura compreender a dinâmica de produção e consumo de informação em um ambiente empresarial, visando identificar atributos relevantes na caracterização das fontes de informação;
- ii. Compreender o funcionamento dos algoritmos existentes na literatura para a recuperação de documentos e pessoas (especialistas) em ambientes empresariais. Extrair destes algoritmos conceitos e idéias que podem ser utilizadas na caracterização adequada das fontes de informação e/ou no aprimoramento dos algoritmos de aprendizagem de máquina utilizados;
- iii. Elencar e verificar a aplicabilidade de atributos, geralmente utilizados em funções de ordenação para domínios genéricos, na caracterização de fontes de informação do domínio empresarial;

- iv. Montar um ambiente de avaliação, utilizando coleções de referência do domínio empresarial, para avaliar a aplicabilidade de algoritmos de aprendizagem de máquina simbólicos na obtenção de funções de ordenação, e;
- v. Avaliar as funções de ordenação encontradas.

1.5 Organização do texto

O texto está organizado da seguinte maneira:

- i. No capítulo 2 são descritos os principais conceitos que fundamentam a área de Recuperação de Informação. O conteúdo descrito neste capítulo é fundamental para a execução da atividade iii, para a definição das coleções de referência utilizadas na atividade iv e para a definição das medidas utilizadas na atividade v do método descrito na seção 1.4;
- ii. No capítulo 3 são descritos a dinâmica de acesso à informação dentro de uma organização e algoritmos existentes na literatura para a recuperação de documentos e pessoas em empresas. Este conteúdo é essencial para a execução das atividades i e ii do método descrito na seção 1.4;
- iii. O capítulo 4 apresenta os detalhes da proposta desta tese, incluindo detalhes de implementação;
- iv. O capítulo 5 descreve os experimentos e os resultados alcançados. Cada resultado obtido também é discutido neste capítulo, e;
- v. O capítulo 6 apresenta as conclusões do trabalho.

2 Recuperação de Informação

O termo Recuperação de Informação (RI) foi cunhado por (MOOERS, 1951) 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 (POLTROCK et al., 2003). 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 (MANNING; RAGHAVAN; SCHÜTZ, 2008).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZE, 2008).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999). 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 (MANNING; RAGHAVAN; SCHÜTZE, 2008).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (LUGO, 2004).

2.1 Modelos de Recuperação de Informação

Segundo Baeza-Yates e Ribeiro-Neto (1999), 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 (BAEZA-YATES; RIBEIRO-NETO, 1999): 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 (BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZ, 2008).

2.1.1 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 2.1). 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 (BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZ, 2008).

Figura 2.1: Representação do resultado de uma expressão booleana conjuntiva (AND)

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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

2.1.2 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 (SALTON, 1971) apud (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 $\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$, 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 $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$ (BAEZA-YATES; RIBEIRO-NETO, 1999).

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:

$$\text{sim}(q, d_j) = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (2.1)$$

Sendo $w_{i,j} \geq 0$ e $w_{i,q} \geq 0$, a similaridade $\text{sim}(d_j, q)$ 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 (BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZE, 2008).

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 (SALTON; BUCKLEY, 1987), tenta balancear o número de ocorrências do termo no documento com o número de documentos onde o termo aparece (equação 2.2).

$$w_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \times \log \frac{N}{n_i} \quad (2.2)$$

onde,

- N = número de documentos da coleção.
- n_i = número de documentos onde a palavra i aparece.
- $\frac{f_{i,j}}{\max_z f_{z,j}}$ = frequência 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 (LUGO, 2004; MANNING; RAGHAVAN; SCHÜTZE, 2008).

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 (SALTON, 1971) apud (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

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.

2.1.3 Modelo probabilístico

O modelo probabilístico, introduzido por Robertson e Jones (1976), 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

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

conjunto de documentos que contem exatamente os documentos relevantes e nenhum não relevante (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (JONES; WALKER; ROBERTSON, 2000).

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 $\bar{R}$, o complemento de R (JONES; WALKER; ROBERTSON, 2000).

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 e assume que a distribuição de termos na coleção seja capaz de informar a relevância provável de um documento para uma consulta (BAEZA-YATES; RIBEIRO-NETO, 1999).

O peso dos termos em um modelo probabilístico são todos binários, por exemplo, $w_{i,j} \in \{0, 1\}^2$ 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 $\bar{R}$ 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 $P(\bar{R}|\vec{d}_j)$ 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:

$$sim(d_j, q) = \frac{P(R|\vec{d}_j)}{P(\bar{R}|\vec{d}_j)} \quad (2.3)$$

Aplicando a regra de Bayes (MITCHELL, 1997) tem-se:

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 .

$$\text{sim}(d_j, q) = \frac{P(\vec{d}_j|R) \times P(R)}{P(\vec{d}_j|\bar{R}) \times P(\bar{R})} \quad (2.4)$$

onde,

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

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

$$\text{sim}(d_j, q) \sim \frac{P(\vec{d}_j|R)}{P(\vec{d}_j|\bar{R})} \quad (2.5)$$

Assumindo a independência entre termos, tem-se:

$$\text{sim}(d_j, q) \sim \prod_{i=1}^M \frac{P(d_{j,i}|R)}{P(d_{j,i}|\bar{R})} \quad (2.6)$$

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

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 (COOPER, 1994; BAEZA-YATES; RIBEIRO-NETO, 1999).

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 frequência 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 (COOPER, 1994; BAEZA-YATES; RIBEIRO-NETO, 1999).

Um exemplo concreto, e de sucesso, da aplicação do modelo probabilístico é a função de ordenação BM25 (BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHA-

VAN; SCHÜTZE, 2008). Esta função de ordenação faz uso da frequência dos termos no documento e também leva em consideração o tamanho dos documentos (equação 2.7).

$$sim(q, d_j) = \sum_{t \in q} \log \frac{N}{n_i} \times \frac{(k_1 + 1) \times f_{t,d}}{k_1((1 - b) + b \times (T_d/T_{med})) + f_{t,d}} \quad (2.7)$$

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} .

2.1.4 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 (MANNING; RAGHAVAN; SCHÜTZE, 2008; KLEINBERG, 1999; BRIN; PAGE, 1998). Um *hyperlink* é uma estrutura comum em documentos no formato HTML. Esta estrutura possui duas propriedades importantes (MANNING; RAGHAVAN; SCHÜTZE, 2008):

- 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 (KLEINBERG, 1999) e PAGERANK (BRIN; PAGE, 1998).

2.1.4.1 HITS

Kleinberg (1999) 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 contém 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 2.2.

Figura 2.2: Conjuntos S e T utilizados pelo algoritmo HITS

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

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

$$a(d_i) = \sum_{d_j \rightarrow d_i} h(d_j) \quad (2.8)$$

$$h(d_i) = \sum_{d_i \rightarrow d_j} a(d_j) \quad (2.9)$$

onde, a representação $d_i \rightarrow d_j$ 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 2.1. Este algoritmo pode ser utilizado para filtrar os n documentos com maior autoridade ou os n documentos com maior hub.

Iterate(G,k)

G : coleção de n documentos

k : um número natural

$z \leftarrow$ um vetor $(1, 1, \dots, 1) \in \mathbb{R}^n$

$a_0 \leftarrow z$

$h_0 \leftarrow z$

for $i = 1, 2, \dots, k$ **do**

 Aplica a equação 2.8 em (a_{i-1}, h_{i-1}) para obter os valores de a'_i

 Aplica a equação 2.9 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 2.1: Algoritmo para cálculo dos hubs e autoridades (KLEINBERG, 1999)

2.1.4.2 PAGERANK

O algoritmo PAGERANK (BRIN; PAGE, 1998) 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:

$$PR(A) = (1 - d) + d \left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \dots + \frac{PR(T_n)}{C(T_n)} \right) \quad (2.10)$$

onde:

- A é um documento.
- $T_{1,2,\dots,n}$ 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 (BRIN; PAGE, 1998), 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 (BRIN; PAGE, 1998), parecido com o apresentado na figura 2.1.

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.

2.1.5 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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

Existe diversos trabalhos na literatura comparando os modelos vetorial e probabilístico (CROFT; LAFFERTY, 2003; SALTON; BUCKLEY, 1987). Alguns destes trabalhos sugerem que o modelo probabilístico supera o modelo vetorial (CROFT; LAFFERTY, 2003) e outros sugerem que o modelo vetorial supera o modelo probabilístico em coleções de documentos genéricos (SALTON; BUCKLEY, 1987). 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 (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Os modelos booleano, vetorial e probabilístico estão, quase que por inteiro, no nível de palavras (LUGO, 2004). Como consequência, diversas pesquisas tentaram responder à seguinte pergunta: a Recuperação de Informação não melhoraria se fossem utilizadas técnicas mais sofisticadas para representar os documentos e as necessidades de informação dos usuários? Muitos têm trilhado esta via, porém segundo (RUSSEL; NORVIG, 2003): “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.

2.2 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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 (LUGO, 2004; BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZE, 2008):

- *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 documentos, 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 2.2 é possível visualizar um exemplo de julgamentos de relevância.

Tabela 2.2: 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 (MANNING; RAGHAVAN; SCHÜTZE, 2008).

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 (LUGO, 2004; BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZE, 2008).

2.2.1 Medidas

Existem diversas medidas que podem ser utilizadas para avaliar o desempenho de um Sistema de Recuperação de Informação. As medidas mais utilizadas são precisão⁴ e cobertura⁵ (BAEZA-YATES; RIBEIRO-NETO, 1999).

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 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 (BAEZA-YATES; RIBEIRO-NETO, 1999):

$$precisao = \frac{|relevantes \cap recuperados|}{|recuperados|} \quad (2.11)$$

$$cobertura = \frac{|relevantes \cap recuperados|}{|relevantes|} \quad (2.12)$$

A figura 2.3 ilustra estes conceitos.

Figura 2.3: Precisão e cobertura (BAEZA-YATES; RIBEIRO-NETO, 1999)

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

⁴Do inglês, *precision*

⁵Do inglês, *recall*

e ter um baixo índice de cobertura, mas teria uma chance razoável de 100% de precisão (RUSSEL; NORVIG, 2003). Uma forma de definir uma média harmônica entre precisão e cobertura é através da medida F^6 (equação 2.13) (MANNING; RAGHAVAN; SCHÜTZE, 2008).

$$F = \frac{2 \times (\text{precisao} \times \text{cobertura})}{\text{precisao} + \text{cobertura}} \quad (2.13)$$

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 resultados (RUSSEL; NORVIG, 2003).

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 (MANNING; RAGHAVAN; SCHÜTZE, 2008).

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.

2.2.1.1 Precisão em n ($P@n$)

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

$$P@n = \frac{r}{n} \quad (2.14)$$

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 $P@n$.

⁶Do inglês, *F-measure*

2.2.1.2 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:

$$AP = \frac{\sum_{n=1}^N P@n \times rel(n)}{r_q} \quad (2.15)$$

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:

$$rel(n) = \begin{cases} 1 & \text{se o } n^{th} \text{ documento é relevante} \\ 0 & \text{caso contrário} \end{cases} \quad (2.16)$$

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

2.2.1.3 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:

$$MRR = \frac{\sum_{i=1}^N \frac{1}{p_i}}{N} \quad (2.17)$$

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 2.3 é igual a:

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

Tabela 2.3: Exemplo, com dados retornados por um algoritmo hipotético, 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

2.2.1.4 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 relevância 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:

$$CG[i] = \begin{cases} G[i] & \text{se } i = 1 \\ CG[i - 1] + G[i] & \text{caso contrário} \end{cases} \quad (2.18)$$

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 $G' = \{3, 2, 3, 0, 0, 1, 2, 2, 3, 0, \dots\}$.

Para a lista G' é retornado a seguinte lista de $CG' = \{3, 5, 8, 8, 8, 9, 11, 13, 16, 16, \dots\}$. 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 (JäRVELIN; KEKÄLÄINEN,

2000):

$$DCG[i] = \begin{cases} G[i] & \text{se } i = 1 \\ DCG[i-1] + \frac{G[i]}{\log i} & \text{caso contrário} \end{cases} \quad (2.19)$$

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

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 (JäRVELIN; KEKÄLÄINEN, 2002):

$$MR[i] = \begin{cases} 3 & \text{se } i \leq m \\ 2 & \text{se } m < i \leq m + l \\ 1 & \text{se } m + l < i \leq m + l + k \\ 0 & \text{caso contrário} \end{cases} \quad (2.20)$$

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 $MR' = \{3, 3, 3, 2, 2, 2, 2, 1, 1, 0, \dots\}$.

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 2.4 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).

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 ($V = \{v_1, v_2, \dots, v_k\}$) de uma função de ordenação qualquer e um vetor DCG com desempenho ideal ($I = \{i_1, i_2, \dots, i_k\}$), o vetor $NDCG$ de V é obtido através da seguinte equação (JäRVELIN; KEKÄLÄINEN, 2002):

Figura 2.4: Exemplo de curva *DCG*

$$NDCG(V) = \{v_1/i_1, v_2/i_2, \dots, v_k/i_k\} \quad (2.21)$$

Na figura 2.5 é possível visualizar um exemplo de curva *NDCG* onde são utilizados os mesmos dados que foram apresentados na figura 2.4.

2.2.2 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*⁷ (de *Text Retrieval Conference*, cuja primeira versão é de 1992) (BAEZA-YATES; RIBEIRO-NETO, 1999; MANNING; RAGHAVAN; SCHÜTZ, 2008).

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 contém 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

⁷<http://trec.nist.gov/>

Figura 2.5: Exemplo de curva *NDCG*

experimento (MANNING; RAGHAVAN; SCHÜTZE, 2008).

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 (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Uma das seções ativas no ano de 2008, chamada *Enterprise Track*, é voltada especialmente para o domínio empresarial. Nos anos de 2005 e 2006 esta seção fez uso de uma coleção com documentos fornecidos pelo consórcio *World Wide Web Consortium* (W3C⁸) e nos anos de 2007 e 2008 esta seção fez uso de uma coleção com documentos da *Commonwealth Scientific and Industrial Research Organization* (CSIRO⁹), sediada na Austrália.

Para construir uma coleção de referência realística para o domínio empresarial é importante que a coleção (documentos, tópicos e julgamentos) seja representativo

⁸<http://www.w3.org>

⁹<http://www.csiro.au>

do ambiente real de uma empresa (BAILEY et al., 2007a). Isto gera dois potenciais problemas (BAILEY et al., 2007a):

- Cada empresa é diferente, sendo assim, não está claro se e quando uma coleção gerada a partir de uma única empresa pode ser utilizada para responder perguntas sobre todo o universo de empresas.
- A maioria dos documentos utilizados nas empresas são confidenciais. Muitas vezes, apenas os colaboradores das empresas têm acesso aos documentos ou apenas um subconjunto de colaboradores das empresas têm acesso a certos documentos. Sendo assim, a coleção de documentos, na maioria dos casos, é criada somente a partir de documentos públicos.

Nas próximas seções serão apresentados detalhes das coleções de referência para o domínio empresarial: W3C e CERC.

2.2.3 Coleção W3C

A coleção de referência W3C é um ambiente de testes para o domínio empresarial formada por uma coleção de documentos gerada a partir de um *crawler* realizado nos sites públicos do consórcio W3C (*.w3.org) em junho de 2004 (CRASWELL; VRIES; SOBOROFF, 2005). Ao total foram indexados 331.037 documentos.

Na tabela 2.4 é possível visualizar detalhes desta coleção de documentos. A coluna *host* significa o *host* onde as páginas foram encontradas, por exemplo: *lists* significa *lists.w3.org*. A única exceção é o valor *outros* que representa diversos *hosts* pequenos.

Tabela 2.4: Detalhes do conjunto de documentos da coleção de referência W3C (CRASWELL; VRIES; SOBOROFF, 2005)

Tipo	Host	Tamanho(GB)	Número de documentos	Média do tamanho dos documentos(KB)
E-mail	lists	1.855	198.394	9.8
Código	dev	2.578	62.509	43.2
Páginas web	www	1.043	45.975	23.8
Wiki web	esw	181	19.605	9.7
Diversos	outros	47	3.538	14.1
Web	people	3	1.016	3.6
	todos	5.7	331.037	18.1

Utilizando esta coleção, nos anos de 2005 e 2006, foram definidas duas tarefas, num total de três experimentos: tarefa de recuperação de e-mails, e; tarefa de re-

cuperação de especialistas (CRASWELL; VRIES; SOBOROFF, 2005; SOBOROFF; VRIES; CRASWELL, 2006).

