UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENG. DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Modelo ontológico relacional *fuzzy* em sistemas de recuperação de informação textual

Dissertação de Mestrado

Autor: **Rachel Carlos Pereira**

Orientador: **Prof. Dr. Fernando Antônio Campos Gomide**

Co-orientador: **Prof. Dr. Ivan Luiz Marques Ricarte**

Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, para preenchimento dos pré-requisitos parciais para obtenção do Título de Mestre em Engenharia Elétrica na área de Engenharia de Computação.

09 de novembro de 2004

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Pereira, Rachel Carlos
R735s Modelo ontológico relacional *fuzzy* em sistemas de recuperação de informação textual / Rachel Carlos Pereira. -- Campinas, SP: [s.n.], 2004.

Orientadores: Fernando Antônio Campos Gomide, Ivan Luiz Marques Ricarte.
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Recuperação da informação. 2. Sistemas de recuperação da informação - Documentos. 3. Lógica difusa. 4. Ontologia. I. Gomide, Fernando Antônio Campos. II. Ricarte, Ivan Luiz Marques. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Resumo

O crescimento exponencial de conteúdo em sistemas de informação faz com que as técnicas utilizadas pelos tradicionais sistemas de busca se tornem cada vez mais inadequadas para recuperação de informações relevantes. Nesse sentido, apresenta-se neste trabalho um novo modelo de recuperação de informação que traz em suas respostas itens de maior relevância do que os obtidos pelos modelos clássicos de recuperação de informação. O modelo proposto, denominado de modelo ontológico relacional *fuzzy*, baseia-se em uma ontologia relacional e conceitos da teoria de conjuntos *fuzzy* para representação do conhecimento e recuperação de informações relevantes. Os resultados obtidos mostram que o desempenho do modelo ontológico é competitivo sob o ponto de vista de cobertura e precisão, quando comparado a abordagens alternativas baseadas em *thesaurus* e redes *fuzzy* para representação e processamento do conhecimento.

Abstract

The exponential growth of contents in information systems raises the need of using better search techniques in information retrieval systems. This work introduces a new information retrieval model which returns a larger number of relevant items than those returned by classical information retrieval models. The proposed model, called fuzzy relational ontological model, is based on a relational ontology and on principles of fuzzy set theory. The fuzzy relational ontological model is used for both, knowledge representation and information retrieval. Experimental results suggest that the fuzzy relational ontological achieves better performance (precision and recall) when compared with two alternative approaches based on thesaurus and fuzzy networks for knowledge representation and processing.

Agradecimentos

À Deus que me deu forças para seguir em frente e chegar até aqui...

À minha família, por me proprocionar a base emocional necessária para vencer as etapas mais complicadas do meu processo de aprendizado. Em especial, à minha mãe Tânia e ao meu avô José Carlos que serão sempre fonte de orgulho, admiração e uma referência para o meu crescimento pessoal e profissional.

Ao meu namorado, amigo e companheiro Clausius, por ser minha fonte de inspiração diária e ter compreendido a minha ausência em tantas ocasiões em que precisei me dedicar integralmente ao mestrado. Pelas palavras de incentivo, gestos de amor e carinho nas horas certas.

Aos professores Fernando Gomide e Ivan Ricarte, pela oportunidade de realização deste trabalho, pela confiança e orientação sempre oportuna.

Aos amigos Andréa, Cláudio, Leandro Ledel, Ludimila e Mábia por estarem ao meu lado nos momentos mais importantes dessa caminhada, oferecendo valiosas sugestões, palavras de incentivo e principalmente amizade e carinho.

À minha princesinha linda, "Maluzinha", pela companhia sempre presente durante todo o período do mestrado, pela colaboração nos ensaios de minhas apresentações e pelo carinho demonstrado através de recepções festivas e muitos beijos.

À Bila, por me apresentar a "fórmula mágica" de levar a vida de maneira leve e tranqüila.

Aos amigos Jackie e Paulinho, Ângelo e Carla, Alessandro (DT) e Carlinha, Jorge, Marcos (DSEE), Irênio (DSEE), Jane, Michel, Marina, Ivana, Gizele, Igor, Ivete, Rossano, Luis Gustavo, Leandro Loss, Alessandro (LCA), Dalton, Débora e Quintino, pela amizade e boa vontade nos momentos em que precisei da ajuda de vocês.

Aos amigos de todas as horas, André, Antônio, Eliene, Léo Calado, Liu (Janú), Marisa, Thales, Railer e Sheily que mesmo distantes mantiveram-se sempre ao meu lado me apoiando e torcendo pelo meu sucesso.

À Capes pelo apoio financeiro.

”Sei que meu trabalho é uma gota no oceano,

mas sem ele, o oceano seria menor”.

Madre Teresa de Calcutá

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação e relevância

Informação atualmente representa um fator chave no sucesso de qualquer tipo de negócio. O crescimento exponencial do conteúdo em sistemas de informação, incluindo a *World Wide Web*, disponibiliza uma quantidade enorme de informações sobre os mais diversos assuntos. Por exemplo, através da *Web* é possível ter acesso rápido e cômodo a serviços como comércio eletrônico, pesquisas de dados, e diversão, que são disponibilizados por pessoas e instituições espalhadas por todo o mundo. Apesar dessas vantagens a maioria dos sistemas de informação ainda apresenta problemas para atender as necessidades de seus usuários. A dificuldade de retornar documentos relevantes em uma pesquisa caracteriza um dos principais problemas dos sistemas de informação (Huberman et al., 1998).

Pesquisar informações a partir de mecanismos de busca é o método mais popular em utilização. Existem vários sistemas projetados para facilitar a pesquisa e recuperação de informações por parte do usuário. Os mais conhecidos são os "sistemas de recuperação de informação" (na *Web*, também chamados de "engenhos de busca") . Esses sistemas tipicamente realizam pesquisas com base em palavras-chave fornecidas pelo usuário. Normalmente, obtém-se como resposta um grande número de *URLs* (*Uniform Resource Locator*), o que força o usuário a dedicar uma quantidade significativa de tempo na análise das informações até encontrar aquelas que realmente são

relevantes. Nesse sentido, pesquisas vêm sendo desenvolvidas, procurando incrementar o processo de classificação da informação, a fim de melhorar a velocidade de recuperação e a relevância dos itens retornados em uma pesquisa.

A teoria de conjuntos *fuzzy* tem sido empregada com sucesso em várias aplicações voltadas para a modelagem de sistemas de recuperação de informação. Dentre elas, pode-se citar: indexação e recuperação de documentos, mineração de dados, sistemas de recomendação, problemas de classificação, recuperação de dados distribuídos, etc (Herrera-Viedma e Pasi, 2003). A iniciativa de se utilizar conjuntos *fuzzy* em sistemas de recuperação de informação está relacionada à necessidade desses sistemas em modelar os aspectos de imprecisão (*imprecision*) e indefinição (*vagueness*) caracterizados no processo de recuperação (Kraft et al., 1998).

Uma informação é considerada imprecisa quando ela não é descrita de forma exata. A imprecisão em um sistema de recuperação de informação caracteriza o processo de indexação de documentos. Já a indefinição ocorre quando não são fornecidas informações suficientes sobre um determinado assunto. Esse aspecto é identificado na etapa de formulação da consulta, em que o usuário normalmente não consegue explicitar claramente suas reais necessidades de informação.

O desenvolvimento deste trabalho tem como principal motivação a seguinte questão: qual estratégia poderia ser usada nos sistemas de recuperação de informação de forma a reduzir o número de itens irrelevantes retornados em uma busca convencional?

1.2 Objetivo do trabalho

O objetivo deste trabalho é apresentar um modelo *fuzzy* de recuperação de informação baseado em uma ontologia relacional inspirada no trabalho de Takagi e Kawase (2001). A expectativa é de que a técnica utilizada se apresente como uma solução viável para reduzir a quantidade excessiva de informações irrelevantes que são recuperadas em uma busca convencional. Além disso, são feitas comparações entre o modelo proposto e os modelos Ogawa (Ogawa et al., 1991) e Horng (Horng et al., 2001), que são dois importantes trabalhos baseados em *thesaurus* e redes

fuzzy, respectivamente, para representação e processamento do conhecimento.

1.3 Organização do trabalho

Este trabalho está organizado da seguinte forma: neste Capítulo foi apresentada uma visão geral do problema a ser investigado, o contexto associado ao tema, as motivações e os objetivos a serem atingidos.

O Capítulo 2 descreve as principais características de um sistema de recuperação de informação destacando os modelos clássicos de recuperação abordados na literatura e as medidas de avaliação de desempenho.

Na seqüência, o Capítulo 3 contém uma revisão de literatura sobre os modelos *fuzzy* de recuperação de informação, enfatizando os modelos Ogawa (Ogawa et al., 1991) e Horng (Horng et al., 2001).

A seguir, o Capítulo 4 descreve as características do modelo ontológico relacional *fuzzy* que consiste no principal objetivo deste trabalho.

No Capítulo 5 são apresentadas questões relacionadas à implementação do sistema desenvolvido. São também analisados e discutidos os resultados do modelo proposto em comparação com as abordagens desenvolvidas por Ogawa e Horng.

Finalmente, o Capítulo 6 contém as conclusões, a síntese das contribuições desta dissertação e as sugestões de trabalhos futuros.

Capítulo 2

Recuperação de informação

A recuperação de informação é uma área da computação que considera métodos e técnicas para representação, armazenamento e recuperação de itens de informação. Seu principal objetivo é facilitar o acesso a itens de informação relevantes às necessidades do usuário. Neste trabalho consideramos documentos como itens de informação. Portanto, questões como "Que documentos são relevantes a uma consulta do usuário?" ou "Qual o grau de semelhança entre dois documentos?" motivam a criação de modelos para interpretar e manipular documentos.

Este capítulo descreve as principais características de um sistema de recuperação de informação destacando os modelos clássicos abordados na literatura e as medidas para avaliação de desempenho.

2.1 Sistemas de recuperação de informação

A recuperação de informação tem como objetivo definir sistemas capazes de proporcionar acesso rápido a documentos que satisfaçam as necessidades do usuário. Grande parte dos sistemas de recuperação de informação permite somente o armazenamento e a recuperação de conteúdos textuais presentes nos documentos. O principal objetivo desses sistemas consiste em selecionar itens (documentos) que sejam relevantes à solicitação do usuário (consulta) (Pasi, 2002).

Um sistema de recuperação de informação é basicamente constituído por duas partes: uma responsável pela indexação e armazenamento de documentos e outra responsável pela recuperação, (Figura 2.1). A parte de indexação e armazenamento é encarregada de representar os documentos de acordo com um modelo e armazenar estas informações em uma base de dados. Já a parte de recuperação é encarregada de responder as solicitações dos usuários com documentos da base de dados que sejam relevantes à sua consulta.

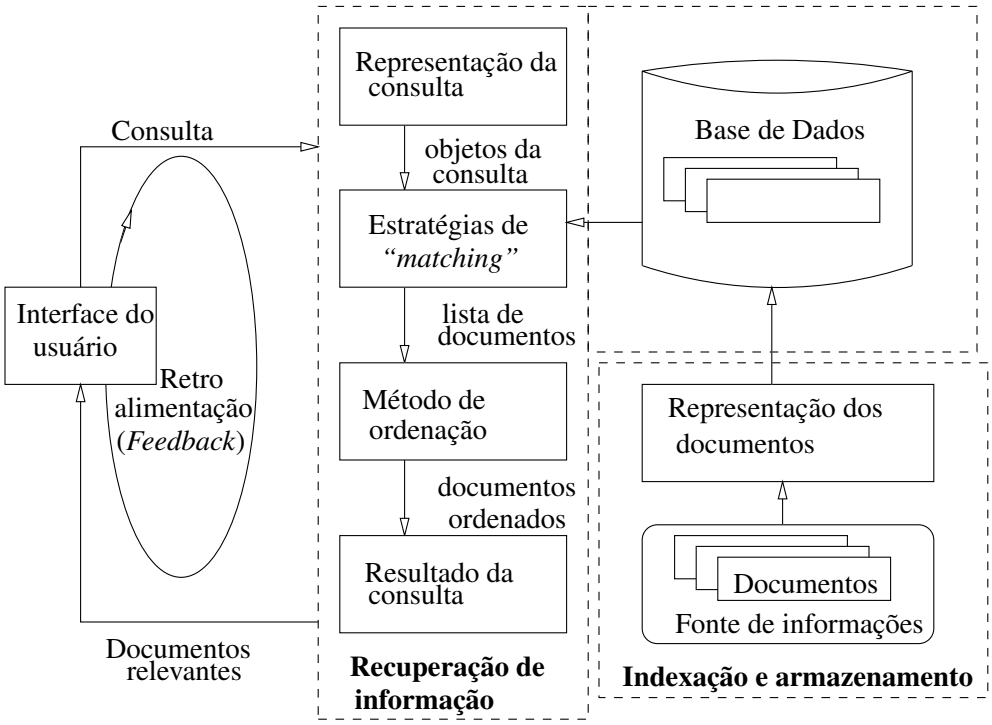


Figura 2.1: Modelo de um sistema de recuperação de informação. Adaptado de Ricarte e Gomide (2001).

Definições

Os principais conceitos relacionados aos sistemas de recuperação de informação são descritos a seguir.

- Palavras-chave

Palavras-chave são palavras com capacidade de descrever semanticamente o conteúdo de um documento. Essas palavras são utilizadas pelos sistemas de recuperação de informação para realizar pesquisas relacionadas a um assunto específico. Elas podem ser combinadas para formar frases-chave, com a finalidade de aumentar a especificidade semântica do termo. Por exemplo, normalmente o número de documentos retornados em uma pesquisa com a palavra "aviação" é bem maior do que o resultado de uma pesquisa por "aviação militar", pois existe maior quantidade de documentos sobre o tema genérico "aviação" do que sobre o tema mais específico "aviação militar". Devido a estas diferenças de quantidade e especificidade dos resultados, a qualidade da resposta no primeiro caso (pesquisa por "aviação") tende a ser menor que no segundo caso (pesquisa por "aviação militar"), do ponto de vista de quem procura por uma página sobre "aviação militar" (Campos e Bax, 2002).

- Documentos relevantes

  Documento é um termo utilizado para denotar um registro textual (um texto). O termo "documentos relevantes" está relacionado ao julgamento do usuário sobre aqueles documentos que melhor expressam a informação desejada.

- Retro-alimentação por relevância (*relevance feedback*)

  Retro-alimentação por relevância é um processo de reformulação de consultas que utiliza informações fornecidas pelo usuário para torná-las mais precisas. Essa técnica só pode ser aplicada após o usuário realizar sua primeira consulta. No ciclo de retro-alimentação por relevância o sistema de recuperação de informação pede que o usuário examine a lista de documentos retornados e selecione aqueles que ele considera mais relevantes. Na seqüência, o sistema altera a consulta do usuário, modificando os termos de acordo com o conteúdo dos documentos selecionados. O efeito esperado é que as consultas subseqüentes passem a retornar documentos cada vez mais relevantes, descartando os irrelevantes (Baeza-Yates e Ribeiro-Neto, 1999).

- Ordenação de documentos (*ranking*)

O resultado esperado de uma pesquisa em um sistema de recuperação de informação consiste de uma lista ordenada composta pelos documentos mais relevantes à consulta do usuário.

Por exemplo, como critério de ordenação de documentos o Google[1] utiliza em sua estrutura um algoritmo denominado *PageRank* que continuamente usa informações do número de páginas que apontam para uma outra página *Web*, ou seja, cada página obtém uma "nota" com base no número de páginas que fizerem referência a ela . Além disso, é utilizado por esse mecanismo de busca os textos dos *links* como descritores da página (Brin e Page, 2002). O algoritmo *PageRank* possibilita a recuperação de documentos na *Web* que ainda não foram indexados. Entretanto, se a consulta do usuário não puder ser relacionada a pelo menos um documento *Web*, a utilização desse mecanismo torna-se inviável (Kruschwitz, 2001).

Um outro exemplo de ordenação consiste na identificação de dois tipos importantes de páginas *Web*: *authorities* e *hubs*. *Authorities* são páginas que contêm informações relevantes sobre um determinado assunto e *hubs* são aquelas que contêm uma coleção de *links* que apontam para os *authorities*. O princípio básico desse método consiste em páginas *hub* bem conceituadas apontarem para muitas páginas *authority*, e páginas *authority* bem conceituadas serem apontadas por muitas páginas *hub* (Lawrence e Giles, 1999; Henzinger, 2000).

2.2 Modelos para recuperação de informação

Segundo Baeza-Yates e Ribeiro-Neto (1999) um modelo de recuperação de informação é definido pela quádrupla [D, Q, f,$R(q_i, d_{doc})$] onde:

- D: conjunto das representações dos documentos na coleção;
- Q: conjunto das representações das consultas;
- f: *framework* para a modelagem dos documentos, consultas e seu relacionamento;

[1] Mecanismo de busca voltado para recuperação de informação na Web

- $R(q_i, d_{doc})$: função de ordenação que associa a cada consulta $q_i \in Q$ e cada $d_{doc} \in D$, um número real que representa a similaridade entre o documento e a consulta.

Os modelos clássicos abordados na literatura para recuperação de informação são os modelos booleano, vetorial e probabilístico. As principais características desses modelos são descritas a seguir. Maiores detalhes podem ser encontrados em Baeza-Yates e Ribeiro-Neto (1999), Wiesman et al. (1997), Russell e Norvig (2003), Jones (1990) e Crestani et al. (1998).

Modelo booleano

O modelo booleano é um modelo de recuperação de informação simples baseado na teoria de conjuntos e na álgebra booleana. Nesse modelo documentos e consultas são representados por conjuntos de palavras.

Ao realizar uma pesquisa o usuário deve especificar na consulta as palavras (elementos) que os documentos (conjunto) resultantes devem apresentar para que sejam retornados. Assim, aqueles documentos que fizerem interseção com a consulta (possuírem as mesmas palavras) serão retornados.

Os operadores "AND" (interseção), "OR" (união) e "NOT" (negação ou exclusão) são operadores comumente usados na formulação de consultas booleanas. Nesse caso o conjunto de documentos relevantes é o conjunto que satisfaz as restrições da consulta.

Apesar da simplicidade, o modelo booleano apresenta alguns inconvenientes:

- a necessidade de informação (consulta do usuário) precisa ser traduzida numa expressão booleana, o que é difícil para os usuários;
- não disponibilizam um método de ordenação (*ranking*) de documentos;
- a busca é baseada em uma decisão binária sem a noção de relação parcial.

Modelo vetorial

O modelo vetorial propõe um *framework* que permite o relacionamento parcial entre documentos e consulta. A partir desse mecanismo, o conjunto de documentos obtidos como resultado pode ser ordenado segundo um critério de relevância baseado no grau de similaridade entre documentos e consulta.

Nesse modelo uma consulta do usuário q e cada documento d_{doc} da coleção são representados pelos vetores $\vec{q}$ e $\vec{d_{doc}}$, respectivamente. Os vetores $\vec{q}$ e $\vec{d_{doc}}$ são formados por todas as palavras indexadas da coleção de documentos e cada palavra possui um valor associado que indica seu peso (grau de importância) para a consulta e documento. Logo, as palavras que não estiverem presentes na consulta recebem grau de importância zero no vetor $\vec{q}$. O mesmo ocorre no vetor $\vec{d_{doc}}$ para as palavras ausentes no documento d_{doc}. Os pesos das outras palavras são calculados através de uma fórmula de importância. Em geral, as fórmulas para calcular o grau de importância de um termo para um documento se baseiam na freqüência do termo no documento. Isso faz com que os termos com peso próximo de um (1) sejam caracterizados como relevantes para o documento e os termos com peso próximo de zero (0) como irrelevantes.

Assim, documentos e consulta no modelo vetorial são representados por vetores t-dimensionais (Figura 2.2), sendo t o número de termos indexados. O grau de similaridade entre cada documento d_{doc} da coleção e a consulta q do usuário pode ser calculado, por exemplo, pelo cosseno do ângulo entre os vetores $\vec{d_{doc}}$ e $\vec{q}$ dado por:

$$sim(d_{doc}, q) = \frac{\sum_{k=1}^{t} W_{k,doc} \times W_{k,q}}{\sqrt{\sum_{k=1}^{t} W_{k,doc}^2} \times \sqrt{\sum_{k=1}^{t} W_{k,q}^2}}$$

sendo $W_{k,doc} \geq 0$ o peso do termo k no documento d_{doc}; $W_{k,q} \geq 0$ o peso do termo k na consulta q, $k = 1, ..., t$, e $sim(d_{doc}, q) \in [0, 1]$.

Depois dos graus de similaridade terem sido calculados, os documentos são listados em ordem decrescente dos valores de $sim(d_{doc}, q)$. Os documentos que obtiverem maior grau de similaridade são considerados os mais relevantes à consulta.

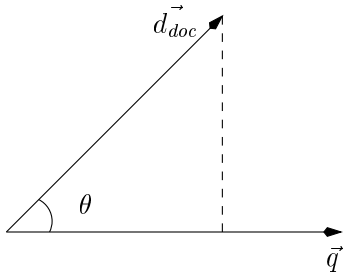


Figura 2.2: O cosseno de θ representa a equação $sim(d_{doc}, q)$.

Modelo probabilístico

O modelo probabilístico baseia-se na teoria da probabilidade para representar documentos e consultas. O princípio básico desse modelo consiste em estimar a probabilidade do usuário encontrar um documento d_{doc} que seja relevante a uma consulta q. O modelo assume que a probabilidade de relevância depende apenas da representação da consulta e do documento. Outra suposição considerada pelo modelo é que existe um subconjunto R de documentos da coleção que são relevantes para a consulta q. Os documentos que não pertencerem a esse subconjunto formam o conjunto $\bar{R}$ dos documentos não relevantes.

