RECUPERAÇÃO DE VÍDEOS INDEXADOS POR CONCEITOS

Christian Danniel Paz Trillo

DISSERTAÇÃO APRESENTADA AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA UNIVERSIDADE DE SÃO PAULO
PARA OBTENÇÃO DO GRAU DE
MESTRE EM CIÊNCIA DA COMPUTAÇÃO

Área de Concentração: **Inteligência Artificial**

Orientadora: **Profa. Dra. Renata Wassermann**

São Paulo, março de 2005

Durante a elaboração deste trabalho o autor recebeu apoio financeiro da CAPES.

RECUPERAÇÃO DE VÍDEOS INDEXADOS POR CONCEITOS

Este exemplar corresponde à redação
final da dissertação devidamente
corrigida e defendida por
Christian Danniel Paz Trillo
e aprovada pela comissão julgadora.

São Paulo, julho de 2005.

Banca examinadora:

- Prof. Dra. Renata Wassermann (Orientadora) - IME-USP
- Prof. Dr. Flávio Soares Correia da Silva - IME-USP
- Prof. Dr. Frederico Luiz Gonçalves de Freitas - UFAL

à minha família.

Agradecimentos

O meu Mestrado será um período da minha vida que tenho certeza nunca esquecerei, mais do que pelos conhecimentos que tive a oportunidade de adquirir pelas coisas que me aconteceram nesses dois anos.

Quero agradecer a Deus por ter me permitido estar aqui apresentando este trabalho. A Marta, por ter sido uma maravilhosa mãe, por ter sido o meu melhor exemplo de vida, e principalmente por todo o carinho e proteção que me deu e que ainda desde o céu me dá. A Daniel por ser um pai exemplar, sempre dando motivação e forças para ser melhor cada dia e lograr as coisas que me proponho. A Danny, meu irmão, porque foi a sua maturidade e a dedicação dele e do meu pai que me deram a confiança para continuar em São Paulo.

Agradeço a Renata pela sua orientação acadêmica, mas sobre todo pela sua amizade e pelo apoio que me deu nos momentos mais complicados durante o Mestrado. A Paula pela idéia que deu origem a este trabalho e pelo importantíssimo tempo dedicado ao desenvolvimento, afinamento e os testes do sistema.

Agradeço profundamente a Cristian, Jesús, Karina, Edu, Thiago, Vladi, Fábio, Claudia, David e Jorge, minha família em São Paulo, que compartilharam comigo a difícil e emocionante experiência de morar longe de casa por mais de dois anos. Tenho muito que agradecer também a Luis Alberto, Eduardo e Paola, por serem meus grandes amigos e por terem estado aí sempre, nos momentos

alegres e nos momentos tristes mesmo nas grandes distâncias que nos separam. Não posso deixar de agradecer ao pessoal, do Brasil e do mundo todo, que conheci em São Paulo: Juliana, Leandro, Betsabé, Erika, Victor Hugo, Lourdes, Gerard, Joan, Iván, Mariana, Karina Pretto, Pedro, e muitos outros que me fizeram sentir como em casa em São Paulo.

Quero também agradecer a Claudia pelas reuniões para discutir nossos trabalhos e gostaria de agradecer ao Prof. Fábio Kon pelo apoio constante, pelas revisões e sugestões e pela confiança no meu trabalho. Aos membros da banca, tanto do exame de qualificação quanto da defesa de Mestrado, pelos seus valiosos aportes e comentários.

Agradeço ao IME e à USP pela oportunidade de fazer meu Mestrado aqui e pelas ótimas condições que me brindaram e agradeço à CAPES pelo apoio financeiro.

Resumo

Neste trabalho, é apresentado um sistema para a busca em uma coleção de vídeos através de consultas escritas em linguagem natural. A coleção de vídeos utilizada foi compilada de um conjunto de entrevistas com a artista brasileira Ana Teixeira, feitas pela especialista do domínio Paula P. Braga. O domínio em que a entrevista é desenvolvida é o da arte contemporânea.

O sistema utiliza uma ontologia do domínio para expandir as consultas com termos relacionados aos originais. Foi aplicada uma bateria de testes, com a qual comprovou-se que a ontologia contribui na melhora da eficiência do sistema em termos da relevância dos documentos recuperados.

O sistema foi projetado para funcionar de forma independente do domínio, simplesmente com a mudança da ontologia e da coleção de vídeos.

Abstract

In this work, we describe a query-based video retrieval system. We use a video collection compiled from interviews with Ana Teixeira, a Brazilian artist. The interviews were made by Paula P. Braga, the domain expert. The interview is developed in the domain of contemporary art.

The system uses a domain ontology to expand the queries with related terms. We tested the system with a battery of queries, and we verified that the ontology contributes to the efficiency improvement in terms of relevance of the retrieved documents.

We designed the system to work in a domain-independent way, allowing moving to another domain by just changing the underlying ontology and the video collection.

Sumário

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	3
1.3	Contribuições	3
1.4	Trabalhos relacionados	4
1.5	Estrutura do documento	6
2	Recuperação de Informação	7
2.1	Definições	7
2.1.1	Problemas na recuperação de informação	8
2.2	Processo de Indexação	9
2.2.1	Remoção de afixos	11
2.2.2	Expansão de termos do índice	13

2.2.3	Cálculo de pesos	13
2.3	Processo de Recuperação	13
2.3.1	Busca booleana	14
2.3.2	Recuperação ordenada	15
2.3.3	Recuperação probabilística	19
2.4	Avaliação de sistemas de recuperação de informação	19
2.5	Considerações finais	21
3	Uso de Ontologias em RI	23
3.1	Definições	23
3.2	Linguagens de representação	24
3.3	Utilização em recuperação de informação	27
3.3.1	Elaboração de ontologias	28
3.3.2	Medidas de similaridade em RI	28
3.4	Ontologia do domínio	30
3.5	Ferramentas para manipulação de ontologias	32
3.5.1	Ferramentas para elaboração	32
3.5.2	Ferramentas para navegação	34
3.6	Considerações finais	34
4	Recuperação de Vídeos: A aplicação	36

4.1	A aplicação	36
4.2	Descrição do sistema	37
4.2.1	Processo de Indexação	37
4.2.2	Processo de Recuperação	41
4.3	Considerações finais	50
5	Resultados	51
5.1	Execução do sistema	51
5.2	Mudança no motor de inferência	53
5.3	Aspectos relevantes no processo de testes do sistema	58
5.4	Testes	59
5.4.1	Testes globais	59
5.4.2	Testes por consulta	62
5.5	Considerações finais	67
6	Conclusões e Trabalhos Futuros	69
6.1	Conclusões	69
6.2	Contribuições do trabalho	70
6.3	Trabalhos futuros	71
A	Consultas de Teste	73

Lista de Figuras

2.1	Estrutura de um arquivo de índice invertido.	10
2.2	Regra de remoção do sufixo “ <i>ais</i> ”.	12
2.3	Distribuição: documentos relevantes/irrelevantes e recuperados/não recuperados [38].	20
3.1	Exemplo de grafo RDF.	24
3.2	Exemplo em RDFS.	25
3.3	Parte da hierarquia abaixo da classe “ <i>obra de arte</i> ” na ontologia do domínio. . . .	32
4.1	Esquema de arquivo XML de configuração.	40
4.2	Algoritmo de geração do arquivo de índice invertido.	42
4.3	Processo de recuperação.	43
4.4	Algoritmo de expansão de consulta.	45
4.5	Algoritmo de cálculo de similaridade entre dois termos.	46
4.6	Algoritmo de recuperação utilizando o modelo de espaço de vetores.	49

5.1	Tela do sistema, apresentando resultados da consulta original e as alternativas de correção ortográfica.	52
5.2	Tela de apresentação dos clips recuperados pelo sistema.	54
5.3	Estrutura de dados simplificada para as inferências.	56
5.4	Tempo de Processamento para as consultas de teste.	61
5.5	Resultados para a consulta “ <i>Não entendi seu trabalho. Explique</i> ”.	64
5.6	Resultados para a consulta “ <i>Por que você não dá então alguma coisa útil ao povo que está passando necessidade?</i> ”.	64
5.7	Resultados para a consulta “ <i>Por que você não dá algo para as pessoas que necessitam?</i> ”.	65
5.8	Resultados para a consulta “ <i>E o povo que participou da sua obra não ganha nada?</i> ”.	66
5.9	Resultados para a consulta “ <i>Como você vende sua obra?</i> ”.	67

Lista de Tabelas

5.1	Configurações de execução dos testes.	59
A.1	Consultas de teste.	73
B.1	Precisão para as consultas de teste.	77
B.2	Cobertura para as consultas de teste.	78
B.3	Medida-F para as consultas de teste.	79

Introdução

O problema de Recuperação de Informação (RI) é um problema clássico em ciência da computação, com uma grande aplicabilidade em diversos domínios do conhecimento humano, mais ainda na atualidade em que a quantidade de informação disponível em computadores cresce rapidamente. Uma das principais motivações para a criação de sistemas de RI é a de facilitar e otimizar a interação do usuário com grandes quantidades de informação.

No presente trabalho, a informação tratada foi um conjunto de entrevistas feitas com a artista contemporânea brasileira Ana Teixeira. As entrevistas foram feitas por Paula P. Braga do Departamento de Filosofia da Universidade de São Paulo, que atuou também como a especialista no domínio nas fases de aquisição do conhecimento requeridas para o desenvolvimento da aplicação. A aplicação desenvolvida como parte do trabalho tenta permitir aos seus usuários satisfazer suas consultas sobre um domínio específico, mas a aplicação foi implementada de um modo independente do domínio, e pode ser usado com diferentes tipos de vídeos, como cursos e seminários.

Um usuário pode entrar com a sua consulta em linguagem natural, e o sistema apresentará um conjunto de fragmentos de vídeo que satisfazem melhor a consulta dele. O processo utiliza uma ontologia de domínio para lidar com o problema da presença de sinônimos, permitindo recuperar os fragmentos de vídeo, maximizando os relevantes e minimizando os irrelevantes para a consulta.

Neste capítulo introdutório, apresentamos o problema, a motivação do trabalho, os objetivos do

mesmo, uma revisão dos trabalhos relacionados e a estrutura do presente documento.

1.1 Motivação

A motivação do presente trabalho foi a necessidade de permitir o acesso a um grande conjunto de dados (vídeos), permitindo que essa interação seja rápida e focada no que o usuário quer ver desse conjunto de dados. O domínio específico escolhido para a aplicação consiste na apresentação de um conjunto de entrevistas filmadas com a artista brasileira Ana Teixeira [30].

Em resposta a essa necessidade, foi desenvolvido de um sistema que permite a visualização desses vídeos como resposta a consultas específicas do usuário feitas em uma linguagem escrita, sem restrições de vocabulário, em um sistema computacional. A aplicação poderia ser montada em um espaço de exibição, como por exemplo uma galeria ou um museu.

As consultas são expandidas para permitir a recuperação dos vídeos não só pelas palavras utilizadas pelo usuário, mas por palavras sinônimas ou semanticamente relacionadas. Para o sistema ter acesso a esse conhecimento, foi utilizada uma ontologia do domínio.

Buscou-se uma implementação independente do assunto tratado nas entrevistas ou vídeos. Para isso, foi desenvolvida uma aplicação para a configuração, na qual se pudesse cadastrar os vídeos, associar conceitos (da ontologia) aos trechos de vídeos e gerar as estruturas para uma recuperação de informação eficiente. Assim, sem maiores mudanças o sistema poderia ser adaptado a outros domínios.

Um dos principais objetivos é comprovar se o uso de ontologias de domínio ajuda ou não na melhoria do desempenho do processo de recuperação em termos da qualidade das respostas mostradas pelo sistema.

1.2 Objetivos

Os principais objetivos do presente trabalho são:

- Implementar um método de recuperação de vídeos indexados por conceitos em um domínio específico, que permita processar consultas feitas em linguagem natural escrita.
- Verificar se o método desenvolvido melhora a recuperação em termos da relevância dos documentos recuperados.