2.2.3.1 Tarefa de recuperação de e-mails

Nesta tarefa são utilizados apenas os documentos do *host lists.w3.org*. Os experimentos realizados dentro desta tarefa são dois: procurar por itens conhecidos (*known item experiment*), e; procurar por documentos que expressam pontos de vista distintos sobre o mesmo assunto (*discussion search experiment*).

O experimento que mede a capacidade de um algoritmo encontrar itens conhecidos é composto por 125 consultas para testes, mais 25 consultas para treinamento do algoritmo. O objetivo deste experimento é simular uma situação onde o usuário procura por um item conhecido, ou seja, um item que ele já teve acesso. O algoritmo avaliado deve recuperar o item conhecido como o primeiro item da lista ou próximo do primeiro item. Neste caso, para cada consulta existe um único documento correto. Um exemplo de tópico para este experimento é apresentado na tabela 2.5.

Tabela 2.5: Exemplo de descrição de tópico para o experimento que procura por itens conhecidos

```
<top>  
<num> Number: KI44 </num>  
<title> James' question about the Webdav in Tomcat </title>  
</top>
```

O objetivo do segundo experimento é simular uma situação onde o usuário procura por documentos que expressam opiniões distintas sobre um mesmo assunto. Para este conjunto de julgamentos existem três tipos de respostas: documento não relevante, documento relevante sem descrição de prós e contras e documento relevante com prós e contras sobre o assunto.

No ano de 2006, a coleção W3C continuou sendo usada na seção *Enterprise Track*, com a adição de novos tópicos e julgamentos de relevância no experimento que simula uma situação onde o usuário procura por documentos que expressam opiniões distintas sobre um mesmo assunto (SOBOROFF; VRIES; CRASWELL, 2006).

No experimento *discussion search* os tópicos foram construídos pensando em situações onde a pessoa procura por argumentos sobre decisões de projeto e de desenvolvimento de software. Neste ano foram adicionados mais 50 tópicos a coleção e a escala de relevância para os julgamentos foi alterada, como apresentado na tabela 2.6.

Tabela 2.6: Escala de relevância para os julgamentos do experimento *discussion search*

0	não relevante
1	relevante, mas não possui argumentos
2	relevante, possui argumento negativo
3	relevante, possui ambos argumentos
4	relevante, possui argumento positivo

2.2.3.2 Tarefa de recuperação de especialistas

Nesta tarefa existe apenas um experimento onde o objetivo é encontrar pessoas (*expert search*), consideradas especialistas, para um determinado tópico. Os algoritmos projetos para esta tarefa podem fazer uso de todos os documentos da coleção para recuperar pessoas consideradas especialistas dentro de um universo de 1092 candidatos (CRASWELL; VRIES; SOBOROFF, 2005).

Os sistemas devem retornar para cada tópico uma lista de especialistas. Além disso, para cada especialista é necessário retornar uma lista ordenada de documentos que justificam a escolha do especialista (SOBOROFF; VRIES; CRASWELL, 2006).

As escalas utilizadas, para os candidatos à especialistas e documentos, nos julgamentos de relevância são apresentados na tabela 2.7.

Tabela 2.7: Escalas utilizadas nos julgamentos de relevância para os candidatos à especialistas e documentos

<i>Candidatos à especialistas</i>	
0	se o candidato não é um especialista
1	não se tem nenhuma classificação sobre o candidato
2	o candidato é um especialista
<i>Documentos de suporte</i>	
0	suporte negativo (o documento indica que a pessoa não é um especialista)
1	não indica nada sobre o candidato
2	suporte positivo (o documento indica que a pessoa é um especialista)

No ano de 2005 os tópicos foram criados a partir dos grupos de trabalho do W3C e os membros de cada grupo foram considerados os especialistas de cada tópico. Ao todo, foram disponibilizadas 10 tópicos para treinamento e 50 para testes, totalizando 60 tópicos (CRASWELL; VRIES; SOBOROFF, 2005).

No ano de 2006 foram selecionados 55 tópicos. Neste ano não foi utilizada nenhuma estrutura pré-definida do W3C para a criação dos julgamentos de relevância. Neste ano os julgamentos foram realizados pelos próprios participantes do TREC, após

a análise dos tópicos e documentos (SOBOROFF; VRIES; CRASWELL, 2006).

2.2.4 Coleção CERC

A coleção de referência CERC (*CSIRO Enterprise Research Collection*)¹⁰ é um ambiente de testes para o domínio empresarial formada por uma coleção de documentos, descrições de tópicos e julgamentos de relevância para documentos e especialistas (BAILEY et al., 2007a). Esta coleção de referência foi criada utilizando documentos e colaboradores da *Australian Commonwealth Scientific and Research Organization* (CSIRO) e foi usada nos anos de 2007 e 2008 no *Enterprise Track* (BAILEY et al., 2007b).

Esta coleção de referência possui 370.715 documentos, totalizando 4.2 GB de informação. O *crawler* para construção desta coleção de documentos foi aplicado em 100 *hosts* diferentes. Na tabela 2.8 é possível visualizar a distribuição do tamanho dos *hosts*, por exemplo, quatorze (14) *hosts* tiveram somente uma (1) página indexada e oito (8) *hosts* têm um tamanho entre 10.000 e 100.000 documentos. O maior *host* é o *anic.ento.csiro.au* com 72.358 documentos. A coleção tem aproximadamente 7.9 milhões de hiperlinks, 95% dos documentos têm um ou mais links de saída com texto associado e 38% dos documentos contêm no mínimo um (1) endereço de e-mail. De toda a coleção de documentos foram extraídos 3.678 e-mails pessoais distintos (BAILEY et al., 2007a).

Tabela 2.8: Distribuição de *hosts* por número de páginas

Número de <i>hosts</i>	14	7	25	31	15	8
Número de documentos	10 ⁰	10 ¹	10 ²	10 ³	10 ⁴	10 ⁵

Os tópicos criados para esta coleção, 50 no ano de 2007 e 77 no ano de 2008, são usados para ambos os experimentos, recuperação de documentos e recuperação de especialistas. Cada descrição de tópico possui uma consulta (*query*), uma descrição da necessidade de informação e alguns exemplos de referências chaves (em média 4 por consulta) (BAILEY et al., 2007b).

Um exemplo de descrição de tópico pode ser visto na tabela 2.9. No caso da recuperação de documentos, os participantes podem submeter soluções chamadas *automáticas* e *com realimentação*¹¹. No caso das soluções chamadas automáticas, as equipes podem fazer uso apenas dos campos *query* e *narr*. Toda equipe pode submeter até cinco (5) soluções, sendo que no mínimo uma solução deverá usar apenas

¹⁰<http://es.csiro.au/cerc/>

¹¹Do inglês, feedback

o campo *query*. No caso das soluções classificadas como soluções com realimentação, as equipes podem fazer uso dos campos *query*, *narr* e dos documentos descritos no atributo *page*. Estes documentos são utilizados para simular a realimentação que o usuário pode fornecer ao Sistema de Recuperação de Informação, indicando quais documentos ele considera relevante para a consulta.

Tabela 2.9: Exemplo de descrição de tópico da coleção CSIRO

```
<top>
<num>CE-003</num>
<query>sustainable agriculture </query>
<narr>
General information about CSIRO's activities
in sustainable agriculture with links to more
specific research areas.
</narr>
<page>CSIRO144-09895603</page>
<page>CSIRO139-13945406</page>
</top>
```

Os primeiros cinquenta tópicos, referentes ao ano de 2007, foram criados por comunicadores científicos da CSIRO. O papel destes comunicadores é promover as capacidades da empresa, interagindo com grupos industriais, agências de governo e o público em geral. A principal tarefa destes comunicadores científicos é criar páginas no portal (Internet) da CSIRO que fornecem uma visão geral sobre determinado tópico, enumerando documentos e pessoas relevantes para o tópico (BAILEY et al., 2007b).

2.2.5 Resumo sobre as coleções de referência para o domínio empresarial

Nas seções 2.2.3 e 2.2.4 foram apresentadas as coleções do W3C e CERC, respectivamente. Com a coleção do W3C foram realizados três experimentos nos anos de 2005 e 2006: recuperação de especialistas; recuperação de itens conhecidos, e; recuperação de documentos com pontos de vista distintos. Com a coleção CERC foram realizados dois experimentos nos anos de 2007 e 2008: recuperação de especialistas, e; recuperação de documentos (tabela 2.10).

Consequentemente, os cenários que as coleções de referência do W3C e CERC permitem avaliar são:

- Situações onde as pessoas procuram por itens conhecidos.

Tabela 2.10: Experimentos realizados no *TREC Enterprise Track*

Experimento	2005	2006	2007	2008
Recuperação de especialistas	X	X	X	X
Recuperação de itens conhecidos	X			
Recuperação de documentos com pontos de vista distintos	X	X		
Recuperação de documentos			X	X

- Situações onde as pessoas procuram por e-mails que expressam pontos de vista distintos sobre o mesmo assunto.
- Situações onde as pessoas procuram por pessoas, consideradas especialistas em determinado assunto.
- Situações onde as pessoas procuram por páginas Web que são referência para determinado tópico.

3 Recuperação de Informações em Empresas

A tarefa de recuperação de informação envolve identificar uma necessidade de informação, formular uma consulta, recuperar informação, avaliar a informação recuperada e aplicar esta informação. A recuperação de informação dentro de uma organização, ou de um ambiente colaborativo de trabalho, envolve estas mesmas atividades, mas também inclui: comunicar a necessidade da informação entre os participantes da equipe, compartilhar a informação recuperada com a equipe e dividir as tarefas de recuperação de informação entre os participantes da equipe (POLTROCK et al., 2003). Na figura 3.1 é apresentado um ciclo que ilustra a relação entre estas atividades.

Figura 3.1: Ciclo de acesso à informação

Assim como um indivíduo, uma equipe deve reconhecer suas necessidades de informação, formular consultas, recuperar informação, avaliar a informação recuperada e utilizar os resultados na solução dos seus problemas. Cada uma destas atividades pode ser realizada colaborativamente ou individualmente (FAGIN et al., 2003).

A necessidade de informação é identificada individualmente enquanto o integrante da equipe trabalha sozinho na execução de uma tarefa, e comunicada a equipe de ma-

neira virtual (i.e., através de e-mail) ou pessoalmente, quando necessário. Os membros das equipes também levantam questões em reuniões ou em conversas não formais (HERTZUM; PEJTERSEN, 2000).

A elaboração de uma consulta acontece de maneira individual e colaborativa. Geralmente, a construção de uma consulta acontece de maneira colaborativa quando o assunto é menos especializado. Quando o assunto é mais técnico, a elaboração da consulta acontece individualmente. Por exemplo, o líder de uma equipe de desenvolvimento de software ao fazer uma apresentação para gerentes, solicita que todos os membros da sua equipe comentem a sua apresentação (HANSEN; JARVELIN, 2005).

As fontes de informação mais comuns são documentos e as pessoas próximas da equipe de desenvolvimento. Os limites de uma equipe geralmente são nebulosos, uma equipe nunca é totalmente fechada (MUKHERJEE; MAO, 2004; FREUND; TOMS; CLARKE, 2005). A recuperação de informação realizada por uma equipe pequena pode estender-se por diversos membros de diversas equipes, com funções e atividades diferentes.

As necessidades de informação geralmente são identificadas e analisadas em reuniões. As equipes coletivamente decidem como resolver uma necessidade de informação. Uma estratégia muito utilizada para solucionar uma necessidade de informação é convidar alguém para uma próxima reunião para falar sobre o tema. A reunião é o local onde, concorrentemente, os membros da equipe podem adquirir informação, disseminar a informação entre a equipe, explorar novos desafios, avaliar a informação adquirida e determinar como aplicá-la (POLTROCK et al., 2003; FREUND; TOMS, 2006).

As atividades de recuperação de informação não são atividades separadas das outras atividades realizadas pela equipe. Não são atividades com início, meio e fim bem definidos. As atividades de recuperação de informação estão inseridas dentro das outras atividades realizadas pela equipe. São atividades realizadas individualmente, de uma maneira informal ou em reuniões (FELDMAN; SHERMAN, 2003; GREIF, 1994).

Sistemas de Recuperação de Informação estão inseridos nesta dinâmica através da etapa “*Recuperar Informação*”, como apresentado na figura 3.1. O objetivo de um Sistema de Recuperação de Informação Empresarial é encontrar informações que possam satisfazer uma necessidade de informação de um determinado usuário ou equipe para completar uma determinada tarefa.

Na próxima seção são apresentados alguns trabalhos que propõe modelos para a recuperação de pessoas, chamadas especialistas, e documentos em um ambiente empresarial.

3.1 Estratégias para a recuperação de informação em empresas

Como dito anteriormente, as fontes de informação mais comuns no domínio empresarial são documentos e pessoas (POLTROCK et al., 2003; HANSEN; JARVELIN, 2005). Sendo assim, dado uma consulta, que é uma representação da necessidade de informação do indivíduo ou da equipe, um Sistema de Recuperação de Informação Empresarial deve retornar referências a todos os documentos e pessoas que possam satisfazer esta necessidade de informação.

A maioria dos Sistemas de Recuperação de Informação disponíveis no mercado não estão preparados para o domínio empresarial. Pois, as suas implementações não levam em consideração algumas características deste domínio, por exemplo (MUKHERJEE; MAO, 2004):

- Normalmente, um grande número de consultas realizadas em uma organização têm um conjunto pequeno de documentos considerados relevantes.
- A resposta correta para uma consulta não necessariamente possui os documentos mais populares, conceito normalmente utilizado para definir a resposta de uma consulta na Internet, por exemplo.
- Muitas das consultas, entre 60% e 80%, feitas dentro de um ambiente empresarial têm como objetivo recuperar algo que o usuário já viu. O usuário lembra de algum atributo do documento e utiliza este atributo como termo em uma consulta.

Devido a grande complexidade do espaço de informações de uma organização, os algoritmos desenvolvidos dentro dos laboratórios de pesquisa não podem ser inseridos em organizações sem um extenso trabalho de engenharia (HAWKING, 2004; DANILOWICZ; NGUYEN, 2002). Em uma organização a maioria das informações está armazenada de forma não estruturada. Estas informações não estruturadas podem estar em páginas HTML ou em qualquer tipo de formato proprietário. Todas as informações não estruturadas mais as informações armazenadas em banco de dados formam o ecossistema de informação de uma organização (MUKHERJEE; MAO, 2004).

Os sistemas disponíveis no mercado não conseguem tratar ao mesmo tempo a diversidade de informações que é encontrada em uma organização: sistemas de gestão de conteúdo, e-mails, documentos do tipo texto, planilhas, lista de contatos, entre outros. Nem todas as informações são armazenadas e indexadas (HAWKING, 2004; DANILOWICZ; NGUYEN, 2002). Ou seja, nem toda informação necessária para uma tarefa

está disponível como um conteúdo indexado. Mesmo se uma organização tem acesso a um grande conjunto de informações relevantes, podem existir casos onde a informação não estará disponível, por exemplo: por questões de segurança ou por questões de direito autoral (MUKHERJEE; MAO, 2004). Em instituições de grande porte, diferentes departamentos têm depósitos isolados de informações que não estão disponíveis para outros departamentos da mesma organização devido a restrições meramente técnicas (HAWKING, 2004).

Além disso, as premissas da área de Recuperação de Informação e os modelos que regem a implementação da maioria dos Sistemas de Recuperação de Informação não consideram a possibilidade da recuperação de referências para pessoas. Sendo que, na maioria das situações, a única forma para identificar pessoas relevantes (especialistas) para uma determinada necessidade de informação é através da análise dos documentos disponíveis (HANSEN; JARVELIN, 2005; BALOG, 2008).

Nas próximas seções são apresentados trabalhos que propõem modelos para a recuperação de pessoas e documentos dentro do contexto empresarial.