Dada uma consulta q, o modelo probabilístico atribui para cada documento d_{doc} da coleção uma medida de similaridade (Eq. 2.1) que calcula a probabilidade do documento d_{doc} ser relevante a consulta q.

$$sim(d_{doc}, q) \sim \sum_{i=1}^{t} W_{i,q} \times W_{i,j} \times \left[log \frac{P(k_i|R)}{1 - P(k_i|R)} + log \frac{1 - P(k_i|\bar{R})}{P(k_i|\bar{R})} \right] \tag{2.1}$$

$W_{i,q} \in \{0, 1\}$ é o peso do termo i para a consulta q; $W_{i,doc} \in \{0, 1\}$, o peso do termo i para o documento d_{doc}; $P(k_i|R)$, a probabilidade do termo k_i estar presente em um documento do conjunto R selecionado aleatoriamente e $P(k_i|\bar{R})$ é a probabilidade do termo k_i estar presente em um documento do conjunto $\bar{R}$ selecionado aleatoriamente.

Uma das dificuldades encontradas na utilização desse modelo consiste em definir um método para calcular o valor inicial das probabilidades de $P(k_i|R)$ e $P(k_i|\bar{R})$. Algumas alternativas são definidas por Baeza-Yates e Ribeiro-Neto (1999), Crestani et al. (1998), Russell e Norvig

(2003) e Wiesman et al. (1997).

Modelo *fuzzy*

O modelo *fuzzy* é um modelo de representação baseado na teoria de conjuntos *fuzzy*. Dentre outros, ele é considerado uma generalização do modelo booleano apropriado para lidar com informações imprecisas.

Para melhor entendimento da teoria de conjuntos *fuzzy* pode-se recorrer à teoria clássica dos conjuntos. Nesta teoria, o conceito de pertinência de um elemento a um conjunto é definido da seguinte forma: elementos de um conjunto A em um determinado universo U simplesmente pertencem ou não pertencem àquele conjunto. Isto pode ser expresso pela função característica:

$$A(x) : U \Rightarrow \{0, 1\}$$

$$A(x) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases}$$

Na teoria de conjuntos *fuzzy*, proposta por Zadeh (1965), generaliza-se a função característica de modo que ela possa assumir um valor no intervalo [0,1]. Assim, um conjunto *fuzzy* A em U é definido por uma função de pertinência $A(x) : U \Rightarrow [0, 1]$, representados por um conjunto de pares ordenados:

$$A = \{x/A(x), x \in U\}$$

Para melhor compreensão do uso do modelo *fuzzy* em sistemas de recuperação de informação, alguns conceitos relacionados à teoria de conjuntos *fuzzy* e lógica *fuzzy* são introduzidos (Pedrycz e Gomide, 1998; Klir e Yuan, 1995).

Operações de conjuntos *fuzzy*

As operações de interseção (Eq. 2.2), união (Eq. 2.3) e complemento (Eq. 2.4) de conjuntos *fuzzy* são definidas como:

$$(A \cap B)(x) = \min[A(x), B(x)] \tag{2.2}$$

$$(A \cup B)(x) = \max[A(x), B(x)] \tag{2.3}$$

$$\bar{A}(x) = 1 - A(x) \tag{2.4}$$

para todo $x \in X$, sendo A e B conjuntos *fuzzy* do universo X.

As operações de interseção *fuzzy* e união *fuzzy* podem ser generalizadas utilizando operadores do tipo t-normas e s-normas, respectivamente.

O operador $\min$ utilizado na Eq. 2.2 é um exemplo de uma t-norma, enquanto que o operador $\max$ da Eq. 2.3 é um exemplo de uma s-norma. Outros exemplos e mais informações sobre t-normas e s-normas podem ser encontrados em Pedrycz e Gomide (1998) e Klir e Yuan (1995).

Operadores lógicos *fuzzy*

Os operadores lógicos *fuzzy* $\wedge$ (AND), $\vee$ (OR) e $\neg$ (NOT) são definidos como:

$$p \wedge q = \min(p, q)$$

$$p \vee q = \max(p, q)$$

$$\neg p = 1 - p$$

sendo p e $q \in [0, 1]$.

Relação *fuzzy*

Uma relação clássica representa a presença ou ausência de associação ou interação entre os elementos de dois ou mais conjuntos. A generalização desse conceito define uma relação *fuzzy* que admite a noção de associação parcial entre os elementos,

$$R\colon X \times Y \to [0,1] \tag{2.5}$$

sendo X e Y universos $R(x,y) \in [0,1]$, $x \in X$ e $y \in Y$.

A definição das operações básicas de interseção (Eq. 2.6), união (Eq. 2.7) e complemento (Eq. 2.8) com relações *fuzzy* são definidas para todo $x \in X$ e $y \in Y$ como:

$$(R \cap W)(x,y) = \min\left[R(x,y), W(x,y)\right] \tag{2.6}$$

$$(R \cup W)(x,y) = \max\left[R(x,y), W(x,y)\right] \tag{2.7}$$

$$\bar{R}(x,y) = 1 - R(x,y) \tag{2.8}$$

sendo R, W relações *fuzzy* no universo $X \times Y$. Quando os universos X e Y são discretos e finitos, uma relação *fuzzy* R é caracterizada por uma matriz $R \in [r_{ij}]$ em que $r_{ij} \in [0,1]$ é o grau da relação entre $x_i \in X$ e $y_j \in Y$.

Composição max-min

Sejam R, G e W relações *fuzzy* definidas no universo de discurso $X \times Z$, $X \times Y$ e $Y \times Z$, respectivamente. A composição max-min da relação G e W, denotada por $G(x,y) \circ W(y,z)$, produz uma relação $R(x,z)$ definida como:

$$R(x,z) = [G \circ W](x,z) = \max_{y \in Y} \min\left[G(x,y), W(y,z)\right] \tag{2.9}$$

para todo $x \in X$ e todo $z \in Z$.

Suponha que os universos de discurso G e W sejam definidos pelas seguintes relações *fuzzy*:

$$G = \begin{bmatrix} 0.5 & 0.2 \\ 0.1 & 0.7 \end{bmatrix}$$

$$W = \begin{bmatrix} 0.3 & 0.6 \\ 0.8 & 0.9 \end{bmatrix}$$

O resultado da composição max-min da relação G e W e dada pela relação R tal que:

$$R = \begin{bmatrix} \vee(\wedge(0.5, 0.3), \wedge(0.2, 0.8)) & \vee(\wedge(0.5, 0.6), \wedge(0.2, 0.9)) \\ \vee(\wedge(0.1, 0.3), \wedge(0.7, 0.8)) & \vee(\wedge(0.1, 0.6), \wedge(0.7, 0.9)) \end{bmatrix} = \begin{bmatrix} 0.3 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \tag{2.10}$$

sendo "$\vee$" o operador $\max$ e "$\wedge$" o operador $\min$.

Fecho transitivo

Uma relação R na teoria clássica de conjuntos é chamada transitiva se e somente se $(x, z) \in R$ sempre que $(x, y) \in R$ e $(y, z) \in R$ para pelo menos um $y \in X$.
Uma relação *fuzzy* R é transitiva se:

$$R(x, z) \geq \max_{y \in Y} \min [R(x, y), R(y, z)] \tag{2.11}$$

A definição de transitividade (Eq. 2.11) é baseada no conceito de composição max-min (Eq. 2.9). Além desta, existem definições alternativas baseadas nos conceitos de t-normas e s-normas (Klir e Yuan, 1995).

O fecho transitivo R_T de uma relação *fuzzy* R pode ser determinado através da execução de um algoritmo simples de três passos:

Passo 1 - $R' = R \cup (R \circ R)$

Passo 2 - Se $R' \neq R$, fazer $R = R'$ e voltar ao passo 1

Passo 3 - Critério de parada: R' igual a R. Logo, $R_T = R'$, sendo R_T o fecho transitivo da relação *fuzzy* R.

O tipo de composição e o conjunto união do passo 1 devem ser compatíveis com a definição de transitividade empregada.

Por exemplo, os passos para calcular o fecho transitivo R_T da relação *fuzzy* R (2.10) é dado por:

Passo 1 -

$$R' = \begin{bmatrix} 0.3 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \cup \left(\begin{bmatrix} 0.3 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0.3 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \right) = \begin{bmatrix} 0.5 & 0.5 \\ 0.7 & 0.7 \end{bmatrix}$$

Passo 2 - $R' \neq R$? Verdade. Logo, $R = R'$;

Passo 1 -

$$R' = \begin{bmatrix} 0.5 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \cup \left(\begin{bmatrix} 0.5 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0.5 & 0.5 \\ 0.7 & 0.7 \end{bmatrix} \right) = \begin{bmatrix} 0.5 & 0.5 \\ 0.7 & 0.7 \end{bmatrix}$$

Passo 3 - R' é igual a R? Verdade. Logo, $R_T = R'$

2.3 Medidas de avaliação

As métricas usadas para avaliar o desempenho e os resultados de um sistema de recuperação de informação são: cobertura (*recall*) e precisão (*precision*). Para que essas medidas sejam relevantes é necessário conhecer bem o conteúdo dos documentos manipulados, ou seja, examinar cada documento da coleção.

Para melhor entendimento, considere uma consulta I do usuário e o conjunto R de documentos relevantes para essa consulta. Seja $|R|$ o número de documentos presentes no conjunto R. Assuma que uma dada estratégia de busca processe a consulta I e retorne um conjunto de

documentos A como resposta. Seja $|A|$ o número de documentos presentes no conjunto A. Seja $|Ra|$ o número de documentos da interseção dos conjuntos R e A. Através da Figura 2.3 pode-se visualizar a relação entre os conjuntos descritos anteriormente.

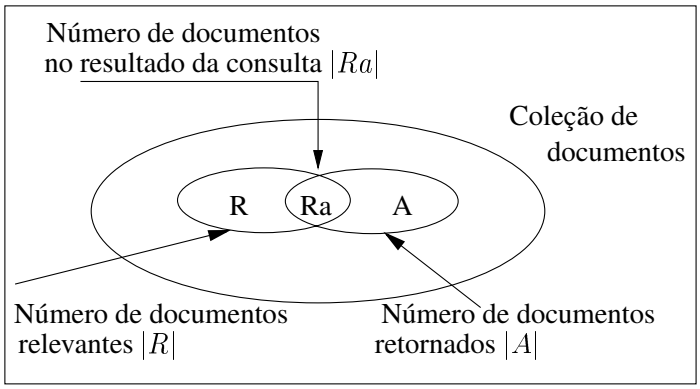


Figura 2.3: Conjuntos R, A e Ra. Adaptado de Baeza-Yates e Ribeiro-Neto (1999).

As medidas de cobertura (*recall*) e precisão (*precision*) são definidas como:

- Cobertura

  É a fração de documentos do conjunto R que foram recuperados.

  $$Cobertura = \frac{|Ra|}{|R|}$$

  Para que essa medida possa ser aplicada, o sistema ou usuário que está analisando o resultado deve saber quantos documentos relevantes à consulta existem na base de dados.

- Precisão

  É a fração de documentos recuperados do conjunto A que são relevantes.

  $$Precisão = \frac{|Ra|}{|A|}$$

  Essa medida avalia a habilidade do sistema em manter os documentos irrelevantes fora do resultado de uma consulta. A precisão é capaz de avaliar o esforço (*overhead*) do usuário para analisar o resultado de uma busca, ou seja, quanto maior a precisão menor o esforço do usuário.

Padrão de 11 níveis de cobertura (*11 standard recall levels*)

Avaliar os resultados de um sistema de recuperação de informação usando a representação de gráficos que exibem os valores da precisão média *versus* cobertura tem sido uma estratégia bastante usada na literatura de recuperação de informação. Esse tipo de representação é normalmente baseado no padrão de 11 níveis de cobertura (*11 standard recall levels*).

Um exemplo mostrando como calcular os valores da precisão média *versus* cobertura para várias consultas q_i distintas é apresentado a seguir (Baeza-Yates e Ribeiro-Neto, 1999).

Seja D_{q_1} o conjunto (Eq. 2.12) formado por todos os documentos relevantes para a consulta q_1.

$$D_{q_1} = \{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\} \tag{2.12}$$

Seja L a lista ordenada dos documentos retornados por um algoritmo de busca para a consulta q_1:

1. $d_{123}*$	6. d_9*	11. d_{38}
2. d_{84}	7. d_{511}	12. d_{48}
3. $d_{56}*$	8. d_{129}	13. d_{250}
4. d_6	9. d_{187}	14. d_{113}
5. d_8	10. $d_{25}*$	15. d_3*

Para analisar a lista de resultados L apenas os itens relevantes à consulta q_1 (itens sinalizados com um "*") são levados em consideração. O documento d_{123}, localizado na posição 1, é relevante para q_1 e corresponde a 10% dos documentos presentes em D_{q_1}. Logo, a precisão nesse caso é de 100% para uma cobertura de 10%. O próximo documento relevante, d_{56}, está localizado na posição 3 da lista L. Nesse ponto, a precisão é de $66,6\%$ (dois documentos relevantes em um total de três) para 20% de cobertura (dois documentos visitados dos dez presentes em D_{q_1}). O terceiro documento relevante é o d_9 (posição 6), com precisão de 50% para 30% de cobertura. Na seqüência, tem-se o documento d_{25} como relevante (posição 10), com precisão de 40% para 40% de cobertura. O último documento relevante retornado é o d_3 (posição 15), com precisão de

$33,3\%$ para 50% de cobertura. Esses valores são representados pela curva da Figura 2.4. Ainda analisando a Figura 2.4 tem-se que a precisão no nível de cobertura maior que 50% diminui para 0%. Isso ocorre porque nem todos os documentos relevantes foram recuperados para a consulta q_1. A curva da precisão *versus* cobertura (Figura 2.4) é baseada no padrão de 11 níveis de cobertura que são 0%, 10%, 20%, ..., 100%. O valor da precisão no nível de cobertura 0% é obtido através do procedimento de interpolação definido na Eq. 2.13.

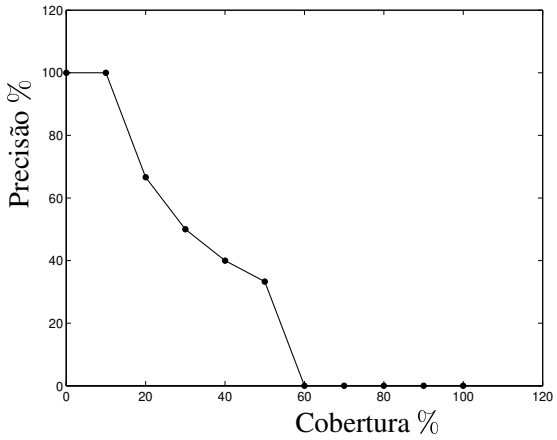


Figura 2.4: Precisão para 11 níveis de cobertura, consulta q_1. Adaptado de Baeza-Yates e Ribeiro-Neto (1999).

Os algoritmos de recuperação de informação são normalmente avaliados para várias consultas distintas. Nesse caso, para cada consulta q_i uma curva com os valores da precisão *versus* cobertura deve ser criada. O desempenho de um algoritmo com base em várias consultas é avaliado pela expressão:

$$\bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

sendo $\bar{P}(r)$ o valor da precisão média no nível r da medida de cobertura, N_q o número de consultas usadas e $P_i(r)$ o valor da precisão no nível r de cobertura para a i-ésima consulta (Baeza-Yates e Ribeiro-Neto, 1999).

Como os níveis de cobertura para cada consulta podem ser diferentes do padrão de 11 níveis, a utilização de um procedimento de interpolação torna-se necessária. Por exemplo, consi-

dere D_{q_2} o conjunto dos documentos relevantes para a consulta q_2:

$$D_{q_2} = \{d_3, d_{56}, d_{129}\}$$

Seja L a lista dos documentos retornados para a consulta q_2 (mesma lista de q_1). Nesse caso, o primeiro documento relevante que aparece na lista de resultado é o d_{56} (posição 3), com precisão de $33,3\%$ para $33,3\%$ de cobertura. O segundo documento relevante é o d_{129} (posição 8), com 25% de precisão para $66,6\%$ de cobertura. O terceiro e último documento relevante é o d_3 (posição 15), com 20% de precisão para 100% de cobertura. O método de interpolação usado para essa situação é definido como segue.

Seja r_j, $j \in \{0, 1, 2, 3, ..., 10\}$, uma referência para o j-ésimo nível de cobertura (ex.: r_5 é uma referência ao nível de cobertura 50%),

$$P(r_j) = \max_{r_j \leq r \leq r_j+1} P(r) \tag{2.13}$$

sendo o valor de interpolação da precisão no j-ésimo nível de cobertura calculado como o maior valor de precisão no intervalo $(j, j+1)$.

A Figura 2.5 exibe a curva interpolada das medidas de precisão *versus* cobertura para a consulta q_2.

A curva com os valores da precisão média *versus* cobertura, considerando as consultas q_1 e q_2, é ilustrada na Figura 2.6. Esse tipo de curva é normalmente usado para comparar o desempenho de algoritmos de busca distintos.

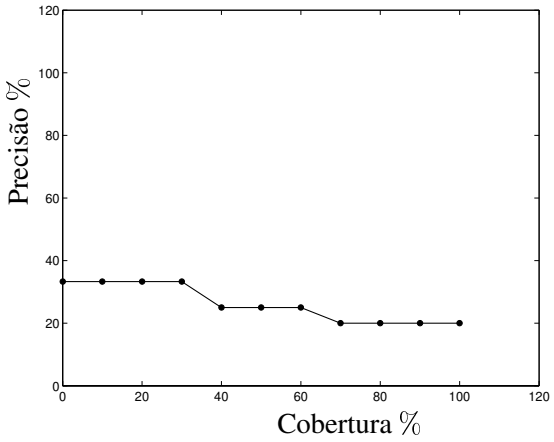


Figura 2.5: Precisão para 11 níveis de cobertura, consulta q_2. Adaptado de Baeza-Yates e Ribeiro-Neto (1999).

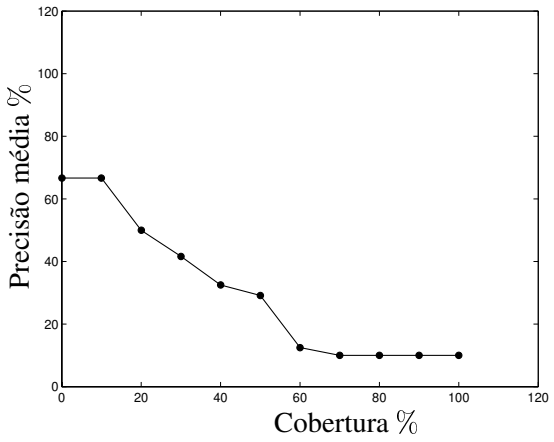


Figura 2.6: Precisão média *versus* cobertura, consultas q_1 e q_2.

2.4 Resumo

Neste capítulo foi apresentada uma visão geral sobre a área de recuperação de informação. Os modelos clássicos de recuperação de informação citados na literatura e suas principais características foram abordados. Além disso, foram resumidas as mais relevantes medidas de avaliação de resultados usadas pelos sistemas de recuperação de informação.

O próximo capítulo aborda o estado da arte sobre o modelo *fuzzy* de recuperação de informação enfatizando os modelos sugeridos por Ogawa et al.(1991) e Horng et al.(2001), modelos estes de especial interesse neste trabalho.

Capítulo 3

Modelos *fuzzy* de Recuperação de Informação

A motivação para a aplicação da teoria de conjuntos *fuzzy* em sistemas de recuperação de informação está relacionada à necessidade desses sistemas em manipular informações imprecisas. Grandes esforços têm sido voltados nos últimos anos para melhorar o desempenho e a qualidade dos resultados dos sistemas de recuperação de informação. Nesse sentido, importantes pesquisas têm sido direcionadas para atuar na modelagem da incerteza e imprecisão que caracteriza o gerenciamento da informação (Herrera-Viedma e Pasi, 2003).

Este capítulo apresenta as principais aplicações da teoria de conjuntos *fuzzy* na modelagem de sistemas de recuperação de informação. Em particular, são detalhados os modelos sugeridos por Ogawa et al.(1991) e Horng et al.(2001).

3.1 Fundamentos

A teoria de conjuntos *fuzzy* tem sido empregada em várias aplicações para modelar sistemas de recuperação de informação. Na indexação de documentos, técnicas *fuzzy* têm sido aplicadas com o objetivo de possibilitar formas de representação de documentos personalizadas e mais específicas do que as geradas pelos procedimentos de indexação tradicionais (Garcés et al., 2003; Bordogna e Pasi, 2002; Pasi, 2002; Bordogna et al., 1993).

Nos sistemas de recomendação, isto é, sistemas que selecionam e sugerem conteúdos de interesse do usuário, os conjuntos *fuzzy* têm mostrado ser uma ferramenta com grande habilidade para avaliar e filtrar os documentos de uma coleção de acordo com as necessidades do usuário (Herrera-Viedma et al., 2003).

Em problemas de classificação, mecanismos de associação *fuzzy* baseados em técnicas de agrupamentos e *thesauri* têm sido usados para melhorar a representação de documentos e consultas (Loia et al., 2003; Haruechaiyasak e Shyu, 2002; Larsen e Yager, 1993).

O foco deste trabalho baseia-se em modelos *fuzzy* que utilizam bases de conhecimento no processo de recuperação de informações relevantes. Uma base de conhecimento pode ser representada, por exemplo, por ontologias *fuzzy*, *thesaurus fuzzy* e redes *fuzzy*. A definição de cada um desses conceitos é introduzida a seguir:

Ontologia *fuzzy*: neste trabalho uma ontologia *fuzzy* é definida como um vocabulário de termos, associados entre si por uma relação *fuzzy* (Eq. 2.5), para representação de um domínio de conhecimento. Alguns dos objetivos das ontologias são permitir que conhecimento de múltiplos "agentes" possa ser compartilhado ou auxiliar as pessoas na compreensão de uma área de conhecimento (Gruber, 1993). Outras definições sobre o conceito de ontologia podem ser encontradas em Gruber (1992) .