Os objetivos específicos traçados são:

- Fazer um estudo bibliográfico para selecionar as técnicas de recuperação de informação (RI) mais adequadas para o problema.
- Projetar a utilização da ontologia como parte do processo de RI.
- Implementar o processo de RI.
- Desenvolver uma ontologia do domínio de arte contemporânea que contemple os conceitos e termos utilizados na entrevista.
- Implementar interfaces visuais para facilitar o processo de indexação e recuperação de informação por parte dos usuários.
- Testar o desempenho da qualidade dos resultados da aplicação utilizando medidas padrão, analisando principalmente a diferença entre a recuperação: (1) sem o uso da ontologia e, (2) com o uso da mesma.

1.3 Contribuições

A principal contribuição do trabalho é a implementação de um sistema de recuperação de informação, no caso, um conjunto de vídeos, que utiliza uma ontologia do domínio para melhorar a eficiência

do mesmo. A implementação foi projetada de maneira independente do domínio, assim a mesma aplicação poderá ser usada com diferentes vídeos, especificando os vídeos e mudando a ontologia para o domínio referido na coleção de vídeos.

O sistema foi testado para uma aplicação específica, com trechos de uma entrevista com a artista brasileira Ana Teixeira, gerando para isso uma ontologia sobre arte contemporânea.

Foi efetuada uma pesquisa sobre distintas ferramentas, tanto para a manipulação de ontologias quanto para processamentos simples de linguagem, como a remoção de afixos (Ver Seção 2.2.1) e a correção ortográfica. Foi comprovada a sua utilidade assim como foi verificada a possibilidade de integração delas em uma única aplicação complexa.

A versão final do sistema, incluindo a ontologia e os clips utilizados junto com a documentação dos mesmos, encontra-se disponível em <http://www.ime.usp.br/~cpaz/ontologies/arte/>.

1.4 Trabalhos relacionados

As ontologias têm sido recentemente usadas em recuperação de informação. O sistema Ontoseek [15] propôs o uso de ontologias para melhorar a qualidade das suas respostas em domínios restritos. O Ontoseek utiliza uma linguagem limitada para representação de conceitos e uma ontologia geral, Wordnet [27], para caçar conceitos e tratar sinônimos. Khan propôs em [21] as idéias de *distância conceitual*¹, e as aplicou na recuperação de dados de áudio usando um mecanismo de expansão de consulta que trabalha com linguagem natural no domínio de esportes.

Um trabalho relacionado com recuperação de vídeo baseado em conteúdo foi desenvolvido na Universidade de Cambridge no Laboratório de Máquinas Inteligentes [5], que apresenta um enfoque estatístico para recuperação de documentos multimídia. Alguns trabalhos utilizam o padrão MPEG-

¹A distância conceitual permite medir o grau de proximidade entre conceitos, quanto eles estão relacionados.

7 para descrição e busca de conteúdo de áudio e vídeo para prover informação ao sistema e explorá-la na hora da recuperação [3].

Um uso de ontologias para recuperação de vídeo foi proposto pelo grupo de pesquisa da Universidade de Amsterdã [40], oferecendo um método interativo para recuperação de vídeo guiando a interação do usuário com informação de domínio extraída de uma ontologia.

Em Janeiro de 2001, o Museu de Arte Krannert em Champaign, nos Estados Unidos, apresentou uma exibição do trabalho do escultor Jacques Lipchitz (1891-1973). Além das esculturas e desenhos do artista, a exposição mostrou um software desenvolvido por Bruce Bassett e a companhia Hystor Systems titulado “*Conversation with Jacques Lipchitz: A Breakthrough in Interactivity*” (Conversa com Jacques Lipchitz: Um salto na interatividade). Este sistema começou a ser desenvolvido na década de setenta e, na sua primeira versão, utilizava fitas VHS manualmente selecionadas e reproduzidas de acordo com a consulta. A versão apresentada em 2001 atualizou este trabalho pioneiro usando vídeos digitalizados. Sendo o trabalho parte de uma iniciativa comercial, não existe informação disponível sobre a sua implementação. De um modo ou de outro, esse sistema é precursor do sistema aqui descrito. Ele rompe com as práticas de exibição tradicionais e provê uma interação muito atraente entre o trabalho do artista e os espectadores, trazendo de volta os pensamentos do escultor já falecido [22, 4].

O presente trabalho utiliza uma ontologia restrita ao domínio da arte contemporânea. Apresenta uma medida de similaridade baseada na mesma, utilizando tanto a informação de hierarquia de conceitos da ontologia quanto os relacionamentos entre eles para estimar a sua proximidade. O trabalho vista também a reutilização através do uso de ferramentas freqüentemente usadas no processamento de ontologias.

1.5 Estrutura do documento

Este texto está estruturado em seis capítulos. O Capítulo 2 apresenta o problema de recuperação de informação (RI), os processos requeridos pelos sistemas de RI e as principais técnicas usadas em cada um deles. O Capítulo 3 descreve o conceito de ontologia, para depois apresentá-lo dentro do contexto da RI, mostrando o uso de ontologias nesse tipo de sistemas. A aplicação desenvolvida e a descrição dos mecanismos utilizados nela são apresentados no Capítulo 4, a execução do sistema e os resultados obtidos dos testes são mostrados no Capítulo 5. O Capítulo 6 apresenta uma discussão sobre os resultados obtidos, as contribuições feitas e os trabalhos futuros a serem desenvolvidos como continuação deste trabalho.

Recuperação de Informação

No presente capítulo será apresentado o problema de Recuperação de Informação. Serão descritos os principais processos envolvidos em um sistema de recuperação de informação: Indexação e Recuperação. Finalmente se mostrará as medidas utilizadas na avaliação desse tipo de sistemas.

2.1 Definições

Recuperação de Informação (RI) pode ser definida como a aplicação de tecnologia computacional à aquisição, organização, armazenamento, recuperação e distribuição de informação [18].

O propósito da RI é satisfazer uma necessidade de informação do usuário, a qual é expressa em uma consulta que procura encontrar documentos relevantes. Essa consulta pode não expressar da melhor forma a necessidade de informação, e pode conter palavras com erros ortográficos ou usadas de forma inadequada. O Sistema de RI deve tentar recuperar os documentos relevantes para a necessidade do usuário baseado na informação fornecida pela consulta.

Uma grande dificuldade na RI é saber quando um documento é relevante ou não para a necessidade do usuário. Esse conceito é básico na RI pois o seu objetivo primordial consiste em recuperar todos os documentos relevantes e a menor quantidade possível de documentos irrelevantes [2].

2.1.1 Problemas na recuperação de informação

Os primeiros sistemas de busca booleana (Ver Seção 2.3.1) foram sistemas de RI baseados em palavras-chave. Esses sistemas recebem uma consulta e devolvem os documentos que contenham literalmente alguma palavra dela.

A linguagem como ela é usada em textos reais, possui várias características complicadas e pouco previsíveis, i.e. exceções, que podem levar os sistemas de RI baseados em palavras-chave a confundir textos relevantes e irrelevantes. As mais comuns destas características são [25]:

- **Sinonímia:** quando existem diversos termos para descrever um mesmo objeto ou conceito, um sistema de recuperação de informação baseado em semelhança de palavras chave só recuperará aqueles documentos em que o objeto ou conceito esteja descrito com os mesmos termos com que a consulta foi elaborada.

Por exemplo, se alguém estiver procurando algum lugar para se hospedar no Rio de Janeiro, poderia colocar em uma consulta: *“pousada no Rio de Janeiro”*. Só os documentos em que a palavra *“pousada”* estiver presente serão recuperados. Os documentos em que a informação de hospedagem esteja referenciada por exemplo pelas palavras: *“hotel”* ou *“pensão”*, não serão recuperados.

- **Polissemia:** uma palavra é polissêmica quando ela tem vários significados, expressando diversos conceitos. Neste caso a recuperação de informação por uma palavra polissêmica pode trazer documentos relacionados não só ao conceito procurado como também aos outros conceitos que são expressos pela mesma palavra. Neste caso, por exemplo, a busca da palavra *“recepção”* no sentido de festa, pode fazer o sistema recuperar documentos relacionados a *“recepção”* no sentido do impacto que alguma coisa (uma obra) tem em uma pessoa (o espectador).

Estes problemas fazem com que os sistemas de recuperação de informação baseados em palavras-

chave tenham um limite no seu desempenho conhecido como “barreira da palavra-chave”¹ [25]. A informação a ser tratada pelo sistema deve ser organizada de modo que possa-se lidar com o problema da sinonímia. A polissemia não será tratada neste trabalho, pois as aplicações que se pretendem abarcar são de domínio restrito, o qual diminui o impacto da polissemia no sistema.

Um Sistema de RI apresenta dois processos principais bem definidos e diferenciados: o Processo de Indexação, no qual a coleção de documentos é indexada para permitir uma recuperação posterior eficiente; e o Processo de Recuperação no qual, dada uma consulta, os documentos mais fortemente relacionados a ela são recuperados.

2.2 Processo de Indexação

Um sistema básico de RI, geralmente, consta de três componentes de entrada principais: a coleção de documentos, o dicionário e um arquivo de índice invertido (ou simplesmente arquivo invertido). A coleção de documentos contém a informação que vai ser recuperada pelo sistema. O dicionário é uma lista em ordem alfabética das palavras que aparecem no banco de dados. O arquivo de índice invertido armazena as ocorrências de cada palavra do dicionário e permite acesso eficiente a grandes bancos de dados. A entrada do processo de indexação é a coleção de documentos e as suas saídas são o dicionário e o arquivo invertido.

A Figura 2.1 apresenta uma fração de um arquivo invertido. A estrutura de uma entrada em um arquivo invertido pode ser [18]:

- **Token:** o token que representa a palavra, ou a raiz da palavra indexada. O processo de obtenção da raiz das palavras será explicado posteriormente na Seção 2.2.1.
- **Contagem de documentos:** o número de documentos em que o token aparece.
- **Contagem de frequência total:** o número de vezes que o token aparece no total de docu-

¹Do inglês *keyword barrier*.

mentos. Indica quão comum é o token na coleção de documentos.

- **Informação por documento:** uma lista de informações associadas às ocorrências do token em cada documento. Para cada documento em que o token aparece, existe uma entrada nessa lista, contendo as seguintes informações:
 - **Contagem de frequência:** quantas vezes o token aparece no documento. Este valor indica, a grosso modo, se o documento é realmente relevante a esse conceito ou não.
 - **Posição:** contem uma lista das posições relativas em que o token aparece no documento.

Figura 2.1: Estrutura de um arquivo de índice invertido.

As *palavras de parada*² são termos que aparecem tão freqüentemente nos textos da coleção que perdem sua utilidade como termos para a busca. Elas não são consideradas na criação do arquivo invertido. Os artigos e preposições são comumente considerados como palavras de parada, por exemplo “um”, “de” e “cada” são algumas das utilizadas no sistema. A lista utilizada neste protótipo foi obtida de Snowball³.

²Do inglês *stopwords* ou *stop list*.

³Snowball é uma biblioteca de recursos, incluindo algoritmos de remoção de afixos e listas de palavras de parada em diversos idiomas. Disponível em <http://snowball.tartarus.org/portuguese/stop.txt>.

Alguns sistemas de RI armazenam, além da coleção, informação sobre as palavras utilizadas nos documentos, que pode ser utilizada para enriquecer o processo de recuperação. Isso permite inferir relações entre as palavras (por exemplo, sinônimos) utilizadas na consulta e as presentes na coleção. Este tipo de informação adicional pode estar em uma Ontologia (Ver Capítulo 3).

Durante a indexação, os sistemas podem executar operações adicionais sobre o texto da coleção. Algumas delas são explicadas a seguir.

2.2.1 Remoção de afixos

Consiste na extração dos radicais das palavras, e é conhecido pelo termo em inglês *Stemming*. Após este processo, as variantes de palavras são associadas ao mesmo radical, e só uma entrada é criada no dicionário para elas. Com isso, a quantidade de armazenamento requerida decresce, pois o número de palavras indexadas diminui.