3.1.1 Recuperação de referências de pessoas

Um dos maiores desafios em organizações comerciais, educacionais ou governamentais é administrar o conhecimento dos seus colaboradores, por exemplo, identificando especialistas para uma determinada necessidade de informação ou tarefa. Encontrar uma pessoa com as habilidades e conhecimentos necessários para executar uma tarefa em uma organização é, geralmente, crucial para o sucesso de um projeto em andamento, por exemplo (MOCKUS; HERBSLEB, 2002).

Reusar conhecimento está fortemente relacionado a explorar melhor o capital humano existente dentro da equipe ou organização, de modo a evitar gastos de tempo e dinheiro na aquisição do conhecimento cada vez que ele for necessário. Porém, o conhecimento nem sempre é algo facilmente acessível, visto que ele pode estar presente apenas como conhecimento tácito entre os colaboradores de uma organização (REICHLING; SCHUBERT; WULF, 2005).

Um sistema que perceba indivíduos que são potenciais colaboradores e os coloque em contato trará um grande benefício ao trabalho desenvolvido por ambos. Afinal, a comunicação direta torna mais fácil e efetivo o compartilhamento dos conhecimentos tácitos e habilidades entre os indivíduos (MOCKUS; HERBSLEB, 2002; BALOG; AZZOPARDI; RIJKE, 2006).

Um sistema de recuperação de especialistas é um sistema que retorna referências

para atores humanos que são identificados como “especialistas” no domínio requisitado (BALOG; AZZOPARDI; RIJKE, 2006).

Abordagens iniciais para o desenvolvimento de sistemas de recuperação de especialistas fazem uso de um sistema para cadastramento do perfil dos colaboradores em um banco de dados (MCDONALD; ACKERMAN, 2000). Este banco de dados é manualmente construído e consultado. Tais abordagens requerem um considerável esforço para a criação e manutenção dos dados. Conseqüentemente, outras iniciativas foram desenvolvidas para automaticamente construir os perfis dos colaboradores a partir de repositórios da organização. A maioria destas abordagens é focada na procura por especialistas em um determinado tópico, com base em documentos bem conhecidos deste tópico (MOCKUS; HERBSLEB, 2002). Outras abordagens, mais recentes, realizam a extração automática do perfil dos colaboradores a partir de uma coleção de documentos heterogêneos, por exemplo, documentos encontrados em uma Intranet de uma organização (BALOG; AZZOPARDI; RIJKE, 2006).

As principais abordagens para recuperação de especialistas podem ser divididas em duas categorias: recuperação a partir do perfil do especialista, e; modelo de dois estágios.

3.1.1.1 Recuperação a partir do perfil do especialista

Nesta abordagem é construído um perfil para cada pessoa da organização. Um perfil é composto por uma lista de palavras com as suas respectivas frequências. Esta lista de palavras é gerada a partir da concatenação do conteúdo encontrado nos documentos ou em janelas que possuem uma referência a pessoa. Geralmente, uma lista com os nomes e e-mails dos especialistas é fornecida para identificar uma citação ao candidato à especialista. No entanto, variantes dos nomes dos especialistas precisam ser consideradas para melhorar o índice de recuperação dos especialistas.

Uma janela é uma seqüência de palavras delimitadas: pelo número de palavras; por meta-informações, tais como as *tags* de páginas HTML, ou; por estruturas semânticas do texto, tais como sentenças e parágrafos.

Nesta abordagem, o Sistema de Recuperação de Informação procura pelos termos da busca nos arquivos dos perfis dos usuários, retornando os perfis mais relevantes para a consulta feita. Neste caso, qualquer função de ordenação pode ser utilizada.

Um exemplo desta abordagem é o trabalho desenvolvido por Fu et al. (2005). Neste trabalho é utilizada uma lista de e-mails pré-determinada para identificar os especialistas nos documentos. Para cada especialista é criado um arquivo que é formado

a partir do conteúdo de todos os documentos onde a referência para o especialista foi encontrada. A recuperação de um especialista para um determinado tópico é realizada procurando-se pelos termos de uma consulta nos arquivos dos especialistas. O trabalho desenvolvido por Fu et al. (2005) foi o trabalho que teve o melhor resultado no TREC de 2005 na tarefa de recuperação de especialistas (ver tabela 3.1).

Tabela 3.1: Melhores resultados da tarefa de recuperação de especialistas do TREC 2005. Os resultados estão ordenados de acordo com o valor do MAP (CRASWELL; VRIES; SOBOROFF, 2005).

Algoritmos	MAP	P@5	P@10	P@20	P@30	P@100	P@1000
Fu et al. (2005)	0.2749	0.4880	0.4520	0.3390	0.2800	0.1142	0.0114
Cao et al. (2005), v. 54	0.2688	0.4080	0.3700	0.3190	0.2753	0.1306	0.0131
Cao et al. (2005), v. 55	0.2600	0.3920	0.3580	0.3150	0.2733	0.1308	0.0131

Os outros algoritmos apresentados na tabela 3.1 utilizam uma abordagem baseada em um modelo de dois estágios. No ano de 2006, os três algoritmos que obtiveram os melhores resultados foram algoritmos que utilizaram um modelo de dois estágios (tabela 3.2). Estes algoritmos serão descritos na próxima seção.

Tabela 3.2: Melhores resultados da tarefa de recuperação de especialistas do TREC 2006. Os resultados estão ordenados de acordo com o valor do MAP. Os algoritmos em *itálico* são algoritmos que tem alguma intervenção humana durante o processo (SOBOROFF; VRIES; CRASWELL, 2006).

Algoritmos	MAP	P@5	P@10	P@20
<i>Zhu et al. (2006)</i>	0.6431	0.8245	0.7347	0.6031
Bao et al. (2006)	0.5947	0.7673	0.7041	0.5694
<i>You et al. (2006)</i>	0.5639	0.7224	0.6551	0.5469
Ru et al. (2006)	0.5564	0.7592	0.6653	0.5459

Na abordagem apresentada por Ru et al. (2006) os candidatos a especialista são ordenados pela relevância dos arquivos que descrevem os candidatos. Cada arquivo de descrição é construído com as palavras que co-ocorrem com o candidato dentro de uma mesma janela de texto em um documento, com tamanho igual a 150 palavras.

Neste mesmo trabalho foi testado a expansão da consulta utilizando os termos encontrados na descrição do tópico. Os resultados encontrados sem a expansão da consulta e os resultados encontrados com a expansão da consulta foram os mesmos.

No ano de 2007 os dois trabalhos com melhores resultados foram apresentados por (FU et al., 2007) e (SHEN et al., 2007). Fu et al. (2007) apresenta uma continuação da abordagem apresentada em (FU et al., 2005), onde para cada candidato à especialista é criado um perfil. No entanto, Shen et al. (2007) apresenta uma alternativa baseada no

modelo de dois estágios. Na tabela 3.3 é possível verificar que os resultados encontrados não têm muita diferença.

Tabela 3.3: Melhores resultados da tarefa de recuperação de especialistas do TREC 2007. Os resultados estão ordenados de acordo com o valor do MAP (BAILEY et al., 2007b).

Algoritmos	MAP	P@5	P@20
Fu et al. (2007)	0.4632	0.2280	0.0910
Shen et al. (2007)	0.4427	0.2360	0.0910

Fu et al. (2007) concentra-se em especial na identificação dos nomes dos especialistas. No TREC 2007 não foi fornecido nenhuma lista de possíveis especialistas. As equipes que participaram do TREC 2007 tiveram que criar mecanismos para identificar nomes e sinônimos nos documentos da coleção.

Para cada candidato a especialista Fu et al. (2007) cria o que é chamado de documento de descrição da pessoa (PDD¹). Para a construção do documento de descrição da pessoa é utilizado o contexto onde uma referência à pessoa é encontrada nos documentos, ou seja, uma janela de palavras. Além disso, são utilizados documentos que são as páginas pessoais dos colaboradores da empresa. Fu et al. (2007) encontrou 477 páginas pessoais, que significa 15.2% do total de candidatos à especialista.

3.1.1.2 Modelo de dois estágios

A maioria dos trabalhos sobre recuperação de especialistas utiliza uma abordagem composta por duas partes: um modelo de relevância, e; um modelo de co-ocorrência. O modelo de relevância representa se um documento é ou não relevante para uma consulta. O modelo de co-ocorrência representa se uma consulta (se os termos de uma consulta) é associada com uma pessoa ou não (BALOG, 2008).

Em Cao et al. (2005) e Bao et al. (2006) o modelo de dois estágios é definido da seguinte maneira:

$$P(e|q) = \sum_d P(d|q)P(e|d, q) \quad (3.1)$$

onde:

- $P(d|q)$ representa o modelo de relevância.
- $P(e|d, q)$ representa o modelo de co-ocorrência.

¹Do inglês, *Person Description Document*

- d representa um documento.
- q significa uma consulta.
- e representa um especialista (pessoa).

O modelo de co-ocorrência é estimado da seguinte maneira:

$$P(e|d, q) = \mu \frac{pf(e, d)}{|d|} \quad (3.2)$$

onde:

- $pf(e, d)$ é a frequência da pessoa e no documento d que satisfaz q .
- $|d|$ é a frequência total de pessoas em d .
- μ é uma forma de suavização que é calculado da seguinte maneira:

$$\mu = \frac{|d|}{|d| + \kappa} \quad (3.3)$$

- κ é o tamanho médio da frequência de pessoas nos documentos da coleção.

Em Cao et al. (2005) e Bao et al. (2006) o modelo de relevância é implementado utilizando a função BM25, como definido na equação 2.7. O que é variável, e testado, em ambos os trabalhos é a utilização de diferentes janelas de texto no modelo de co-ocorrência. As janelas de texto consideradas são as mesmas definidas anteriormente: tamanho de palavras, blocos semânticos e blocos gerados a partir de metadados.

Em (CAO et al., 2005) os melhores resultados foram obtidos com janelas de 50 palavras. Em Bao et al. (2006) os melhores resultados foram obtidos utilizando as *tags* <TITLE>, <H1> e <H2> ao criar as janelas. Segundo Zhu et al. (2006), ao aumentar o tamanho da janela aumenta-se o número de associações entre pessoas e termos da consulta. No entanto, um ruído maior ao modelo é introduzido.

Em uma Intranet de uma organização alguns documentos são mais populares que outros. Segundo Zhu et al. (2006), esta relação de popularidade pode auxiliar na identificação de especialistas em um tópico. Por isso, em Zhu et al. (2006) o modelo de relevância $P(d|q)$ é alterado como visto na equação 3.4

$$P(d|q) = P_{content}(d|q) + w \times \frac{1}{PageRank(d|q)} \quad (3.4)$$

onde $P_{content}(d|q)$ é o resultado da função de ordenação BM25, w é o peso dado a popularidade do documento e $PageRank(d|q)$ é a autoridade dada para o documento d através do uso do algoritmo PAGERANK.

Em (YOU et al., 2006) o modelo de relevância implementado é a função BM25 e o modelo de co-ocorrência utiliza é a concatenação de várias janelas baseadas em meta-informações dos documentos: janela do título do documento (<TITLE>), janela do cabeçalho do documento (<HEAD>) e janela de bloco do documento (<H1> e <H2>).

Em (SHEN et al., 2007) o modelo de relevância utiliza a função BM25 e o modelo de co-ocorrência é o número de ocorrências do especialista no documento ($pf(e, d)$). Apesar da simplicidade, os resultados alcançados por (SHEN et al., 2007) não são muito diferentes dos resultados alcançados por (FU et al., 2007) na tarefa de recuperação de especialistas do TREC 2007 (tabela 3.3).

3.1.2 Recuperação de documentos

Vários autores colocam que o objetivo de um Sistema de Recuperação de Informação para o domínio empresarial é satisfazer a necessidade de informação de um usuário para que ele possa completar uma tarefa (BAILEY et al., 2007b; BALOG, 2008; FREUND; TOMS; CLARKE, 2005; HERTZUM; PEJTERSEN, 2000).

No TREC em 2005 (CRASWELL; VRIES; SOBOROFF, 2005), 2006 (SOBOROFF; VRIES; CRASWELL, 2006) e 2007 (BAILEY et al., 2007b, 2007a) criaram-se coleções para avaliar situações onde o usuário procura por itens conhecidos, procura por documentos com ponto de vista distintos sobre determinado tópico e procura por documentos que forneçam uma visão geral sobre determinado tópico. A tabela 2.10 sumariza esta informação.

Em 2005 os trabalhos que obtiveram os melhores resultados na tarefa onde o objetivo é procurar por itens conhecidos foram os trabalhos (MACDONALD et al., 2005), (CRASWELL; ZARAGOZA; RODERTSON, 2005) e (OGILVIE; CALLAN, 2005) (ver tabela 3.4). Nesta tarefa, os únicos documentos utilizados foram e-mails da coleção W3C. Todos os grupos fizeram uso dos metadados dos e-mails. Entretanto, cada grupo utilizou uma abordagem diferente para fazer uso dos metadados e procurar por itens conhecidos neste conjunto de e-mails (CRASWELL; VRIES; SOBOROFF, 2005).

Craswell, Zaragoza e Rodertson (2005) utilizam uma função de ordenação baseada na função BM25, chamada BM25F, que considera pesos diferentes para os campos ou fragmentos de um e-mail. Primeiro, para calcular o valor de BM25F, é definida a frequência normalizada de um termo para cada campo do e-mail:

Tabela 3.4: Melhores resultados obtidos na tarefa para recuperação de itens conhecidos (CRASWELL; VRIES; SOBOROFF, 2005)

Algoritmos	MRR	S@10
Macdonald et al. (2005)	0.621	0.784
Craswell, Zaragoza e Rodertson (2005)	0.613	0.816
Ogilvie e Callan (2005)	0.601	0.816

$$\bar{x}_{d,f,t} = \frac{x_{d,f,t}}{(1 + B_f(\frac{l_{d,f}}{l_f} - 1))} \quad (3.5)$$

onde:

- f significa um fragmento do e-mail e pode assumir um dos seguintes valores *SUBJECT*, *BODY* ou *QUOTED*. *SUBJECT* é o título do e-mail, *BODY* é o corpo do e-mail e *QUOTED* é um trecho de um e-mail anterior que foi utilizado no e-mail analisado. A utilização de um trecho *QUOTED* é muito comum quando um e-mail é utilizado para responder outro e-mail ou quando um e-mail é re-encaminhado (opção *forward*).
- $x_{d,f,t}$ é a frequência do termo t no campo f do documento d .
- $l_{d,f}$ é o tamanho do campo f no documento d .
- l_f é o tamanho médio do campo f para todos os documentos da coleção.
- B_f é um parâmetro dependente de campo similar ao parâmetro b da equação *BM25* (equação número 2.7). B_f pode variar de 0 à 1. Se $B_f = 0$ então o valor de $\bar{x}_{d,f,t}$ não é normalizado.

As frequências calculadas a partir da equação 3.5 podem ser combinadas para obter um valor final da frequência de um termo t em um documento d :

$$\bar{x}_{d,t} = \sum_f W_f \times \bar{x}_{d,f,t} \quad (3.6)$$

onde W_f é um valor parametrizável que significa o peso do campo ou do fragmento do e-mail.

A função de ordenação *BM25F* é definida da seguinte maneira:

$$BM25F(d, q) = \sum_{t \in q \cap d} \frac{\bar{x}_{d,t}}{K_1 + \bar{x}_{d,t}} \times \frac{x_{d,t}}{\max_z x_{d,z}} \quad (3.7)$$

onde K_1 é um valor parametrizável e $\frac{x_{d,t}}{\max_z x_{d,z}}$ é o valor da frequência normalizada do termo t no documento d .

Durante os experimentos realizados por Craswell, Zaragoza e Rodertson (2005) percebeu-se que os melhores resultados encontrados foram quando $K_1 = 2.0$, $W_{subject} = 20.0$, $W_{body} = 0.68$ e $W_{quoted} = 0.7$. Ou seja, os experimentos mostram que o campo *SUBJECT* tem um peso maior que os campos *BODY* e *QUOTED*. Além disso, os resultados apresentados por (CRASWELL; ZARAGOZA; RODERTSON, 2005) mostram que ao utilizar pesos uniformes para os campos os valores para MRR encontrados são próximos de 0.45, enquanto que os melhores valores para MRR encontrados são próximos de 0.6.