Thesaurus fuzzy: um *thesaurus* pode ser definido como um dispositivo de controle terminológico usado na tradução da linguagem natural dos documentos, dos indexadores ou dos usuários para uma "linguagem do sistema mais restrita" (Campos et al., 2002). Um *thesaurus fuzzy* é formado por uma lista estruturada de termos, associados semanticamente entre si por uma relação *fuzzy* (Eq. 2.5), que pode ser usada na recuperação de informações relevantes. As informações semânticas de um *thesaurus* incluem termos sinônimos, termos relacionados ao conceito principal, estrutura hierárquica de definição dos conceitos, etc. Mais informações sobre os *thesaurus* podem ser encontradas em de Jesus (2002).

Redes *fuzzy*: é uma representação, no formato de uma matriz, que define uma rede formada por arcos e nós. Cada nó pode representar um conceito, no caso de uma rede de conceitos

(Horng et al., 2001), ou um documento, no caso de uma rede de citações (Nomoto et al., 1990). Os arcos da rede *fuzzy* definem uma relação *fuzzy* (Eq. 2.5) entre cada par de nós.

Várias abordagens utilizando modelos *fuzzy* baseados em conhecimento são encontradas na literatura (Tomiyama et al., 2003; Takagi e Kawase, 2001; Takagi e Tajima, 2001; Widyantoro e Yen, 2001; Horng et al., 2001; Chen et al., 2001; Klir e Yuan, 1995; Chen e Wang, 1995; Ogawa et al., 1991; Nomoto et al., 1990). Algumas delas são descritas a seguir.

No modelo proposto por Takagi e Kawase (2001), o domínio de conhecimento é representado por uma ontologia *fuzzy*. Essa ontologia é definida como uma organização de palavras em que um conceito é explicado por outros conceitos, ou seja, a ontologia nesse caso é formada por um conjunto de palavras que expressam os vários significados de um conceito. Para definir a região de palavras, na ontologia *fuzzy*, que representa o significado de um conceito de acordo com o contexto de interesse essa abordagem propõe o uso dos conjuntos *fuzzy* conceituais (CFS — *Conceptual Fuzzy Set*). Para testar o efeito dessa técnica dois algoritmos de recuperação de dados são descritos. O primeiro é uma aplicação voltada para a recuperação de imagens. Já o segundo considera um agente que recomenda programas de TV de acordo com as preferências do usuário. Informações adicionais sobre essa abordagem podem ser encontradas em Takagi e Tajima (2001) e Tomiyama et al. (2003).

Uma outra aplicação usando o conceito de ontologia *fuzzy* define um mecanismo de refinamento de consultas. O processo de refinamento consiste em uma técnica onde o sistema de recuperação apresenta ao usuário uma lista de termos como sugestão para substituir os termos de entrada. A ontologia *fuzzy* funciona como uma base de conhecimento que sugere novos termos relacionados aos termos da consulta. Sua estrutura é definida pelas palavras indexadas da coleção de documentos e o grau de relacionamento entre elas. Dois tipos de relacionamento *fuzzy* são fornecidos pela ontologia: generalização e especialização. Esses relacionamentos definem uma relação *fuzzy* (Eq. 2.5). Durante o processo de recuperação o usuário deve selecionar um dos termos sugeridos pela ontologia para substituir o termo antigo. O objetivo dessa técnica é que o processamento da nova consulta seja capaz de retornar documentos mais relevantes (Widyantoro e Yen, 2001).

O modelo Horng (Horng et al., 2001) de recuperação é baseado em um *thesaurus* representado por uma rede de conceitos *fuzzy*. Quatro possíveis relacionamentos entre os conceitos podem ser derivados da rede *fuzzy*: generalização, especialização, associação *fuzzy* positiva e associação *fuzzy* negativa. A descrição detalhada do modelo é apresentada na Seção 3.3.

Uma outra abordagem usando o conceito de redes *fuzzy* (Nomoto et al., 1990) propõe um sistema de recuperação baseado em uma rede de citações. A consulta do usuário, nesse caso, é representada por um documento. Após receber a consulta o sistema segue os seguintes passos:

1. Recupera todos os documentos referenciados ou que referenciam o documento de origem;
2. Constrói a rede de citações *fuzzy*;
3. Determina o grau da relação, $\mu \in [0, 1]$, entre os documentos recuperados e o documento de origem;
4. Ordena os resultados pelo valor μ (ordem decrescente) e apresenta-os ao usuário.

Os detalhes sobre a execução dos passos 1, 2, 3 e 4 são apresentados em Nomoto et al. (1990).

Klir e Yuan (1995) tratam os *thesaurus fuzzy* como uma importante relação na recuperação de informação. A idéia básica consiste em expandir os termos da consulta com termos relacionados obtidos de um *thesaurus*. Um modelo *fuzzy* de recuperação baseado no uso de um *thesaurus* pode ser definido como segue.

Seja A um conjunto *fuzzy* em X representando uma consulta e seja T um *thesaurus fuzzy*. O resultado da composição de A e T é um novo conjunto *fuzzy* em X (conjunto B) que representa a expansão do conjunto A com novos termos relacionados. Logo,

$$B = A \circ T \tag{3.1}$$

sendo $\circ$ a composição max-min.

O conjunto *fuzzy* D definido em Y representa o conjunto de documentos recuperados e é obtido a partir da composição do conjunto B e da relação de relevância R (relaciona conceitos e

documentos). Logo,

$$D = B \circ R \tag{3.2}$$

O processo *fuzzy* de recuperação de informação é representado pelas composições (3.1) e (3.2).

O modelo Ogawa (Ogawa et al., 1991) de recuperação baseia-se em um *thesaurus* construído a partir de uma matriz de correlação de palavras e operações *fuzzy*. A descrição do modelo é apresentada na Seção 3.2.

Os modelos *fuzzy* de recuperação de informação desenvolvidos por Ogawa (Ogawa et al., 1991) e Horng (Horng et al., 2001) foram escolhidos para serem comparados com o modelo ontológico relacional *fuzzy* proposto neste trabalho (Capítulo 4). Essa escolha justifica-se pela necessidade de se avaliar o modelo ontológico com outros modelos *fuzzy* baseados em abordagens diferentes para representação do conhecimento (*thesaurus* e redes *fuzzy*) e recuperação da informação. A descrição dos modelos Ogawa e Horng é apresentada a seguir nas Seções 3.2 e 3.3.

3.2 Modelo Ogawa

O sistema proposto por Ogawa et al. (1991) utiliza um modelo de representação *fuzzy* que faz uso de uma matriz de correlação de palavras usada na recuperação de informações relevantes ao usuário. Esta matriz representa o quanto dois termos ocorrem simultaneamente dentro do conjunto de documentos e é elaborada com base no conjunto de termos indexados. Desta forma, quanto mais dois termos aparecem simultaneamente em um conjunto de documentos, maior o valor $W_{ij} \in [0, 1]$ e a relação entre os termos i e j. A matriz de correlação de palavras é obtida através de:

$$W_{i,j} = \frac{N_{i,j}}{N_i + N_j - N_{i,j}}$$

sendo $W_{i,j}$ o grau de correlação entre as palavras-chave k_i e k_j; $N_{i,j}$ o número de documentos que contém a palavra-chave k_i e a palavra-chave k_j; N_i o número de documentos que contém a palavra-chave k_i; N_j o número de documentos que contém a palavra-chave k_j.

A partir da matriz de correlação de palavras são gerados índices *fuzzy* para cada termo indexado da coleção de documentos. O índice é definido pela seguinte expressão:

$$\mu_{doc,j} = 1 - \prod_{k_k \in A_{doc}} (1 - W_{j,k}) \tag{3.3}$$

sendo $\mu_{doc,j}$ o índice *fuzzy* do termo k_j no documento doc; $W_{j,k}$ o grau de correlação entre as palavras chave k_j e k_k; doc é um documento; j é uma palavra-chave; k representa uma palavra-chave presente no documento doc; k_k é cada uma das palavras-chave presente no documento doc; A_{doc} é o conjunto das palavras-chave do documento doc.

O valor do índice *fuzzy* gerado especifica o grau de compatibilidade entre o termo j e o documento doc. Logo, um documento neste modelo é representado por um conjunto de palavras-chave e os respectivos graus de compatibilidade com o documento.

Para determinar quais documentos estão relacionados a uma determinada consulta são consideradas duas situações: a primeira delas refere-se à busca do usuário por uma única palavra-chave. Neste caso o índice *fuzzy* descrito pela Eq.3.3 é usado apenas no processo de ordenação dos documentos exibidos como resultado. A segunda situação ocorre quando o usuário, na formulação da consulta, utiliza mais de uma palavra-chave ligada pelos operadores lógicos OR, AND e NOT. Nesse caso, a consulta é convertida para a forma normal conjuntiva. Uma consulta q na forma normal conjuntiva é escrita como:

$$q = c(1) \wedge ... \wedge c(N)$$

$$c(h) = k_1 \vee ... \vee k_{nh} \vee \neg k_{nh+1} \vee ... \vee \neg k_{nh+mh}$$

sendo $\wedge$, $\vee$ e $\neg$ os operadores lógicos AND, OR e NOT; k_i representa a i-ésima palavra na consulta; N o número de componentes; $n_h \geq 1$, $m_h \geq 1$ e $n_h + m_h \geq 1$ para o h-ésimo componente da consulta q; $h = 1, ..., N$.

O processo de recuperação é executado em 3 etapas:

1. Geração dos índices fuzzy (Eq. 3.3)

2. Cálculo do valor de relevância para cada componente $c(h)$ da consulta:

$$c_{doc}(h) = 1 - \left(\prod_{k_j \in q(h)^+} S_{doc,j} \right) \left(\prod_{k_j \in q(h)^-} \mu_{doc,j} \right) \tag{3.4}$$

sendo $S_{doc,j} \equiv 1 - \mu_{doc,j} = \prod_{k_k \in A_i} (1 - W_{j,k})$; $c(h)$ o h-ésimo componente da consulta; $q(h)^+$ o conjunto de palavras presentes em h sem o operador NOT e $q(h)^-$ o conjunto de palavras com o operador NOT.

3. Cálculo do valor de relevância da consulta:

$$c_{doc} = \prod_{h=1}^{N} c_{doc}(h) \tag{3.5}$$

sendo N o número de componentes na consulta.

Outra característica dessa abordagem está relacionada ao método de aprendizagem, baseado no $feedback$ do usuário, incorporado ao modelo. Tal discussão não é considerada neste trabalho.

3.3 Modelo Horng

Horng et al. (2001) definem um modelo de recuperação de informação baseado em uma rede de conceitos *fuzzy* usada como base de conhecimento na recuperação de informação. A rede de conceitos *fuzzy* é formada por nós (*nodes*) e arcos (*links*). Cada nó representa um conceito ou um documento, e a cada arco entre dois nós é associada uma tupla (μ, r) que representa o grau de relevância e o tipo de relacionamento entre dois nós. Baseado na arquitetura da rede de conceitos *fuzzy*, quatro tipos de relacionamento podem ser derivados: generalização *fuzzy* (G), especialização *fuzzy* (S), associação *fuzzy* positiva (P) e associação *fuzzy* negativa (N). Um conceito é considerado uma generalização de outro conceito quando o primeiro engloba todos os significados do segundo.

O relacionamento de associação *fuzzy* positiva relaciona conceitos sinônimos ou que possuem significado similar em algum contexto. Por outro lado, o relacionamento de associação *fuzzy* negativa relaciona conceitos antônimos ou complementares (Chen et al., 2001).

A construção automática da rede de conceitos *fuzzy* segue os seguintes passos:

1. Extrair as palavras da coleção de documentos através de algum mecanismo de indexação.

2. Calcular o peso das palavras nos documentos:

$$W_{t,doc} = \frac{\left(0.5 + 0.5\frac{tf_{doc,t}}{\max\ tf_{doc,k}}\right) \log \frac{N}{df_t}}{\max_{j=1,2,\ldots,L} \left\{ \left(0.5 + 0.5\frac{tf_{doc,t}}{\max\ tf_{doc,k}}\right) \log \frac{N}{df_j} \right\}} \tag{3.6}$$

   sendo $tf_{doc,t}$ o valor da freqüência da palavra t no documento doc; df_t o número de documentos que contém a palavra t; L o número de palavras contidas no documento doc; N o número de documentos armazenados na base de dados.

   Analisando o numerador da Eq. 3.6 temos que $\frac{tf_{doc,t}}{\max\ tf_{doc,k}}$ é uma medida da freqüência relativa da palavra t em relação a palavra mais freqüente no documento doc. Essa freqüência relativa é ajustada de forma a ficar no intervalo [0.5, 1.0]. O total é multiplicado por $\log \frac{N}{df_t}$ que representa a importância da palavra t na busca. Logo, o peso de uma palavra em um documento é uma medida relativa do quanto uma palavra está no contexto do documento e da busca.

3. Calcular o peso das palavras para os conceitos:

$$W_{t,con} = \frac{\sum_{i=1}^{m} W_{t,doc}}{m}$$

   sendo m o número de documentos pertencentes ao conceito con.

4. Calcular o grau de relevância entre conceitos e documentos:

$$W_{doc,con} = \frac{\sum_{v=1}^{k} W_{v,doc}}{n}$$

sendo n o número de palavras contidas no conceito c; k o número de palavras comuns no documento doc e conceito con.

5. Determinar o relacionamento e o grau de relevância entre os conceitos.

   O método usado para determinar o tipo de relacionamento e o grau de compatibilidade entre os conceitos da rede *fuzzy* baseia-se na matriz de relevância E,

$$E = \begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_y \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_y \end{array} \\ \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1y} \\ e_{21} & e_{22} & \cdots & e_{2y} \\ \vdots & \vdots & \ddots & \vdots \\ e_{y1} & e_{y2} & \cdots & e_{yy} \end{bmatrix} \end{array}$$

   em que e_{ij} é o grau de relevância entre os conceitos c_i e c_j, $e_{ij} \in [0, 1]$.

   A matriz de relevância E representa inicialmente apenas os relacionamentos de generalização *fuzzy* e associação *fuzzy* positiva. Os outros relacionamentos são obtidos através de regras, definidas no passo 6, baseadas na matriz E^*, sendo $E^* = E_T$ o fecho transitivo da matriz E.

   A fórmula para calcular o grau de generalização *fuzzy* entre dois conceitos é dada por:

$$G(c_i, c_j) = \frac{\sum_{k=1}^{p} \min W_{k,i}, W_{k,j}}{\sum_{k=1}^{p} W_{k,j}}$$

   sendo $W_{k,i}$ o peso da palavra k para o conceito c_i; $W_{k,j}$ o peso da palavra k para o conceito c_j; p o número total de palavras indexadas da coleção de documentos (passo 1); $G\left(c_i, c_j\right) \in [0, 1]$.

   O relacionamento de generalização *fuzzy* entre dois conceitos vai existir quando $G\left(c_i, c_j\right)$ for maior que γ e $G\left(c_j, c_i\right)$ for menor que γ, sendo $\gamma \in [0, 1]$ um limiar definido pelo

projetista do sistema. Se $G(c_i, c_j)$ e $G(c_j, c_i)$ forem ambos maior ou igual a γ, então o relacionamento entre os conceitos é de associação *fuzzy* positiva:

$$P(c_i, c_j) = P(c_j, c_i) = \min(G(c_i, c_j), G(c_j, c_i))$$

6. Construção da rede de conceitos *fuzzy*

   A rede de conceitos *fuzzy* é formada por quatro matrizes U_r, uma para cada tipo de relacionamento. A matriz de relevância U_r é uma matriz *fuzzy*,

$$U_r = \begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_y \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_y \end{array} \\ \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1y} \\ u_{21} & u_{22} & \cdots & u_{2y} \\ \vdots & \vdots & \ddots & \vdots \\ u_{y1} & u_{y2} & \cdots & u_{yy} \end{bmatrix} \end{array}$$

   em que u_{ij} é o grau de relevância entre os conceitos c_i e c_j quando eles estão associados por um tipo de relacionamento $r \in \{P, N, G, S\}$, $u_{ij} \in [0, 1]$.

   Para determinar o tipo de relacionamento e o grau de associação *fuzzy* entre os conceitos, quatro regras foram definidas (Horng et al., 2001):

   - **Regra 1:** Quando o conceito c_i não é antecessor nem sucessor do conceito c_j, mas c_i e c_j têm um antecessor comum, c_h, que é sucessor do conceito c_k, então o relacionamento entre os conceitos c_i e c_j é de associação *fuzzy* positiva, com grau de relevância igual a $\min(E^*(c_h, c_i), E^*(c_h, c_j))$.
   - **Regra 2:** Quando o conceito c_i não é antecessor nem sucessor do conceito c_j e não existe nenhum relacionamento de associação *fuzzy* positiva entre c_i e c_j e o antecessor mais próximo do conceito c_i e c_j é o conceito c_k, então o relacionamento entre os conceitos c_i e c_j é de associação *fuzzy* negativa, com grau de relevância $\min(E^*(c_k, c_i), E^*(c_k, c_j))$.

- **Regra 3:** Se o conceito c_i é um antecessor do conceito c_j, então o conceito c_i é uma generalização *fuzzy* do conceito c_j com grau de relevância $E^*(c_i, c_j)$.
- **Regra 4:** Se o conceito c_i é um sucessor do conceito c_j, então o conceito c_i é uma especialização *fuzzy* do conceito c_j com grau de relevância $E^*(c_j, c_i)$.

A representação da consulta no modelo Horng tem o seguinte formato:

$$q = \{c_t, [c_1, r_1, x_1], [c_2, r_2, x_2], ..., [c_y, r_y, x_y]\},$$

em que c_t representa o domínio de busca; c_i representa um conceito da rede de conceitos *fuzzy*; $r_i \in [G, S, P, N]$ define um tipo de relacionamento e $x_i \in [0, 1]$ indica o grau de relevância desejado pelo usuário do conceito c_i em um documento; $1 \leq i \leq y$, e y é o número de conceitos presentes na rede *fuzzy*.

Na formulação da consulta q, se $x_i = 0$, então o documento desejado pelo usuário não deve possuir o conceito c_i. Além disso, se algum conceito for negligenciado pelo usuário, ou seja, a presença ou ausência desse conceito no documento não influencia o resultado, então esse conceito deve ser rotulado com o símbolo "-". As variáveis x_i e r_i quando $c_i =$ "-", não deverão ser inicializadas.

Durante o processamento de uma consulta o sistema utiliza um mecanismo que possibilita a recuperação de documentos adicionais que não estão diretamente relacionados aos termos de entrada. Esse mecanismo define um algoritmo de expansão do vetor consulta que adiciona novos conceitos da rede *fuzzy* à consulta do usuário e conseqüentemente documentos mais relevantes são recuperados. O algoritmo de expansão do vetor consulta é apresentado em (Algoritmo 3.1, Apêndice).

Por fim, o grau de satisfação dos documentos para a consulta é calculado pela Eq.3.7.

Seja $\overline{D_{doc}}$ o vetor de relevância para o doc-ésimo documento da coleção e $\overline{q^\circ}$ o vetor con-

sulta expandido representados por:

$$\overline{D_{doc}} = < s_{doc,1}, s_{doc,2}, ..., s_{doc,y} >$$

$$\overline{q^{\circ}} = < x_1, x_2, ..., x_y >$$

sendo $s_{doc,i} \in [0,1]$ o grau de relevância, definido no sistema, do conceito c_i no documento d_{doc}; $x_i \in [0,1]$ o grau de relevância, desejado pelo usuário, do conceito c_i no documento d_{doc}; $1 \leq i \leq y$; $1 \leq doc \leq m$; y é o número de conceitos e m é o número de documentos. O grau de satisfação $DS(d_{doc})$ de um documento d_{doc} para uma consulta q é avaliado como:

$$DS(d_{doc}) = \frac{\sum_{\overline{q^{\circ}} \neq \text{“}-\text{''}\ e\ i=1...y} W\left(s_{doc,i}, x_i\right)}{k} \tag{3.7}$$

sendo $DS(d_{doc}) \in [0,1]$; $1 \leq doc \leq y$; k é o número de conceitos da rede *fuzzy* usados pelo usuário na formulação da consulta e $W(s_{ij}, xj) = 1 - |s_{doc,i} - x_i|$ o grau de similaridade entre $s_{doc,i}$ e x_i. O símbolo “-” é usado para denotar os termos negligenciados pelo usuário na formulação da consulta. Quanto maior o valor de $DS(d_{doc})$ maior é o grau de satisfação do documento d_{doc} para a consulta q.

3.4 Resumo

Este capítulo apresentou uma visão das principais aplicações da teoria de conjuntos *fuzzy* na modelagem de sistemas de recuperação de informação. Dentre os modelos pesquisados, destacaram-se os modelos *fuzzy* de recuperação baseados em ontologias, *thesaurus* e redes *fuzzy*. Em particular, foram apresentados os modelos Ogawa (Ogawa et al.,1991) e Horng (Horng et al.,2001).

O próximo capítulo, como proposta principal deste trabalho, apresenta as principais características do modelo ontológico relacional *fuzzy* usado na recuperação de informação.

Capítulo 4

Modelo ontológico relacional *fuzzy*

Este capítulo propõe um modelo ontológico relacional *fuzzy*, que é a principal contribuição deste trabalho.

A definição de uma ontologia relacional *fuzzy* é apresentada na seção 4.2. Na seção 4.3 são descritas as características e o funcionamento do modelo ontológico relacional *fuzzy* em um sistema de recuperação de informação textual.

4.1 Ontologia relacional *fuzzy*

O modelo ontológico relacional *fuzzy* define uma ontologia estruturada em duas camadas. A camada 1 é formada por nomes de categorias. Já a camada 2 contém palavras relacionadas às categorias da camada 1. Os nomes das categorias e palavras foram definidos de acordo com o conteúdo de cada documento da coleção.