Por exemplo as palavras “*cumprido*”, “*cumpriu*” e “*cumprirá*” são associadas ao mesmo radical “*cumpr*” que, como neste caso, não precisa ser uma palavra válida, mas permite associar todas as derivações ao mesmo radical. Com essa associação é possível fazer com que consultas por palavras derivadas do mesmo radical recuperem os mesmos resultados, o que normalmente é válido pois documentos relacionados com palavras derivadas são relevantes para as mesmas consultas.

Um algoritmo de remoção de afixos simples aplica heurísticas para detectar os pontos dos quais os afixos podem ser extraídos. Essas heurísticas obedecem a regras simples, como a detecção de marcas que normalmente são associadas com afixos. Por exemplo, a terminação “*mente*” é utilizada na formação de advérbios no português, assim como a terminação “*inho*” é comum na formação de diminutivos dos substantivos. Essas regras e as suas exceções são especificadas para o algoritmo. Um exemplo de exceção para a terminação “*inho*” é a palavra “*caminho*”, em que “*inho*” não é sufixo e sim parte da palavra.

A maioria desses algoritmos heurísticos removem só sufixos, pois sufixos mudam a catego-

ria sintática da palavra, mas conservam o sentido da palavra em que são colocados. Um algoritmo muito utilizado baseado em regras heurísticas é o algoritmo de Porter [31], que remove as terminações morfológicas mais comuns das palavras, e foi inicialmente criado para palavras em inglês. Posteriormente, Porter desenvolveu a Snowball, uma linguagem para manipulação de cadeias de caracteres que pode ser utilizado para criar algoritmos de remoção de afixos gerando programas em Java, ou ANSI C. Existem algoritmos prontos para diversos idiomas, incluindo o português do Brasil⁴.

Em 2001, foi desenvolvido um algoritmo de remoção de afixos para português [29] composto por uma série de regras aplicadas sequencialmente. Estas regras permitem a remoção do plural, o feminino, a forma adverbial *mente*, o aumentativo e diminutivo e algumas terminações comuns para nomes e verbos respectivamente. Cada regra especifica o tipo de sufixo, o sufixo que será removido, o menor tamanho do radical que pode ficar após a remoção, o sufixo substituto do removido, e as exceções à regra, sendo esses dois últimos opcionais. Por exemplo, a Figura 2.2 mostra a entrada da regra “ais”.

Tipo de Sufixo: Plural
Sufixo: ais
Menor tamanho do radical que pode ficar após a remoção do sufixo: 1
Sufixo substituto: al.
Exceções: cais, mais.

Figura 2.2: Regra de remoção do sufixo “ais”.

Existe uma implementação disponível do algoritmo em ANSI C⁵ e uma versão dele em Java foi implementada para ser utilizada neste trabalho.

⁴Os algoritmos estão disponíveis em <http://snowball.tartarus.org/>.

⁵Disponível em <http://www.cs.mdx.ac.uk/research/PhDArea/rs1p/RSLP.htm>.

2.2.2 Expansão de termos do índice

Durante a criação do arquivo invertido, os termos não presentes no texto mas que estejam relacionados aos presentes podem ser indexados também. A relação ou similaridade entre os termos está definida em uma estrutura associada à coleção, que pode ser uma ontologia, e sua utilização será detalhada no Capítulo 3. A expansão normalmente não é efetuada durante a indexação, pois sua aplicação faria com que o índice crescesse muito, o que poderia afetar o rendimento do sistema de maneira desnecessária.

2.2.3 Cálculo de pesos

Para o cálculo da relevância de um documento são usados alguns valores que associam cada termo do arquivo invertido aos documentos nos quais ele aparece. Esses valores podem ser calculados na hora da consulta ou podem ser armazenados no arquivo invertido para evitar sobrecarregar o processo de consulta. Na próxima seção explicará-se o cálculo dos pesos, pois ele varia dependendo do tipo de sistema de RI empregado.

2.3 Processo de Recuperação

Este processo começa quando a consulta do usuário é apresentada ao sistema. São três os passos básicos presentes nesta etapa: o processamento da consulta, o cálculo da relevância dos documentos e a ordenação dos resultados de acordo com a relevância para serem apresentados ao usuário.

Existem basicamente, na literatura de RI, três enfoques para a recuperação [18]: a busca booleana, a recuperação ordenada e a recuperação probabilística. Eles serão apresentados a seguir.

2.3.1 Busca booleana

Neste tipo de busca o usuário faz a consulta na coleção conectando palavras mediante operadores lógicos (AND, OR, e NOT) e é o tipo de busca freqüentemente usado nos motores de busca. Um motor de busca retorna o conjunto de documentos na coleção que satisfaz a consulta do usuário. Por exemplo, a consulta “obra AND arte” deverá retornar todos os documentos contendo os termos “obra” e “arte”, i.e.: $docs_{obra} \cap docs_{arte}$ em que $docs_{palavra}$ é o conjunto dos documentos em que “palavra” aparece. Do mesmo modo a consulta “mostra OR bienal” recupera os documentos em que aparecem ou a palavra “mostra” ou a palavra “bienal”: $docs_{mostra} \cup docs_{bienal}$.

Um operador freqüentemente permitido neste tipo de consultas é o operador de proximidade de termos [18]. A consulta “obra /2 arte” deve recuperar os documentos em que as palavras “obra” e “arte” aparecem no máximo com uma distância de duas palavras uma da outra. Por exemplo, o sistema não recuperaria documentos em que essas palavras aparecessem dentro da frase: “a obra dele usa um novo estilo de arte” para essa consulta.

Esse tipo de busca pode ser melhorada incrementando sinônimos à consulta para usar maior quantidade de termos em um processo chamado de expansão de consulta (Ver Seção 4.2.2). Por exemplo, dada a consulta “obra AND arte” ela pode ser expandida a “(obra OR trabalho) AND arte”. Com isso, documentos que usam sinônimos para se referir ao assunto, podem ser recuperados.

As buscas booleanas podem aproveitar o processo de remoção de afixos detalhado na Seção 2.2.1, removendo os afixos tanto na hora da criação do arquivo invertido quanto no processamento da consulta. Desse modo, a consulta por “obra AND arte” poderia recuperar documentos que utilizem a frase “as obras de arte foram apresentadas em ...” por exemplo, pois removendo os afixos de “obra” e de “obras” se chega na mesma raiz.

Apesar dessas melhoras, as buscas booleanas apresentam alguns problemas: elas devolvem conjuntos de resultados muito grandes, a lógica para efetuar consultas é complexa, devolvem um conjunto desordenado de resultados, e além disso oferecem uma recuperação booleana, i.e. o do-

cumento é ou não é relevante, não existe a noção de grau de relevância. Além disso apresentam um desempenho pobre em coleções com poucos documentos ou com documentos muito diferentes [18]. Este tipo de buscas é adequado para usuários experientes, mas é difícil para usuários esporádicos.

2.3.2 Recuperação ordenada

Os sistemas de busca na Web frequentemente são baseados em uma técnica que ordena os resultados da busca com base na distribuição de frequência dos termos da consulta na coleção de documentos. Por exemplo, um documento que contém várias ocorrências de uma palavra que não é comum na coleção de documentos, provavelmente é relevante para uma consulta que contenha tal palavra. Desse modo, palavras comuns são consideradas com menor importância na ordenação dos resultados.

Esse tipo de busca é usualmente empregado em interfaces de busca nas quais se permite aos usuários entrar com consultas em uma linguagem irrestrita, isto é, sem operadores ou uma sintaxe formal. Para permitir isso, os motores removem as palavras de parada e realizam algumas operações sobre as outras palavras, sendo a mais comum dentre elas a remoção de afixos. Neste enfoque, os documentos e as consultas são tratados como vetores no espaço multi-dimensional definido pelos termos existentes na coleção, e em que a frequência de aparição dos termos é o valor escalar associado ao vetor na dimensão do termo. Assim, o cálculo da similaridade entre a consulta e cada documento dá um valor de relevância do documento para com a consulta. Finalmente, os documentos são ordenados de acordo com a medida de similaridade deles com a consulta.

O modelo de espaço de vetores

Seja D uma coleção de n documentos, $d_1, d_2, \dots, d_n$, nos quais aparecem m diferentes termos, $t_1, t_2, \dots, t_m$. Os termos t_i definem um espaço m -dimensional, em que um documento d é representado como um vetor:

$$d = (w_{1,d}, w_{2,d}, \dots, w_{m,d}) \quad (2.1)$$

sendo $w_{i,d}$ o peso do i -ésimo termo no documento d . O peso de um termo em um documento representa a importância que ele tem no documento, e é uma estimativa de quanto o documento é relevante para esse termo. Do mesmo modo, uma consulta q pode ser representada como um vetor:

$$q = (w_{1,q}, w_{2,q}, \dots, w_{m,q}) \quad (2.2)$$

em que $w_{i,q}$ é o peso do i -ésimo termo na consulta q , este peso representa a presença do termo na consulta sendo esse valor binário. O grau de similaridade entre dois documentos, ou entre um documento e uma consulta no espaço vetorial, em sua forma mais simples, pode ser o produto escalar dos vetores que os representam[34]:

$$\text{sim}(d, q) = \sum_{t=1}^m w_{t,d} \cdot w_{t,q} \quad (2.3)$$

Para efeitos de exemplo, consideremos $w_{t,d}$ como o número de vezes que o termo t aparece no documento d . Suponha uma coleção com três documentos, nos quais aparecem os termos: “obra”, “de”, “arte”, “tratado”, “música”, “moderna”, “contemporânea” e “Gross” que representarão as dimensões de 1 a 8 do espaço dimensional que elas definem. Os documentos são:

d_1 : “obras de arte”

d_2 : “tratado de música moderna e de música contemporânea”

d_3 : “obra moderna de Gross”

Considerando um processo de remoção de afixos em que o radical “obra” é extraído da palavra “obras” em d_1 , os documentos podem ser vistos como vetores no espaço vetorial definido pelos termos como:

$$d_1 = (1, 1, 1, 0, 0, 0, 0, 0)$$

$$d_2 = (0, 2, 0, 1, 2, 1, 1, 0)$$

$$d_3 = (1, 1, 0, 0, 0, 1, 0, 1)$$

Pode-se ver que as palavras de parada, como “de” trazem pouca informação que permita discernir a relevância ou não relevância de algum documento, pois provavelmente aparecerá na maioria

dos documentos. Seja uma consulta:

$q = \text{"quais são as obras de Carmela Gross"}$

a consulta, depois do processo de remoção de afixos, também pode ser representada no espaço vetorial pelo vetor:

$$q = (1, 1, 0, 0, 0, 0, 0, 1)$$

O processo de recuperação ordenada, deve calcular a similaridade de cada documento com a consulta, segundo a fórmula apresentada anteriormente:

$$\text{sim}(d_1, q) = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 = 2$$

$$\text{sim}(d_2, q) = 0 \cdot 1 + 2 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 = 2$$

$$\text{sim}(d_3, q) = 1 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 3$$

Dando como resultado que o documento d_3 é o mais similar com a consulta, o qual é, segundo a nossa intuição, certo. Além disso, segundo o cálculo de similaridade, d_1 e d_2 são igualmente similares à consulta, mas podemos observar que d_2 obteve o seu grau de similaridade pela presença da palavra “de”, o tópico principal do documento é aparentemente “música”, mesmo assim a presença de uma palavra de parada pode causar esse tipo de erros na recuperação.

Existem termos que, dentro do contexto da coleção, são muito comuns e que assim como as palavras de parada não são bons discriminantes de relevância de documentos. Esses termos podem ser detectados conhecendo a frequência total deles na coleção e esse dado pode ser usado para diminuir o peso do termo no documento. Assim, uma fórmula muito empregada para o cálculo do peso de um termo no documento, chamada de TF-IDF é [18]:

$$w_{t,d} = tf_{t,d} \cdot idf_{t,d} \quad (2.4)$$

em que $tf_{t,d}$ (*term frequency*) é o número de aparições do termo t no documento d e $idf_{t,d} = \log\left(\frac{n}{n_t} + 1\right)$ (*inverse document frequency*) mede a raridade relativa do termo na coleção e permite saber se ele é ou não um bom discriminante. n_t é o número de documentos em que o termo t

aparece.