Ogilvie e Callan (2005) também combinam evidências de vários campos de um e-mail. Em (OGILVIE; CALLAN, 2005), as mensagens de e-mail são ordenadas através da probabilidade de uma mensagem de e-mail possuir os termos t_i de uma consulta q . Neste trabalho, uma função de ordenação é definida como:

$$P(q|\theta) = \prod_{i=1}^{|q|} P(t_i|\theta) \quad (3.8)$$

onde $P(t_i|\theta)$ é definido de tal maneira:

$$\begin{aligned} P(t_i|\theta) = & W_{text} \times P(t_i|\theta_{text}) \\ & + W_{subject} \times P(t_i|\theta_{subject}) \\ & + W_{thread} \times P(t_i|\theta_{thread}) \\ & + W_{sub-thread} \times P(t_i|\theta_{sub-thread}) \\ & + W_{collection} \times P(t_i|\theta_{collection}) \end{aligned} \quad (3.9)$$

onde W_x é o peso atribuído a parte x do e-mail e cada um dos modelos $P(t_i|\theta_x)$ são definidos da seguinte maneira:

$$P(t_i|\theta_x) = \frac{f_{t_i,x}}{l_x} \quad (3.10)$$

onde:

- $f_{t_i,x}$ é a frequência do termo t_i no campo x .
- l_x : é o tamanho do campo x , ou seja, o número de palavras que existem no campo x .

Os campos considerados são: *text* que significa o texto do e-mail; *subject* que representa o título do e-mail; *thread* que significa todo o texto de todos os e-mails da *thread*; *sub-thread* que representa todo o texto do e-mail e todos os e-mails de resposta, e; *collection* que significa todo o texto de todos os e-mails da coleção.

Os participantes do TREC de 2007, na tarefa de recuperação de documentos, submeteram 43 propostas classificadas como automáticas e 15 propostas classificadas como com realimentação. Nas propostas classificadas como automáticas as equipes participantes podem fazer uso apenas dos campos *query* e *narr* da descrição do tópico (tabela 2.9). Nas propostas classificadas como com retorno, os participantes podem fazer uso dos campos *query*, *narr* e dos documentos descritos no atributo *page* (BAILEY et al., 2007b; SHEN et al., 2007).

Entre as propostas submetidas como automáticas, os melhores resultados foram apresentados por (SHEN et al., 2007), (KOLLA; VECHTOMOVA, 2007) e (FAN; HUANG, 2007). Na tabela 3.1.2 é possível visualizar os resultados encontrados utilizando as medidas MAP, NDCG e P@20.

Tabela 3.5: Melhores resultados na tarefa de recuperação de documentos do TREC 2007 na categoria de propostas automáticas.

Algoritmos	MAP	NDCG	P@20
(SHEN et al., 2007)	0.422	0.743	0.527
(FAN; HUANG, 2007)	0.416	0.730	0.513
(KOLLA; VECHTOMOVA, 2007)	0.388	0.707	0.508

Shen et al. (2007) submeteram três propostas no TREC 2007. A primeira e a segunda propostas são baseadas na função BM25. A única diferença é o conteúdo da consulta. Na primeira proposta a consulta é formada apenas pelo conteúdo do campo *query*, enquanto que na segunda proposta o conteúdo do campo *query* é expandido utilizando o conteúdo do campo *narr*. O processo de expansão de consultas remove todas as *stop-words* do campo *narr*, adicionando o que restou a consulta.

A terceira proposta é baseada no algoritmo PAGERANK. O algoritmo PAGERANK é utilizado para reordenar os resultados encontrados pela segunda proposta. São escolhidos os 200 primeiros documentos retornados para a construção de um grafo, onde os nodos são os documentos (páginas web) e as arestas são os hyperlinks entre os documentos. Shen et al. (2007) aplicam o algoritmo PAGERANK para reordenar os documentos e apresentar uma nova lista de documentos para cada tópico.

Os resultados encontrados por (SHEN et al., 2007) são apresentados na tabela 3.6. A partir destes resultados é possível ver que o uso do algoritmo PAGERANK não melho-

rou o desempenho da solução. O algoritmo que obteve o melhor desempenho, como visto na tabela, foi a proposta que utiliza a função de ordenação BM25 com expansão de consultas.

Tabela 3.6: Resultados dos experimentos realizados por Shen et al. (2007) na tarefa de recuperação de documentos no TREC 2007

Algoritmos	MAP	MRR
(SHEN et al., 2007), proposta 1	0.3970	0.8384
(SHEN et al., 2007), proposta 2	0.4048	0.8158
(SHEN et al., 2007), proposta 3	0.1746	0.3493

Fan e Huang (2007) focaram o seu trabalho no pré-processamento dos dados e na expansão de consultas. Para o pré-processamento dos dados foram utilizadas algumas heurísticas para a filtragem de documentos. O objetivo da aplicação das heurísticas era eliminar documentos considerados irrelevantes para uma tarefa onde o objetivo é procurar por documentos que forneçam uma visão geral sobre determinado tópico. Na maioria dos casos, as heurísticas estavam associadas ao formato dos documentos, por exemplo: documentos que tinham apenas um link para outro documento não poderiam ser considerados documentos que fornecem uma visão sobre determinado tópico, portanto, estes documentos eram eliminados do índice.

Os trabalhos (FAN; HUANG, 2007) e (KOLLA; VECHTOMOVA, 2007) também fizeram uso do algoritmo BM25 com expansão de consultas. Ambos os trabalhos utilizaram o conteúdo do atributo *narr* para expandir as consultas e ambos os trabalhos relataram uma melhora significativa de desempenho ao usar abordagens com expansão de consultas.

3.2 Considerações sobre Recuperação de Informações em Empresas

Os problemas relacionados com a recuperação de informação em ambientes empresariais podem ser divididos em dois grupos:

- **Problemas de Engenharia:** são os problemas relacionados com: integração e interoperabilidade de dados e sistemas; definição de processos formais para o acesso à informação certificada, e; aquisição e armazenamento de informação, ou seja, transformação de conhecimento tácito em explícito.
- **Problemas Conceituais:** são os problemas relacionados com a concepção de modelos de recuperação de informação adequados para o ambiente empresarial

- entender as necessidades de informação dos usuários e entender as características das fontes de informação de uma organização para então desenvolver modelos e algoritmos apropriados para a recuperação de informação em empresas. Estes problemas visam responder algumas perguntas, entre elas: (i) Quais são as características das fontes de informação existentes em ambientes empresariais? (ii) Como utilizar estas características para aprimorar os modelos e os sistemas de recuperação de informação aplicáveis neste domínio? (iii) O que caracteriza uma função de ordenação eficiente para o ambiente empresarial?

Os problemas conceituais podem ser divididos em duas sub-classes:

- **Recuperação de Especialistas:** os trabalhos que atuam nesta classe de problema estão preocupados em criar funções de ordenação eficientes para a recuperação de referências de especialistas.
- **Recuperação de Documentos:** os trabalhos que atuam nesta classe de problema estão preocupados em criar funções de ordenação eficientes para a recuperação de documentos em um ambiente empresarial.

Este capítulo descreveu algumas das iniciativas para a recuperação de especialistas e documentos em ambientes empresariais. A maioria das referências utilizadas para esta revisão de literatura foram obtidas a partir de dois grandes fóruns da área: *Text Retrieval Conference (TREC)*² e *ACM SIGIR Conference on Research and Development in Information Retrieval*³, entre os anos de 2003 e 2008. A partir desta revisão foi possível sintetizar algumas das características das soluções adotadas.

Foi visto que as principais abordagens para a recuperação de especialistas podem ser divididas em duas categorias: recuperação a partir do perfil do especialistas, e; modelo de dois estágios.

Na abordagem baseada no perfil do especialista, o Sistema de Recuperação de Informação procura pelos termos da busca nos arquivos dos perfis dos usuários. A forma para construção do perfil que retornou os melhores resultados foi utilizando o janelamento pelo número de palavras. As funções de ordenação utilizadas foram, basicamente, formadas pela função de ordenação BM25 com expansão de consultas e sem expansão de consultas. Os resultados encontrados sem a expansão da consulta e os resultados encontrados com a expansão da consulta não apresentaram diferenças significativas.

²<http://trec.nist.gov/>

³<http://www.sigir.org/>

A maioria dos trabalhos sobre recuperação de especialistas utilizam uma abordagem composta por duas partes, chamada de modelo de dois estágios. As partes deste modelo são: um modelo de relevância, e; um modelo de co-ocorrência. Geralmente, para o modelo de relevância são utilizadas ou a função BM25 ou a função BM25 com PAGERANK (equação 3.4). Para o modelo de co-ocorrência são testadas diferentes janelas de texto. As janelas de texto consideradas são as mesmas utilizadas nas soluções baseadas no perfil do especialista: tamanho de palavras, blocos semânticos e blocos gerados a partir de metadados. As funções utilizadas para medir a co-ocorrência variam desde funções mais elaboradas, como a fórmula apresentada na equação 3.2, até simples funções que apenas calculam a frequência de um especialista em um documento.

Os resultados encontrados até o momento não são conclusivos. Não sabe-se dizer se o uso da função BM25 de forma isolada é melhor que o uso da função BM25 com a função PAGERANK para a construção do modelo de relevância. Não se tem indícios suficientes para afirmar qual é a melhor equação para o modelo de co-ocorrência, nem que tipo de janela deve-se utilizar. Apesar do modelo de dois estágios ser mais utilizado que a abordagem baseada no perfil do especialista, também não pode-se afirmar que o modelo de dois estágios é melhor que a abordagem baseada no perfil do especialista.

Na tarefa de recuperação de documentos diversas equipes experimentaram métodos de Recuperação de Informação utilizados no ambiente Web, por exemplo o algoritmo PAGERANK. Os resultados encontrados indicam que o comportamento do domínio empresarial é diferente do ambiente Web. Os métodos geralmente utilizados com sucesso no ambiente Web tiveram um desempenho abaixo do esperado nas coleções CERC e W3C.

Estes resultados só vem a colaborar com a afirmação colocada por Mukherjee e Mao (2004): *"a resposta correta para uma consulta em um ambiente empresarial não necessariamente possui os documentos mais populares"*.

De forma geral, os trabalhos utilizados na tarefa de recuperação de documentos que utilizaram algum método de expansão de consultas reportaram algum tipo de melhora nos resultados.

O uso de funções de ordenação que consideram pesos diferentes para os campos ou fragmentos de um e-mail foi a abordagem mais utilizada para a atividade onde o objetivo era procurar por e-mails conhecidos. Os experimentos realizados mostraram que o campo SUBJECT é o que representa melhor o conteúdo do e-mail para esta tarefa.

4 Proposta e Implementação

Este trabalho avalia o desempenho do uso árvores de decisão como função de ordenação para documentos e pessoas em ambientes empresariais. Para tanto, este trabalho propõe construir dois tipos de funções de ordenação para o ambiente empresarial:

- Árvores de decisão capazes de ordenar documentos para uma dada consulta, e;
- Árvores de decisão capazes de ordenar referências de pessoas para uma dada consulta.

Ambas as árvores de decisão serão construídas utilizando o algoritmo C4.5 (QUINLAN, 1996). O algoritmo C4.5 é um algoritmo de aprendizagem de máquina simbólico capaz de gerar uma descrição de item relevante, na forma de árvore de decisão, a partir das informações contidas em um conjunto de treinamento.

No contexto deste trabalho, o objetivo principal desta descrição é tomar o lugar da função de ordenação de um Sistema de Recuperação de Informação. Ou seja, dada uma nova consulta, a descrição gerada pelo algoritmo de aprendizagem de máquina deve fornecer a relevância de cada documento e pessoa armazenados no índice do Sistema de Recuperação de Informação.

Além disso, espera-se que a representação na forma de árvore de decisão possa contribuir para a compreensão da razão da utilização de determinados atributos na composição da função de ordenação e explicar porquê um documento ou pessoa foi considerado relevante ou não.

Nas próximas seções são apresentados os atributos utilizados na caracterização de documentos e pessoas, incluindo as justificativas para as decisões tomadas. Na última seção deste capítulo são apresentados os detalhes de implementação do algoritmo C4.5.

4.1 Caracterização dos documentos

Para a caracterização dos documentos, os mesmos precisam ser transformados em uma representação lógica qualquer. Uma forma muito utilizada é transformar um documento (figura 4.1) em um vetor de palavras (equação 4.1).

... Esta disciplina tem como objetivo apresentar os principais conceitos da área de Inteligência Artificial, caracterizar as principais técnicas e métodos, e implementar alguns problemas clássicos desta área sob um ponto de vista introdutório.

A estratégia de trabalho, o conteúdo ministrado e a forma dependerão dos projetos selecionados pelos alunos. Inicialmente, os alunos deverão trazer os seus Projetos de Conclusão de Curso, identificar intersecções entre o projeto e a disciplina, e propor atividades para a disciplina. ...

Figura 4.1: Exemplo de um documento

$$\vec{d}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,n}) \quad (4.1)$$

Na equação 4.1 os termos $p_{i,j}$ são as palavras encontradas nos documentos da coleção. O valor para cada termo pode ser calculado de diversas maneiras, por exemplo:

- *Booleana* - se a palavra aparece ou não no documento (1 ou 0).
- *Por frequência do termo* - a frequência com que a palavra aparece no documento (normalizada ou não).
- *Ponderação tf-idf* - o peso é proporcional ao número de ocorrências do termo no documento e inversamente proporcional ao número de documentos onde o termo aparece (equação 2.2).

Em todos os idiomas existem átomos (palavras) que não significam muito e aparecem em quase todos os documentos com uma frequência alta. Estas palavras são chamadas de *stop words* e é aconselhável que sejam eliminadas da representação dos documentos. Esta eliminação acontece através de um filtro de *stop words* (MANNING; SCHÜTZE, 2003). Um exemplo de palavras filtradas neste processo é apresentado na figura 4.2.

disciplina objetivo apresentar principais conceitos área Inteligência Artificial caracterizar principais técnicas métodos implementar alguns problemas clássicos área ponto vista introdutório

Figura 4.2: Texto do exemplo da figura 4.1 após filtro de *stop words*

Outra questão importante no tratamento preliminar dos textos é a decisão sobre reduzir ou não as diferentes formas de palavras em radicais (por exemplo, *analisar*, *análise*, *Análises*, *analizou*, *analizando*, seriam todos representados pelo radical *ana-lis*).

Assim sendo, um processo completo de transformação de documentos em uma representação como apresentada na equação 4.1 envolve os seguintes passos: filtro de *stop words*, algoritmo de *stemming* (PORTER, 1997) e cálculo do peso da palavra no documento (por exemplo, ponderação tf-idf). O resultado deste processo sobre uma coleção de documentos é um índice na forma de uma tabela como apresentado na tabela 4.1.

Documento	apresent	form	tecnic	caracteriz	...	problem
d_1	0.33	0.33	0.33	0.33	...	0.4
d_2	0	0.5	0.2	0.33	...	0.2
d_3	1	0.6	0	0	...	0.1
d_4	0.4	0.3	0.33	0.4	...	0.66
d_5	...	...	...	...	...	...
d_n	1	0.4	0.1	0.1	...	0.33

Tabela 4.1: Exemplo de uma coleção processada

Neste trabalho, em cada documento da coleção foi aplicado o filtro de *stop words*, algoritmo de *stemming* e calculado o peso da palavra no documento utilizando o valor da frequência da palavra no documento ($f_{i,j}$), que é a simples contagem das ocorrências da palavra no documento.

Os atributos escolhidos para caracterizar os documentos foram identificados a partir de duas fontes: (i) funções de ordenação ou medidas consideradas clássicas e genéricas, ou seja, aplicáveis em qualquer domínio, e; (ii) funções de ordenação ou medidas utilizadas em outras propostas para o domínio empresarial.

Os atributos utilizados para caracterizar os documentos do domínio empresarial foram:

- *Somatório de tf-idf*: levando-se em consideração que uma consulta q é formada por uma lista de termos $\{t_1, t_2, \dots, t_m\}$, o valor deste atributo para um documento d_j é a soma dos valores tf-idf dos termos presentes na consulta:

$$R_{tfidf}(q, d_j) = \sum_{t_i \in q} p_{i,j} \quad (4.2)$$

onde, $p_{i,j}$ é definido na equação 2.2.