A Figura 4.1 ilustra uma ontologia relacional *fuzzy* formada pelas categorias c_1 e c_2 e pelas palavras p_1, p_2 e p_3. Cada categoria c_j é relacionada a uma palavra p_i por um grau de associação *fuzzy*, $r_{ij} \in [0, 1]$. A Figura 4.1 ilustra também os documentos, d_1, d_2, d_3, na ontologia relacional *fuzzy*.

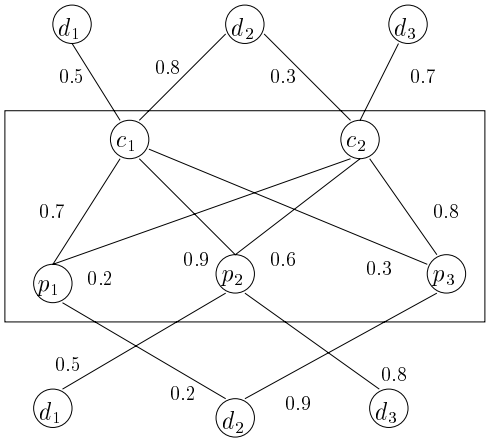


Figura 4.1: Exemplo de uma ontologia relacional *fuzzy*

A ontologia relacional *fuzzy* é um relação representada por uma matriz de relevância R definida como:

$$R = \begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_m \end{array} \\ \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix} \end{array}$$

sendo $1 \leq i \leq n$, n o número de palavras da segunda camada da ontologia; $1 \leq j \leq m$, m o número de categorias da primeira camada; $r_{ij} \in [0, 1]$ o grau de relevância ou da relação entre a palavra p_i e a categoria c_j.

Note que R é uma relação *fuzzy* definida em $P \times C$ onde $P = \{p_1, p_2, \cdots, p_i, \cdots, p_n\}$ e $C = \{c_1, c_2, \cdots, c_j, \cdots, c_m\}$ cujos elementos são enfatizados na coluna à esquerda e na linha no topo da matriz.

4.2 Características do modelo ontológico

O sistema escolhido para dar suporte ao desenvolvimento do modelo ontológico relacional *fuzzy* implementa um mecanismo de busca voltado para recuperação de informação textual (Figura 4.2). Essa escolha justifica-se pelo fato desses sistemas serem largamente utilizados para consultas e apropriados para implementação.

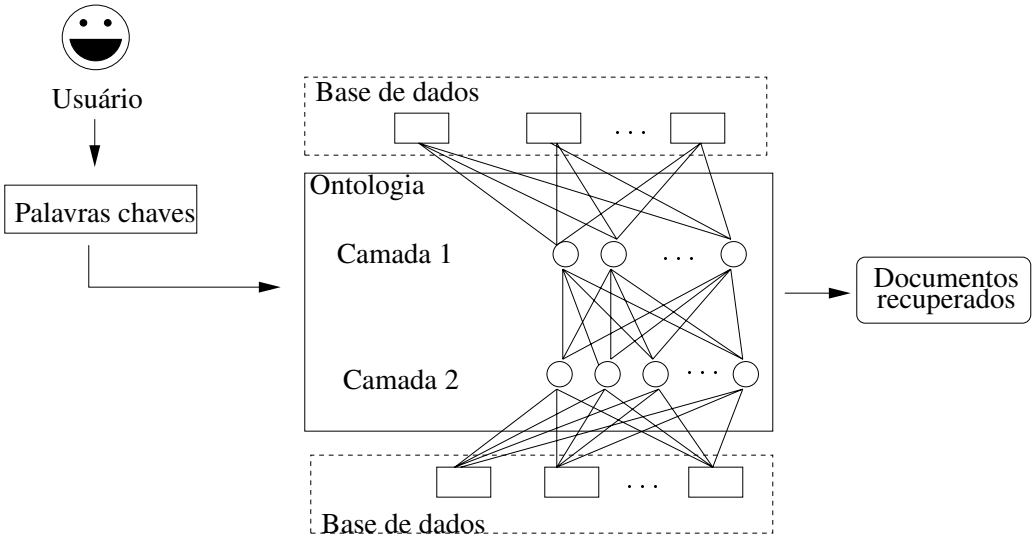


Figura 4.2: Arquitetura do sistema de recuperação de informação.

Na Seção 2.3 foram definidas as principais características de um modelo de recuperação de informação. Com base nisso, será apresentado a seguir a descrição de cada uma dessas características para o modelo ontológico relacional *fuzzy*.

Representação da consulta

Seja $Q = \{p_i, c_j\}$ o conjunto formado por todas as palavras e categorias presentes na consulta do usuário conectados pelos operadores AND e OR, no caso de consultas compostas.

Uma consulta q é representada pelos vetores $x = [x_1, x_2, \cdots, x_i, \cdots, x_n]$, $1 \leq i \leq n$ e $y = [y_1, y_2, \cdots, y_j, \cdots, y_m]$, $1 \leq j \leq m$, tais que:

$$x_i = \begin{cases} 1 & se\ p_i \in q \\ 0 & c.c. \end{cases} \tag{4.1}$$

$$y_j = \begin{cases} 1 & se\ c_j \in q \\ 0 & c.c. \end{cases} \tag{4.2}$$

Representação dos documentos

Seja D o conjunto de documentos da coleção, $D = \{d_1, d_2, \cdots, d_{doc}, \cdots, d_u\}$, P o conjunto de palavras, $P = \{p_1, p_2, \cdots, p_i, \cdots, p_n\}$ e C o conjunto de categorias, $C = \{c_1, c_2, \cdots, c_j, \cdots, c_m\}$. A representação dos documentos é dada pelas matrizes T_p,

$$T_p = \begin{array}{c} \\ d_1 \\ d_2 \\ \vdots \\ d_u \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & \cdots & p_n \end{array} \\ \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{u1} & \alpha_{u2} & \cdots & \alpha_{un} \end{bmatrix} \end{array} \tag{4.3}$$

sendo $\alpha_{doc,i} \in [0, 1]$ o grau de compatibilidade[2] entre o documento d_{doc} e a palavra p_i;$1 \leq doc \leq u$; $1 \leq i \leq n$, e T_c:

$$T_c = \begin{array}{c} \\ d_1 \\ d_2 \\ \vdots \\ d_u \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_m \end{array} \\ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{u1} & \beta_{u2} & \cdots & \beta_{um} \end{bmatrix} \end{array} \tag{4.4}$$

sendo $\beta_{doc,j} \in [0, 1]$ o grau de compatibilidade entre o documento d_{doc} e a categoria c_j; $1 \leq doc \leq u$; $1 \leq j \leq m$.

[2]O grau de compatibilida mede o quanto um termo (palavra ou categoria) caracteriza o conteúdo de um documento

Recuperação de informação usando o modelo ontológico relacional *fuzzy*

Definição 4.3.1: Seja x um vetor que representa as palavras presentes em uma consulta q, conforme (4.1). Seja R uma ontologia relacional *fuzzy*. O resultado da composição de x e R é o conjunto *fuzzy* G_c que representa as categorias relacionadas às palavras presentes na consulta q. Logo,

$$G_c = x \circ R \tag{4.5}$$

Definição 4.3.2: Seja y um vetor que representa as categorias presentes em uma consulta q, conforme (4.2). Seja R uma ontologia relacional *fuzzy*. O resultado da composição de R e y é o conjunto *fuzzy* G_p que representa as palavras relacionadas às categorias presentes na consulta q. Logo,

$$G_p = R \circ y \tag{4.6}$$

As composições max-min (4.5) e (4.6) foram baseadas em (3.1) do modelo *fuzzy* apresentado por Klir e Yuan (1995).

Para determinar quais documentos são relevantes para uma dada consulta três situações devem ser consideradas:

1. Quando o conjunto Q é formado apenas por elementos do conjunto P, ou seja, a consulta do usuário é composta apenas por palavras relacionadas às categorias.

2. Quando o conjunto Q é formado apenas por elementos do conjunto C, ou seja, a consulta do usuário é composta apenas por nomes de categoria.

3. Quando o conjunto Q é formado por elementos do conjunto C e por elementos do conjunto P, ou seja, a consulta do usuário é formada por categorias e palavras.

Ordenação de documentos

Seja $F_p = [f_{p_1}, f_{p_2}, \cdots, f_{p_i}, \cdots, f_{p_n}]$, $1 \leq i \leq n$, um conjunto *fuzzy* derivado de G_p (4.6), e seja $F_c = [f_{c_1}, f_{c_2}, \cdots, f_{c_j}, \cdots, fc_m]$, $1 \leq j \leq m$, um conjunto *fuzzy* derivado de G_c (4.5), tais que:

$$f_{p_i} = \begin{cases} g_{p_i} & se\ g_{p_i} > z_1 \\ 0 & c.c. \end{cases}$$

$$f_{c_j} = \begin{cases} g_{c_j} & se\ g_{c_j} > z_1 \\ 0 & c.c. \end{cases}$$

sendo $g_{p_i} \in G_p$ (4.6); $g_{c_j} \in G_c$ (4.5); z_1 um limiar determinado pelo projetista do sistema.Esse limiar é definido como um nível a partir do qual são selecionados os documentos relevantes para uma consulta.

Definição 4.3.3: O resultado da composição de F_p e T_p (4.3) é o conjunto *fuzzy* V_{DP} que representa o grau de relevância de cada documento da coleção para a palavra p_i. Logo,

$$V_{DP} = F_p \circ T_p \tag{4.7}$$

Definição 4.3.4: O resultado da composição de F_c e T_c (4.4) é o conjunto *fuzzy* V_{DC} que representa o grau de relevância de cada documento da coleção para a categoria c_j. Logo,

$$V_{DC} = T_c \circ F_c^T \tag{4.8}$$

sendo F_c^T a matriz transposta de F_c.

Como critério de ordenação os valores dos vetores V_{DP} e V_{DC} são colocados em ordem decrescente.

4.2.1 Descrição dos métodos para recuperação de informação

Dois métodos são propostos para processar consultas considerando as situações 1, 2 e 3. Como será visto, o segundo método é considerado um caso particular do primeiro. Durante o processo de recuperação de documentos, os métodos 1 e 2 utilizam a composição max-min (2.9) para encontrar as categorias (4.5) e/ou palavras (4.6) da ontologia relacional *fuzzy* que estejam relacionadas aos termos da consulta. Além disso, esse mecanismo é também usado no momento de calcular o grau de relevância dos documentos da coleção para a consulta do usuário (4.7, 4.8). O limiar z_1 é usado na seleção das categorias e/ou palavras mais relacionadas aos termos da consulta, enquanto que o limiar z_2 seleciona os documentos que serão exibidos ao usuário. A descrição dos passos de execução de cada método para as situações 1, 2, e 3 é apresentada a seguir. Cada situação é ilustrada com um exemplo baseado na ontologia relacional *fuzzy* da Figura 4.1 em que:

$$R = \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix}$$

$$T_c = \begin{bmatrix} 0.5 & 0 \\ 0.8 & 0.3 \\ 0 & 0.7 \end{bmatrix}$$

$$T_p = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.2 & 0 & 0.9 \\ 0 & 0.8 & 0 \end{bmatrix}$$

Os valores das matrizes R, T_c e T_p tanto neste exemplo como nos resultados experimentais do Capítulo 5 foram definidos pelo projetista do sistema.

Método 1

Situação 1. $Q = \{p_i\}$, $1 \leq i \leq n$ e $1 \leq |Q| \leq n$

(a) Calcular $Gc = [g_{c_j}]$, conforme (4.5),

(b) Selecionar as categorias $c_j \in G_c$ com $g_{c_j} > z_1$, sendo $z_1 \in [0, 1]$ um limiar definido pelo projetista do sistema e $g_{c_j} \in [0, 1]$ o resultado da composição max-min para a categoria c_j,

(c) Recuperar os documentos da base de dados que pertencem às categorias do passo (b),

(d) Calcular V_{DC}, conforme (4.8),

(e) Ordenar os valores do vetor V_{DC} em ordem decrescente,

(f) Exibir a lista de documentos do passo (c) na ordem definida pelo vetor V_{DC} (passo e) com $v_{DC} > z_2$, sendo z_2 um limiar definido pelo projetista do sistema.

Exemplo 1: $Q = \{p_2\}$, $z_1 = 0.7$, $z_2 = 0.2$, logo $q = p_2$

(a)

$$G_c = x \circ R = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.6 \end{bmatrix}$$

(b) Categorias com $g_{c_j} > 0.7 : c_1$

(c) Documentos que pertencem à categoria $c_1 : d_1$ e d_2

(d)

$$V_{DC} = T_c \circ F_c^T = \begin{bmatrix} 0.5 & 0 \\ 0.8 & 0.3 \\ 0 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0.9 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.8 \\ 0 \end{bmatrix}$$

(e)

$$V_{DC}\, ordenado = \begin{bmatrix} 0.8 \\ 0.5 \\ 0 \end{bmatrix}$$

(f) Resultado com $v_{DC} > 0.2 : d_2, d_1$

Situação 2. $Q = \{c_j\}$, $1 \leq j \leq m$ e $1 \leq |Q| \leq m$

(a) Calcular $Gp = [g_{p_i}]$, conforme (4.6),

(b) Selecionar as palavras $p_i \in G_p$ com $g_{p_i} > z_1$, sendo $z_1 \in [0, 1]$ um limiar definido pelo projetista do sistema e $g_{p_i} \in [0, 1]$ o valor da composição max-min para a palavra p_i,

(c) Recuperar os documentos da base de dados relacionados às palavras do passo (b),

(d) Calcular V_{DP}, conforme (4.7),

(e) Ordenar os valores do vetor V_{DP} em ordem decrescente,

(f) Exibir a lista de documentos do passo (c) na ordem definida pelo vetor V_{DP} (passo e) com $v_{DP} > z_2$, sendo z_2 um limiar definido pelo projetista do sistema.

Exemplo 2: $Q = \{c_1\}$, $z_1 = 0.6$, $z_2 = 0.75$, logo $q = c_1$

(a)

$$G_p = R \circ y = \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.3 \end{bmatrix}$$

(b) Palavras com $g_{p_i} > 0.6 : p_1$ e p_2

(c) Documentos relacionados às palavras p_1 e $p_2 : d_1, d_2$ e d_3

(d)

$$V_{DP} = T_p \circ F_p = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.2 & 0 & 0.9 \\ 0 & 0.8 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.7 \\ 0.9 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.8 \end{bmatrix}$$

(e)

$$V_{DP}\, ordenado = \begin{bmatrix} 0.8 \\ 0.5 \\ 0.2 \end{bmatrix}$$

(f) Resultado com $v_{DP} > 0.75 : d_3$.

Situação 3. $Q = \{p_i, c_j\}$, $1 \leq i \leq n$, $1 \leq j \leq m$

(a) Dividir a consulta do usuário Q em dois subconjuntos: $Q_1 = \{p_i\}$ e $Q_2 = \{c_j\}$,

(b) Calcular $Gc = [g_{c_j}]$ para os elementos de Q_1, conforme (4.5),

(c) Calcular $Gp = [g_{p_i}]$ para os elementos de Q_2, conforme (4.6),

(d) Selecionar as categorias $c_j \in G_c$ (passo b) com $g_{c_j} > z_1$, sendo $z_1 \in [0, 1]$ um limiar definido pelo projetista do sistema e $g_{c_j} \in [0, 1]$ o resultado da composição max-min para a categoria c_j,

(e) Selecionar as palavras $p_i \in G_p$ (passo c) com $g_{p_i} > z_1$, sendo $z_1 \in [0, 1]$ um limiar definido pelo projetista do sistema e $g_{p_i} \in [0, 1]$ o valor da composição max-min para a palavra p_i,

(f) Recuperar os documentos da base de dados que pertencem às categorias do passo (d),

(g) Recuperar os documentos da base de dados relacionados às palavras do passo (e),

(h) Selecionar os documentos do passo (f) que pertencem à(s) categoria(s) presente(s) em Q_2,

(i) Selecionar os documentos do passo (g) relacionados à(s) palavra(s) presente(s) em Q_1,

(j) Se os elementos da consulta Q estiverem conectados pelo operador lógico AND selecionar os documentos que fazem interseção nos passos (h) e (i); caso seja o operador lógico OR pegar todos os documentos dos passos (h) e (i) sem repetição,

(k) Calcular V_{DP}, conforme (4.7),

(l) Calcular V_{DC}, conforme (4.8),

(m) Comparar os valores do vetor V_{DP} e V_{DC} considerando o maior valor de compatibilidade,

(n) Ordenar o resultado do passo (m) em ordem decrescente,

(o) Exibir a lista de documentos do passo (j) ordenada de acordo com os valores do passo (n) maiores que z_2, sendo z_2 um limiar definido pelo projetista do sistema.

Exemplo 3: $Q = \{p_3\ OR\ c_2\}$, $z_1 = 0.4$, $z_2 = 0.75$, logo $q = p_3\ OR\ c_2$

(a) $Q_1 = \{p_3\}$, $Q_2 = \{c_2\}$

(b)

$$G_c = x \circ R = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.8 \end{bmatrix}$$

(c)

$$G_p = R \circ y = \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.8 \end{bmatrix}$$

(d) Categorias com $g_{c_j} > 0.4 : c_2$

(e) Palavras com $g_{p_i} > 0.4 : p_2$ e p_3

(f) Documentos que pertencem à categoria $c_2 : d_2$ e d_3

(g) Documentos relacionados às palavras p_2 e $p_3 : d_1$, d_2 e d_3

(h) Documentos (passo f) que pertencem às categorias de $Q_2 : d_2$ e d_3

(i) Documentos (passo g) relacionados às palavras de $Q_1 : d_2$

(j) Operador lógico OR : d_2 e d_3

(k)

$$V_{DC} = T_c \circ F_c^T = \begin{bmatrix} 0.5 & 0 \\ 0.8 & 0.3 \\ 0 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.3 \\ 0.7 \end{bmatrix}$$

(l)

$$V_{DP} = T_p \circ F_p = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.2 & 0 & 0.9 \\ 0 & 0.8 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0.6 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.8 \\ 0.6 \end{bmatrix}$$

(m)

$$V_{DP,DC}\, ordenado = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.5 \end{bmatrix}$$

(n) Resultado $v_{DP,DC} > 0.75 : d_2$

Método 2

Situação 1. $Q = \{p_i\}$, $1 \leq i \leq n$ e $1 \leq |Q| \leq n$

(a) Calcular $Gc = [g_{c_j}]$, conforme (Eq. 4.5),

(b) Selecionar as categorias $c_j \in G_c$ com $g_{c_j} > z_1$ e armazená-las no conjunto J_c, sendo $z_1 \in [0, 1]$ um limiar definido pelo projetista do sistema e $g_{c_j} \in [0, 1]$ o resultado da composição max-min para a categoria c_j,

(c) Criar para cada categoria do passo (b) uma nova consulta q^* no formato "$q^* = c_k\, AND\, p_i$", sendo k a k-ésima categoria do conjunto J_c e $tam = |J_c|$,

(d) Executar o seguinte algoritmo:

```
for (k=1 to tam)
 for (i=1 to n)
   Executar q* conforme consulta 3, método 1
   Armazenar os documentos na lista L
```

(e) Recuperar todos os documentos da lista L (passo d) sem repetição,

(f) Calcular V_{DC}, conforme (Eq. 4.7),

(g) Ordenar os valores do vetor V_{DC} em ordem decrescente

(h) Exibir a lista de documentos do passo (e) na ordem definida pelo vetor V_{DC} com $v_{DC} > z_2$, sendo z_2 um limiar definido pelo projetista do sistema.

Exemplo 4: $Q = \{p_3\}$, $z_1 = 0.5$, $z_2 = 0.2$, logo $q = p_3$

(a)

$$G_c = x \circ R = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.8 \end{bmatrix}$$

(b) Categorias com $g_{c_j} > 0.5 : c_2$

(c) "$q^* = c_2\, AND\, p_1$", "$q^* = c_2\, AND\, p_2$", "$q^* = c_2\, AND\, p_3$"

(d) L = d_2 - d_3 - d_2

(e) Documentos da lista L: d_2 e d_3

(f)

$$V_{DC} = T_c \circ F_c^T = \begin{bmatrix} 0.5 & 0 \\ 0.8 & 0.3 \\ 0 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.3 \\ 0.7 \end{bmatrix}$$

(g)

$$V_{DC}\, ordenado = \begin{bmatrix} 0.7 \\ 0.3 \\ 0 \end{bmatrix}$$

(h) Resultado com $v_{DC} > 0.2 : d_3, d_2$

Situação 2. $Q = \{c_j\}$, $1 \leq j \leq m$ e $1 \leq |Q| \leq m$

(a) Calcular $Gp = [g_{p_i}]$, conforme (4.6),

(b) Selecionar as palavras $p_i \in G_p$ e $g_{p_i} > z_1$ e armazená-las no conjunto J_p , sendo $z_1 \in [0, 1]$ um limiar definido pelo projetista do sistema e $g_{p_i} \in [0, 1]$ o resultado da composição max-min para a palavra p_i,

(c) Criar para cada palavra do passo (b) uma nova consulta q^* no formato "$q^* = p_k \, AND \, c_j$", sendo k a k-ésima palavra do conjunto J_p e $tam = |J_p|$.

(d) Executar o seguinte algoritmo:

```
for (k=1 to tam)
 for (j=1 to m)
   Executar q* conforme consulta 3, solução 1
   Armazenar os documentos na lista L
```

(e) Recuperar todos os documentos da lista L (passo d) sem repetição,

(f) Calcular V_{DP}, conforme (4.7),

(g) Ordenar os valores do vetor V_{DP} em ordem decrescente

(h) Exibir a lista de documentos do passo (e) na ordem definida pelo vetor V_{DP} com $v_{DP} > z_2$, sendo z_2 um limiar definido pelo projetista do sistema.