Uma medida de similaridade derivada da apresentada na Equação 2.3 é a similaridade do cosseno, que consiste no cálculo do cosseno do ângulo formado pelos vetores dos quais se deseja achar a similaridade [17].

$$sim(d, q) = \frac{\sum_{t=1}^m w_{t,d} \cdot w_{t,q}}{\sqrt{\sum_{t=1}^m w_{t,d}^2 \cdot \sum_{t=1}^m w_{t,q}^2}} \quad (2.5)$$

Podemos perceber que quando esses vetores coincidem, i.e. são iguais, formam ângulo de 0 graus e portanto o cosseno deles é máximo, 1. Este modelo tem sido usado em vários experimentos de recuperação de informação, entre os quais se destaca o SMART⁶ que é um dos maiores esforços na pesquisa na área de recuperação de informação.

O modelo estendido de espaço de vetores

O modelo estendido de espaço de vetores [39] permite considerar o vetor como uma coleção de sub-vetores. Esses sub-vetores são usados para representar diferentes aspectos dos documentos da coleção. Por exemplo, um dos sub-vetores pode representar os pesos do documento considerando o texto completo, enquanto outro sub-vetor pode representar o documento como uma coleção de palavras-chave manualmente associadas.

A similaridade total neste modelo estendido é calculada como a soma ponderada das similaridades nos sub-vetores.

$$sim(d, q) = \sum_{sub-vetor\ i} \alpha_i \cdot sim_i(d_i, q_i) \quad (2.6)$$

em que d_i e q_i são os i -ésimos sub-vetores do documento e da consulta respectivamente, e α_i é a importância do sub-vetor i na similaridade total. A similaridade usada em cada sub-vetor pode

⁶<ftp://ftp.cs.cornell.edu/pub/smart/>

ser a similaridade coseno apresentada anteriormente.

$\sum_{sub-vetor\ i} \alpha_i$ deve ser 1.

Em média os sistemas de recuperação que utilizam o modelo de espaço de vetores apresentam uma maior performance que os sistemas de recuperação booleana, no sentido que devolvem ao usuário um conjunto de documentos mais relevantes, além de apresentá-los em uma ordem que reflete a relevância deles para a necessidade de informação do usuário [37].

2.3.3 Recuperação probabilística

O enfoque de recuperação probabilística tenta formalizar as idéias por trás da recuperação ordenada em termos de teoria de probabilidades. A probabilidade a ser avaliada é a de um documento ser relevante para a consulta [18, 11].

Esse enfoque, normalmente tem um desempenho menor do que o que utiliza espaço de vetores, mas apresenta uma base teórica melhor fundamentada, e sua saída é um ranking da probabilidade de relevância dos documentos para a consulta do usuário e não da similaridade com a consulta como no caso do espaço de vetores [10].

2.4 Avaliação de sistemas de recuperação de informação

A avaliação dos Sistemas de RI é fundamental para o estabelecimento de um ponto de comparação entre os sistemas sendo desenvolvidos. A idéia é medir a capacidade do sistema de satisfazer a necessidade de informação do usuário em termos da relevância dos documentos recuperados.

Existem duas medidas padrão para a avaliação da eficiência, e ambas estão baseadas em relações entre os conjuntos de documentos relevantes e não relevantes, assim como os conjuntos de documentos recuperados e não recuperados (Ver Figura 2.3).

	Relevantes	Não Relevantes	
Recuperados	$A \cap B$	$\bar{A} \cap B$	B
Não Recuperados	$A \cap \bar{B}$	$\bar{A} \cap \bar{B}$	$\bar{B}$
	A	$\bar{A}$	N

Figura 2.3: Distribuição: documentos relevantes/irrelevantes e recuperados/não recuperados [38].

Com base nessa distribuição, podemos definir [38]:

- **Precisão:** É a razão entre o número de documentos recuperados pelo sistema que são relevantes para a consulta e o número total de documentos recuperados.

$$P = \frac{\text{Número de documentos relevantes recuperados}}{\text{Número total de documentos recuperados}} = \frac{|A \cap B|}{|B|} \quad (2.7)$$

Por exemplo, se os sistema recuperou 6 documentos para uma consulta, em que 3 de eles foram relevantes, a medida de precisão do sistema é de 0,5 ou 50%. A polissemia pode produzir baixas taxas de precisão no caso de alguns documentos irrelevantes serem recuperados.

- **Cobertura (*Recall*):** Podem existir muitos documentos na coleção que o usuário considera relevante, mas só alguns deles serão recuperados pelo sistema. A medida de cobertura é a razão do número de documentos relevantes recuperados e o total de documentos relevantes na coleção.

$$R = \frac{\text{Número de documentos relevantes recuperados}}{\text{Número total de documentos relevantes na coleção}} = \frac{|A \cap B|}{|A|} \quad (2.8)$$

A cobertura é difícil de se medir, pois requer o conhecimento do número total de documentos relevantes no banco de dados, que tem que ser determinado manualmente. Para fazer essa medida, normalmente se estabelece um limite superior baseado na seleção de documentos que deveriam ter sido recuperados mas não foram. Quanto maior for o esforço manual para

achar esses documentos, mais precisa é a medida de cobertura. A sinonímia pode induzir uma baixa cobertura, pois é possível que não sejam recuperados alguns documentos relevantes que estão relacionados só a sinônimos das palavras utilizadas na consulta.

As medidas de precisão e cobertura são independentes e a melhora de uma delas pode implicar perda na outra. Um sistema pode ter precisão de 100% se ele devolve só um documento da coleção que com certeza é relevante. Uma cobertura de 100% é obtida por um sistema que devolve a qualquer consulta todos os documentos da coleção. Para evitar esse problema, utiliza-se uma medida que combina precisão e cobertura, chamada de Medida-F [38].

$$F = \frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}} \quad (2.9)$$

Em que $0 < \alpha < 1$ indica a importância relativa da precisão para o usuário. O caso especial em que $\alpha = 0,5$, que dá a mesma importância para a precisão e a cobertura, resulta sendo a média harmônica delas:

$$F = \frac{2 \cdot R \cdot P}{R + P} \quad (2.10)$$

A vantagem do uso da Medida-F é que uma melhora nela indica uma melhora tanto da precisão quanto da cobertura.

Usualmente na avaliação dos sistemas de RI são consideradas as medidas de cobertura em n valores diferentes entre 0.0 e 1.0, usualmente em intervalos de 0.1, fazendo $n = 11$. Essa avaliação é conhecida como Média de Precisão de n pontos (*n-point average precision*).

2.5 Considerações finais

Nesse capítulo foi apresentado o problema de recuperação de informação, junto com as principais componentes que compõem um sistema de RI. Foram descritas também as medidas frequentemente utilizadas na avaliação desses sistemas.

Pode-se ressaltar assim que esse tipo de sistemas é muito utilizado na atualidade, devido à sua aplicabilidade. Existem, por exemplo, muitos trabalhos relacionados a sistemas de pesquisa na Web, que é a maior coleção de documentos no mundo. Mesmo assim, apesar dos grandes avanços na área, os usuários ainda têm dificuldades na recuperação de informação relevante e na obtenção de respostas rápidas. Uma pergunta que fica em aberto na área é a de quanto o melhor entendimento do comportamento do usuário deve afetar o projeto de sistemas de RI [2].

Uso de Ontologias em RI

O conceito de *ontologia* é bastante discutido, e aqui será utilizado o conceito associado com compartilhamento de conhecimento. No presente capítulo será apresentada a justificativa da utilização de ontologias em aplicações de RI. Também serão apresentados os mecanismos para utilizar ontologias em sistemas de RI, e as medidas utilizadas para definir a similaridade entre conceitos expressos em uma ontologia. Finalmente, será apresentada a ontologia desenvolvida para a aplicação.

3.1 Definições

Uma das definições mais conhecidas do termo indica que uma ontologia é a “*especificação explícita de uma conceitualização*”[14]. Esta definição é aceita no entorno relacionado com compartilhamento de conhecimento em sistemas baseados em conhecimento. Assim, uma ontologia descreve todos os conceitos que existem no domínio do sistema e esses conceitos junto com os relacionamentos entre eles formam o vocabulário dela. Os relacionamentos impõem uma estrutura no domínio e restringem as possíveis interpretações dos conceitos.

As ontologias dão uma descrição concisa, comum e declarativa de informação semântica, idealmente independente da representação semântica ou dos modelos conceituais. Os modelos de domínio ampliam a acessibilidade da informação permitindo que várias ontologias pertençam a

vários grupos de usuários [20]. É muito importante nas aplicações que utilizam ontologias, que elas capturem os elementos do domínio que sejam de interesse do usuário da aplicação e que expressem toda a informação necessária para a aplicação.

3.2 Linguagens de representação

Uma das características principais das ontologias é que elas permitem compartilhar informação de um domínio, que pode ser utilizada por diversas aplicações. Um requisito para compartilhamento é a padronização da linguagem de representação. Nesse sentido, existem diversas linguagens que foram surgindo para a representação de ontologias que serão apresentadas nesta seção [9].

RDF¹ [6] foi desenvolvido pelo W3C² para a criação de metadados que permitam a descrição de recursos na Web. Um recurso é qualquer coisa que possa ser identificada, física (um país ou um continente) ou lógica (um item de informação em uma página web). RDF é uma linguagem de especificação que permite definir grafos que estabelecem, através de propriedades e restrições de valor, relações entre recursos ou entre recursos e valores.

Na Figura 3.1 se mostra um grafo RDF, que contém os recursos Colômbia, e América do Sul, relacionados pela propriedade continente. Além disso, Colômbia tem outra propriedade: população, cujo valor é 42.310.775.

Figura 3.1: Exemplo de grafo RDF.

¹Resource Definition Framework.

²World Wide Web Consortium.

serão utilizadas na implementação. A seguir são apresentadas as principais componentes de OWL que foram utilizadas neste trabalho e as suas correspondentes construções em OWL:

- **Classe:** Como em RDFS, é um descritor de um conjunto de objetos, abstratos ou do mundo real agrupados por compartilhar algumas propriedades, por exemplo: Objeto Geográfico, País, ou Continente. Uma classe é representada pela construção **owl:Class**. A classe **owl:Thing** é a que agrupa todos os objetos. A seguinte sentença em OWL declara a classe Objeto Geográfico:

```
<owl:classrdf:ID='`Objeto_Geográfico`' />
```

- **Especialização:** Uma classe X é uma especialização de uma outra classe Y se todos os indivíduos que fazem parte de X pertencem a Y também, e se diz que X é uma sub-classe de Y . Esse tipo de relação se conhece como *é-um*⁵.

Com a relação de especialização podem se formar hierarquias de classes. Por exemplo, a classe País é uma especialização de Objeto Geográfico. A construção **rdf:subClassOf** é utilizada para denotar especialização em OWL. Todas as classes OWL são sub-classes de **owl:Thing**. A seguir o exemplo de criação da classe País como sub-classe de Objeto Geográfico.

```
<owl:classrdf:ID='`País`'>
  <rdfs:subClassOf>
    <owl:classrdf:about='`#Objeto_Geográfico`'>
      </rdfs:subClassOf>
    </owl:class>
```

- **Relações:** Como as propriedades definidas em RDFS, as relações permitem estabelecer propriedades que vinculam uma classe (ou um indivíduo) com outros. Essas relações são denotadas por um nome e, como qualquer relação, seu comportamento é restringido pela especificação do seu domínio e sua imagem.

⁵Do inglês *is-a*.