- *Somatório da frequência normalizada dos termos da consulta*: este atributo retorna a soma da frequência normalizada dos termos da consulta q encontrados no documento d_j :

$$R_{tf_{norm}}(q, d_j) = \sum_{t_i \in q} \frac{f_{i,j}}{\max_z f_{z,j}} \quad (4.3)$$

- *Somatório da frequência não normalizada dos termos da consulta*: este atributo retorna a soma da frequência não normalizada dos termos da consulta q encontrados no documento d_j :

$$R_{tf}(q, d_j) = \sum_{t_i \in q} f_{i,j} \quad (4.4)$$

- *Cosseno do ângulo de dois vetores* ($R_{cosseno}$): este atributo retorna o valor da similaridade entre o documento d_j e uma consulta q calculando o cosseno do ângulo sobre os vetores $\vec{d}_j$ e $\vec{q}$, construídos utilizando a ponderação tf-idf. Esta função de ordenação que pertence ao modelo vetorial é descrita na equação 2.1.
- *Distância euclidiana entre dois vetores*: este atributo retorna o valor da distância entre os mesmos vetores $\vec{d}_j$ e $\vec{q}$ utilizando a fórmula da distância euclidiana. Esta função de ordenação também pertence ao modelo vetorial e é definida da seguinte forma:

$$R_{euclidiana}(q, d_j) = |\vec{q} - \vec{d}_j| = \sqrt{\sum_{i=1}^n (p_{q,i} - p_{j,i})^2} \quad (4.5)$$

- *Função de ordenação BM25* (R_{BM25}): este atributo retorna o valor que a função de ordenação BM25 encontra para o documento d_j dado a consulta q . Esta função de ordenação foi escolhida para compor o conjunto de atributos porque trata-se de um exemplo de sucesso da aplicação do modelo probabilístico. A função BM25 é definida na equação 2.7.
- *Função de ordenação BM25 com expansão da consulta* ($R_{BM25'}$): este atributo retorna o valor que a função de ordenação BM25 encontra para o documento d_j levando em consideração todos os termos $\{t_1, t_2, \dots, t_m\}$ que aparecem na consulta q , mais todos os termos que aparecem na descrição da necessidade de informação. A equação utilizada para a função BM25 é a mesma do atributo anterior, ou seja, a equação 2.7. Esta função de ordenação foi utilizada em algumas propostas para a recuperação de documentos no domínio empresarial, obtendo bons resultados.

- *Função de ordenação* PAGERANK ($R_{pagerank}$): este atributo retorna o valor que representa a autoridade do documento d_j na coleção. A forma de cálculo deste valor é apresentada na equação 2.10. A justificativa para a utilização deste atributos deve-se ao fato de alguns trabalhos na área de recuperação de documentos no domínio empresarial utilizarem o valor retornado pelo algoritmo PAGERANK na composição das funções de ordenação.

Por fim, o conjunto de treinamento é criado a partir dos julgamentos de relevância (tabela 2.2), substituindo a referência ao documento pelos valores mencionados acima. Um exemplo do resultado desta transformação pode ser visto na tabela 4.2.

Tabela 4.2: Extrato do conjunto de treinamento para o problema onde o objetivo é encontrar uma hipótese que determina se um documento é ou não relevante.

Consulta	R_{tfidf}	...	$R_{cosseno}$	R_{bm25}	$R_{bm25'}$	$R_{pagerank}$	Relevância
1	0.06761	...	0.01557	0.03836	1.84878	0.14387	2
1	0.0	...	0.0	0.0	2.02189	0.0	1
...	...	...	...	...	...	...	...
50	0.00444	...	0.00143	0.00241	1.91903	0.00175	0

4.2 Caracterização das pessoas

Na seção 3.1.1 é relatado que as principais abordagens para recuperação de especialistas podem ser divididas em duas categorias: recuperação a partir do perfil do especialista, e; modelo de dois estágios.

Na abordagem baseada no perfil do especialista o Sistema de Recuperação de Informação procura pelos termos da busca nos arquivos dos perfis das pessoas. Um perfil é composto por uma lista de palavras com as suas respectivas frequências. Esta lista de palavras é gerada a partir da concatenação do conteúdo encontrado nos documentos ou em janelas que possuem uma referência de pessoa.

Para identificar uma referência a uma pessoa em um documento, são utilizados diversos métodos: uso de expressão regular para identificação de e-mails, uso de reconhecedores de entidades nomeadas (FREITAG; MCCALLUM, 2000) para identificação de nomes e listas descrevendo a relação entre o nome de uma pessoa e os seus e-mails.

Neste trabalho, uma lista com os nomes e e-mails dos especialistas será utilizada para identificar uma citação ao candidato à especialista. Existirão dois tipos de perfis:

- Um composto pela concatenação do conteúdo integral de todos os documentos onde aparece uma referência ao candidato à especialista.

- Um composto pela concatenação do conteúdo dos parágrafos onde aparece uma referência ao candidato à especialista.

As funções de ordenação utilizadas por trabalhos que optam por ordenar candidatos a partir de perfis de usuários são, basicamente, formadas pela função de ordenação BM25 com expansão de consultas e sem expansão de consultas. Sendo que, a função de ordenação BM25 com expansão de consultas leva em consideração todos os termos que aparecem na consulta e todos os termos que aparecem na descrição da necessidade de informação, enquanto que, na função de ordenação BM25 sem expansão de consultas apenas os termos da consulta são considerados.

As combinações possíveis, vindas da abordagem baseada no perfil de usuário, para criação de atributos são:

- Utilização da função de ordenação BM25 sem expansão de consultas para a ordenação de perfis gerados a partir do conteúdo completo de todos os documentos onde aparece uma referência ao candidato à especialista ($BM25_1$).
- Utilização da função de ordenação BM25 com expansão de consultas para a ordenação de perfis gerados a partir do conteúdo completo de todos os documentos onde aparece uma referência ao candidato à especialista ($BM25_2$).
- Utilização da função de ordenação BM25 sem expansão de consultas para a ordenação de perfis gerados a partir do conteúdo dos parágrafos onde aparece uma referência ao candidato à especialista ($BM25_3$).
- Utilização da função de ordenação BM25 com expansão de consultas para a ordenação de perfis gerados a partir do conteúdo dos parágrafos onde aparece uma referência ao candidato à especialista ($BM25_4$).

Na tabela 4.3 é apresentado um resumo sobre a composição dos atributos $BM25_1$, $BM25_2$, $BM25_3$ e $BM25_4$.

Tabela 4.3: Resumo sobre a composição dos atributos $\{BM25_1, \dots, BM25_4\}$

	$BM25_1$	$BM25_2$	$BM25_3$	$BM25_4$
BM25 sem expansão	•		•	
BM25 com expansão		•		•
Documento completo	•	•	•	•
Apenas os parágrafos			•	•

No modelo de dois estágios, o modelo de relevância representa se um documento é ou não relevante para uma consulta e o modelo de co-ocorrência representa se os

termos da consulta estão associados com uma pessoa ou não. Geralmente, para calcular o valor de relevância de um documento para uma consulta é utilizada a função de ordenação BM25 com ou sem expansão de consulta. Para calcular o valor de co-ocorrência entre os termos da consulta e uma pessoa são utilizadas algumas medidas, tais como:

- $pf(e, d)$: frequência da pessoa e no documento d .
- $|d|$: frequência total de pessoas no documento d .

Baseado nas implementações dos modelos de relevância e co-ocorrência apresentados na seção 3.1.1, foram propostos outros quatro atributos para caracterizar uma pessoa:

- Média da relevância dos documentos onde a pessoa aparece, utilizando a função de ordenação BM25 sem expansão das consultas para o cálculo da relevância ($\overline{BM25_{docs}}$).
- Média da relevância dos documentos onde a pessoa aparece, utilizando a função de ordenação BM25 com expansão das consultas para o cálculo da relevância ($\overline{BM25'_{docs}}$).
- Número de documentos da coleção onde uma referência a pessoa aparece ($n(e, D)$).
- Média da frequência com que as referências a pessoas aparecem em documentos onde existe, no mínimo, uma referência à pessoa ($\overline{f_{pessoa}}$).

Por fim, o conjunto de treinamento é criado a partir dos julgamentos de relevância, substituindo a referência a pessoa pelos valores dos oito atributos mencionados acima.

4.3 Criação das funções de ordenação

Para a criação das funções de ordenação foi utilizado um sistema de aprendizagem de máquina supervisionado do tipo TDIDT (*Top Down Induction of Decision Trees*). Um sistema de aprendizagem de máquina supervisionado é um programa (*indutor*) capaz de induzir uma descrição de um conceito a partir de um conjunto de exemplos previamente conhecidos e rotulados com suas respectivas classes (tabela 4.2). Essa descrição é uma generalização dos exemplos fornecidos ao sistema e é, normalmente, referenciada na literatura como hipótese. O objetivo principal de uma hipótese é, dado um novo exemplo cuja classe é desconhecida, predizer sua classe (MITCHELL, 1997).

Entrada: Conjunto de Exemplos D .

Saída: Árvore de Decisão (Hipótese h).

- 1 Se todos os exemplos tem o mesmo resultado para a função sendo aprendida então retorna um nodo folha com este valor;
- 2 Cria um nodo de decisão N e escolhe o melhor atributo A para este nodo;
- 3 Para cada valor V possível para A :
 - 3.1 cria uma aresta em N para o valor V ;
 - 3.2 cria um subconjunto D_V de exemplos onde $A = V$;
 - 3.3 liga a aresta com o nodo que retorna da aplicação do algoritmo considerando os exemplos D_V .
- 4 Os passos 1, 2 e 3 são aplicados recursivamente para cada novo subconjunto de exemplos de treinamento.

Tabela 4.4: Funcionamento dos algoritmos do tipo TDIDT (RUSSEL; NORVIG, 2003)

Algoritmos do tipo TDIDT (*Top Down Induction of Decision Trees*) geram hipóteses no formato de árvores de decisão (figura 4.3). Uma árvore de decisão é uma estrutura de dados definida recursivamente como (WITTEN; FRANK, 2005):

- um nó folha que corresponde a uma classe ou;
- um nó de decisão que contém um teste sobre algum atributo. Para cada resultado do teste existe uma aresta para uma sub-árvore. Cada sub-árvore tem a mesma estrutura que a árvore.

Figura 4.3: Exemplo de árvore de decisão que pode ser construída a partir do conjunto de treinamento da tabela 4.2

Para realizar a classificação de um novo exemplo, uma árvore de decisão toma como entrada um objeto ou situação descritos por um conjunto de atributos e retorna uma decisão - o valor de saída previsto, de acordo com a entrada. Os algoritmos do tipo TDIDT funcionam como descrito na tabela 4.4.

Um dos pontos principais do algoritmo é escolher qual o melhor atributo para colocar em um nodo (item 2). O atributo escolhido irá determinar a profundidade da árvore. Na próxima seção é visto como é realizado a escolha do atributo.

4.3.1 Seleção de Atributos

A idéia básica da seleção de atributos é avaliar todos os atributos disponíveis para divisão e selecionar aquele que tenha maior probabilidade de melhorar o desempenho de predição da árvore. Essa abordagem tende a gerar árvores menores com maior poder de predição (MITCHELL, 1997).

Para encontrar o atributo que tenha maior probabilidade de melhorar o desempenho de predição da árvore é utilizado o conceito de entropia. Quanto menor o valor da entropia de um nó, mais organizado estarão os exemplos desse nó de acordo com o valor do atributo meta. O valor da entropia de um determinado atributo é dado pela equação 4.6:

$$Entropia(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right) \quad (4.6)$$

onde A é o atributo, v é a quantidade de valores distintos de A , p é a quantidade de exemplos positivos e n é a quantidade de exemplos negativos. O cálculo para determinar a quantidade de informação para representar uma determinada hipótese, dado um conjunto de exemplos, é fornecido pela equação 4.7:

$$I(p, n) = -\frac{p}{p + n} \log_2 \left(\frac{p}{p + n}\right) - \frac{n}{p + n} \log_2 \left(\frac{n}{p + n}\right) \quad (4.7)$$

A quantidade de informação necessária para representar uma hipótese menos a entropia de um determinado atributo gera o Ganho de Informação do atributo (equação 4.8).

$$Ganho(A) = I(p, n) - Entropia(A) \quad (4.8)$$

4.3.2 Atributos com valores contínuos em algoritmos do tipo TDIDT

A maioria das aplicações reais de aprendizagem de máquina, incluindo o problema tratado por este trabalho, precisam manipular dados contínuos. Entretanto, os algoritmos do tipo TDIDT (QUINLAN, 1986) apenas conseguem manipular valores nominais - caso do algoritmo ID3 (QUINLAN, 1988). Desta maneira, os atributos com valores contínuos devem ser discretizados antes de serem selecionados para utilização por um algoritmo TDIDT.

Em princípio, a discretização é apenas uma condição lógica (usando um ou mais

atributos) que serve para particionar os dados em no mínimo dois sub-conjuntos. A principal questão é onde particionar o conjunto, ou seja, que valor de corte¹ utilizar.

Fayyad e Irani (1992) sugerem um método para selecionar o valor de corte dinamicamente. Ou seja, eliminando a necessidade de discretizar os atributos antes de serem selecionados para utilização. Basicamente, para um atributo A que possui valores contínuos, o algoritmo pode criar dinamicamente um novo atributo booleano A_c que é verdadeiro se $A(x) < c$ e falso se $A(x) \geq c$.

Para selecionar o valor de c é utilizado o conceito de entropia para encontrar o valor que proporciona o maior ganho de informação. Desta maneira, enquanto um algoritmo gera uma árvore de decisão, além de testar o melhor atributo para cada nodo, ele deverá também testar o melhor valor para discretizar o atributo. Este método é utilizado pelo algoritmo C4.5 (QUINLAN, 1996).

4.3.3 Poda da árvore

Ao induzir uma função de ordenação, a partir dos exemplos disponíveis, é possível que a função de ordenação seja muito específica para o conjunto de treinamento utilizado. Como o conjunto de treinamento é apenas uma amostra de todas as tuplas $\langle consulta, documento, relevância \rangle$, é possível criar funções de ordenação que melhorem o seu desempenho no conjunto de treinamento, enquanto pioram o desempenho em exemplos diferentes daqueles pertencentes ao conjunto de treinamento. Neste caso, diz-se que a função de ordenação (hipótese) ajusta-se em excesso ao conjunto de treinamento ou que houve um sobre-ajuste² (REZENDE, 2003).

A poda da árvore é uma técnica para lidar com o problema do sobre-ajuste. A idéia geral consiste em aprender uma função de ordenação mais genérica a partir do conjunto de treinamento, melhorando o desempenho em consultas não vistas.

A poda da árvore pode ser do tipo descendente ou ascendente. Na poda descendente (*top-down pruning*), ou pré-poda (*pre pruning*), podem ser utilizados os seguintes critérios de parada: limite da profundidade da árvore, limite na quantidade de fragmentação (por exemplo, não dividir um nó se o número de casos ficar abaixo de determinado limite) ou significância estatística. Na poda ascendente (*bottom-up pruning*), também chamada de pós-poda (*post pruning*), uma grande árvore é gerada e então os ramos são cortados de maneira reversa usando um critério de seleção de modelo (WITTEN; FRANK, 2005; QUINLAN, 1993).

¹Do inglês, *cut point*

²Do inglês, *overfitting*

4.3.4 Uso de árvores de decisão para ordenação de elementos

A estrutura básica de uma árvore de decisão permite classificar novos exemplos. Para realizar a classificação de um novo exemplo, uma árvore de decisão toma como entrada um objeto ou situação descritos por um conjunto de atributos e retorna uma classe. Esta capacidade de classificação é útil para muitos problemas. No entanto, neste trabalho é necessário que a árvore de decisão consiga ordenar um conjunto de exemplos. Para que uma árvore de decisão possa ordenar um conjunto de exemplos é necessário que a sua estrutura seja alterada de alguma maneira.