Exemplo 5: $Q = \{c_2\}$, $z_1 = 0.7$, $z_2 = 0.75$, logo $q = c_2$, logo $q = c_2$

(a)

$$G_p = R \circ y = \begin{bmatrix} 0.7 & 0.2 \\ 0.9 & 0.6 \\ 0.3 & 0.8 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.8 \end{bmatrix}$$

(b) Palavras com $g_{p_i} > 0.7 : p_3$

(c) "$q^* = p_3 \, AND \, c_1$", "$q^* = p_3 \, AND \, c_2$"

(d) L = d_2 - d_2

(e) Documentos da lista L: d_2

(f)

$$V_{DP} = T_p \circ F_p = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.2 & 0 & 0.9 \\ 0 & 0.8 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.2 \\ 0.6 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.8 \\ 0.6 \end{bmatrix}$$

(g)

$$V_{DP}\, ordenado = \begin{bmatrix} 0.8 \\ 0.6 \\ 0.5 \end{bmatrix}$$

(h) Resultado com $v_{DP} > 0.75 : d_2$.

Situação 3. $Q = \{p_i, c_j\}$, $1 \leq i \leq n$, $1 \leq j \leq m$

Idem a consulta 3, método 1

4.3 Resumo

Este capítulo apresentou as principais características do modelo ontológico relacional *fuzzy* desenvolvido para sistemas de recuperação de informação textual. Além disso, dois métodos para a recuperação de informações relevantes foram propostos.

No capítulo seguinte, os dois métodos baseados no modelo ontológico relacional *fuzzy* são avaliados e comparados aos modelos Ogawa e Horng, descritos no Capítulo 3.

Capítulo 5

Aplicação e análise

Este capítulo apresenta uma discussão sobre os resultados do modelo ontológico relacional *fuzzy* em comparação com os modelos Ogawa e Horng, apresentados no Capítulo 3.

A Seção 5.2 formula o problema através da descrição do domínio da aplicação e discute algumas decisões de implementação. A Seção 5.3 apresenta os experimentos computacionais. E por fim, na Seção 5.4 discute-se os resultados fornecidos pelos métodos comparando-os com base nas medidas de desempenho de cobertura (*recall*) e precisão (*precision*), e tempo de execução.

5.1 Formulação do problema

Definição do domínio

O domínio da aplicação, desenvolvida neste trabalho, é definido por um conjunto de 100 artigos científicos relacionados ao assunto "Inteligência Computacional".

Cada artigo da coleção é classificado em uma ou mais categorias e associado a uma ou mais palavras relacionadas às categorias (processo de indexação). No total foram definidas seis categorias e cinqüenta e cinco palavras para serem usadas na ontologia relacional *fuzzy*.

Dentre as categorias e palavras têm-se:

1. categorias: *"Information Retrieval"*, *"Fuzzy Logic"*, *"Genetic Algorithm"*, *"Agent"*, *"Neural Network"*, *"Web Search"*.

2. palavras: *"query"*, *"keyword"*, *"imprecision"*, *"fuzzy rule"*, *"fitness function"*, *"crossover"*, *"multi-agent system"*, *"neuron"* , *"image retrieval"*, etc.

Os termos indexados da coleção de artigos foram também usados na geração das redes conceituais *fuzzy* do modelo Horng e thesaurus *fuzzy* do modelo Ogawa.

Decisões de implementação

O sistema de recuperação de informação usando o modelo ontológico relacional *fuzzy* foi desenvolvido na linguagem JSP (*Java Server Page*) e *serlvet*, sobre a plataforma Unix, utilizando o Tomcat como servidor de *web* e o PostgresSQL como servidor de banco de dados.

JSP e *servlets* são tecnologias baseadas na linguagem de programação Java para desenvolvimento de aplicações dinâmicas para *web*. A união dessas duas tecnologias tem proporcionado soluções atraentes à programação *web* resolvendo algumas limitações apresentadas pelas linguagens *web* tradicionais (ex.: ASP, PHP).

Vantagens das tecnologias JSP e *servlets*:

- Facilidade de uso
- Independência de plataforma
- Código aberto
- Integração com APIs Java
- Orientação a objetos
- Separação do conteúdo dinâmico da apresentação

Mais informações sobre as linguagens JSP e *servlets* podem ser encontradas em Wutka (2000) e Kurniawan (2002).

O PostgresSQL é um sistema gerenciador de banco de dados relacional, implementado em C/C++, que utiliza o padrão SQL (*Structured Query Language*) para acessar e manipular os dados armazenados. Esse SGBD possui a vantagem de ser um sistema multiplataforma, podendo ser executado sobre os sistemas operacionais como Unix, Linux, Windows 95/98/ME e NT/2000. Outra característica do PostgresSQL é possuir código aberto e distribuição livre.

O Tomcat é um servidor de *web* baseado no protocolo HTTP, desenvolvido como parte do projeto Jakarta da *Apache Software Foundation* (http://jakarta.apache.org). Uma característica importante do Tomcat é poder ser executado como um servidor *stand-alone* ou em conjunto com outros servidores, como Apache e IIS. Além disso, é um servidor de distribuição livre e código aberto.

A arquitetura comumente usada para execução de aplicações para *web* é a arquitetura de três camadas, na qual o *browser* na máquina cliente corresponde à camada de apresentação, o servidor de *web* à camada de aplicação, e o servidor de banco de dados à camada persistente (Figura 5.1).

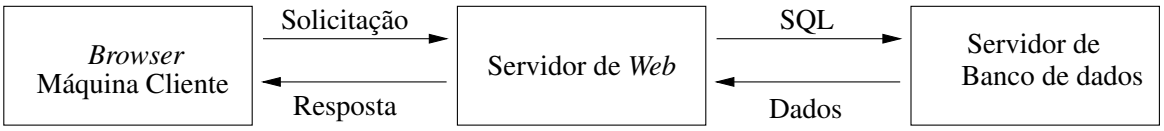


Figura 5.1: Arquitetura de três camadas

A camada de apresentação é responsável pela interface do sistema e é a única camada visível ao usuário. Essa camada permite a solicitação de páginas ou tarefas que devem ser processadas na camada de aplicação.

A camada de aplicação é encarregada de interpretar e processar solicitações e, em seguida, enviar as respostas para a máquina cliente. Além disso, essa camada também atende às requisições que executam alguma operação (inserção, remoção ou modificação) na camada persistente. A

execução dessas operações é definida por comandos escritos em SQL.

A camada persistente é responsável pela manipulação dos dados, armazenamento físico dos objetos em uma base permanente, e pela conexão com o SGBD.

A arquitetura de três camadas (Figura 5.1) é baseada no paradigma MVC (*Model-View-Controller*) que fornece uma maneira de dividir a funcionalidade de uma aplicação separando o conteúdo de geração do conteúdo de apresentação. Nesse paradigma uma aplicação é dividida em três seções (Wutka, 2000; Kurniawan, 2002):

Modelo, responsável pelo armazenamento, manipulação e geração de dados.

Visão, camada de interface com o usuário usada para receber os dados de entrada e apresentar o resultado.

Controlador, responsável por controlar e mapear as ações.

Uma aplicação *web* baseada no paradigma MVC pode ser indicada pela presença de um *servlet* controlador que recebe todas as requisição do *browser* e encaminha cada requisição a uma das páginas JSP (Figura 5.2). As páginas JSP são encarregadas da parte de interface de dados (visão). Já a parte de manipulação de dados (modelo) são definidas por classes Java.

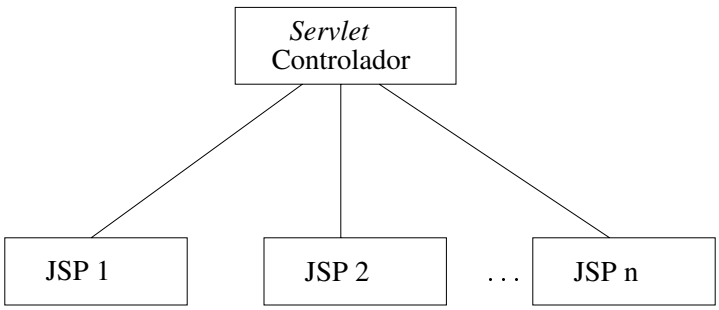


Figura 5.2: Servlet controlador

A aplicação desenvolvida neste trabalho foi baseada no paradigma MVC descrito anteriormente.

5.2 Experimentos computacionais

Nesta seção são exibidos experimentos computacionais para avaliação do modelo ontológico relacional *fuzzy* em comparação com as abordagens propostas por Ogawa e Horng. A análise de cada experimento é baseada nas medidas de avaliação, cobertura (*recall*) e precisão (*precision*), definidas na Seção 2.4.

Para os experimentos aqui apresentados consideraram-se três casos distintos de consultas do usuário. O primeiro e segundo caso são consultas simples formadas por nomes de categorias e palavras, respectivamente. Já o terceiro caso são consultas compostas conectadas pelo operador lógico AND.

Cada situação é representada por três tipos de gráficos com $z_2 = \{0.2, 0.75, 0.95\}$, sendo z_2 um limiar definido pelo projetista do sistema. No primeiro gráfico (Figuras (a)) são comparados os resultados de cobertura (*recall*) e precisão (*precision*) para os métodos 1 e 2. Os dois gráficos seguintes exibem o comportamento do método 1 (Figuras (b)) e método 2 (Figuras (c)) em comparação com as abordagens propostas por Ogawa e Horng.

As curvas dos gráficos foram baseadas em um padrão de 21 níveis de cobertura (0%, 5%, 10%, 15%, ..., 100%). Esse padrão é uma modificação do "*11 standard recall levels*" descrito na Seção 2.3. Nesse caso, o método de interpolação é definido como segue.

Seja r_j, $j \in \{0, 1, 2, 3, ..., 20\}$, uma referência para o j-ésimo nível de *recall* (ex.: r_9 é uma referência ao nível de *recall* 40%):

$$P(r_j) = \max_{r_j \leq r \leq r_j+1} P(r)$$

sendo o valor de interpolação da precisão no j-ésimo nível de cobertura calculado como o maior valor de precisão no intervalo $(j, j+1)$.

Resultados experimentais 1

As Figuras 5.3, 5.4 e 5.5 ilustram os resultados das medidas de cobertura e precisão média para cinco consultas distintas formadas pelos seguintes nomes de categoria: "*Information Retrieval*", "*Genetic Algorithm*", "*Fuzzy Logic*", "*Agent*" e "*Neural Network*".

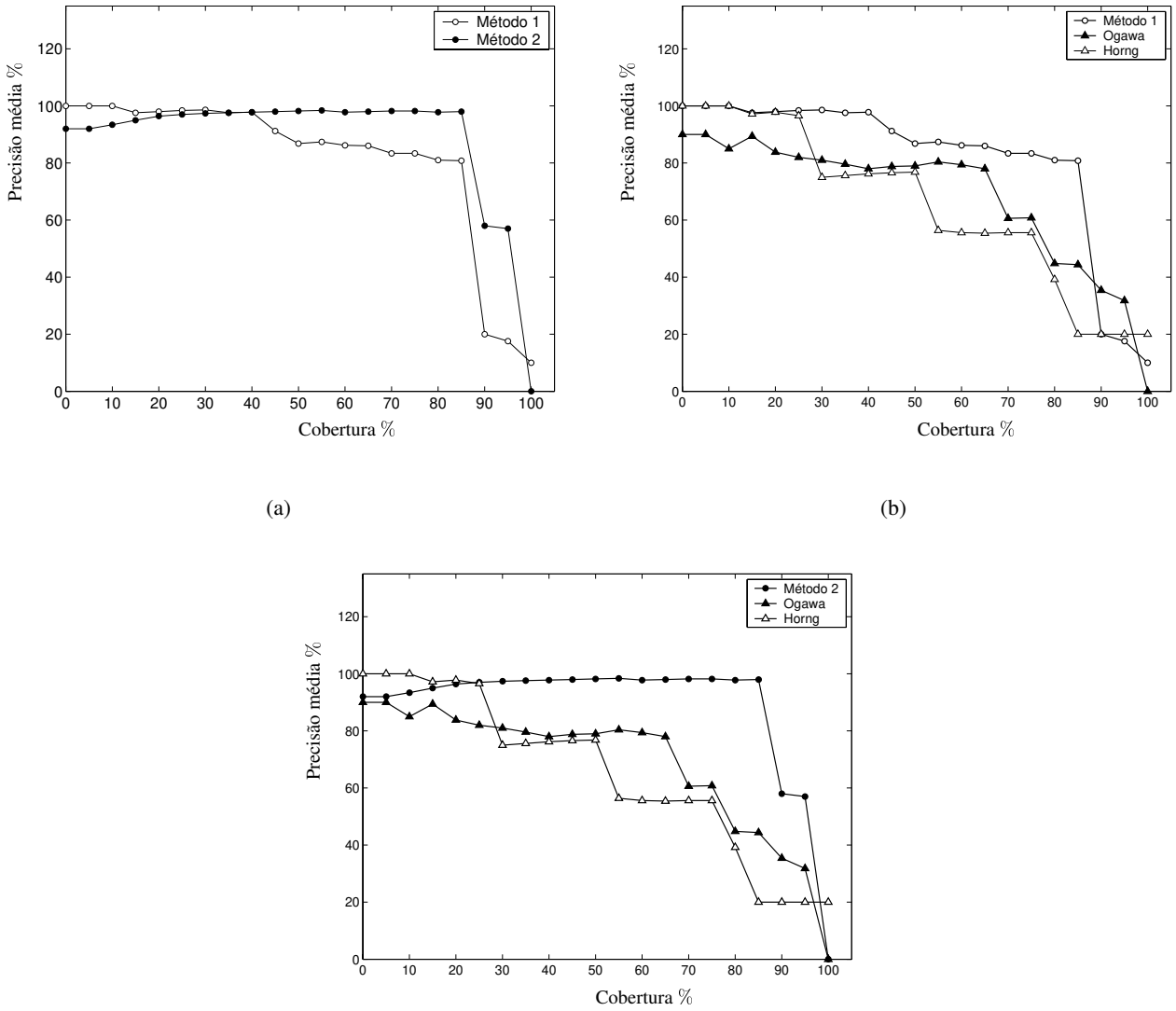


Figura 5.3: Cobertura e precisão média para consultas formadas por nomes de categoria, $z_2 = 0.2$.

Com base nos gráficos da Figura 5.3, pode-se verificar que o valor da precisão média do método 1 (Figura 5.3(b)) e método 2 (Figura 5.3(c)) foi, em geral, mais elevado que os algoritmos de Ogawa e Horng. Na Figura 5.3(a) o método 2 apresentou-se estável nos pontos de cobertura $\in [40, 90]$. Exatamente nesses pontos, o método 2 exibiu valores da medida de precisão média superiores ou iguais aos do método 1 (Figura 5.3(a)).

As Tabelas 5.1, 5.2, 5.3 e 5.4 ilustram o comportamento particular de cada algoritmo em três pontos específicos.

Tabela 5.1: Pontos de desempenho - Método 1, $z_2 = 0.2$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	100	2	2
Desempenho médio	80	81	25-45	20-35
Alto valor de cobertura	100	10	40-69	21-38

Tabela 5.2: Pontos de desempenho - Método 2, $z_2 = 0.2$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	15	95	4-82	4-7
Desempenho médio	85	98	21-38	21-37
Alto valor de cobertura	95	57	23-43	22-39

Tabela 5.3: Pontos de desempenho - Ogawa, $z_2 = 0.2$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	90	2-4	2
Desempenho médio	75	60.8	25-39	19-32
Alto valor de cobertura	95	31.8	34-39	22-32

Tabela 5.4: Pontos de desempenho - Horng, $z_2 = 0.2$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	100	2	2
Desempenho médio	55	56.4	8-28	8-24
Alto valor de cobertura	100	20	8-43	8-42

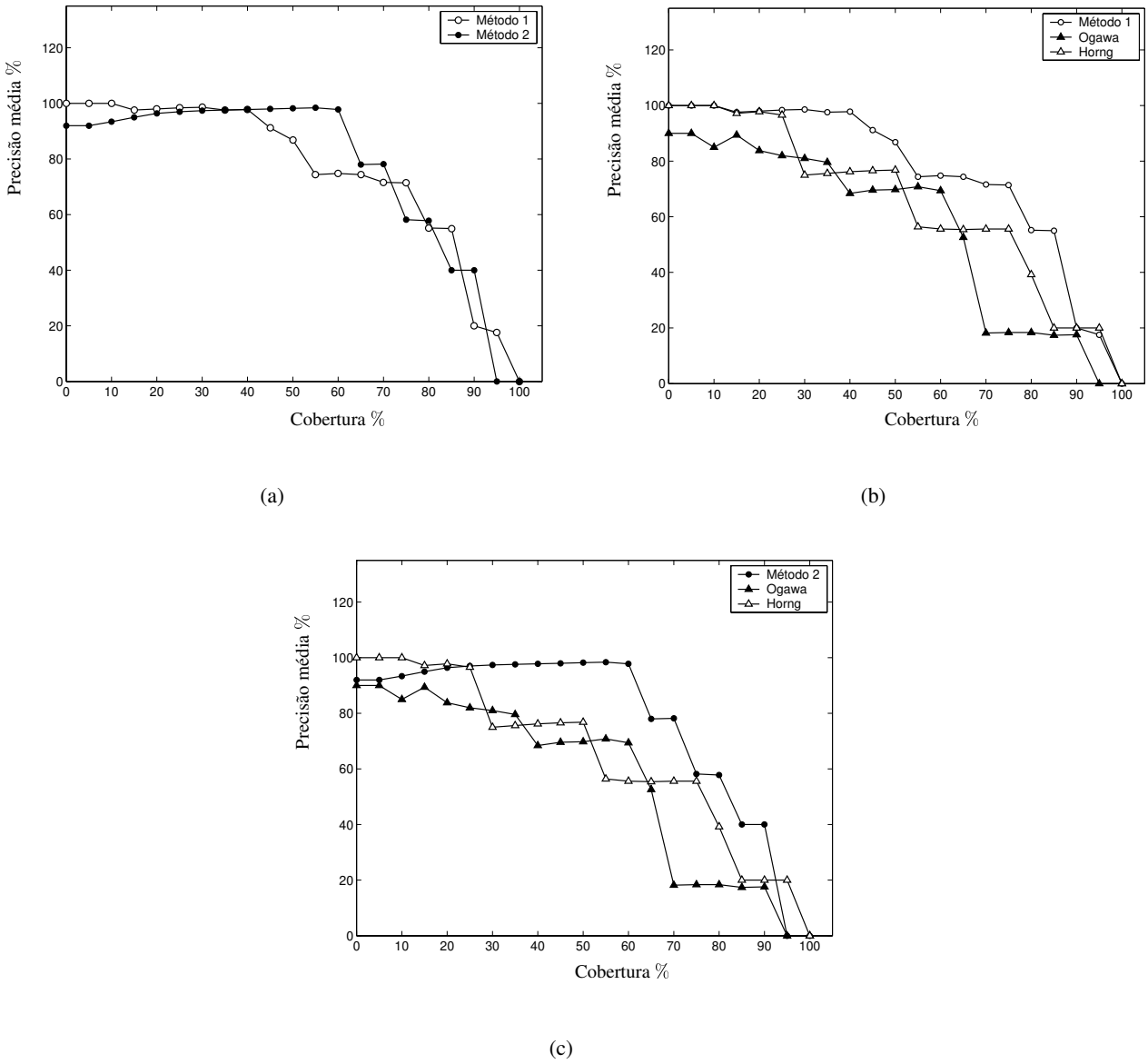


Figura 5.4: Cobertura e precisão média para consultas formadas por nomes de categoria, $z_2 = 0.75$.

Para $z_2 = 0.75$ (Figura 5.4) os resultados das buscas começam a sofrer variações; algumas consultas apresentam bom desempenho (ex.: cobertura $= 95$ e precisão $= 100$), ao passo que outras recuperam poucos itens relevantes (ex.: cobertura $= 36$ e precisão $= 45$). Isso justifica a redução do valor da precisão média em alguns pontos das curvas.

Nesse caso, os valores da medida de precisão média do método 2 (Figura 5.4(a)) começam

a diminuir quando a cobertura atinge o valor 60. A partir daí, as duas curvas da Figura 5.4(a) passam a ter comportamentos similares. Em comparação às abordagens de Ogawa e Horng, o método 1 apresentou melhor desempenho (Figura 5.4(b)). Já em comparação com o método 2 (Figura 5.4(c)) o algoritmo Horng exibiu valores mais elevados da medida de precisão média nos pontos de cobertura $\in [0, 25]$ e cobertura > 92.

Com o aumento do valor de z_2 para 0.75 algumas consultas passaram a retornar zero documentos como resultado. O comportamento de cada método, quando $z_2 = 0.75$, é mostrado nas Tabelas 5.5, 5.6, 5.7 e 5.8.