Por exemplo, a relação `continente` tem como domínio a classe `País` e como imagem a classe `Continente`. `rdf:Property` é utilizada para expressar relações. OWL oferece dois tipos de relações: `owl:ObjectProperty` indica que a relação associa dois indivíduos, enquanto `owl:DatatypeProperty` permite associar um indivíduo a um valor. O seguinte fragmento OWL cria a propriedade `continente` descrita anteriormente.

```
<owl:ObjectProperty rdf:ID='continente'>
  <rdfs:range rdf:resource='#Continente' />
  <rdfs:domain rdf:resource='#País' />
</owl:ObjectProperty>
```

- **Indivíduos:** Um indivíduo é uma instância de uma classe. Um indivíduo pode pertencer a mais de uma classe em OWL. Por exemplo `Colômbia` é um indivíduo da classe `País`, e sua declaração em OWL é apresentada a seguir:

```
<País rdf:ID='Colômbia'>
  <continente rdf:resource='#América_do_Sul' />
  <população rdf:datatype='http://www.w3.org/2001/XMLSchema:integer'>
    42310775</população>
</País>
```

- **Classes equivalentes:** Duas classes são equivalentes se agrupam os mesmos indivíduos. Esta construção serve para expressar termos alternativos (sinônimos), e em OWL é expressa por `owl:equivalentClass`. Por exemplo, `País` tem como sinônimo `Nação`, para o qual se estabelece uma equivalência entre essas classes. Em OWL essa equivalência se expressa como:

```
<owl:class rdf:ID='Nação'>
  <owl:equivalentClass>
    <owl:class rdf:about='#País'>
  </owl:equivalentClass>
</owl:class>
```

3.3 Utilização em recuperação de informação

As ontologias são utilizadas em recuperação de informação principalmente para lidar com o problema da sinonímia, já que com elas é possível representar os conceitos do domínio que definem, e

cada conceito pode ser associado a um termo e aos seus sinônimos.

3.3.1 Elaboração de ontologias

As ontologias representam o conhecimento do domínio, e esse conhecimento nem sempre é explícito, freqüentemente é implícito, e só pode ser extraído das pessoas especialistas no domínio utilizando um processo de aquisição de conhecimento.

A existência de ferramentas simples para elaboração de ontologias (Ver Seção 3.5) facilitam o processo de aquisição. Estas ferramentas permitem, entre outras coisas, a edição visual das hierarquias de conceitos e das relações. Assim, o especialista pode utilizar esse tipo de ferramentas, com menor participação do desenvolvedor que normalmente não conhece o domínio.

Na área de pesquisa de Engenharia de Ontologias [7, 28], é conhecido que o desenvolvimento de ontologias é uma atividade iterativa, e em cada iteração deve se avaliar o progresso e validá-lo. Isso inclui a verificação de se a ontologia representa o domínio de forma adequada e completa para a aplicação, e se é adequada para ser reutilizada em outras aplicações.

3.3.2 Medidas de similaridade em RI

A utilização de ontologias em RI está fortemente ligada ao conceito de *similaridade*, pois as ontologias permitem definir graus de similaridade ou proximidade semântica entre termos (palavras). Assim, pode-se recuperar documentos que utilizam palavras similares às utilizadas nas consultas, sem necessidade de contar com as mesmas palavras como quando se trabalha só com palavras-chave.

Existem diversos trabalhos que propõem medidas de similaridade entre termos em uma hierarquia de conceitos, alguns deles são:

- Uma medida de similaridade em função inversa da distância entre dois conceitos de uma

hierarquia foi apresentada em [32]. Nesse trabalho, dada uma rede de conceitos enlaçadas por relações do tipo *é-um*, a distância entre dois conceitos é o mínimo número de arestas (relações) que separam os dois conceitos. Essa medida pode ser generalizada para encontrar a similaridade entre dois conjuntos de conceitos, que no caso de RI podem ser consultas ou documentos, por exemplo. Assim, essa medida pode ser usada para calcular a similaridade entre um documento e uma consulta sem utilizar o modelo de espaço de vetores.

- WordNet [27] é um banco de dados léxico projetado para ser usado por programas de computador, diferentemente da maioria dos dicionários eletrônicos, cujo objetivo é atender necessidades humanas. Mostra nomes, verbos, adjetivos e advérbios organizados em conjuntos de sinônimos e estabelece algumas relações semânticas entre palavras destacando a hiponímia (sub-classes) e a hiperonímia (super-classes).

Em [33] foi proposta uma medida de similaridade que combina a informação léxica obtida do WordNet com a probabilidade de ocorrência das palavras na coleção. A medida de similaridade é dependente da coleção. A similaridade de dois conceitos (C_1 e C_2) é inversamente proporcional à probabilidade do conceito C (a palavra representativa ou qualquer dos seus sinônimos) aparecer na coleção, onde C é um conceito que domina os conceitos C_1 e C_2 , i.e. que está acima deles na hierarquia de hiponímia de WordNet (ou qualquer outra hierarquia em geral).

- Em [19] foi apresentado um enfoque para medida de similaridade semântica entre conceitos utilizando uma hierarquia e estatísticas de uma coleção. Este enfoque combina a proposta de contagem de arestas de distância entre dois conceitos, e a proposta de utilização das probabilidades de ocorrência na coleção. Esse enfoque foi empregado em desambiguação de sentido das palavras, e pode ser utilizado também em recuperação de informação.
- Uma proposta mais geral foi apresentada em [24, 1]. Se define a similaridade entre dois conceitos (C_1 e C_2) como a razão entre a quantidade de informação necessária para descrever a parte comum entre C_1 e C_2 , e a quantidade de informação necessária para descrever C_1 e C_2 por completo. Um dos exemplos apresentados para validar a utilidade da medida proposta

baseia-se em uma hierarquia e uma distribuição de probabilidade de ocorrência para cada objeto da hierarquia.

A similaridade entre dois conceitos, por exemplo Ator e Crítico é:

$$Sim(Ator, Critico) = \frac{\log(P(Profissao))}{\log(P(Ator)) + \log(P(Critico))}$$

Onde Profissão é uma super-classe de Ator e de Crítico tal que nenhuma super-classe de ambas é super-classe de Profissão também, e $P(Conceito)$ é a probabilidade de ocorrência do *Conceito*.

Como será detalhado no Capítulo 4, o cálculo de similaridade entre dois conceitos na ontologia utilizado na aplicação é baseado na proposta de [24], pelo fato dela aproveitar a existencia de uma hierarquia (a ontologia) e uma distribuição de probabilidade (a contagem de frequência dos documentos na coleção).

Além do cálculo baseado na hierarquia, a medida de similaridade utilizada neste trabalho, mede a proximidade de dois conceitos por um segundo fator que depende de uma contagem das relações que os unem. Isso permite fazer com que, por exemplo, os conceitos Colômbia e América do Sul (associados pela relação continente descrita anteriormente), tenham um grau de similaridade maior do que teriam só pela hierarquia. A relação que existe entre eles permite incrementar um segundo fator, que será descrito no Capítulo 4.

3.4 Ontologia do domínio

A ontologia desenvolvida contém informação sobre o domínio, e o processo de elaboração dela é detalhado a seguir. Na elaboração de uma ontologia utilizada em recuperação de informação devem-se contemplar os principais conceitos (palavras-chave) utilizados nos documentos da coleção e conceitos que em geral ajudam a distinguir a relevância dos documentos.

Para limitar o mundo da arte a um sistema de conceitos e categorias compreensível pelo computador, o processo de desenvolvimento começou pela modelagem do mundo da arte com termos próprios desse mundo, como: “*instituição*”, “*obra de arte*”, “*local*”, “*texto*” e detalhando cada um desses blocos em sub-classes em uma relação *é-um*. Assim, uma instituição poderia ser um museu, uma galeria, um centro cultural, etc. Após isso e da extração de palavras-chave dos documentos da coleção por parte da especialista do domínio, essas palavras-chave foram analisadas para verificar se elas apareciam como conceitos ou relações na ontologia. Assim, surgiu a necessidade de criar super-classes adicionais tais como “*material*” e “*poder*”. A seguir fez-se uma listagem dos sinônimos para as palavras presentes na ontologia, considerando os termos relevantes, e que poderiam ser utilizados nas consultas do sistema. Os sinônimos foram cadastrados como comentários de cada classe, para facilitar a tarefa do especialista do domínio. Durante a indexação (Ver Seção 4.2.1), esses comentários são transformados em classes equivalentes.

O desenvolvimento dos relacionamentos entre classes e a adição de instâncias ao modelo foram os últimos passos do desenvolvimento dessa primeira versão da ontologia. A especificação de instâncias na ontologia foi um passo importante, pois muitas afirmações feitas durante a entrevista tratam de indivíduos e não das classes propriamente ditas.

Uma parte da hierarquia presente abaixo da classe “*obra de arte*” na ontologia é mostrada na Figura 3.3.

Esse processo de desenvolvimento foi iterativo, e espera-se ter gerado uma ontologia reutilizável no domínio da arte contemporânea que possa ser enriquecida à medida em que mais artistas forem entrevistados. Pela construção do sistema espera-se também que o sistema possa ser utilizado com essas outras entrevistas sem ter um esforço de desenvolvimento adicional muito grande.

Figura 3.3: Parte da hierarquia abaixo da classe “*obra de arte*” na ontologia do domínio.

3.5 Ferramentas para manipulação de ontologias

As principais motivações para o uso de ontologias são a reutilização e a padronização no compartilhamento de informação entre agentes inteligentes. Dado que muitas comunidades utilizam ontologias, existem diversas ferramentas que permitem a elaboração e navegação de ontologias em formatos padrão como OWL e RDF.

3.5.1 Ferramentas para elaboração

Existe uma ampla variedade de ferramentas para elaboração de ontologias, cada uma focada em etapas diferentes do processo de desenvolvimento de ontologias e apresentando vantagens e desvantagens cada uma. Dessas ferramentas, se destacam como ambientes integrados de desenvolvimento [7]: Protegé, OntoEdit e WebODE.

- **Protegé**[13]: Protegé evoluiu de uma pequena aplicação para construção de aplicações de aquisição de conhecimento no domínio médico para uma plataforma extensível de desenvolvimento de ontologias. Protegé é multi-plataforma e permite exportar o resultado em diferentes formatos, como RDF, DAML+OIL, CLIPs e OWL. Existem várias comunidades estendendo esta ferramenta através do seu mecanismo de *plugins*: melhorando a interface com o usuário, suportando a importação em diversos formatos [23], ou permitindo o armazenamento em bancos de dados.

As características principais do Protegé são: escalabilidade e extensibilidade devido à sua arquitetura de plugins, a integração dos serviços que oferece, a facilidade de utilização para os usuários, a disponibilidade do software, a quantidade de formatos dos quais pode importar e/ou exportar ontologias e os motores de inferência que podem ser integrados nele.

- **OntoEdit** [36] OntoEdit é um ambiente que suporta o desenvolvimento e manutenção de ontologias utilizando ferramentas visuais. Está construído sobre um modelo interno que pode-se guardar em XML. Uma característica importante do OntoEdit é que permite trabalho colaborativo.
- **WebODE** [8]: é um ambiente de trabalho integrado que facilita o desenvolvimento de ontologias, o raciocínio sobre ontologias e a importação/exportação com outras ferramentas baseadas em ontologias. Oferece, do mesmo modo que Protegé e OntoEdit, um modelo genérico para a modelagem de ontologias que pode ser exportado a diversos formatos, pois definem as entidades principais para a definição de uma ontologia.

Dessas ferramentas foi selecionada o Protegé, já que se buscava uma ferramenta amigável, em que a especialista do domínio pudesse criar a ontologia diretamente, e Protegé oferece uma maior facilidade para a criação visual de ontologias.

3.5.2 Ferramentas para navegação

Para integrar a ontologia com a aplicação, foi preciso utilizar uma API que permita explorar ontologias em formato OWL e realizar inferências, por exemplo, para determinar se uma classe é sub-classe de outra. Assim, foram avaliadas duas ferramentas recentes que cumprem essas características: Jena e Racer.