Existem diversos trabalhos na literatura que propõem a extensão de árvores de decisão para implementar a capacidade de ordenar exemplos (TOTH; PATAKI, 2007). A maioria destes trabalhos propõem que o *ranking* seja obtido pela ordenação do valor de confiança da classificação dada pela árvore de decisão. A forma usual é utilizando a frequência das classes em cada folha da árvore (PROVOST; DOMINGOS, 2003):

$$p_i = \frac{n_i}{N} \quad (4.9)$$

onde, n_i é o número de exemplos da classe c_i que pertencem a folha analisada e N é o número total de exemplos de teste que estão nesta folha.

Além disso, pode-se utilizar técnicas de suavização para melhorar a previsão do valor de confiança (LING; YAN, 2003):

$$p_i = \frac{n_i + 1}{N + C} \quad (4.10)$$

onde, C é o número de classes.

Desta maneira, o valor retornado por uma árvore de decisão para um novo exemplo é o valor da classe encontrada somando-se o valor da confiança deste exemplo pertencer a classe específica, por exemplo.

4.3.5 Configurações do algoritmo C4.5

O algoritmo utilizado para construir as funções de ordenação foi o C4.5. A implementação do C4.5 utilizada foi a disponível no ambiente RAPIDMINER (MIERSWA et al., 2006). Os parâmetros configuráveis desta implementação são:

- O critério de seleção de atributos.
- Se existe ou não pós-poda e qual o nível de confiança utilizado para o cálculo da

pós-poda da árvore, e;

- O limite da profundidade da árvore para situações onde existe pré-poda.

O critério de seleção de atributos adotado foi o ganho de informação (equação 4.8). Para a poda da árvore optou-se pela pré-poda porque é mais rápida que a pós-poda. O método utilizado para pré-poda foi limitar a profundidade da árvore.

Para determinar o limite da profundidade da árvore foi implementado um algoritmo iterativo que recebe um conjunto de treinamento D , um conjunto de testes T e uma medida, chamada de P , para cálculo da precisão do modelo encontrado. Este algoritmo executa o algoritmo C4.5 diversas vezes, passando como parâmetro o conjunto de treinamento D e incrementando o limite de profundidade a cada vez que o algoritmo C4.5 é executado.

O algoritmo C4.5 retorna uma hipótese (função de ordenação) H que é avaliada pela função $Testa(H, T, P)$, utilizando o conjunto de testes T e a medida de precisão P . Para P podem ser utilizadas duas medidas: precisão do modelo ou precisão média ponderada do modelo.

A precisão do modelo é obtida através da divisão do número de exemplos do conjunto de teste corretamente classificados pelo número de exemplos do conjunto de teste. A precisão muitas vezes é chamada também de acurácia e pode ser interpretada como a porcentagem de exemplos no conjunto de teste que são corretamente classificados pela hipótese (MITCHELL, 1997).

A precisão média ponderada do modelo é obtida através da média ponderada das medidas de precisão de cada classe. A precisão de uma classe é a divisão do número de exemplos corretamente classificados daquela classe pelo número de exemplos que o classificador considerou como sendo da classe analisada. Esta medida é utilizada em situações onde uma classe tem um peso maior que outras classes (WITTEN; FRANK, 2005). Por exemplo, no caso da coleção de referência CERC para recuperação de documentos, que possui valores de relevância entre dois e zero, a classe dois (muito relevante) tem peso quatro, a classe um (relevante) tem peso dois e a classe zero (não relevante) tem peso um nos experimentos realizados neste trabalho.

O valor retornado pela função $Testa(H, T, P)$ é utilizado no critério de parada da iteração do algoritmo 4.5. Quando o critério de parada é aceito, o algoritmo iterativo retorna a hipótese que obteve a melhor taxa de precisão.

O funcionamento do algoritmo 4.5 consiste em incrementar o limite de profundidade da árvore de decisão até encontrar uma hipótese com uma precisão menor do que

Determina Árvore (D,T,P): H

Precisão = 0.0;

Limite = 1.0;

H = C4.5(D, Limite);

Precisão' = Testa(H, T, P);

while Precisão' $\geq$ Precisão **do**

Limite = Limite + 1;

H' = H;

Precisão = Precisão';

H = C4.5(D, Limite);

Precisão' = Testa(H, T, P);

end while**return** H';**Tabela 4.5:** Algoritmo iterativo para identificação da hipótese com melhor precisão

a última hipótese encontrada.

Ao optar pela precisão média ponderada espera-se que a função de ordenação construída consiga identificar melhor os documentos mais relevantes. Ao escolher pela precisão do modelo esperar-se que a função de ordenação construída tenha um valor de MAP mais alto.

Ao final do processo de criação da função de ordenação, tem-se uma árvore de decisão que para cada novo par $\langle consulta, item \rangle$ atribui uma relevância para o *documento*. Esta relevância é calculada levando em consideração a classe identificada pela árvore de decisão e o valor de confiança da classificação (equação 4.10).

Para a função de ordenação de documentos, a árvore de decisão pode classificar um documento usando os valores 0, 1 ou 2. Se a árvore de decisão classificar o documento com os valores 1 ou 2 então a relevância do documento para a consulta informada será a classificação obtida mais a confiança da classificação, que é um valor real entre 0 e 1. Se a árvore de decisão classificar o documento com o valor 0, que significa não relevante, então a relevância do documento para consulta informada será a classificação obtida mais o complemento da confiança da classificação.

Para a função de ordenação de pessoas, a árvore de decisão pode classificar uma pessoa usando os valores 0 ou 1. Neste caso a idéia para encontrar o valor que representa a relevância de uma pessoa é a mesma que a utilizada para encontrar o valor que representa a relevância de um documento. Para o valor que representa uma pessoa não relevante (0) é somado o complemento da confiança da classificação. Para o valor que representa uma pessoa relevante (1) é somado a confiança da classificação. A função para cálculo da relevância de um item é apresentada na equação 4.11.

$$rel(c_i, cc_i) = \begin{cases} c_i + (1 - cc_i) & \text{se } c_i = 0 \\ c_i + cc_i & \text{se } c_i > 0 \end{cases} \quad (4.11)$$

onde cc_i é a confiança da classificação, definido em 4.10.

Assim, ao invés de utilizar apenas a classificação encontrada pela árvore de decisão, utiliza-se a classificação encontrada mais o valor de confiança da classificação. Com esta alteração é possível criar uma ordem de relevância ao invés de apenas categorizar os itens.

5 Resultados e Discussões

Neste capítulo são apresentados:

- A metodologia utilizada para avaliar se as funções de ordenação criadas obtêm melhores resultados que outras funções de ordenação.
- O ambiente de avaliação criado para avaliar a aplicabilidade de algoritmos de aprendizagem de máquina na obtenção de funções de ordenação.
- Os resultados obtidos com o uso do algoritmo C4.5 na criação de funções de ordenação para documentos e pessoas. Cada um dos resultados obtidos é também discutido.

5.1 Função de ordenação de documentos

Nesta seção são descritos: a metodologia e o ambiente para avaliação das funções de ordenação de documentos, e; os resultados encontrados pelas duas versões implementadas do algoritmo C4.5:

- Limitado pela precisão do modelo (RD_{c45p}), e;
- Limitado pela precisão média ponderada do modelo (RD_{c45m}).

5.1.1 Metodologia e ambiente para avaliação

O conjunto de dados que formam o ambiente de avaliação é composto pelos oito atributos utilizados para caracterizar um documento, por uma descrição do tópico e pela relevância - que pode assumir o valor dois, um ou zero (tabela 4.2). Este ambiente foi gerado a partir da coleção de referência CERC. Desta coleção de referência foram utilizados os julgamentos de relevância para a tarefa de recuperação de documentos.

No total, são 33.814 exemplos. Destes, 3.730 são exemplos com relevância igual a dois, 4.073 são exemplos com relevância igual a um e 26.011 são exemplos com relevância igual a zero. Os valores dos atributos utilizados para caracterizar os documentos

são sumarizados na tabela 5.1, utilizando: as medidas de média e desvio padrão, e; o valor mínimo e valor máximo encontrados no conjunto de dados.

Tabela 5.1: Valores de média, desvio padrão, mínimo e máximo dos atributos utilizados no conjunto de dados para a tarefa de recuperação de documentos

Atributo	Média	Desvio Padrão	Mínimo	Máximo
R_{tfidf}	0.102	0.150	0.0	1.381
R_{tfnorm}	0.114	0.161	0.0	1.414
R_{tf}	10.633	39,488	0.0	3622.0
R_{coseno}	0.080	0.120	0.0	0.976
$R_{euclidiana}$	1.546	0.291	0.0	2.236
R_{bm25}	0.166	0.238	0.0	2.532
$R_{bm25'}$	0.961	1.284	0.0	13.062
$R_{pagerank}$	0.840	1.119	0.0	3.0

Para a avaliação das abordagens descritas neste trabalho foi utilizado o conceito *5-fold cross validation* (WITTEN; FRANK, 2005). O conjunto de treinamento, composto por 33.814 exemplos, foi dividido em cinco partes de maneira aleatória. Para cada algoritmo foram executados cinco vezes o processo de aprendizagem da hipótese (função de ordenação) sobre quatro partes do conjunto de treinamento. A função de ordenação identificada foi aplicada sobre a parte restante, calculando-se o $P@n$, MAP e $NDCG$. Para cada processo executado, a parte utilizada no teste era modificada. Os valores finais para $P@n$, MAP e $NDCG$ são a média de todos os valores encontrados nas cinco execuções.

Os resultados encontrados pelas funções RD_{c45p} e RD_{c45m} são comparados com os resultados das funções de ordenação mais utilizadas para a recuperação de documentos: BM25 (R_{bm25}), BM25 com expansão de consultas ($R_{bm25'}$) e BM25 com PAGERANK (R_{bm25_pr}). As funções de ordenação R_{bm25} e $R_{bm25'}$ são definidas na seção 4.1. A função de ordenação R_{bm25_pr} é uma das funções implementadas em (SHEN et al., 2007). Esta função reordena os resultados encontrados pela função de ordenação BM25 utilizando o valor de PAGERANK dos documentos.

5.1.2 Resultados encontrados

Na figura 5.1 é apresentada a curva $NDCG$, com o intervalo da 1ª à 50ª posição, para as funções de ordenação BM25 (R_{bm25}), BM25 com expansão de consultas ($R_{bm25'}$), BM25 com PAGERANK (R_{bm25_pr}), C4.5 com limite definido pela precisão (RD_{c45p}) e C4.5 com limite definido pela precisão média ponderada do modelo (RD_{c45m}).

Figura 5.1: Curva NDCG das funções de ordenação para documentos - intervalo da 1^a à 50^a posição

Para deixar mais clara a diferença entre as funções de ordenação, principalmente entre as primeiras posições do *ranking*, é apresentada na figura 5.2 a mesma curva *NDCG* com o intervalo da 1^a à 10^a posição apenas e na tabela 5.2 são descritos os valores de *NDCG@1* e *NDCG@2* das funções de ordenação avaliadas.

Tabela 5.2: Valores de *NDCG@1* e *NDCG@2* das funções de ordenação para documentos

	R_{bm25}	$R_{bm25'}$	R_{bm25_pr}	RD_{c45p}	RD_{c45m}
<i>NDCG@1</i>	0.585	0.665	0.173	0.653	0.684
<i>NDCG@2</i>	0.548	0.618	0.132	0.584	0.589

Na figura 5.3 é apresentada a curva $P@n$, com o intervalo da 1^a à 50^a posição para todas as funções analisadas. Para o cálculo de $P@n$, documentos com relevância igual a 2 foram considerados relevantes e o restante foi considerado como não relevante.

Da mesma forma como foi feito para a curva *NDCG*, para deixar mais clara a diferença entre as funções de ordenação, principalmente entre as primeiras posições do *ranking*, é apresentada na figura 5.4 a mesma curva $P@n$ com o intervalo da 1^a à 10^a posição apenas.

Figura 5.2: Curva NDCG das funções de ordenação para documentos - intervalo da 1ª à 10ª posição

Figura 5.3: Curva P@n das funções de ordenação para documentos - intervalo da 1ª à 50ª posição

Figura 5.4: Curva $P@n$ das funções de ordenação para documentos - intervalo da 1ª à 10ª posição

Na tabela 5.3 são apresentados os valores de MAP das funções de ordenação avaliadas e na tabela 5.4 são apresentados os valores de $P@1$ até $P@3$.

Tabela 5.3: Valores de MAP das funções de ordenação para documentos

	R_{bm25}	$R_{bm25'}$	R_{bm25_pr}	RD_{c45p}	RD_{c45m}
MAP	0.508	0.578	0.258	0.472	0.428

Tabela 5.4: Valores de $P@1$ até $P@3$ das funções de ordenação para documentos

	R_{bm25}	$R_{bm25'}$	R_{bm25_pr}	RD_{c45p}	RD_{c45m}
$P@1$	0.6840	0.7560	0.2000	0.7520	0.7800
$P@2$	0.6380	0.7140	0.1720	0.6880	0.6880
$P@3$	0.6173	0.7000	0.1920	0.6600	0.6386

A função de ordenação RD_{c45m} , criada a partir do algoritmo C4.5 com poda da árvore limitada pela precisão média ponderada do modelo, tem um desempenho ligeiramente superior na primeira posição do *ranking* em relação às outras funções avaliadas. No entanto, a sua precisão cai de maneira acentuada à medida que as posições do *ranking* vão avançando. A função de ordenação RD_{c45p} , criada a partir do algoritmo C4.5 com poda da árvore limitada pela precisão do modelo, tem um MAP maior que

a função de ordenação RD_{c45m} . Estes dados colaboram com a hipótese levantada, durante a execução do trabalho, que é possível controlar a função de ordenação construída pelo algoritmo C4.5 a fim de otimizar a precisão nas primeiras posições do *ranking* ou otimizar a média das precisões (MAP).

Através das curvas $NDCG$ (figura 5.1) e $P@n$ (figura 5.3) e do valor de MAP (tabela 5.3) é possível constatar o desempenho superior da função de ordenação $R_{bm25'}$ (BM25 com expansão de consulta). As únicas situações onde esta função de ordenação tem um desempenho pior do que outra função de ordenação é no $P@1$ e $NDCG@1$, onde a função de ordenação RD_{c45m} apresenta valores superiores (tabelas 5.2 e 5.4).

Normalmente, um grande número de consultas realizadas em uma organização procuram por apenas um documento (o documento certo, não um documento bom). Isto poderia ser uma justificativa para alegar que a função de ordenação RD_{c45m} fornece resultados melhores que a função de ordenação $R_{bm25'}$. No entanto, a diferença entre os resultados de $P@1$ e $NDCG@1$ das duas funções de ordenação é muito pequena para ser considerada relevante.

Alguns resultados apresentados neste trabalho coincidem com os resultados de outros trabalhos propostos para o domínio empresarial. As coincidências são: o bom desempenho de uma função de ordenação que faz uso de uma abordagem baseada em expansão de consultas, e; o mau desempenho de funções de ordenação que fazem uso do valor da autoridade do documento na coleção: R_{bm25pr} , por exemplo.

5.1.3 Análise qualitativa da árvore de decisão

As funções de ordenação construídas pelo algoritmo C4.5 são representadas na forma de árvores de decisão. As árvores geradas têm uma profundidade entre 6 e 10 níveis. Infelizmente, quando uma árvore de decisão possui tantos níveis assim é difícil fazer a análise da mesma. No entanto, entre as funções de ordenação geradas, existe um padrão na escolha dos primeiros atributos. Os atributos dos quatro primeiros níveis são sempre os mesmos.

Na figura 5.5 é possível visualizar os quatro primeiros níveis da árvore de decisão encontrada para o problema de recuperação de documentos. São apresentados os nós intermediários (atributos) e as folhas. Em cada folha da árvore é apresentado o número de exemplos com classificação igual a dois, um e zero que existem em cada sub-conjunto que a folha representa. Além disso, foram feitos cortes, na forma de quadrados com linhas intermitentes, na árvore de decisão para representar a quantidade de exemplos que cada nó intermediário abrange. Por exemplo, o sub-conjunto de exem-

plos que satisfaz a restrição $R_{bm25'} > 0.598$ tem 9030 exemplos da classe zero, 2896 exemplos da classe um e 3247 exemplos da classe dois. O sub-conjunto de exemplos que satisfaz a restrição $R_{bm25'} > 0.598 \wedge R_{tfidf} > 0.175$ tem 2704 exemplos da classe zero, 1417 exemplos da classe um e 2054 exemplos da classe dois.