Tabela 5.5: Pontos de desempenho - Método 1, $z_2 = 0.75$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	100	2	2
Desempenho médio	70	71.6	0-41	0-30
Alto valor de cobertura	95	17.6	0-34	0-30

Tabela 5.6: Pontos de desempenho - Método 2, $z_2 = 0.75$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	10	93.4	3-7	3-5
Desempenho médio	70	78.2	0-33	0-29
Alto valor de cobertura	90	40	0-29	0-29

Tabela 5.7: Pontos de desempenho - Ogawa, $z_2 = 0.75$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	90	2-4	2
Desempenho médio	65	52.6	0-33	0-28
Alto valor de cobertura	90	17.6	0-32	0-28

Tabela 5.8: Pontos de desempenho - Horng, $z_2 = 0.75$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	100	2	2
Desempenho médio	55	56.4	8-28	8-24
Alto valor de cobertura	95	20	0-41	0-41

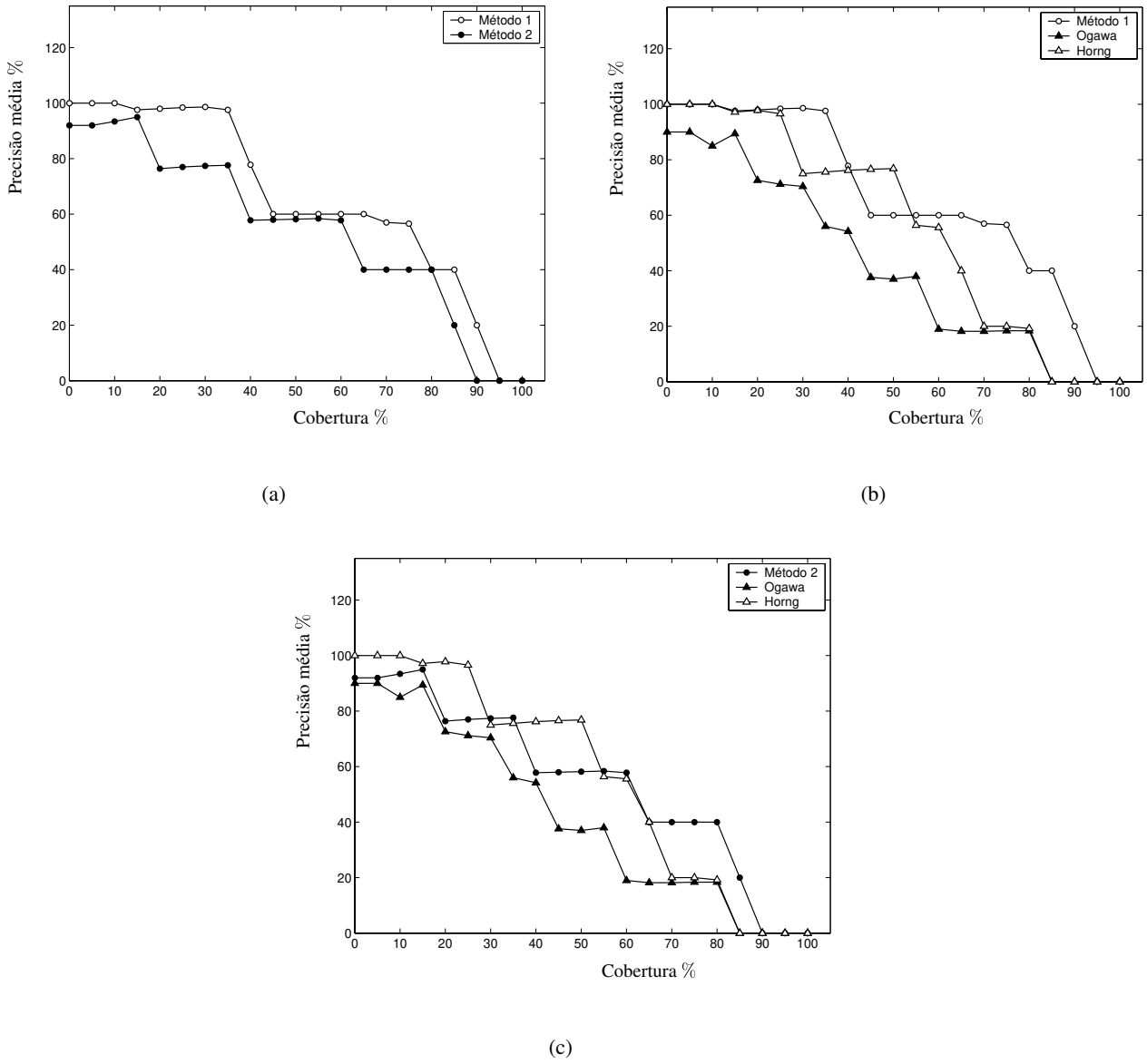


Figura 5.5: Cobertura e precisão média para consultas formadas por nomes de categoria, $z_2 = 0.95$.

Com o valor de $z_2 = 0.95$ percebe-se uma queda nos valores da precisão média em todos os algoritmos (Figura 5.5). Isso ocorre por causa da redução de documentos relevantes retornados em algumas consultas e a posição que esses documentos assumem na listagem do resultado.

Analisando os gráficos da Figura 5.5 tem-se que o método 1 apresentou melhor desempenho que os demais (Figuras 5.5(a) e 5.5(b)), embora nos pontos de cobertura $\in [40, 50]$ a aborda-

gem Ogawa tenha tido um comportamento mais favorável (Figura 5.5(b)). Já os resultados obtidos com o método 2 não foram tão bons como nos casos anteriores ($z_2 = 0.2$ e $z_2 = 0.75$), principalmente em relação ao método 1 (Figura 5.5(a)) e Horng (Figura 5.5(c)).

O comportamento de cada método, quando $z_2 = 0.95$, é exibido nas Tabelas 5.9, 5.10, 5.11 e 5.12.

Tabela 5.9: Pontos de desempenho - Método 1, $z_2 = 0.95$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	100	2	2
Desempenho médio	60	60	0-30	0-26
Alto valor de cobertura	90	20	0-28	0-28

Tabela 5.10: Pontos de desempenho - Método 2, $z_2 = 0.95$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	15	95	4-9	4-7
Desempenho médio	60	57.8	0-28	0-26
Alto valor de cobertura	85	20	0-37	0-37

Tabela 5.11: Pontos de desempenho - Ogawa, $z_2 = 0.95$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	15	89.4	5-8	4-72
Desempenho médio	45	37.6	0-22	0-20
Alto valor de cobertura	80	18.4	0-27	0-25

Tabela 5.12: Pontos de desempenho - Horng, $z_2 = 0.95$ e consultas com nomes de categoria

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	100	2	2
Desempenho médio	55	56.4	8-28	8-24
Alto valor de cobertura	80	19.2	0-26	0-25

Resultados experimentais 2

Aqui são apresentados os experimentos computacionais (Figuras 5.6, 5.7 e 5.8) para as consultas formadas por palavras relacionadas às categorias. São elas: "*fuzzy set*", "*web agent*", "*search engine*" e "*hybrid system*".

Através dos gráficos da Figura 5.6 pode-se perceber valores baixos da precisão média principalmente no caso de Horng (Figuras 5.6(b) e 5.6(c)).

O comportamento do algoritmo Ogawa mostrou-se favorável em dois momentos particulares: cobertura $\in [0, 15]$ e cobertura $\in [90, 100]$. O primeiro caso indica a presença de documentos relevantes no início da resposta ao usuário. O segundo caso, quando a cobertura $= 100$, ilustra que para uma das consultas todos os documentos relevantes foram retornados (Figuras 5.6(b) e 5.6(c)).

O método 2 apresentou melhor desempenho que as abordagens de Ogawa e Horng (Figuras 5.6(c)). Já o método 1 exibiu valores maiores da medida de precisão média (Figura 5.6(b)) quando a cobertura $= 20$ e nos intervalos $[35, 75]$ e $[80, 95]$.

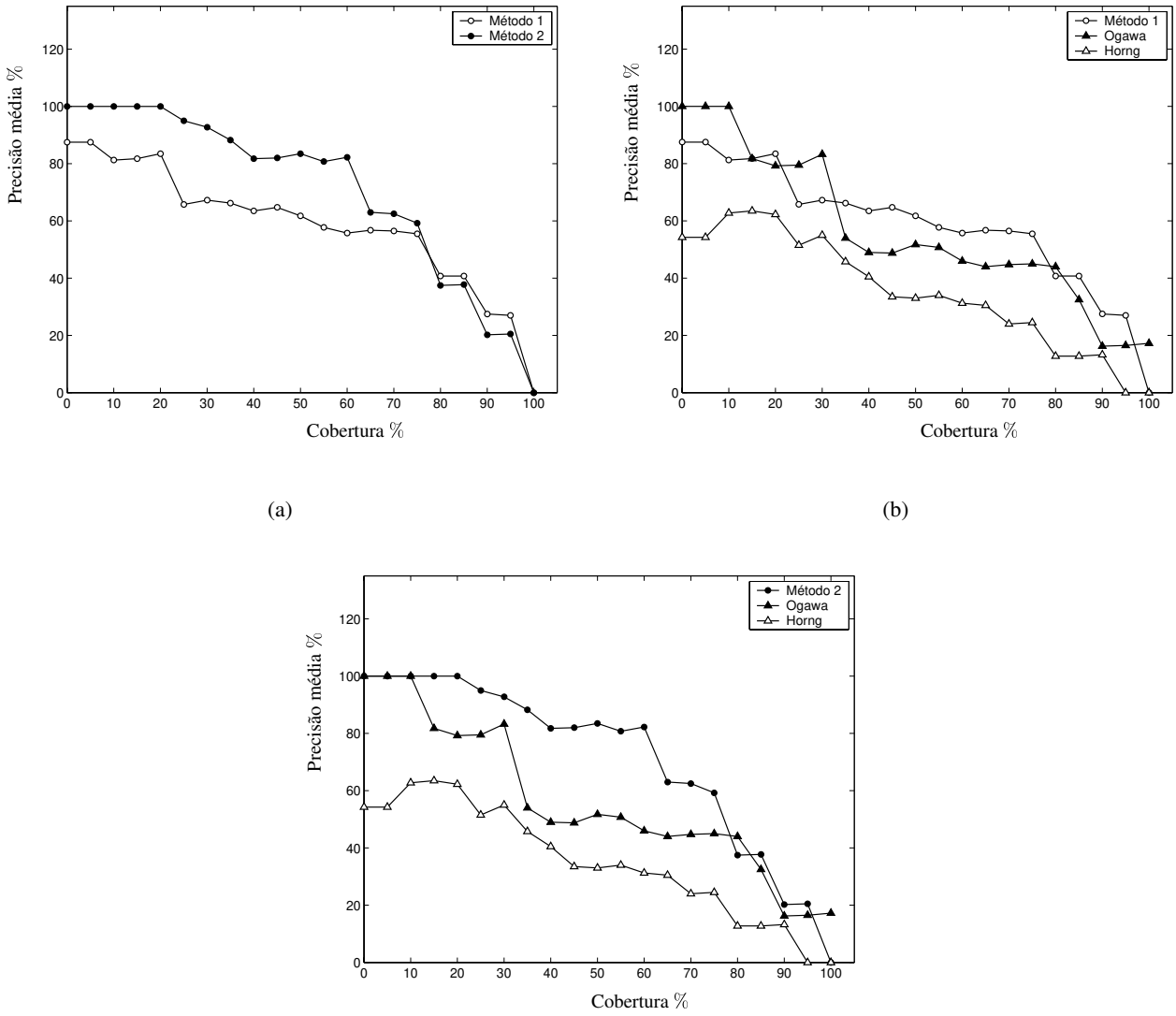


Figura 5.6: Cobertura e precisão média para consultas formadas por palavras, $z_2 = 0.2$.

As Tabelas 5.13, 5.14, 5.15 e 5.16 exibem o comportamento de cada método em três pontos específicos quando $z_2 = 0.2$.

Tabela 5.13: Pontos de desempenho - Método 1, $z_2 = 0.2$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	87.5	1-4	1-2
Desempenho médio	55	57.8	10-45	9-12
Alto valor de cobertura	95	27	0-70	0-18

Tabela 5.14: Pontos de desempenho - Método 2, $z_2 = 0.2$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	20	100	3-5	3-5
Desempenho médio	65	63	0-16	0-15
Alto valor de cobertura	95	20.5	0-20	0-20

Tabela 5.15: Pontos de desempenho - Ogawa, $z_2 = 0.2$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	10	100	2-3	2-3
Desempenho médio	50	51.8	0-17	0-12
Alto valor de cobertura	100	17.3	0-35	0-24

Tabela 5.16: Pontos de desempenho - Horng, $z_2 = 0.2$ e consultas formadas por palavras

Pontos de desempenho	***Recall***	***Precision***	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	15	63.5	3-7	2-4
Desempenho médio	40	40.5	0-18	0-10
Alto valor de cobertura	90	13.3	0-41	0-21

Para $z_2 = 0.75$ o comportamento do método 1 (Figuras 5.7(a) e 5.7(b)) e de Horng (Figuras 5.7(b) e 5.7(c)) são idênticos ao anterior (Figura 5.6).

Nesse caso, os métodos 1 e 2 (Figura 5.7(a)) apresentaram comportamentos similares nos pontos de cobertura $\in [40, 60]$ e cobertura $\in [80, 100]$, sendo o desempenho do método 2 melhor nos pontos iniciais da curva (cobertura $\in [0, 40]$).

Um aspecto interessante a ser observado na abordagem de Ogawa (Figuras 5.7(b) e 5.7(c)) é a redução do valor da precisão média nos pontos inicias da curva em comparação com a abor-

dagem anterior (Figura 5.6). Isso ocorreu porque uma das consultas apresentou cobertura $= 0$ e precisão $= 0$, ou seja, nenhum item relevante foi retornado.

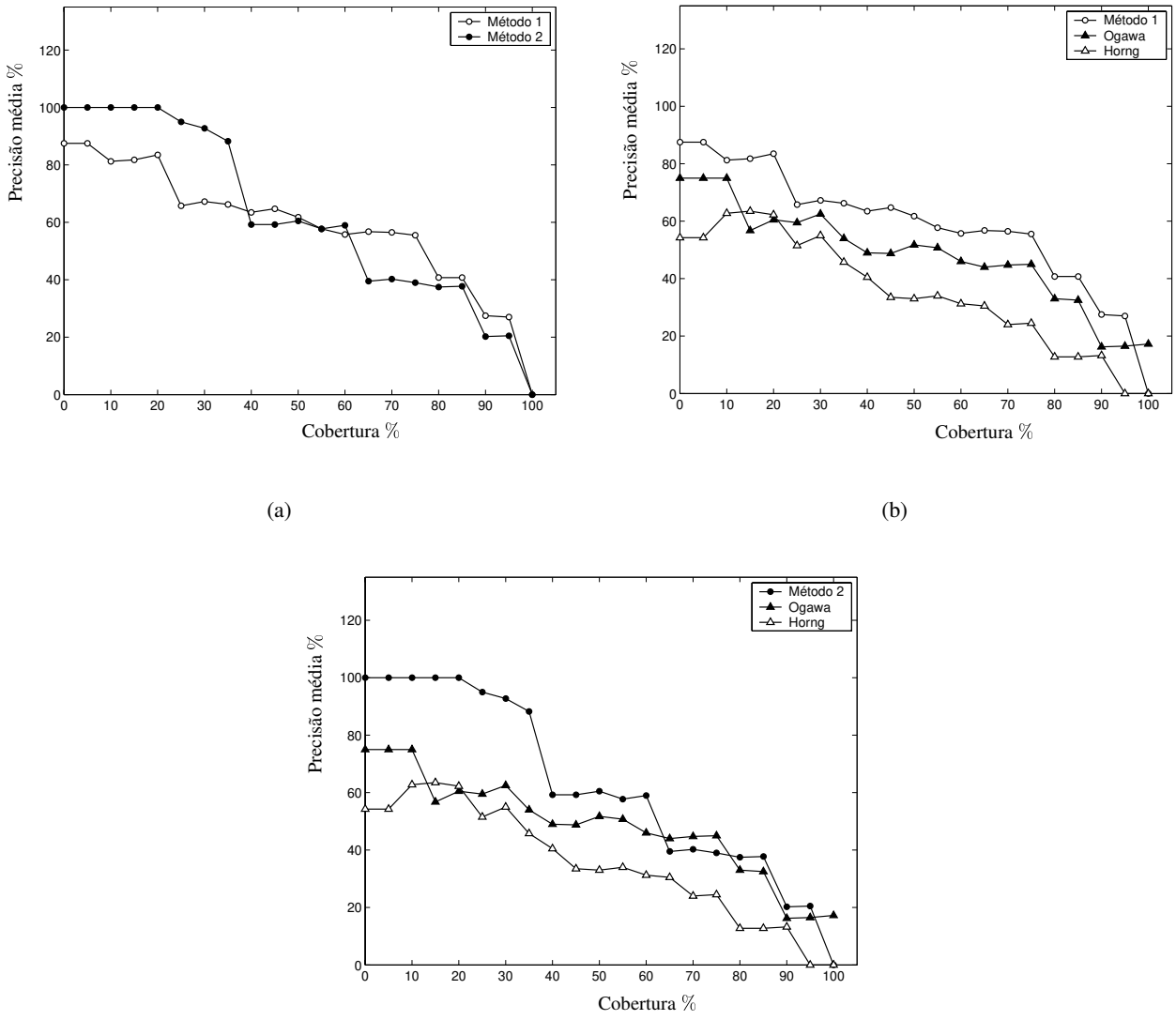


Figura 5.7: Cobertura e precisão média para consultas formadas por palavras, $z_2 = 0.75$.

O comportamento do método 2 e de Ogawa, quando $z_2 = 0.75$, são mostrados nas Tabelas 5.17 e 5.18.

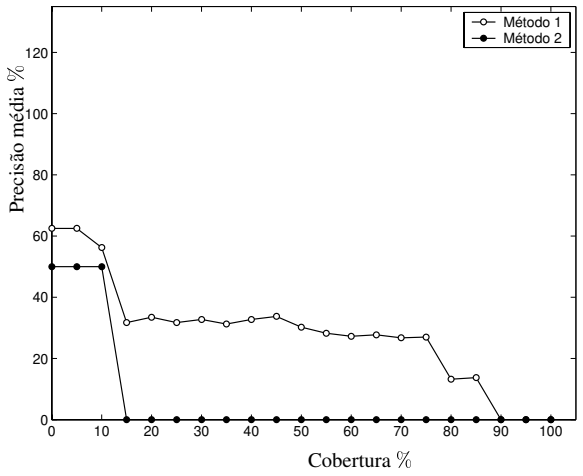
Tabela 5.17: Pontos de desempenho - Método 2, $z_2 = 0.75$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	20	100	3-5	3-5
Desempenho médio	60	59	0-13	0-13
Alto valor de cobertura	95	20.5	0-20	0-20

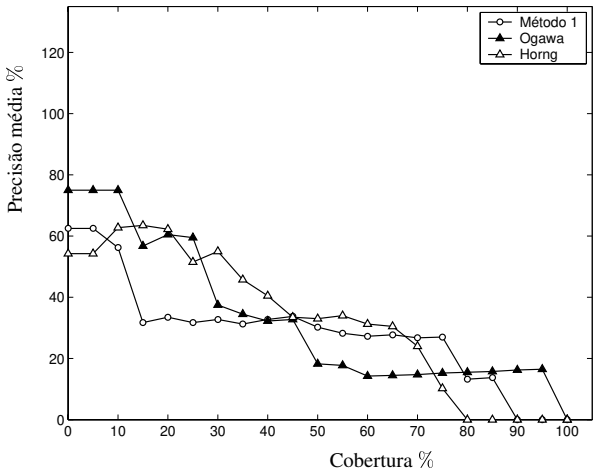
Tabela 5.18: Pontos de desempenho - Ogawa, $z_2 = 0.75$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	10	75	0-3	0-3
Desempenho médio	50	51.8	0-17	0-12
Alto valor de cobertura	100	17.3	0-35	0-24

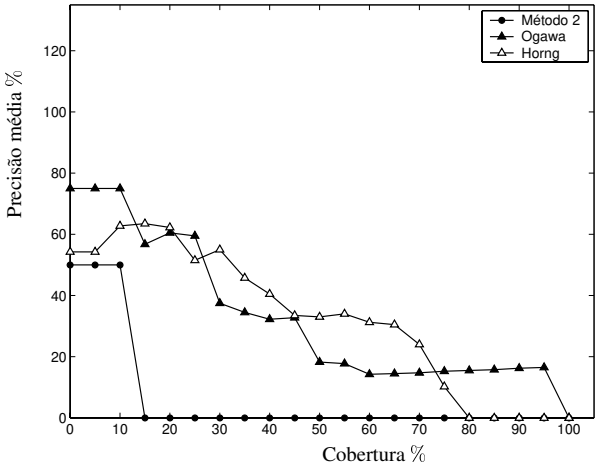
Com $z_2 = 0.95$ o método 2 (Figura 5.8(a)) sofre uma queda acentuada nos valores das medidas de cobertura e precisão média. Essa diminuição ocorreu devido ao fato de duas consultas apresentarem cobertura $= 0$ e precisão $= 0$. O método 1 também apresentou baixo desempenho em relação as abordagens de Ogawa e Horng. Apesar disso, pode-se verificar alguns pontos no gráfico que exibem comportamento positivo para esse algoritmo (Figura 5.8(b)).



(a)



(b)



(c)

Figura 5.8: Cobertura e precisão média para consultas formadas por palavras, $z_2 = 0.95$.

O comportamento dos algoritmos para $z_2 = 0.95$ é apresentado nas Tabelas 5.19, 5.20, 5.21 e 5.22.