- **Jena**⁶: Jena é uma API Java para acessar recursos em RDF, e permite raciocinar sobre ontologias em OWL e outras linguagens. Jena é usado pelo Protegé para a manipulação de ontologias. Possui funcionalidade para ler e escrever RDF e OWL, e diversos motores de inferência para manipular ontologias e que oferecem a possibilidade de trabalhar com os dados na memória para ter melhor desempenho na resolução das consultas.
- **Racer** [16]: Racer é um motor de inferência sobre documentos RDF e OWL (e outros formatos), com funcionalidade para realizar consultas e modificar a estrutura de ontologias. Oferece suporte para lógica de descrições (mecanismo de representação de conhecimento baseado em formalismos lógicos), e realiza diversas validações de consistência em ontologias expressas em formatos OWL e DAML. RACER foi implementado como um programa servidor e oferece uma interface cliente para Java.

Na aplicação foi usado Jena pela sua simplicidade, principalmente o fato de ser uma API instalada no cliente e que não precisa configuração para a distribuição da aplicação final.

3.6 Considerações finais

Uma ontologia permite a representação de informações sobre um domínio para aplicações computacionais, com a vantagem de representar o conhecimento de modo que permita sua reutilização.

⁶<http://www.hp1.hp.com/news/2004/jan-mar/jena2.1.html>.

As ontologias podem ser utilizadas como uma rede de conceitos em RI para obter a similaridade entre termos e recuperar documentos com termos relacionados aos termos utilizados na consulta.

Uma ontologia no domínio da arte contemporânea foi desenvolvida pela especialista do domínio utilizando Protegé. A API Jena foi utilizada para explorar o conteúdo da ontologia tanto para o processo de indexação quanto para o processo de recuperação.

Recuperação de Vídeos: A aplicação

Neste Capítulo, será apresentada a aplicação desenvolvida para recuperação de informação em um banco de dados de vídeos. O problema, assim como a arquitetura do sistema e os processos de indexação e recuperação utilizados serão apresentados a seguir.

4.1 A aplicação

Nessa Seção, será estabelecido o problema que levou à implementação da aplicação de RI. Foram filmadas entrevistas com Ana Teixeira, uma artista brasileira, cujo trabalho está relacionado com intervenções urbanas. A idéia da proposta foi permitir a exibição desses vídeos em espaços tais como museus ou galerias de arte, permitindo ao espectador formular suas próprias consultas sobre os trabalhos em exibição da artista. O sistema permite que o usuário digite consultas em linguagem natural, e deve mostrar os fragmentos da entrevista que mais se aproximam de uma resposta à consulta feita. Além disso, o sistema apresenta funcionalidades adicionais, como por exemplo a possibilidade de apresentar imagens estáticas de trabalhos durante a apresentação do vídeo.

Os espaços de exibição usualmente oferecem aos espectadores somente textos de parede ou catálogos e panfletos que utilizam a linguagem especializada da teoria da arte. Esses textos muitas vezes intimidam o espectador leigo e não o incentivam a desenvolver seu próprio pensamento

e questões sobre as obras de arte. No caso das obras de arte contemporânea, que permitem a participação do espectador na criação da obra, essas exposições requerem um alto grau de interação com o espectador, que pode ser facilitada com o tipo de aplicação implementado.

As entrevistas com Ana Teixeira foram feitas por Paula P. Braga, conhecedora do domínio da arte contemporânea, e principal desenvolvedora da ontologia. As entrevistas foram gravadas em formato digital e serviram como base para o desenvolvimento da primeira versão da ontologia de arte contemporânea usada na aplicação.

A aplicação foi projetada para permitir sua utilização com outras entrevistas e em outros domínios. Conseguiu-se isto elaborando um modelo genérico em que os trechos dos vídeos são independentes da ontologia, e a ontologia seja uma componente externa manipulada por um mecanismo também genérico. Assim, o processo de indexação (Seção 4.2.1) recebe essas duas entradas e gera as estruturas requeridas, independentemente do domínio em que elas estejam.

4.2 Descrição do sistema

O sistema foi desenvolvido utilizando Java, devido à portabilidade entre plataformas que os sistemas desenvolvidos em Java oferecem, além da disponibilidade de ferramentas para a manipulação de ontologias e vídeos requeridos para a aplicação. A seguir apresentaremos a estrutura do sistema, a qual, sendo um sistema de RI e como foi apresentado no Capítulo 2, pode ser dividida em dois processos principais: Indexação e Recuperação.

4.2.1 Processo de Indexação

O processo de indexação permite preparar as estruturas requeridas para as buscas a serem feitas no processo de recuperação. O processo de indexação implementado recebe quatro entradas principais: a coleção de vídeos, os conceitos associados aos vídeos, os recursos adicionais e a ontologia do

domínio.

a. Coleção

A coleção de vídeos da entrevista em formato digital constitui a entrada principal do sistema. No caso da aplicação implementada, os vídeos se encontram em formato MPEG. Esses vídeos devem estar fragmentados e cada fragmento deve conter uma fala, e na implementação atual esses fragmentos são os apresentados como resposta às consultas do usuário. Pretende-se em um trabalho futuro evitar a necessidade da fragmentação manual dos vídeos, permitindo a manipulação dos fragmentos indicando os períodos de tempo associados a cada um deles.

Além dos vídeos, é requerida uma transcrição das falas apresentadas nos vídeos, em formato texto, sobre a qual será feita a recuperação. Essa transcrição foi feita manualmente, mas esse é um processo que poderia ser automatizado em um trabalho futuro também, utilizando as ferramentas de reconhecimento de fala adaptadas para a língua em que o vídeo está, o português do Brasil.

b. Conceitos associados aos documentos da coleção

Uma lista de palavras-chave pode ser manualmente associada a cada fragmento de vídeo. Essas palavras-chave deveriam estar associadas a conceitos na ontologia do domínio (que faz parte do processo de indexação também). As palavras-chave podem ser usadas no processo de recuperação no lugar da transcrição, ou junto com ela, como será visto na Seção 4.2.2.

c. Recursos associados

Uma lista de recursos (imagens) que podem ser associadas a fragmentos de vídeos para serem mostradas durante a sua exibição ao usuário como uma informação adicional ao que está sendo falado no vídeo. Por exemplo, quando a artista fala de uma obra dela, a imagem do vídeo é trocada por fotos da obra, ilustrando assim o que está sendo falado.

d. Ontologia do domínio

É uma ontologia em formato OWL, contemplando os principais conceitos do domínio em que a entrevista é desenvolvida. A idéia foi fazer essa ontologia legível para a especialista que a

desenvolveu, portanto algumas simplificações foram permitidas. Por exemplo, os sinônimos dos conceitos foram escritos como comentários RDF dentro da ontologia.

Essa ontologia é processada durante a indexação, para permitir fazer as inferências necessárias na aplicação. A ontologia foi descrita na Seção 3.4.

O cadastro de todos os itens apresentados como entrada ao processo de indexação é feito através de uma aplicação de configuração, cujo principal objetivo é facilitar a geração das saídas do processo de indexação. A saída do processo de indexação consiste de três elementos: um arquivo XML de configuração, a ontologia processada e os arquivos de índice invertido.

a. Arquivo XML de configuração

Esse arquivo XML apresenta as informações gerais da aplicação final de consulta, principalmente os parâmetros a serem utilizados. Por exemplo, os nomes de arquivos de saída gerados.

Para cada vídeo, ele armazena o nome de arquivo, um nome descritivo (a ser mostrado para o usuário), a duração do vídeo e o nome do arquivo de texto com a transcrição da fala do vídeo. Para cada recurso, o arquivo contém o nome de arquivo, um nome descritivo e uma legenda a ser mostrada na sua exibição.

Além disso, armazena tanto as associações entre os vídeos e as palavras-chave representativas dos mesmos, quanto as associações dos vídeos com os recursos, incluindo o instante de início da apresentação do recurso e o tempo que o recurso será mostrado.

Um esquema do arquivo XML é apresentado na Figura 4.1.

b. Ontologia processada

Uma nova ontologia é criada baseada na original para conseguir fazer as inferências necessárias para a aplicação. Nos comentários de classe na ontologia original é especificada uma lista de sinônimos. Em caso do sinônimo ainda não existir na ontologia processada, uma classe nova é criada para ele. A classe que representa o sinônimo é adicionada como equivalente.

Figura 4.1: Esquema de arquivo XML de configuração.

Além disso, para cada instância de uma classe C qualquer da ontologia, cria-se uma classe com o mesmo nome, como sub-classe de C . Assim, na ontologia processada existe uma classe Colômbia, sub-classe da classe País. Para cada propriedade da instância, uma propriedade de classe é criada também. Com isso, consegue-se realizar inferências sobre a hierarquia sem precisar de condições especiais para o processamento de instâncias na hora da recuperação.

Todas essas alterações na ontologia são feitas para minimizar o tempo de processamento do cálculo de similaridade durante o processo de recuperação. Todo esse processo é realizado utilizando a API Jena.

c. Arquivos de índice invertido

Um arquivo de índice invertido é gerado com base no texto completo das transcrições ignorando as palavras de parada. O índice contém as raízes das palavras, obtidas através do processo de remoção de afixos, apresentado na Seção 2.2.1. O algoritmo de criação do índice é apresentado na Figura 4.2.

O algoritmo recebe como entrada a coleção de documentos, um algoritmo de remoção de afixos e a ontologia do domínio. Para cada um dos documentos se adiciona uma entrada ao índice, na qual manterá-se a contagem de frequência dos termos por cada documento.

Para lidar com termos compostos, utiliza-se um esquema simples de validação. Para cada

termo (*token*), do texto transcrito, associado ao documento, uma validação é feita para comprovar se ele junto com os próximos termos no texto geram um termo da ontologia. Por exemplo, os nomes de pessoas podem ser identificados assim. No caso de um termo ser composto, ele é tratado sem alterações, caso contrário, o algoritmo de remoção de afixos é aplicado no termo. Se o termo já validado não conta com uma entrada no índice, uma é criada para ele e um contador de ocorrências dentro da entrada é incrementado.

Em resumo, para cada entrada de documento é armazenada uma contagem dos termos que aparecem nele, e para cada termo é armazenada uma lista dos documentos em que ele aparece, sendo este último o índice invertido.

Também é criado um arquivo de índice invertido baseado somente nas palavras-chave, construído do mesmo modo que o índice das transcrições.

4.2.2 Processo de Recuperação

Para o processo de recuperação foi criada uma aplicação visual que permite a interação dos espectadores do museu. Nela o usuário pode entrar com sua consulta, obter uma listagem dos vídeos que mais aproximam a resposta à sua consulta e visualizá-los. O processo de recuperação tem a estrutura, mostrada na Figura 4.3, que será detalhado no restante desta seção.

Pré-processador

Um processo inicial de detecção de palavras com erros ortográficos é aplicado na consulta utilizando a API Jazzy¹. Esta API requer de um dicionário para oferecer as possíveis correções. Foram utilizados um dicionário do português do Brasil² e um dicionário específico do domínio gerado de forma automática em base aos termos da ontologia. A saída do processo é a lista de palavras que

¹Jazzy é uma API Java para correção de erros ortográficos, disponível em: <http://sourceforge.net/projects/jazzy>.

²Disponível em <http://www.ime.usp.br/~ueda/br.ispell/>.

Algoritmo CriaIndice(*colecão*, *stemmer*, *ontologia*)

1. **Para todos os documentos** *doc* **da** *colecão*

(a) *entrada_doc* = *indice.adicionaEntradaDoc(doc.id)*;

(b) **Para todos os tokens** *tok* **do** *doc.textocompleto*

i. **Se** *tok* **com o(s) seguinte(s) token(s) do texto geram um token** *tok_ont*
da ontologia

A. *tok* = *tok_ont*;

ii. **Senão**

A. *tok* = *stemmer.removeafixos(tok)*;

iii. *entrada_doc.contaOcorrencia(doc, tok)*;

iv. **Se** *tok* **ainda não está em** *indice.entradasTermos*

A. *indice.adicionaEntradaTermo(tok)*;

v. *entrada_termo = indice.pegaEntradaTermo(tok)*;

vi. *entrada_termo.contaOcorrencia(doc, tok)*;

2. **Retorna** *indice*;

Figura 4.2: Algoritmo de geração do arquivo de índice invertido.