Figura 5.5: Extrato da árvore de decisão gerada pelo algoritmo C4.5 para o problema de recuperação de documentos

Após análise da árvore de decisão (figura 5.5) observa-se que o atributo $R_{bm25'}$, que está na raiz da árvore, consegue separar bem os exemplos com classificação igual a dois: 3247 (87%) que satisfazem $R_{bm25'} > 0.598$, e; 483 (13%) que satisfazem $R_{bm25'} \leq 0.598$. No entanto, depois desta divisão nenhum outro atributo consegue encontrar uma separação adequada entre o conjunto de exemplos existente a fim de classificar adequadamente um documento.

5.2 Função de ordenação de pessoas

Nesta seção são descritos: a metodologia e o ambiente para avaliação das funções de ordenação de especialistas, e; os resultados encontrados pelo algoritmo C4.5. Para a construção de funções de ordenação de especialistas foram testadas duas implementações do algoritmo C4.5:

- Limitado pela precisão do modelo (RP_{c45p}), e;

- Limitado pela precisão média ponderada do modelo (RP_{c45m}).

No entanto, as funções de ordenação encontradas pelas duas versões foram exatamente as mesmas. Sendo assim, para simplificar a descrição dos resultados, a função de ordenação de especialistas construída a partir do algoritmo C4.5 será chamada simplesmente de RP_{c45} .

5.2.1 Metodologia e ambiente para avaliação

O conjunto de dados que forma o ambiente de avaliação é composto pelos oito atributos utilizados para caracterizar uma pessoa, por uma descrição do tópico e pela relevância - que pode assumir o valor 1 ou 0. Este ambiente de avaliação foi gerado a partir da coleção de referência CERC. Desta coleção de referência foram utilizados os julgamentos de relevância para a tarefa de recuperação de especialistas.

O conjunto de julgamentos de relevância para a recuperação de especialistas possui apenas 152 registros, todos positivos. O objetivo destes julgamentos de relevância é dizer quem é considerado especialista para cada um dos 50 tópicos utilizados na coleção de referência. Alguns tópicos possuem apenas um especialista, enquanto que outros tópicos possuem dois ou três especialistas. Em todos os documentos da coleção de referência é possível identificar 3.490 referências de pessoas. Nos julgamentos de relevância são encontradas 150 pessoas distintas.

Para a construção do conjunto de treinamento pode-se assumir que as 3.340 pessoas restantes não são consideradas especialistas em nenhum dos 50 tópicos existentes na coleção. Portanto, o conjunto de treinamento do ambiente de avaliação aqui utilizado poderia ser composto pelos 152 exemplos positivos, e pela combinação entre as 3.340 pessoas que não são consideradas especialistas distribuídas pelos 50 tópicos existentes na coleção, totalizando 167.000 exemplos negativos no conjunto de treinamento. Ao todo, o conjunto de treinamento possuiria 167.152 exemplos. Entretanto, afirmar que as 3.490 pessoas restantes não são especialistas em nenhum dos 50 tópicos pode ser um tanto equivocado, podendo gerar um conjunto de treinamento com muito ruído.

Para evitar a construção de um conjunto de treinamento com muito ruído, optou-se pela seleção aleatória de pares, tópico versus pessoa, para formar 400 exemplos negativos. Os tópicos utilizados foram selecionados a partir de todos os 50 tópicos disponíveis e as pessoas utilizadas foram selecionadas a partir das 3.490 pessoas que não estão presentes nos 152 julgamentos de relevância.

O conjunto de treinamento utilizado para a tarefa de recuperação de especialistas

possui 552 exemplos, 152 positivos e 400 negativos. Os valores dos atributos utilizados para caracterizar as pessoas são sumarizados na tabela 5.5, utilizando: as medidas de média e desvio padrão, e; o valor mínimo e valor máximo encontrados no conjunto de dados.

Tabela 5.5: Valores de média, desvio padrão, mínimo e máximo dos atributos utilizados no conjunto de dados para a tarefa de recuperação de pessoas

Atributo	Média	Desvio Padrão	Mínimo	Máximo
$BM25_1$	0.537	0.828	0.0	4.864
$BM25_2$	1.048	0.941	0.0	5.243
$BM25_3$	0.268	0.101	0.0	0.624
$BM25_4$	1.790	0.411	0.0	2.891
$\overline{BM25_{docs}}$	0.211	0.220	0.0	0.946
$\overline{BM25'_{docs}}$	1.334	1.205	0.0	6.592
$n(e, D)$	86.638	74.884	1.0	477.0
f_{pessoa}	2.129	0.522	1.0	3.768

5.2.2 Resultados encontrados

Na figura 5.6 é apresentada a curva $P@n$ para as funções de ordenação:

- $BM25_1$: função de ordenação BM25 sem expansão de consultas para a ordenação de perfis gerados a partir do conteúdo completo de todos os documentos onde aparece uma referência ao candidato à especialista.
- $BM25_2$: função de ordenação BM25 com expansão de consultas para a ordenação de perfis gerados a partir do conteúdo completo de todos os documentos onde aparece uma referência ao candidato à especialista.
- $BM25_3$: função de ordenação BM25 sem expansão de consultas para a ordenação de perfis gerados a partir do conteúdo dos parágrafos onde aparece uma referência ao candidato à especialista.
- $BM25_4$: função de ordenação BM25 com expansão de consultas para a ordenação de perfis gerados a partir do conteúdo dos parágrafos onde aparece uma referência ao candidato à especialista.
- RP_{c45} : função de ordenação criada pelo algoritmo C4.5.

Para deixar mais clara a diferença entre as funções de ordenação, na tabela 5.6 são apresentados os valores de $P@1$, $P@2$ e $P@3$ das funções de ordenação avaliadas. Na tabela 5.7 são apresentados os valores de MAP das funções de ordenação avaliadas.

Figura 5.6: Curva $P@n$ das funções de ordenação para especialistas

Tabela 5.6: Valores de $P@1$ até $P@3$ das funções de ordenação para especialistas

	$BM25_1$	$BM25_2$	$BM25_3$	$BM25_4$	RP_{c45}
$P@1$	0,6902	0,7156	0,6966	0,6966	0,8191
$P@2$	0,1228	0,1355	0,1261	0,1354	0,2002
$P@3$	0,0194	0,0278	0,0214	0,0320	0,0624

Tabela 5.7: Valores de MAP das funções de ordenação de especialistas

	$BM25_1$	$BM25_2$	$BM25_3$	$BM25_4$	RP_{c45}
MAP	0,7275	0,7450	0,7335	0,7413	0,8307

A partir dos resultados apresentados é possível ver que a função de ordenação RP_{c45} tem um desempenho superior em relação as outras funções avaliadas. Também, é possível constatar que as outras funções avaliadas possuem um desempenho muito similar entre elas, fato igualmente encontrado por (SOBOROFF; VRIES; CRASWELL, 2006) e (BAILEY et al., 2007b).

5.2.3 Análise qualitativa da árvore de decisão

As árvores geradas têm uma profundidade entre 4 e 8 níveis, com pequenas diferenças a partir do quinto nível em algumas versões. Os atributos dos quatro primeiros níveis são sempre os mesmos. Na figura 5.7 é possível visualizar os quatro primeiros níveis da árvore de decisão encontrada para o problema de recuperação de especialistas. São apresentados os nós intermediários (atributos) e as folhas. Em cada folha da árvore é apresentado o número de exemplos positivos (1) e negativos (0) que existem em cada sub-conjunto que a folha representa.

Figura 5.7: Extrado da árvore de decisão gerada pelo algoritmo C4.5 para o problema de recuperação de especialistas

A partir da análise desta árvore percebe-se que:

- O atributo com maior ganho de informação é $\overline{f_{pessoa}}$, pois é o atributo que aparece na raiz da árvore de decisão, e;
- Uma das folhas da árvore de decisão (marcada com uma borda mais grossa na figura 5.7) possui 138 exemplos positivos, o que corresponde a 90.8% dos exemplos positivos do conjunto completo. Provavelmente, as condições usadas para chegar até neste sub-conjunto conseguem caracterizar muito bem especialistas para cada tópico.

Na figura 5.8 é possível ver a expansão deste ramo. Ao adicionar mais uma condição $n(e, D) > 50$ obtém-se um sub-conjunto com 138 exemplos positivos e 64 exemplos negativos, o que corresponde a somente 16% dos exemplos negativos.

Figura 5.8: Árvore de decisão com destaque para o ramo que leva ao sub-conjunto com maior número de exemplos positivos

Da análise desta árvore pode-se concluir que uma forma adequada para identificar se uma pessoa é especialista pode ser através da seguinte regra lógica:

$$(\overline{f_{pessoa}} > 2.093) \wedge (BM25_4 > 1.613) \wedge (\overline{BM25'_{docs}} > 0.072) \wedge (n(e, D) > 50) \quad (5.1)$$

Lembrando que:

- $\overline{f_{pessoa}}$ é a média da frequência com que as referências a pessoas aparecem em documentos onde existe, no mínimo, uma referência a pessoa.
- $BM25_4$ é o valor retornado pela função de ordenação BM25 com expansão de consultas para a ordenação de perfis gerados a partir do conteúdo dos parágrafos onde aparece uma referência ao candidato à especialista.
- $\overline{BM25'_{docs}}$ é a média da relevância dos documentos onde a pessoa aparece, utilizando a função de ordenação BM25 com expansão de consultas.
- $n(e, D)$ é o número de documentos da coleção onde a pessoa e aparece.

Ou seja, a regra 5.1 diz que uma pessoa é considerada especialista em algo se aparecer em muitos documentos, aparecer muitas vezes nos documentos e os documentos onde aparece têm uma relevância alta para a consulta. Isto é ligeiramente diferente das propostas existentes na literatura e pode servir de subsídio para o desenvolvimento de novas funções de ordenação.

O uso da árvore de decisão, ou de uma linguagem de representação compreensível por pessoas, permite explicar porque uma pessoa foi considerada especialista para um tópico. Por exemplo, uma explicação possível seria: "*... a pessoa p estava presente em n documentos, quando presente em um documento a mesma é citada em média m vezes e o conjunto de documentos onde a pessoa é citada tem uma relevância média de $x \dots$* ".

5.3 Detalhes de implementação

Para a manipulação das coleções de referência, desenvolvimento do ambiente de avaliação, implementação do algoritmo indutor de funções de ordenação no formato de árvores de decisão, criação dos algoritmos para avaliação e impressão dos resultados foi utilizado uma máquina servidor SUN: com dois processadores *dual core* de 64 bits, memória de 4 GB RAM, e; sistema operacional Linux.

Foram manipulados 370.715 documentos, aproximadamente 4.2 GB de dados, e gerados conjuntos de treinamento com aproximadamente 80 MB de dados no formato de tabelas esparsas. Para a implementação dos manipuladores de corpus e consultas, geradores de tabelas de treinamento e componentes para avaliação dos resultados foram utilizadas as linguagens de programação JAVA¹, PYTHON², PERL³ e AWK⁴.

A linguagem de programação JAVA foi utilizada na criação dos manipuladores de corpus e nos geradores de tabela de treinamento. A versão final destes componentes possui aproximadamente 85 classes. A linguagem de programação PERL foi utilizada na implementação dos algoritmos para avaliação dos resultados, reutilizando parte dos algoritmos implementados em (LIU et al., 2007). A linguagem de programação PYTHON foi utilizada em alguns momentos para a prototipação de alguns algoritmos. A versão final destes algoritmos foi implementada em JAVA, visando o ganho de desempenho (tempo e espaço de processamento). A linguagem de programação AWK foi utilizada em alguns SCRIPTS devido aos seus recursos que facilitam a manipulação

¹<http://java.sun.com>

²<http://www.python.org>

³<http://www.perl.org>

⁴<http://www.gnu.org/manual/gawk/gawk.html>

de arquivos do tipo texto.

6 Considerações Finais

Neste trabalho foi feita a avaliação do desempenho do uso de árvores de decisão como função de ordenação para documentos e pessoas em ambientes empresariais.

Os atributos utilizados para caracterizar as fontes de informação de um ambiente empresarial foram escolhidos baseando-se em: algoritmos existentes na literatura para a recuperação de informação em ambientes empresariais, e funções de ordenação para domínios genéricos. Não foi utilizado nenhum atributo identificado a partir do levantamento bibliográfico sobre a dinâmica de produção e consumo de informações em um ambiente empresarial, por não ter sido identificado nenhum atributo relevante que já não tivesse sido utilizado pelos algoritmos existentes na literatura.

Para a caracterização de documentos foram utilizados oito atributos, todos também utilizados em propostas para a recuperação de documentos em ambiente empresarial e pertencentes aos modelos vetorial, probabilístico e baseado em *hyperlinks*. Os atributos utilizados foram: (i) soma dos valores *tf-idf* dos termos presentes na consulta e encontrados nos documentos; (ii) soma da frequência normalizada dos termos da consulta encontrados no documento; (iii) soma da frequência não normalizada dos termos da consulta encontrados no documento; (iv) o valor da similaridade entre o documento e a consulta calculado usando o cosseno do ângulo sobre os vetores do documento e da consulta; (v) o valor da distância entre os mesmos vetores utilizando o conceito de distância euclidiana; (vi) o valor que a função de ordenação BM25 encontra para o documento dado uma consulta; (vii) o valor que a função de ordenação BM25 encontra para o documento levando em consideração todos os termos da consulta mais todos os termos que aparecem na descrição da necessidade de informação; e, (viii) o valor que representa a autoridade (PAGERANK) de cada documento na coleção.

Para a caracterização de pessoas foram utilizados oito atributos. Quatro atributos são oriundos da abordagem baseada no perfil do usuário e os outros atributos foram criados a partir de conceitos existentes no modelo de dois estágios. Para os atributos oriundos da abordagem baseada no perfil do usuário foram criados dois tipos de perfis e sobre estes perfis foram aplicadas duas funções de ordenação, gerando quatro

combinações distintas. Os atributos que foram criados a partir de conceitos existentes no modelo de dois estágios são: média da relevância dos documentos onde a pessoa aparece utilizando a função de ordenação BM25, com e sem expansão da consulta; números de documentos da coleção onde uma referência à pessoa aparece; e, média da frequência com que as referências a pessoas aparecem em documentos.

Neste trabalho foi proposto uma alteração na forma como o resultado da relevância de um item é retornado pela árvore de decisão. Ao invés de utilizar apenas a classificação encontrada pela árvore de decisão, utiliza-se a classificação encontrada mais o valor de confiança da classificação. Com esta alteração é possível criar uma ordem de relevância ao invés de apenas categorizar os itens.

6.1 Discussão dos resultados

O uso do algoritmo C4.5 para a construção de funções de ordenação mostrou-se parcialmente efetivo. Para a tarefa de recuperação de documentos a árvore de decisão gerada não trouxe resultados bons. Porém, constatou-se que é possível controlar a forma de construção da árvore de decisão a fim de otimizar a precisão nas primeiras posições do *ranking* ou otimizar a média das precisões (*MAP*).

Para a tarefa de recuperação de pessoas, o algoritmo C4.5 obteve uma árvore de decisão que consegue resultados melhores que todas as outras funções de ordenação avaliadas. A árvore de decisão gerada obteve um $P@1$ igual a 0,9, enquanto que a média de $P@1$ das outras funções de ordenação avaliadas foi de 0,7. O *MAP* obtido pela árvore de decisão foi 0,83, enquanto que a média do *MAP* das outras funções de ordenação foi de 0,74.

A estrutura simbólica utilizada para representar a função de ordenação contribuiu para a compreensão da composição dos diversos atributos utilizados na caracterização dos documentos e pessoas, permitindo propor novas funções de ordenação, por exemplo. Além disso, o uso da árvore de decisão, ou de uma linguagem de representação compreensível por pessoas, permite explicar porque uma pessoa foi considerada especialista para um tópico.