Tabela 5.19: Pontos de desempenho - Método 1, $z_2 = 0.95$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	5	62.5	1-4	1-2
Desempenho médio	30	32.8	0-15	0-7
Alto valor de cobertura	85	13.75	0-58	0-17

Tabela 5.20: Pontos de desempenho - Método 2, $z_2 = 0.95$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	10	50	0-3	0-3
Desempenho médio	0	0	0	0
Alto valor de cobertura	0	0	0	0

Tabela 5.21: Pontos de desempenho - Ogawa, $z_2 = 0.95$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	10	100	0-10	0-6
Desempenho médio	35	34.5	0-15	0-9
Alto valor de cobertura	95	16.5	0-33	0-22

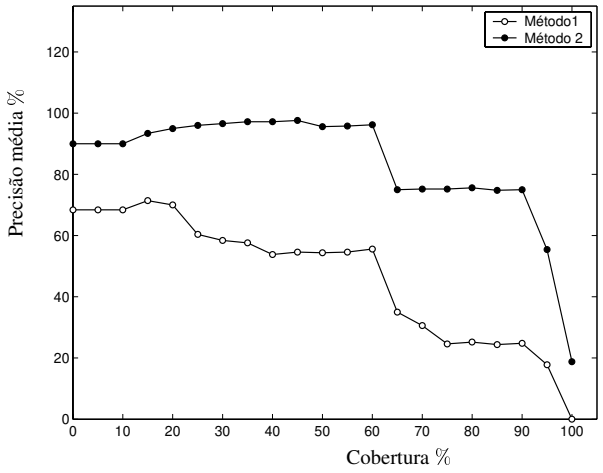
Tabela 5.22: Pontos de desempenho - Horng, $z_2 = 0.95$ e consultas formadas por palavras

Pontos de desempenho	**Cobertura**	**Precisão**	**Itens recuperados**	**Itens relevantes**
Alto valor de precisão	15	63.5	3-7	2-4
Desempenho médio	40	40.5	0-18	0-10
Alto valor de cobertura	75	10.3	0-35	0-18

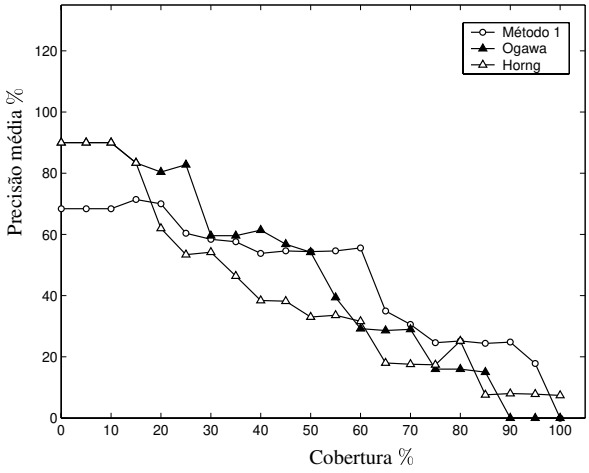
Resultados experimentais 3

Os resultados computacionais para consultas compostas conectadas pelo operador lógico AND são apresentados nas Figuras 5.9, 5.10 e 5.11. As consultas foram as seguintes: "*Agent* AND *Information Retrieval*", "*Fuzzy Logic* AND *Information Retrieval*", "*Information Retrieval* AND *Search Engine*", "*Agent* AND *Genetic Algorithm*" e "*Genetic Algorithm* AND *Hybrid System*".

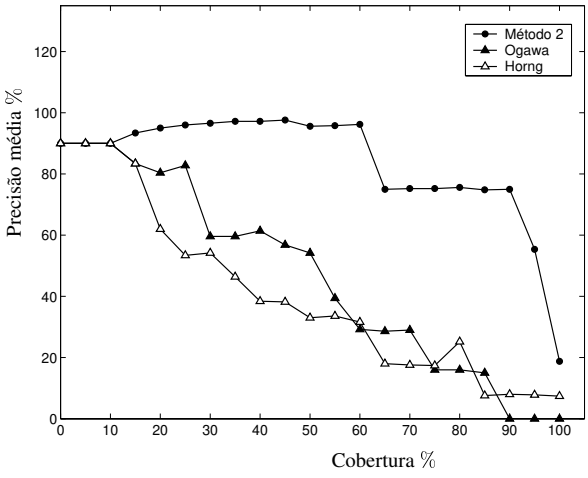
Para $z_2 = 0.2$ (Figura 5.9) o método 2 apresentou melhor desempenho que as outras abordagens (Figuras 5.9(a) e 5.9(c)). Já o método 1 exibiu valores maiores da medida de precisão média que as soluções de Ogawa e Horng nos pontos de cobertura $\in [50, 97]$ (Figura 5.9(b)).



(a)



(b)



(c)

Figura 5.9: Cobertura e precisão média para consultas compostas, $z_2 = 0.2$.

O comportamento particular de cada algoritmo, em três pontos específicos, pode ser visualizado nas Tabelas 5.23, 5.24, 5.25 e 5.26.

Tabela 5.23: Pontos de desempenho - Método 1, $z_2 = 0.2$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	15	71.4	3-32	3-5
Desempenho médio	55	54.6	9-40	6-10
Alto valor de cobertura	95	17.8	0-18	0-16

Tabela 5.24: Pontos de desempenho - Método 2, $z_2 = 0.2$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	45	97.6	3-8	3-8
Desempenho médio	75	75.2	0-14	0-13
Alto valor de cobertura	100	18.8	0-17	0-16

Tabela 5.25: Pontos de desempenho - Ogawa, $z_2 = 0.2$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	20	90	3-6	2-4
Desempenho médio	55	54.2	0-14	0-10
Alto valor de cobertura	85	15	0-12	0-9

Tabela 5.26: Pontos de desempenho - Horng, $z_2 = 0.2$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	10	90	2-4	2
Desempenho médio	35	46.4	0-19	0-6
Alto valor de cobertura	100	7.4	0-43	0-16

Quando z_2 assume o valor 0.75 o método 2 é ainda considerado o melhor em comparação com os métodos 1 (Figura 5.10(a)), Ogawa e Horng (Figura 5.10(c)). O método 1 apresenta valores mais elevados de precisão média nos pontos de cobertura $= 20$ e intervalos $[55, 75]$ e $[85, 95]$.

Ao analisar o ponto de cobertura $= 80$ do algoritmo Horng, é observado um aumento acentuado do valor da precisão média. Isso caracteriza a presença de vários documentos relevantes em seqüência.

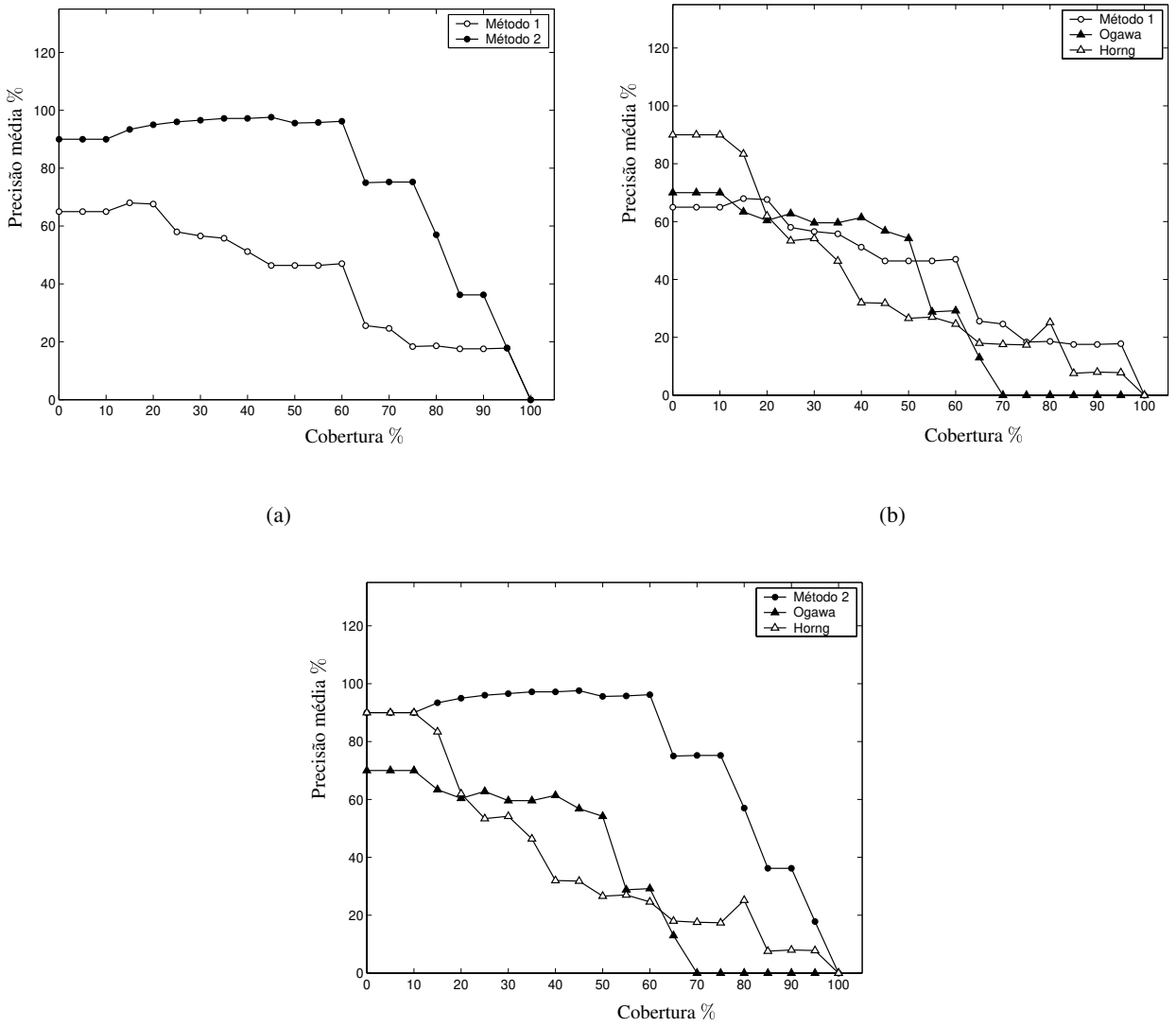


Figura 5.10: Cobertura e precisão média para consultas compostas, $z_2 = 0.75$.

O comportamento dos algoritmos, quando $z_2 = 0.75$ é apresentado nas Tabelas 5.27, 5.28, 5.29 e 5.30.

Tabela 5.27: Pontos de desempenho - Método 1, $z_2 = 0.75$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor precisão	15	68	0-8	0-3
Desempenho médio	45	46.4	0-25	0-8
Alto valor de cobertura	95	17.8	0-18	0-16

Tabela 5.28: Pontos de desempenho - Método 2, $z_2 = 0.75$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	40	97.6	4-7	3-7
Desempenho médio	75	75.2	0-14	0-13
Alto valor de cobertura	95	17.8	0-18	0-16

Tabela 5.29: Pontos de desempenho - Ogawa, $z_2 = 0.75$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	10	70	0-5	0-2
Desempenho médio	50	54.2	0-35	0-9
Alto valor de cobertura	65	13	0-17	0-11

Tabela 5.30: Pontos de desempenho - Horng, $z_2 = 0.75$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	10	90	2-4	2
Desempenho médio	40	32	0-16	0-7
Alto valor de cobertura	95	7.8	0-38	0-15

Com $z_2 = 0.95$ os métodos 1 e 2 (Figura 5.11) sofrem uma queda acentuada nos valores da precisão média. O desempenho do método 1 é inferior ao das abordagens Ogawa, nos pontos de cobertura $\in [0, 35]$, e Horng, nos ponto de cobertura $\in [0, 60]$ (Figura 5.11(b)). Já o comportamento do método 2 é melhor que o método 1 no intervalo de cobertura $\in [0, 40]$. Depois disso os dois algoritmos exibem o mesmo comportamento (Figura 5.11(a)).

As abordagens Ogawa e Horng apresentaram, no geral, melhores valores de precisão média que a solução 2 (Figura 5.11(c)).

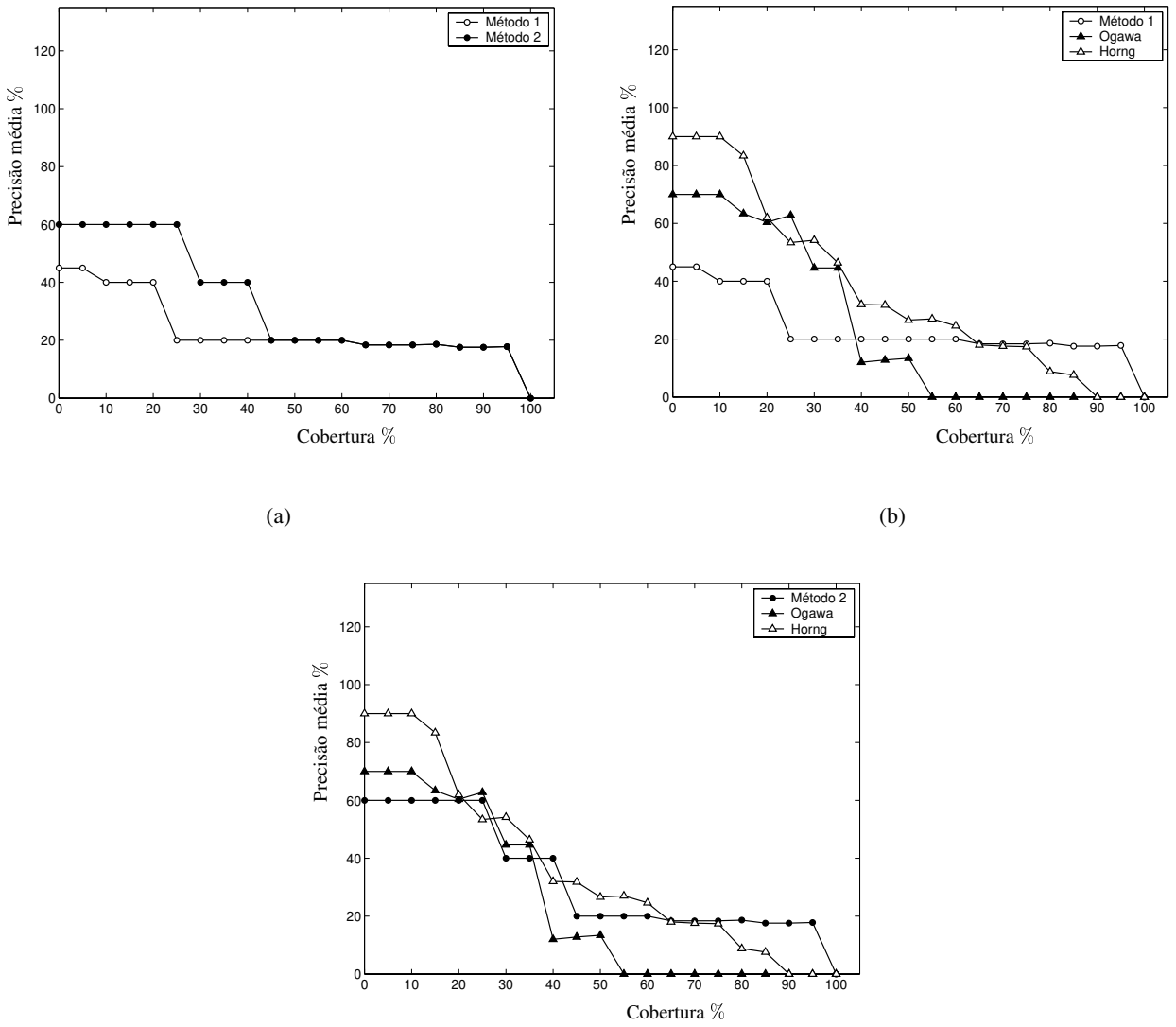


Figura 5.11: Cobertura e precisão média para consultas compostas, $z_2 = 0.95$.

O comportamento de cada algoritmo, quando $z_2 = 0.95$, é resumido nas Tabelas 5.31, 5.32, 5.33 e 5.34.

Tabela 5.31: Pontos de desempenho - Método 1, $z_2 = 0.95$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	5	44	0-4	0-3
Desempenho médio	25	20	0-5	0-5
Alto valor de cobertura	95	17.8	0-18	0-16

Tabela 5.32: Pontos de desempenho - Método 2, $z_2 = 0.95$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	10	60	0-17	0-2
Desempenho médio	40	40	0-6	0-6
Alto valor de cobertura	95	17.8	0-9	0-9

Tabela 5.33: Pontos de desempenho - Ogawa, $z_2 = 0.95$ e consultas compostas

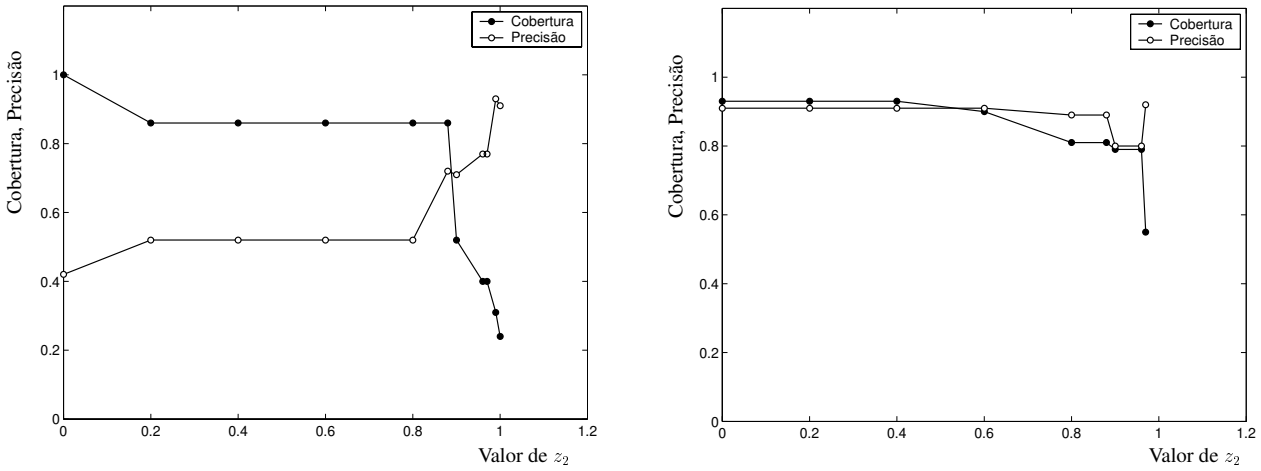
Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	10	70	0-5	0-2
Desempenho médio	35	44.6	0-26	0-6
Alto valor de cobertura	50	13.4	0-18	0-12

Tabela 5.34: Pontos de desempenho - Horng, $z_2 = 0.95$ e consultas compostas

Pontos de desempenho	Cobertura	Precisão	Itens recuperados	Itens relevantes
Alto valor de precisão	10	90	2-4	2
Desempenho médio	40	32	0-16	0-7
Alto valor de cobertura	85	7.6	0-34	0-13

5.3 Desempenho e análise

As Figuras 5.12, 5.13 e 5.14 mostram o relacionamento entre o valor de z_2 e as medidas de cobertura e precisão para o método 1 (Figuras (a)) e o método 2 (Figuras (b)). Cada figura representa esse relacionamento para um tipo de consulta. Na Figura 5.12 a consulta foi realizada com um nome de categoria ("*Information Retrieval*"); na Figura 5.13 com uma palavra ("*Fuzzy Set*") e na Figura 5.14 com dois nomes de categorias conectados com o operador AND ("*Fuzzy Logic* AND *Information Retrieval*").



(a) (b)

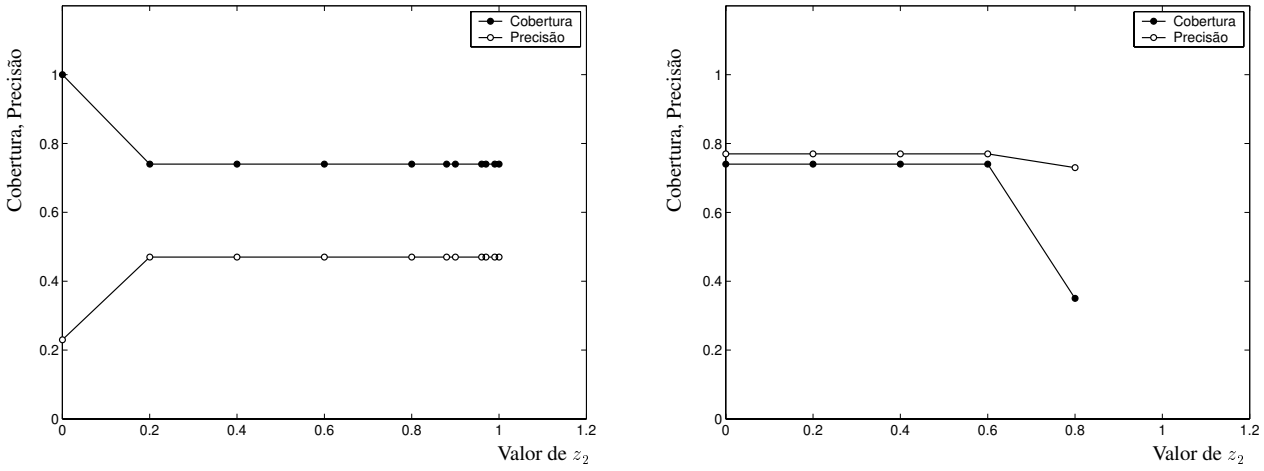
Figura 5.12: Resultado da consulta "*Information Retrieval*".

No método 1 (Figura 5.12(a)) as medidas de cobertura e precisão estão inversamente relacionadas, ou seja, à medida que o valor de z_2 aumenta, o valor de cobertura tende a diminuir e a precisão a aumentar. Já no método 2 (Figura 5.12(b)) as duas medidas apresentaram valores inicias altos com variações na medida de cobertura quando $z_2 > 0.4$ e medida de precisão quando $z_2 > 0.9$.

Para a consulta "*Information Retrieval*" o desempenho do método 2 foi melhor que o do método 1.

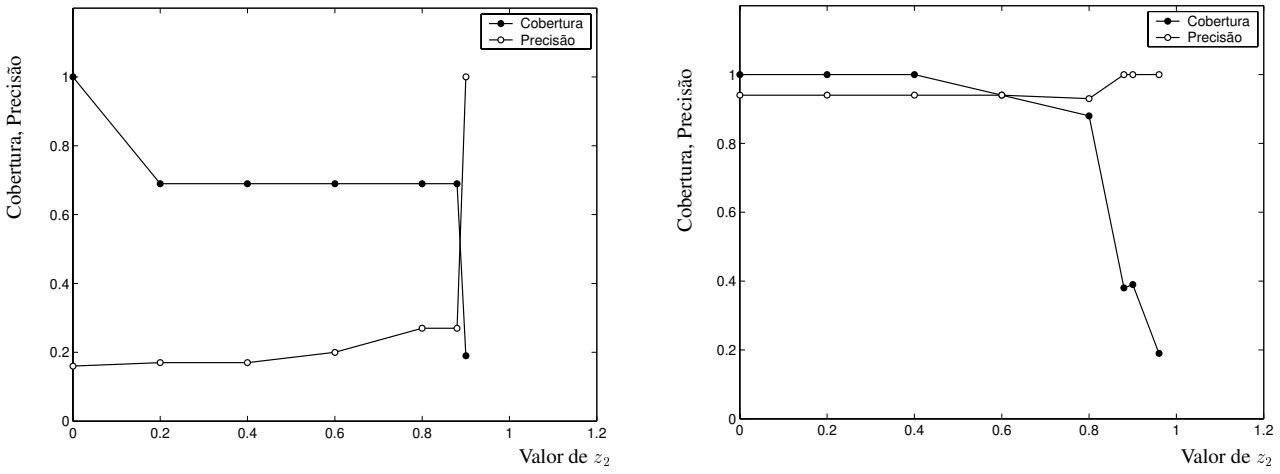
No gráfico 5.13(a) as medidas de cobertura e precisão variam até o valor de z_2 atingir 0.2. Depois disso, esses valores tornam-se estáveis até $z_2 = 1$.