Figura 4.3: Processo de recuperação.

possam ter sido escritas de forma incorreta, e sugestões de correção para as mesmas. A sugestão é feita só na apresentação dos resultados recuperados da consulta utilizando a consulta original. O processo de remoção de afixos descrito na Seção 2.2.1 é aplicado nessa etapa sobre a consulta, sendo removidas antes disso as palavras de parada presentes na consulta.

Nessa etapa também são associados os pesos aos termos da consulta no vetor de consulta como descrito na Seção 2.3.2. Uma contagem das aparições das palavras é feita, considerando como um termo cada palavra, à exceção daqueles termos contíguos que estão presentes na ontologia como termos compostos. Além disso, a ontologia é utilizada para detectar os principais tópicos da consulta. As palavras da consulta presentes na ontologia recebem maior peso (0,5 adicional à contagem de aparições) pois elas tem muita relação com o que foi falado na entrevista. Por exemplo, na consulta: *“Por que você expõe sua obra se ela acontece mesmo na rua?”*, as palavras *“expor”* (na forma infinitiva expor), *“obra”*, *“acontece”* (na forma de acontecimento), e *“rua”* fazem parte da ontologia e são muito representativas do que a consulta quer recuperar. Além disso, o peso da palavra na consulta com menor número de aparições na coleção é incrementado em 0,5 também. No caso do exemplo, a palavra *“rua”* é a que menos aparece na coleção, por tanto ela recebe o peso maior.

Expansão da consulta

O processo de expansão da consulta consiste na atualização de pesos dos termos na consulta. Pode-se incrementar os pesos iniciais atribuídos na etapa anterior ou, atribuir pesos novos aos termos do índice mais relacionadas à consulta.

O algoritmo de expansão da consulta é apresentado na Figura 4.4. A similaridade de cada termo no índice representa a similaridade entre o termo e a consulta:

$$sim(t, q) = \sum_{t_i \in q} w_{i,q} \times sim(t_i, t) \quad (4.1)$$

onde $w_{i,q}$ é o peso do termo t_i na consulta q e $sim(t_i, t)$ é a similaridade entre o termo t_i e o termo t . O peso a ser expandido é uma média ponderada da similaridade do termo com cada um dos termos da consulta original:

$$w_{ex}(t, q) = \frac{sim(t, q)}{\sum_{t_i \in q} w_{i,q}} \quad (4.2)$$

Os pesos expandidos dos *max_exp* termos com maior similaridade são adicionados aos pesos originais da consulta gerando os pesos da consulta expandida.

A função **SimTermos** da linha 3.b.1. da Figura 4.4 é baseada na medida de similaridade proposta em [24], e que foi apresentada na Seção 3.3.2. O algoritmo associado é apresentado na Figura 4.5 e recebe dois termos, a ontologia e o índice, devolvendo como saída um valor entre 0 e 1 que indica o grau de similaridade entre os termos.

O algoritmo inicialmente pega as classes da ontologia associadas ao termo. O termo pode ter mais de uma classe associada pela remoção de afixos. Por exemplo, o termo “contemporân” está associado à classe Contemporânea especificada como instância da classe Época, e à classe Contemporâneo como sinônimo da classe Não Convencional sub-classe da classe Material.

Algoritmo Expansão.Consulta(*consulta*, *indice*, *ontologia*)

1. $peso_consulta = 0$;
2. **Para todos os termos** u **da** *consulta*
 - (a) $peso_consulta += consulta(u).peso$;
3. **Para todos os termos** t **do** *indice*
 - (a) $sim(t) = 0$;
 - (b) **Para todos os termos** u **da** *consulta*
 - i. $sim(t) += consulta(u).peso \times \mathbf{SimTermos}(t, u, ontologia, indice)$;
 - (c) $w_{ex}(t).peso = sim(t) / peso_consulta$;
4. **Para todos os termos** t **da** *consulta*
 - (a) **Se** $w_{ex}(t)$ **é** um dos *max_exp* termos com maior peso
 - i. $consulta_expandida(t) = consulta(t)$;
 - ii. $consulta_expandida(t).peso += w_{ex}(t).peso$;
5. **Retorna** *consulta*;

Figura 4.4: Algoritmo de expansão de consulta.

Algoritmo SimTermos(*term1*, *term2*, *onto*, *indice*)

1. **Se** *term1* == *term2* **Retorna** 1.0;
2. *classes1* = *onto*.pegaClasses(*term1*), *classes2* = *onto*.pegaClasses(*term2*);
3. **Se** *classes1* é vazio ou *classes2* é vazio **Retorna** 0.0;
4. **Se** *classes1* $\cap$ *classes2* não é vazio **Retorna** 1.0;
5. *classes_sup* = *onto*.pegaClassesSuperioresComuns(*classes1*, *classes2*);
6. **Se** *classes_sup* não é vazio
 - (a) $probterm1 = \frac{indice.pegaFrequencia(classes1)}{indice.totalDocs}$;
 - (b) $probterm2 = \frac{indice.pegaFrequencia(classes2)}{indice.totalDocs}$;
 - (c) $probtermsup = \frac{indice.pegaFrequencia(classes_sup)}{indice.totalDocs}$;
 - (d) **Se** $\log(probterm1 \times probterm2) = 0$
 - i. $atermo = \frac{2 \times \log(probtermsup)}{\log(probterm1 \times probterm2)}$;
7. (*totalProps1*, *propsRelacionadas1*) = *onto*.contarProps(*classes1*, *classes2*);
8. (*totalProps2*, *propsRelacionadas2*) = *onto*.contarProps(*classes2*, *classes1*);
9. **Se** *totalProps1* + *totalProps2* > 0
 - (a) $btermo = \frac{propsRelacionadas1 + propsRelacionadas2}{totalProps1 + totalProps2}$;
10. **Retorna** ($\beta \times atermo$) + ((1 - β) $\times btermo$);

Figura 4.5: Algoritmo de cálculo de similaridade entre dois termos.

Se algum dos significados dos termos concordam, i.e. são sinônimos, o algoritmo devolve 1. Em caso de algum dos dois termos não terem significados presentes na ontologia, devolve 0, pois não se conta com critérios para estabelecer uma similaridade entre os termos.

A seguir o algoritmo calcula dois fatores de similaridade, o primeiro baseado na hierarquia de conceitos e o segundo nos relacionamentos estabelecidos na ontologia como pode-se observar na Equação 4.3. O primeiro fator contém o cálculo da medida proposta em [24], onde *classes_sup* é o conjunto das classes que tem os significados de *term1* e *term2* como sub-classes na ontologia.

O segundo fator mede o grau de relacionamento mediante uma contagem das propriedades que cada uma das classes associadas aos termos tem, e quantas dessas propriedades relacionam uma classe com a outra, $props(t_1, t_2)$ é o numero de propriedades que tem t_1 como domínio e t_2 como imagem, e $props(t_1)$ é o número de propriedades que tem t_1 como domínio. A razão da quantidade de propriedades que relacionam as classes e a quantidade total de propriedades de ambas classes é o segundo fator. A média ponderada dos fatores por um parâmetro β é a medida de similaridade dos termos.

$$sim(t_1, t_2) = \beta \times \frac{2 \times \log P(sup)}{\log P(t_1) + \log P(t_2)} + (1 - \beta) \times \frac{props(t_1, t_2) + props(t_2, t_1)}{props(t_1) + props(t_2)} \quad (4.3)$$

no caso de ambos os termos t_1 e t_2 estarem presentes na ontologia, e

$$sim(t_1, t_2) = 0 \quad (4.4)$$

caso contrário.

Todo o cálculo de similaridade é feito utilizando o motor de inferência Jena, cuja funcionalidade utilizada desde a aplicação foi encapsulada utilizando o padrão *Facade*³ para facilitar a utilização de outros motores, como pode ser visto na Seção 5.2.

³ O Padrão *Facade* provê uma interface unificada para um conjunto de interfaces em um subsistema. *Facade* define uma interface de alto nível que torna o sistema mais fácil de usar [12].

Recuperação

Após a expansão da consulta, a consulta expandida é utilizada no processo de recuperação baseado no modelo estendido de espaço de vetores apresentado na Seção 2.3.2. O algoritmo de recuperação é mostrado na Figura 4.6. Ele recebe uma consulta e um índice, e devolve uma lista de documentos com um valor de similaridade associado a cada um deles. Esse algoritmo é chamado uma vez para cada índice utilizado (texto completo e palavras-chave), e os resultados obtidos em cada chamada são somados, e ponderados com um peso α , que é parâmetro do sistema.

A recuperação calcula o cosseno do ângulo que formam o vetor que representa a consulta e o vetor que representa cada documento.

$$cos_{doc,q} = \sum_{t_i \in doc} w_{i,q} \times w_{i,doc} \quad (4.5)$$

Para a medida de similaridade levar em consideração o tamanho da consulta e do documento, o cálculo é:

$$sim_{doc,q} = \frac{cos_{doc,q}}{\sqrt{\sum_{t_i \in doc} w_{i,doc} \times \sum_{t_j \in q} w_{j,q}}} \quad (4.6)$$

Com isso uma lista de documentos relacionados à consulta é criado, e os max_doc documentos mais similares à consulta são mostrados para o usuário. Este modelo é conhecido como modelo de espaço de vetores [34].

Reprodutor de vídeo

A aplicação mostra os resultados da consulta como uma lista dos vídeos que foram recuperados ordenados pela similaridade que eles apresentam com a consulta. Uma das componentes do sistema

Algoritmo Recupera(*consulta*, *indice*)

1. $peso_total_consulta = 0$;
2. **Para todos os termos** *termo* **na** *consulta*
 - (a) $entrada = indice.pegarEntradaTermo(termo)$;
 - (b) **Se** *entrada* **é nula** **processar seguinte termo**
 - (c) $peso_consulta = (consulta(termo).peso)^2$;
 - (d) $peso_total_consulta += peso_consulta$;
 - (e) **Para todos os documentos** *doc* **na** *entrada*
 - i. $docs_analizados(doc).cos += peso_consulta \times entrada(doc).peso$;
3. **Para todos os documentos analisados** *doc* **em** *docs_analisados*
 - (a) $entrada = indice.pegarEntradaDoc(doc)$;
 - (b) **Para todas os termos** *termo* **em** *entrada*
 - i. $docs_analizados(doc).peso_doc += (entrada(termo).peso)^2$;
 - (c) $docs_analizados(doc).sim = \frac{docs_analizados(doc).COS}{\sqrt{docs_analizados(doc).peso_doc \times peso_total_consulta}}$;
4. **Retorna** *docs_analisados*;

Figura 4.6: Algoritmo de recuperação utilizando o modelo de espaço de vetores.

de recuperação é o reprodutor de vídeo implementado utilizando o JMF⁴ para exibir os vídeos e dar funcionalidades de reproduzir, pausar e interromper o vídeo. Esta componente também permite navegar entre os vídeos recuperados e mostrar as imagens associadas aos vídeos nos tempos correspondentes cadastrados na interface administrativa.

4.3 Considerações finais

Nesse capítulo, foi apresentada a aplicação desenvolvida para a recuperação de vídeos indexados por conceitos. A aplicação utiliza um conjunto de entrevistas com a artista brasileira Ana Teixeira e é uma proposta para melhorar a interação entre o visitante de espaços de exibição e as obras do artista, permitindo ao visitante fazer consultas em linguagem natural, sendo apresentados os vídeos que respondam a consulta feita pelo usuário.

Para o desenvolvimento do sistema foi utilizada uma ontologia, para a expansão da consulta do usuário. Essa expansão usa a informação de hierarquia e dos relacionamentos da ontologia. O processo de recuperação de informação foi feito utilizando o modelo estendido de espaço de vetores. O resultado do processo de recuperação é uma lista ordenada dos vídeos que satisfazem a consulta do usuário e que podem ser visualizados através de uma interface desenvolvida utilizando JMF.