A partir da análise da árvore de decisão utilizada como função de ordenação para pessoas foi possível entender que uma pessoa é considerada especialista em algum tópico se ela aparecer em muitos documentos, aparecer muitas vezes nos documentos e os documentos onde aparece têm uma relevância alta para a consulta. Apesar destas características aparecerem em inúmeras funções de ordenação para pessoas, entre as funções de ordenação descritas no levantamento bibliográfico deste trabalho, nenhuma

função de ordenação faz uso dos três atributos simultaneamente.

Alguns resultados apresentados neste trabalho coincidem com os resultados de outros trabalhos propostos para o domínio empresarial. As coincidências são: (i) o bom desempenho de funções de ordenação que fazem uso de uma abordagem baseada em expansão de consultas; e, (ii) o mau desempenho de funções de ordenação que fazem uso do conceito de autoridade ou popularidade, conceitos amplamente empregados em soluções voltadas para a WEB.

Durante a execução deste trabalho foram desenvolvidos dois ambientes de testes para a validação de algoritmos de aprendizagem de máquina na construção de funções de ordenação para o domínio empresarial. Os ambientes de testes são para a tarefa de recuperação de documentos e para a tarefa de recuperação de pessoas. Estes ambientes de testes podem ser integrados em outros ambientes de testes como, por exemplo, o ambiente desenvolvido por (LIU et al., 2007).

É importante salientar que este trabalho não diz respeito apenas a recuperação de documentos, mas diz respeito a recuperação de entidades (documentos e pessoas). Na revisão bibliográfica realizada neste estudo não foi encontrado na literatura nenhum trabalho cujo objetivo fosse aplicar aprendizagem de máquina na construção de funções de ordenação de pessoas. O fato dos resultados obtidos pela árvore de decisão utilizada na ordenação de pessoas ter tido bom resultado abre precedente para que esta mesma abordagem possa ser utilizada na ordenação de outros tipos de entidades, como arquivos de áudio e vídeo, por exemplo.

6.2 Trabalhos futuros

Entre as limitações deste trabalho e sugestões para trabalhos futuros, inclui-se: (i) a necessidade da realização de testes envolvendo outros algoritmos de aprendizagem de máquina; e, (ii) a necessidade de uma análise minuciosa da correlação entre as medidas de precisão da área de Recuperação de Informação com as medidas de precisão da área de Aprendizagem de Máquina.

Além disso, durante o desenvolvimento deste trabalho percebeu-se que algumas ferramentas e conceitos atualmente discutidos na academia e no mercado, tais como ontologias e taxonomias, podem ser utilizados para melhorar a gestão da informação em ambientes empresariais. Por exemplo, é comum o emprego de taxonomias em ambientes empresariais visando a organização das informações. Ao mesmo tempo, um dos resultados encontrados neste trabalho foi a constatação do bom desempenho de funções de ordenação que fazem uso de abordagens baseadas em expansão de consul-

tas. Trabalhos futuros podem experimentar a utilização de um sistema de recuperação de informação que expande as consultas do usuário de acordo com o conteúdo da taxonomia ou ontologia, visando uma recuperação mais eficiente no ambiente empresarial.

Outra tecnologia que tem sido muito empregada atualmente são os chamados Wiki. Segundo (O'LEARY, 2008), um Wiki permite que usuários possam: (i) adicionar novo conteúdo; (ii) ligar um conteúdo ao outro através de hiperlinks; (iii) editar conteúdo já existente; (iv) organizar e estruturar o conteúdo; (v) visualizar o conteúdo, e; (vi) acessar o histórico de contribuições. As vantagens do uso de um Wiki como ferramenta para edição colaborativa de documentos são: (i) Wikis geram uma rede de conhecimento ligando pessoas ao conteúdo criado; (ii) Wikis auxiliam na criação de documentos consensuais, e; (iii) através de Wikis é fácil gerar informações a partir de diversas fontes. Sendo assim, outra linha de trabalho futuro é avaliar o quanto que as meta-informações contidas em um Wiki contribuem para a identificação de documentos relevantes e especialistas para uma dada requisição de informação.

Referências

- BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. [S.l.]: ACM Press, 1999.
- BAILEY, P.; CRASWELL, N.; SOBOROFF, I.; VRIES, A. P. de. The csiro enterprise search test collection. *SIGIR Forum*, v. 41, n. 2, p. 42–45, 2007.
- BAILEY, P.; VRIES, A. P. de; CRASWELL, N.; SOBOROFF, I. Overview of the trec 2007 enterprise track. In: *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. [s.n.], 2007. Disponível em: <<http://trec.nist.gov/pubs/trec16/t16proceedings.html>>.
- BALOG, K. *People Search in the Enterprise*. Tese (Doutorado) — Dutch Research School for Information and Knowledge Systems, 2008.
- BALOG, K.; AZZOPARDI, L.; RIJKE, M. de. Formal models for expert finding in enterprise corpora. In: *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. Seattle, Washington, USA: ACM Press, 2006. p. 43–50.
- BAO, S.; DUAN, H.; ZHOU, Q.; XIONG, M.; CAO, Y.; YU, Y. Research on expert search at enterprise track of trec 2006. In: *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*. [S.l.: s.n.], 2006.
- BERTINO, E.; RAMAMRITHAM, K. Data dissemination on the web. *IEEE Internet Computing*, May/June 2004.
- BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, v. 30, n. 1-7, p. 107–117, 1998.
- CAO, Y.; LIU, J.; BAO, S.; LI, H. Research on expert search at enterprise track of trec 2005. In: *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*. [S.l.: s.n.], 2005.
- COOPER, W. S. 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*. New York, NY, USA: Springer-Verlag, 1994. p. 242–247. Disponível em: <<http://portal.acm.org/citation.cfm?id=188562>>.
- CRASWELL, N.; VRIES, A. de; SOBOROFF, I. Overview of the trec 2005 enterprise track. In: VOORHEES, E. M.; BUCKLAND, L. P. (Ed.). *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*. Gaithersburg, Maryland: [s.n.], 2005. Disponível em: <<http://trec.nist.gov/>>.
- CRASWELL, N.; ZARAGOZA, H.; RODERTSON, S. Microsoft cambridge at trec-14: Enterprise track. In: *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*. [S.l.: s.n.], 2005.

- CROFT, W. B.; LAFFERTY, J. (Ed.). *Language Modeling for Information Retrieval*. Springer-Verlag, 2003. 264 p. Disponível em: <<http://www.google.com/books?id=p1qBAYTqOsYC>>.
- DANILOWICZ, C.; NGUYEN, H. C. Using user profiles in intelligent information retrieval. In: HACID, M.-S.; RÁŠ, Z. W.; ZIGHED, D. A.; KODRATOFF, Y. (Ed.). *Foundations of Intelligent Systems. 13th International Symposium*. Lyon, France: Springer-Verlag, 2002. p. 223–231.
- FAGIN, R.; KUMAR, R.; MCCURLEY, K. S.; NOVAK, J.; SIVAKUMAR, D.; TOMLIN, J. A.; WILLIAMSON, D. P. Searching the workplace web. In: *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: [s.n.], 2003. p. 366–375.
- FAN, Y.; HUANG, X. York university at trec 2007: Enterprise document search. In: *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. [S.l.: s.n.], 2007.
- FAYYAD, U. M.; IRANI, K. B. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, v. 8, p. 87–102, 1992.
- FELDMAN, S.; SHERMAN, C. *The high cost of not finding information. Technical Report 29127*. [S.l.], 2003.
- FREITAG, D.; MCCALLUM, A. Information extraction with HMM structures learned by stochastic optimization. In: *AAAI/IAAI*. [S.l.: s.n.], 2000. p. 584–589.
- FREUND, L.; TOMS, E. G. Enterprise search behaviour of software engineers. In: *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 2006. p. 645–646.
- FREUND, L.; TOMS, E. G.; CLARKE, C. L. Modeling task-genre relationships for ir in the workplace. In: *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 2005. p. 441–448.
- FU, Y.; XUE, Y.; ZHU, T.; LIU, Y.; ZHANG, M.; MA, S. Thuir at trec 2007: Enterprise track. In: *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. [S.l.: s.n.], 2007.
- FU, Y.; YU, W.; LI, Y.; LIU, Y.; ZHANG, M.; MA, S. Thuir at trec 2005: Enterprise track. In: *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*. [S.l.: s.n.], 2005.
- GREIF, I. Desktop agents in group-enabled products. *Commun. ACM*, ACM Press, New York, NY, USA, v. 37, n. 7, p. 100–105, 1994. ISSN 0001-0782.
- HANSEN, P.; JARVELIN, K. Collaborative information retrieval in an information-intensive domain. *Information Processing and Management*, v. 41, n. 5, p. 1101–1119, September 2005.
- HAWKING, D. Challenges in enterprise search. In: *ADC '04: Proceedings of the 15th Australasian database conference*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004. p. 15–24. Disponível em: <<http://portal.acm.org/citation.cfm?id=1012297>>.

HERTZUM, M.; PEJTERSEN, A. M. The information-seeking practices of engineers: searching for documents as well as for people. *Information Processing and Management*, v. 36, n. 5, p. 761–778, September 2000.

IDC, I. D. C. *European Knowledge Management Fact Book*. [S.l.], 2000.

JOACHIMS, T.; LI, H.; LIU, T.-Y.; ZHAI, C. Learning to rank for information retrieval (lr4ir 2007). *SIGIR Forum*, v. 41, n. 2, p. 58–62, 2007.

JONES, K. S.; WALKER, S.; ROBERTSON, S. E. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, v. 36, n. 6, p. 779–808, 2000.

JÄRVELIN, K.; KEKÄLÄINEN, J. 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*. New York, NY, USA: ACM Press, 2000. p. 41–48.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, v. 20, n. 4, p. 422–446, 2002.

KLEINBERG, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, v. 46, n. 5, p. 604–632, 1999.

KOLLA, M.; VECHTOMOVA, O. Enterprise search: Identifying relevant sentences and using them for query expansion. In: *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. [S.l.: s.n.], 2007.

LING, C. X.; YAN, R. J. Decision tree with better ranking. In: *Proceedings of International Conference on Machine Learning (ICML 2003)*. [S.l.: s.n.], 2003.

LIU, T.-Y.; XU, J.; QIN, T.; XIONG, W.; LI, H. Letor: Benchmark dataset for research on learning to rank for information retrieval. In: *LR4IR 2007, in conjunction with SIGIR 2007*. [s.n.], 2007. Disponível em: <<http://research.microsoft.com/users/LETOR/>>.

LUGO, G. A. G. *Um Modelo de Sistemas Multiagentes para Partilha de Conhecimento utilizando Redes Sociais Comunitárias*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, Abril 2004.

MACDONALD, C.; HE, B.; PLACHOURAS, V.; OUNIS, I. University of glasgow at trec 2005: Experiments in terabyte and enterprise tracks with terrier. In: *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*. [S.l.: s.n.], 2005.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *An Introduction to Information Retrieval*. Cambridge University Press, 2008. Disponível em: <<http://www-csli.stanford.edu/hinrich/information-retrieval-book.html>>.

MANNING, C. D.; SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. [S.l.]: MIT Press, 2003.

MCDONALD, D. W.; ACKERMAN, M. S. Expertise recommender: a flexible recommendation system and architecture. In: *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*. [S.l.]: ACM Press, 2000. p. 231–240.

MIERSWA, I.; WURST, M.; KLINKENBERG, R.; SCHOLZ, M.; EULER, T. Yale: Rapid prototyping for complex data mining tasks. In: UNGAR, L.; CRAVEN, M.; GUNOPULOS, D.; ELIASSI-RAD, T. (Ed.). *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006. p. 935–940. ISBN 1-59593-339-5.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill, 1997.

MOCKUS, A.; HERBSLEB, J. D. Expertise browser: a quantitative approach to identifying expertise. In: *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*. New York, NY, USA: ACM Press, 2002. p. 503–512.

MOOERS, C. N. Zatoeodmg applied to mechanical organization of knowledge. *American Documentation*, v. 2, p. 20–32, 1951.

MUKHERJEE, R.; MAO, J. Enterprise search: Tough stuff. *Queue*, v. 2, n. 2, p. 36–46, 2004.

OGILVIE, P.; CALLAN, J. Experiments with language models for know-item finding of e-mail messages. In: *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*. [S.l.: s.n.], 2005.

O'LEARY, D. E. Wikis: from each according to his knowledge. *IEEE Computer Society*, p. 34–41, 2008.

POLTROCK, S.; GRUDIN, J.; DUMAIS, S.; FIDEL, R.; BRUCE, H.; PEJTERSEN, A. M. Information seeking and sharing in design teams. In: *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*. New York, NY, USA: ACM Press, 2003. p. 239–247.

PORTER, M. F. An algorithm for suffix stripping. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 313–316, 1997.

PROVOST, F.; DOMINGOS, P. Tree induction for probability based ranking. *Machine Learning*, v. 52, n. 3, p. 199–215, 2003.

QIN, T.; LIU, T.-Y.; ZHANG, X.-D.; WANG, D.-S.; XIONG, W.-Y.; LI, H. Learning to rank relational objects and its application to web search. In: *WWW '08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2008. p. 407–416.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, p. 81–106, 1986.

QUINLAN, J. R. Knowledge acquisition for knowledge-based systems. In: _____. [S.l.]: Academic Press, 1988. cap. Simplifying Decision Trees.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. [S.l.]: Morgan Kaufmann Publishers, 1993.

QUINLAN, J. R. Induction of decision trees. *Journal of Artificial Intelligence Research*, v. 4, p. 77–90, 1996.

REICHLING, T.; SCHUBERT, K.; WULF, V. Matching human actors based on their texts: design and evaluation of an instance of the expertfinding framework. In: *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*. Sanibel Island, Florida, USA: ACM Press, 2005. p. 61–70.

- REZENDE, S. O. (Ed.). *Sistemas Inteligentes: Fundamentos e Aplicações*. [S.l.]: Editora Manole, 2003.
- ROBERTSON, S. E.; JONES, K. S. Relevance weighting of search terms. *Journal of the American Society for Information Science and Technology*, v. 27, n. 3, p. 129–146, 1976. Disponível em: <<http://portal.acm.org/citation.cfm?id=106783>>.
- RU, Z.; LI, Q.; XU, W.; GUO, J. Bupt at trec 2006: Enterprise track. In: *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*. [S.l.: s.n.], 2006.
- RUSSEL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. 2. ed. [S.l.]: Prentice-Hall, 2003. ISBN 0-13-790395-2.
- SALTON, G. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, 1971. Disponível em: <<http://portal.acm.org/citation.cfm?id=1102022>>.
- SALTON, G.; BUCKLEY, C. *Term Weighting Approaches in Automatic Text Retrieval*. [S.l.], 1987. Disponível em: <<http://portal.acm.org/citation.cfm?id=866292>>.
- SHEN, H.; CHEN, G.; CHEN, H.; LIU, Y.; CHENG, X. Research on enterprise track of trec 2007. In: *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*. [S.l.: s.n.], 2007.
- SOBOROFF, I.; VRIES, A. P. de; CRASWELL, N. Overview of the trec 2006 enterprise track. In: *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*. [s.n.], 2006. Disponível em: <<http://trec.nist.gov/pubs/trec15/t15proceedings.html>>.
- TOTH, N.; PATAKI, B. On classification confidence and ranking using decision trees. In: *Intelligent Engineering Systems (INES 2007)*. [S.l.: s.n.], 2007. p. 133–138.
- TROTMAN, A. Learning to rank. *Information Retrieval*, v. 8, p. 359–381, 2005.
- VELOSO, A. A.; ALMEIDA, H. M.; GONÇALVES, M. A.; JR., W. M. Learning to rank at query-time using association rules. In: *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. [S.l.]: ACM Press, 2008. p. 267–274.
- WITTEN, I. H.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Second. [S.l.]: Elsevier, 2005.
- YOU, G.; LU, Y.; LI, G.; YIN, Y. Ricoh research at trec 2006: Enterprise track. In: *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*. [S.l.: s.n.], 2006.
- ZHU, J.; SONG, D.; RÜGER, S.; EISENSTADT, M.; MOTTA, E. The open university at trec 2006 enterprise track expert search task. In: *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*. [S.l.: s.n.], 2006.