Por outro lado, na Figura 5.13(b) a situação se inverte, as medidas de cobertura e precisão possuem valores iniciais constantes, sofrendo variações para valores de z_2 maiores que 0.6. Quando z_2 atinge o valor 0.88 não são encontrados itens relevantes no resultado da consulta.



(a) (b)

Figura 5.13: Resultado da consulta "*Fuzzy Set*".



(a) (b)

Figura 5.14: Resultado da consulta "*Fuzzy Logic* AND *Information Retrieval*".

Para a consulta "*Fuzzy Logic* AND *Information Retrieval*" (Figura 5.14) o método 2 (Figura 5.14(b)) exibe um comportamento melhor que o método 1 até o ponto em que $z_2 = 0.8$. A partir daí, o valor da medida de cobertura sofre uma redução acentuada enquanto que a medida

de precisão atinge seu valor máximo, 1. Isso caracteriza a presença de poucos itens relevantes no resultado, mas todos eles localizados no ínicio da resposta ao usuário.

O método 1, por outro lado, mantém o valor da medida de cobertura estável nos pontos em que $z_2 \in [0.2, 0.88]$. Em seguida, a precisão aumenta para 1 e a cobertura reduz para 0.19.

Quando $z_2 = 0.96$ nenhum item relevante é retornado pelos dois algoritmos.

Tempo de execução dos algoritmos

A análise do tempo de execução de um algoritmo é baseado no custo de cada operação realizada, em função do tamanho da entrada. Em geral, são consideradas apenas as operações de maior custo, também denominadas de operações fundamentais.

A Tabela 5.35 apresenta a medidas de complexidade temporal, para o pior caso, do método 1, método 2, Ogawa e Horng.

Tabela 5.35: Análise da complexidade temporal

Algoritmos	**Consultas simples**	**Consultas compostas**
Método 1	$O(m.n)$	$O(m.n)$
Método 2	$O(m^2.n)$ ou $O(m.n^2)$	$O(m.n)$
Ogawa	$O(t)$, $t \rightarrow m + n$	$O(t)$, $t \rightarrow m + n$
Horng	$O(y^2)$, $y \rightarrow m$	$O(y^2)$, $y \rightarrow m$

Esses valores foram determinados para consultas simples (um termo) e consultas compostas (dois ou mais termos conectados pelos operadores AND ou OR).

Nos métodos 1 e 2 o tamanho da entrada é definido por um total de m nomes de categorias e n palavras relacionadas às categorias. O procedimento que calcula a composição *fuzzy* max-min (G_c (4.5) e G_p (4.6)) foi considerada a operação de maior custo na execução desses métodos.

No modelo Ogawa, o tamanho da entrada é t, sendo $t = m+n$. As operações fundamentais para esse algoritmo foram as operações responsáveis pela execução dos passos 2 (Eq. 3.4) e 3 (Eq. 3.5) do processo de recuperação.

No modelo Horng, o tamanho da entrada é y, sendo $y = m$ o número de conceitos de rede *fuzzy*. O algoritmo de expansão do vetor consulta foi a operação de maior custo na abordagem Horng (Algoritmo 3.1).

Discussão sobre os modelos *fuzzy*

A partir dos experimentos realizados, algumas questões relacionadas aos modelos *fuzzy* abordados neste trabalho podem ser observadas:

- O modelo Horng mostrou ser adequado para pesquisas referentes a área do assunto de interesse. Por exemplo, para uma consulta sobre "*Fuzzy Logic*" além dos documentos sobre esse assunto foram encontrados documentos sobre "*Genetic Algorithm*" e "*Neural Network*", que são conteúdos relacionados a área de "Inteligência Computacional" e não especificamente ao assunto "*Fuzzy Logic*". Isso explica os baixos valores da precisão média, para esse modelo, nos experimentos realizados.

  Outra característica importante do modelo Horng diz respeito ao processo de criação da rede conceitual *fuzzy* que é baseada na freqüência dos termos indexados nos documentos. A partir da utilização desse procedimento alguns inconvenientes foram identificados:

  1. O custo computacional para a geração da rede conceitual *fuzzy* é muito alto quando há constantes atualizações (inclusão ou remoção) de documentos na base de dados: cada atualização demanda a construção de uma nova rede conceitual;
  2. O fato de um documento conter a palavra em seu conteúdo mostrou não ser uma boa forma de indexação, pois o mesmo pode citar o termo, mas não tratar deste assunto especificamente.

- No modelo Ogawa é necessário que a base de dados esteja consistente, ou seja, bem estruturada, homogênea/uniforme para que o processo de indexação seja válido. Uma base de dados homogênea/uniforme é aquela em que há equilíbrio entre o número de documentos

de uma determinada área em relação a outra. A falta de uniformidade causaria uma baixa relação entre os termos da área que tem menor quantidade de documentos ocasionando baixo índices *fuzzy* para estes termos. Uma resposta a consultas com estes termos poderia resultar em documentos pouco relevantes ao assunto de interesse, e conseqüentemente, baixos valores das medidas de cobertura e precisão.

Como no modelo Horng, o modelo Ogawa também apresenta desvantagens na geração da base de conhecimento (matriz de correlação de palavras) quando a base de dados é dinâmica, já que seu processo de indexação é baseado na presença ou ausência do termo no documento. Se a base de dados for estática ou sofrer poucas alterações ao longo do tempo a matriz de correlação de palavras é vantajosa, pois ela é carregada em memória apenas uma vez e é utilizada em todas as consultas sem a necessidade de recriá-la. Essa característica inviabiliza a utilização desse modelo em domínios como o da *web*.

O desempenho desse algoritmo sob o ponto de vista do tempo de execução ($O(n)$) foi considerado o melhor em comparação com as outras abordagens.

- O mecanismo de indexação adotado para o modelo ontológico relacional *fuzzy*, diferente das abordagens Ogawa e Horng, é baseado na seleção de termos que especificam o conteúdo do documento. A definição desses termos ocorre no momento em que o artigo é cadastrado no sistema, sendo de responsabilidade do usuário que está efetuando o cadastro fornecer as informações corretas sobre o assunto tratado pelo artigo.

  Outra vantagem em relação aos modelos Ogawa e Horng, diz respeito ao custo computacional para atualização (inclusão ou remoção) da base de dados. Nesse caso, o custo é bem inferior, em relação às outras abordagens, pois não há a necessidade de recriar a base de conhecimento (ontologia relacional *fuzzy*) cada vez que um documento for incluído ou removido.

  Por outro lado, o custo computacional com base no tempo de execução do método 2, proposto para este modelo, foi o maior em comparação com os outros algoritmos. Apesar disso, nos resultados retornados pelo modelo ontológico, pôde-se observar a freqüência de

documentos relevantes no início da resposta ao usuário, caracterizando os valores altos da medida de precisão nos experimentos computacionais.

Dentre as soluções propostas para o modelo ontológico, o método 2 mostrou ser adequado para consultas com valores baixos e médios de z_2. Nos casos em que z_2 apresentou valores altos o desempenho desse método foi inferior ao das outras abordagens em todas as situações. Já o método 1, apesar de nem sempre exibir valores elevados para a medida de precisão, apresentou altos valores de cobertura em todas as situações propostas, independente do valor de z_2. Esse método obteve seu melhor desempenho nas consultas formadas por nomes de categoria.

5.4 Resumo

Neste capítulo, foram apresentados os resultados de desempenho fornecidos pelos métodos considerados neste trabalho. Estes resultados foram analisados e discutidos segundo os critérios de desempenho das medidas de cobertura e precisão e medida de complexidade temporal.

Apesar do alto custo computacional apresentado pelo modelo ontológico (em particular, o método 2), seus resultados com base nas medidas de cobertura e precisão se mostraram favoráveis quando comparados ao das abordagens Ogawa e Horng.

Capítulo 6

Conclusões

6.1 Considerações Gerais

A busca por documentos em sistemas de informação hoje em dia é uma das tarefas mais comuns e representa também uma das mais frustrantes (Hu et al., 2001) para muitos usuários. O crescimento contínuo do número de documentos, tanto em quantidade como em variedade, vem fazendo com que as técnicas utilizadas pelos tradicionais sistemas de busca se tornem cada vez mais inadequadas para recuperação de informações relevantes. Nesse sentido, várias pesquisas voltadas à recuperação de dados motiva o desenvolvimento de novas técnicas relacionadas à inteligência computacional para auxiliar no processo de recuperação, filtragem e avaliação da informação desejada.

Dentre as técnicas existentes tem-se a teoria de conjuntos *fuzzy* que tem apresentado resultados satisfatórios quando aplicada em sistemas que representam e gerenciam a imprecisão (*imprecision*) e indefinição (*vagueness*) caracterizados no processo de recuperação de informação (Pasi, 2002; Herrera-Viedma e Pasi, 2003).

Com base nisso, foi desenvolvido neste trabalho um modelo de recuperação de informação que propõe a utilização de uma ontologia relacional *fuzzy*.

A partir dos resultados experimentais do Capítulo 5 pode-se concluir que o modelo on-

tológico mostrou ser uma alternativa satisfatória para recuperação de informação, comparando-se com as abordagens Ogawa e Horng, pelas seguintes razões:

- O mecanismo de indexação adotado para este modelo, diferente das abordagens Ogawa e Horng, é baseado na seleção de termos que especificam o conteúdo do documento;
- O custo computacional para atualização (inclusão ou remoção) da base de dados é bem inferior quando comparado as outras abordagens, pois no caso do modelo ontológico não existe a necessidade de recriar a base de conhecimento (ontologia relacional *fuzzy*) cada vez que um documento for incluído ou removido da base de dados;
- O modelo ontológico obteve um desempenho promissor, determinando a presença predominante de itens relevantes nos resultados das consultas realizadas. Pode-se perceber também a redução de itens irrelevantes nos resultados das consultas quando comparado as abordagens Ogawa e Horng.

6.2 Contribuições

A principal contribuição deste trabalho é a proposta, desenvolvimento e avaliação de um modelo ontológico relacional *fuzzy* para sistemas de recuperação de informação textual. Esse modelo demonstrou solucionar algumas limitações apresentadas pelas abordagens Ogawa e Horng no que diz respeito ao mecanismo de indexação de documentos e ao custo computacional para geração da base de conhecimento.

Uma outra contribuição desse projeto é a arquitetura definida para esse sistema (Figura 4.2), que pode ter sua ontologia desenvolvida para outros domínios de busca. A partir do sistema de busca foi possível testar e validar o modelo ontológico relacional *fuzzy*.

6.3 Trabalhos Futuros

Com relação à continuidade deste trabalho ficam os seguintes casos a serem testados ou aplicados:

- Desenvolver um algoritmo para construção automática da ontologia relacional *fuzzy*, baseado em técnicas de Inteligência Artificial, capaz de definir o grau de relacionamento *fuzzy* entre categorias e palavras;
- Verificar o desempenho do modelo ontológico quando aplicado em bases com grande volumes de dados (ex.: valores entre 5.000 e 10.000);
- Estender a ontologia relacional *fuzzy* para outros domínios de aplicação, como por exemplo, agropecuária, medicina, etc;
- Testar o modelo proposto em sistemas de recuperação de informação complexos (presença de arquivos de áudio e imagens na base de dados);
- Desenvolver métodos de adaptação e aprendizagem que sejam capazes de compreender o perfil do usuário para melhor atender às suas necessidades.

Referências Bibliográficas

Baeza-Yates, R. e Ribeiro-Neto (1999). *Modern Information Retrieval*, ACM Press / Addison Wesley, EUA.

Bordogna, G., Carrara, P. e Pasi, G. (1993). A fuzzy document representation supporting user adaptation in information retrieval, *Proceeding of the 2nd IEEE International Conference on Fuzzy Systems,* pp. 974–979.

Bordogna, G. e Pasi, G. (2002). Flexible querying of web documents, *Proceedings of the 2002 ACM Symposium on Applied Computing,* pp. 675–680.

Brin, S. e Page, L. (2002). The anatomy of a large-scale hypertextual web search engine. Disponível em: http://www-db.stanford.edu/pub/papers/google.pdf. Acesso em: 11/09/2004.

Campos, F. e Bax, M. P. (2002). Como os mecanismos de busca da web indexam páginas html. Disponível em: http://cubaeci.paradigma.com.br:8088/Bax/Publis/ComoMaquinasBuscaIndexamPaginasWeb.pdf. Acesso em: 11/09/2004.

Campos, M. L. A., Brasil, M. I., Coelho, B. A. S. e Bastos, D. R. (2002). Vocabulário sistematizado: A experiência da Fundação Casa Rui Barbosa. Disponível em: http://www.casaruibarbosa.gov.br/irene_brasil/vocabulario.pdf. Acesso em: 11/09/2004.

Chen, S.-M., Horng, Y.-J. e Lee, C.-H. (2001). Document retrieval using fuzzy-valued concept networks, *IEEE Transactions on Systems, Man, and Cybernetics - part B:Cybernetics,* **31**(1): 111–118.

Chen, S.-M. e Wang, J.-Y. (1995). Document retrieval using knowledge-based fuzzy information retrieval techniques, *IEEE Transactions on Systems, Man, and Cybernetics,* **25**(5): 1–6.

Crestani, F., Lalmas, M., Rijsbergen, C. J. V. e Campbell, I. (1998). Is this document relevant? probably: a survey of probabilistic models in information retrieval, *ACM Computing Surveys,* **30**(4): 528–552.

de Jesus, J. B. M. (2002). Tesauro: um sistema de representação do conhecimento em sistemas de recuperação de informação. Disponível em: http://www.ndc.uff.br/textos/jerocir_tesauros.pdf. Acesso em: 11/09/2004.

Garcés, P. J., Olivas, J. A. e Romero, F. P. (2003). Conceptual matching in web search using fis-crm for representing documents, *Third Conference of the European Society for Fuzzy Logic and Technology,* pp. 63–66.

Gruber, T. R. (1992). Toward principles for the design of ontologies used for knowledge sharing. Disponível em: http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-04.html. Acesso em: 10/09/2004.

Gruber, T. R. (1993). A transaction approach to portable ontologies. knowledge acquisition, pp. 199–220. Disponível em: http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html. Acesso em: 10/09/2004.

Haruechaiyasak, C. e Shyu, M.-L. (2002). Identifying topics for web documents through fuzzy association learning, *International Journal of Computational Intelligence and Applications,* pp. 1–8.

Henzinger, M. (2000). Link analysis in web information retrieval, *Bulletin of the Technical Committee on Data Engineering,* **23**(3): 1–6.

Herrera-Viedma, E. e Pasi, G. (2003). Fuzzy approaches to access information on the web: recent developments and research trends, *Third Conference of the European Society for Fuzzy Logic and Technology,* pp. 25–31.

Herrera-Viedma, E., Peis, E., Herrera, J. C. e k. Anaya (2003). Evaluating the informative quality of web documents using fuzzy linguistic techniques, *Third Conference of the European Society for Fuzzy Logic and Technology,* pp. 32–37.

Horng, Y.-J., Chen, S.-M. e Lee, C.-H. (2001). Automatically constructing multi-relationship fuzzy concept networks in fuzzy information retrieval systems, *IEEE International Fuzzy Systems Conference,* pp. 606–609.

Hu, W.-C., Chen, Y., Schmalz, M. S. e Ritter, G. X. (2001). An overview of world wide web search technologies. Disponível em: http://citeseer.nj.nec.com/hu01overview.html. Acesso em: 11/09/2004.

Huberman, B. A., Pirolli, P. L. T., Pitkow, J. E. e Lukose, R. M. (1998). Strong regularties in world wide web surfing, *Science,* **280**(5360): 95–97.

Jones, K. S. (1990). Information retrieval and artificial intelligence, *Artificial Intelligence,* **114**: 257–281.

Klir, G. J. e Yuan, B. (1995). *Fuzzy Sets And Fuzzy Logic*, Prentice Hall : Upper Saddle River, EUA.

Kraft, D. H., Bordogna, G. e Pasi, G. (1998). Information retrieval systems: Where is the fuzz?, *IEEE InternationalConference on Fuzzy system* pp. 1367–1372.

Kruschwitz, U. (2001). Exploiting structure for intelligent web search, *Proceedings of the 34th Hawaii International Conference on System Sciences,* pp. 1–9.

Kurniawan, B. (2002). *Java para a web com servltes JSP e EJB*, Ciência Moderna Ltda, Rio de Janeiro, Brasil.

Larsen, H. L. e Yager, R. R. (1993). The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems, *IEEE Transactions on Systems, Man, and Cybernetics,* **23**(1): 31–41.

Lawrence, S. e Giles, C. L. (1999). Searching the web: General and scientific information access, *IEEE Communications,* **37**(1): 116–122.

Loia, V., Pedrycz, W. e Senatore, S. (2003). Proximity fuzzy clustering for web context analysis, *Third Conference of the European Society for Fuzzy Logic and Technology,* pp. 59–62.

Nomoto, K., Wakayama, S., Kirimoto, T., Ohashi, Y. e Kondo, M. (1990). A document retrieval system based on citations using fuzzy graphs, *Fuzzy Sets and Systems,* **38**: 207–222.

Ogawa, Y., Morita, T. e Kobayashi, K. (1991). A fuzzy document retrieval system using the keyword connection matrix and a learning method, *Fuzzy Sets and Systems,* **39**: 163–179.

Pasi, G. (2002). Flexible information retrieval: some research trends, *Mathware and Soft Computing,* **9**: 107–121.

Pedrycz, W. e Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, Cambridge, Massachusetts, EUA.

Ricarte, I. L. M. e Gomide, F. (2001). A reference sofware model for intelligent information search, *Proc. of the BISC Int. Workshop on Fuzzy Logic and the Internet,* pp. 80–85.

Russell, S. J. e Norvig, P. (2003). *Artificial Intelligence: A Modern Approach. Second Edition*, Prentice Hall, EUA.

Takagi, T. e Kawase, K. (2001). A trial for data retrieval using conceptual fuzzy sets, *IEEE Transactions on Fuzzy Systems,* **9**(4): 497–505.

Takagi, T. e Tajima, M. (2001). Proposal of a search engine based on conceptual matching of text notes, *International Workshop on Fuzzy Logic and the Internet,* pp. 53–58.

Tomiyama, T., Ohgaya, R., Shinmura, A., Kawabata, T. e Takagi, T. (2003). Concept-based web communities for $google^{TM}$ search engine, *Proceedings of the 12th IEEE International Conference on Fuzzy Systems,* pp. 1122–1128.

Widyantoro, D. H. e Yen, J. (2001). A fuzzy ontology-based abstract search engine and its user studies, *Proceedings of the 10th IEEE International Conference on Fuzzy Systems,* pp. 1291–1294.

Wiesman, F., Hasman, A. e van de Herik, H. J. (1997). Information retrieval: an overview of system characteristics, *International Journal of Medical Informatics,* **47**: 5–26.

Wutka, M. (2000). *Special edition using java server pages and servlets*, Que, EUA.

Zadeh, L. A. (1965). Fuzzy sets, *Information and Control,* **8**: 338–353.

Apêndice

Algoritmo de expansão do vetor consulta - Modelo Horng

Algoritmo 3.1: **Algoritmo de expansão do vetor consulta - Modelo Horng**

for $i = 1$ to y **do**

if $r_i = P$ **then**

for $j = 1$ to y **do**

Se o antecessor mais próximo do conceito c_i e conceito c_j for um sucessor do conceito c_k e $U_p\,(c_i, c_j) > 0$, então

$$\overline{q_j^\circ} = \begin{cases} \max\left(x_j, U_p\left(c_i, c_j\right)\right) & se\ x_j \neq \text{“}-\text{”} \\ U_p(c_i, c_j) & se\ x_j = \text{“}-\text{”} \end{cases}$$

end for

end if

if $r_i = N$ **then**

for $j = 1$ to y **do**

Se o antecessor mais próximo do conceito c_i e conceito c_j for o conceito c_k e $U_N\,(c_i, c_j) > 0$, então

$$\overline{q_j^\circ} = \begin{cases} \max\left(x_j, U_N\left(c_i, c_j\right)\right) & se\ x_j \neq \text{“}-\text{”} \\ U_N(c_i, c_j) & se\ x_j = \text{“}-\text{”} \end{cases}$$

end for

end if

if $r_i = G$ **then**

for $j = 1$ to y **do**

Se $U_G\,(c_i, c_j) > 0$, então

$$\overline{q_j^\circ} = \begin{cases} \max\left(x_j, U_G\left(c_i, c_j\right)\right) & se\ x_j \neq \text{“}-\text{”} \\ U_G(c_i, c_j) & se\ x_j = \text{“}-\text{”} \end{cases}$$

end for

end if

if $r_i = S$ **then**

for $j = 1$ to y **do**

Se $U_S\,(c_i, c_j) > 0$, então

$$\overline{q_j^\circ} = \begin{cases} \max\left(x_j, U_S\left(c_i, c_j\right)\right) & se\ x_j \neq \text{“}-\text{”} \\ U_S(c_i, c_j) & se\ x_j = \text{“}-\text{”} \end{cases}$$

end for

end if

end for