⁴Java Media Framework. Ver <http://java.sun.com/products/java-media/jmf/>.

Resultados

Neste capítulo, será apresentado o funcionamento do sistema, através de uma execução do mesmo, detalhando os passos realizados para obter a resposta. A seguir, descreve-se a estrutura de dados utilizada para melhorar o desempenho do mecanismo de expansão de consulta.

São discutidos alguns aspectos do processo de melhora da ontologia e das palavras-chave associadas aos clips. O processo de elaboração e execução dos testes do sistema é descrito. Foi elaborada uma bateria de testes junto com a especialista do domínio, e os testes foram executados e avaliados. São mostrados os resultados em termos de tempo de processamento, precisão e cobertura, justificando a substituição do motor de inferência por uma estrutura de dados mais simples.

5.1 Execução do sistema

O sistema descrito no capítulo 4, permite ao usuário realizar consultas e apresenta como resposta para ele uma listagem dos vídeos que melhor satisfazem a consulta. A seguir, apresentamos a execução do sistema para uma consulta selecionada da bateria de testes que será apresentada na Seção 5.4.

A consulta utilizada para a demonstração foi: “*Como você vende sua obra?*”, mas foi apresentada como: “*Como você vende sua obr?*”, possuindo dois erros ortográficos. O sistema tenta

recuperar os documentos com a consulta original, mas não consegue achar nenhum. Além disso, propõe para o usuário as alternativas de correção para as palavras da consulta que não encontra nos seus dicionários (Ver Seção 4.2.2). No caso, propõe: “*bonde*”, “*pende*”, “*tende*”, “*rende*”, “*benze*”, “*fende*” e “*vende*” para “*bende*” e, “*obra*” e “*oba*” para “*obr*”. Dessas propostas o usuário pode-se guiar para refazer a consulta. A Figura 5.1 mostra uma tela do sistema, com os resultados de execução da consulta com erros ortográficos e a apresentação das alternativas de correção.

Figura 5.1: Tela do sistema, apresentando resultados da consulta original e as alternativas de correção ortográfica.

Com a consulta reformulada pelo usuário: “*Como você vende sua obra?*”, o sistema executa o processo de expansão de consulta e o de recuperação. No exemplo não utilizaremos o índice gerado

pelas transcrições dos vídeos, só o gerado pelas palavras-chave .

Da consulta, o sistema identifica duas palavras relevantes: “vende” e “obra”, ficando após a remoção de afixos os termos “vend” e “obr”. Devido à existência da classe Venda na ontologia, e a que ele aparece com menor frequência na coleção, o termo “vend” obtém peso 2 na consulta. Para todos os termos do índice, é calculada a similaridade deles com a consulta. Devido ao peso do termo “vend”, os termos com maior similaridade com a consulta, são os associados com ele: {“mercado”: 0,667, “barraca”: 0,667, “camelô”: 0,667, “galeria”: 0,667} e que passam a formar parte da consulta expandida. Todos esses termos estão relacionados com “venda” na ontologia.

O clip: Processo - encomendas, possui como palavras-chave “mercado” e “venda”, com o qual a similaridade dele com a consulta é alta. Já os outros clips só tem como palavras-chave um dos termos da consulta expandida, ou dois, mas sem incluir “vend” que possui o maior peso na consulta. Os documentos com maior similaridade são recuperados.

Finalmente, a lista de vídeos ordenadas por similaridade com a consulta expandida é apresentada no sistema como é mostrado na Figura 5.2. O usuário pode selecionar qualquer um dos vídeos e assisti-lo ou assistir todos de forma seqüencial.

5.2 Mudança no motor de inferência

Como foi descrito na Seção 4.2.2, o motor de inferência Jena foi utilizado para a expansão da consulta utilizando os termos similares na ontologia, durante o processo de recuperação. Jena oferece diversas configurações otimizadas para distintos tipos de aplicação, delas foram avaliadas três:

- OWL_MEM: Oferece um processamento de consulta simples, só consulta a hierarquia especificada, não infere relações transitivas por exemplo.
- OWL_MEM_TRANS_INF: Oferece um mecanismo de inferência que explora as relações de

Figura 5.2: Tela de apresentação dos clips recuperados pelo sistema.

sub-classe e super-classe de modo transitivo. Assim, consegue responder consultas do tipo A é sub-classe indireta de B . Mesmo assim, ele não infere relações transitivas que incluem equivalência de classes. Por exemplo, se A é equivalente a B , e C é sub-classe de A , à consulta para saber se C é sub-classe de B , o motor responde que não.

- **OWL_MEM_RULE_INF:** Mecanismo de inferência mais complexo que explora as relações transitivas incluindo classes equivalentes, solucionando o problema apresentado como falha da configuração anterior. O problema desse motor, é que ocupa muita memória e em muitas consultas simples ele ocasiona uma exceção de memória insuficiente na máquina virtual. Além disso, as inferências são consideravelmente mais lentas do que com o motor transitivo.

Dado o problema do motor `OWL_MEM_RULE_INF`, se optou por utilizar o `OWL_MEM_TRANS_INF` mas, para conseguir realizar as inferências sobre as classes equivalentes, forçou-se que cada classe *A* equivalente de *B* tivesse as mesmas sub-classes e super-classes, não precisando de inferência nesse sentido. Esse processo de cópia da hierarquia é executado durante o pré-processamento da ontologia na indexação (4.2.1).

Mesmo assim, o Jena trabalha sobre um modelo complexo que inclui, além das restrições básicas que a aplicação utiliza, classes de união, intersecção, herança de propriedades, e outros. Isso faz com que as consultas sejam lentas, portanto decidiu-se implementar uma estrutura simplificada que permitisse fazer as mesmas inferências de modo mais rápido.

A estrutura básica é apresentada na Figura 5.3. Consta de duas listas, a primeira é uma lista simples das classes ou conceitos da ontologia. Cada classe possui os seguintes atributos:

- um nome, o original na ontologia,
- uma lista dos termos equivalentes ou sinônimos,
- a frequência do termo (e dos termos equivalentes) na coleção,
- a somatória das frequências na coleção de todos os termos na hierarquia abaixo da classe,
- um conjunto das suas sub-classes diretas,
- um conjunto de todas as suas sub-classes, diretas e indiretas, junto com a distância à qual elas se encontram,
- uma lista das classes com a qual estão diretamente relacionadas na ontologia, contendo além disso o número de propriedades que as enlaçam (Não apresentado na Figura).

Dado que a relação de especialização é transitiva, o conjunto de sub-classes indiretas contém todas as classes que estão relacionadas à classe original através das suas sub-classes diretas. A distância entre uma classe e sua sub-classe indireta é o número de classes no menor caminho que as

Figura 5.3: Estrutura de dados simplificada para as inferências.

une. Essa distância é útil, pois dadas as equivalências de classes, aparecem relações indiretas que não refletem o que é expresso na ontologia. Por exemplo, a classe `Material` tem como sinônimo `Processo`, que também é uma classe, sub-classe da classe `Evento`. Desse modo, as sub-classes de `Material`, como `maça`, são vinculadas à classe `Evento`, sendo que essa informação não se reflete na ontologia original. Para evitar isso, um limite foi estabelecido para considerar como sub-classe só aquelas que possuem uma distância menor do que um limite especificado ¹.

Uma otimização adicional que se conseguiu com essa estrutura, foi a possibilidade de armazenar as frequências dos termos na própria estrutura, evitando a utilização do índice durante o processo de expansão da consulta.

A segunda lista, é um mapa que consta de pares (*termo, lista de equivalências*), em que para cada termo se mantém uma lista dos termos (como cadeias de caracteres) e classes equivalentes ao termo. Com isso consegue-se acesso em tempo constante em média² à informação para cada termo, o qual permite melhorar consideravelmente o desempenho da expansão.

Esta estrutura é gerada durante o processo de indexação, e é portanto executada uma vez só. Desse modo, é necessário executar o processo de indexação (geração de índice na interface administrativa) sempre que a ontologia ou a coleção (incremento de vídeos, recursos ou palavras chaves) sejam alteradas.

Como será apresentado na Seção 5.4, a substituição do Jena por um mecanismo simples e adaptado ao problema permitiu a melhora no tempo de resposta. A inclusão desta modificação necessitou pouco esforço, pois além da utilização do padrão *Facade* (Ver Seção 4.2.2), utilizou-se o padrão de projeto *Template Method*³ para o cálculo de similaridade, pois só alguns dos passos do algoritmo são dependentes do motor.

¹Na aplicação utilizou-se 5 por ser a maior distância na hierarquia original.

²A estrutura `HashMap` de Java, garante esse desempenho.

³*Template Method* define o esqueleto de um algoritmo em uma operação, delegando a implementação para as sub-classes. Este padrão permite que as sub-classes redefinam certos passos de um algoritmo sem modificar a estrutura do mesmo [12].

Mesmo assim, a implementação utilizando um motor de inferência sobre a ontologia em um formato padrão, é importante pois esses motores, no caso o Jena, são constantemente otimizados e garantem compatibilidade com versões anteriores.

5.3 Aspectos relevantes no processo de testes do sistema

Como foi descrito no Capítulo 3, uma ontologia foi desenvolvida inicialmente para o domínio de arte contemporânea, com ênfase nos assuntos tratados na entrevista utilizada na aplicação. Do mesmo modo, foram associadas palavras-chave a cada clip, segundo a importância que têm nele.

Essa ontologia e as palavras-chave foram as entradas para um processo iterativo de refinação, executada com a especialista do domínio. Nesse processo, foram percebidas algumas características recorrentes.

Deve-se associar palavras-chave específicas aos clips. O uso de palavras genéricas faz com que a precisão do sistema decresça. Exemplos de palavras-chave genéricas são: “*trabalho*” ou “*evento*”, que correspondem as classes mais gerais na ontologia.

Existiu a necessidade de incluir na ontologia alguns termos não pertencentes ao domínio da aplicação. Um exemplo específico, foi o das críticas, requereu-se incluir termos como “*adorável*”, “*detestável*” e “*medíocre*” como possíveis resultados de crítica, pois o sistema não conseguia associar esses conceitos. Assim, se percebeu a necessidade de uma ontologia geral do português, como o Wordnet, para evitar colocar alguns conceitos pouco específicos do domínio na ontologia.

5.4 Testes

Um conjunto de 46 clips foram cadastrados na interface administrativa, os quais somam aproximadamente uma hora e meia de entrevistas filmadas com Ana Teixeira. Em termos de RI, a coleção de documentos é bastante pequena.

Os testes foram executados para o sistema em diversas configurações, como é mostrado na Tabela 5.1. Essas configurações, variam o fato da consulta ser ou não expandida, i.e. se inclui termos adicionais na consulta ou não, o motor de inferência sendo usado para essa expansão, e o peso que associa a cada índice, o correspondente às palavras-chave e às transcrições.

Mnemônico	Expansão	Motor	$w_{palavras-chave}$	$w_{Transcr}$
noexpand	Não	-	1, 0	0, 0
ed-palch	Sim	Modif.	1, 0	0, 0
jena-palch	Sim	Jena	1, 0	0, 0
ed-ambos-05	Sim	Modif.	0, 5	0, 5
jena-ambos-05	Sim	Jena	0, 5	0, 5

Tabela 5.1: Configurações de execução dos testes.

Foi feito um conjunto de testes propostos pela especialista do domínio, constando de 50 consultas em linguagem natural (Ver Apêndice A). Os testes foram elaborados levando em consideração a perspectiva do possível usuário do sistema: o visitante do museu de arte na exposição de Ana Teixeira.

5.4.1 Testes globais

As 50 consultas foram executadas sob todas as configurações, e foram medidas as variáveis: Tempo de Processamento, Precisão, Cobertura e Medida-F, para elas. Foram fixados dois limites neste conjunto de testes: (1) mínima similaridade aceita para considerar um documento relevante, fixado

em 0,1 e; (2) máximo número de documentos a serem recuperados, fixado em 5. Esses valores foram escolhidos experimentalmente. Os resultados são apresentados a seguir.

Na Figura 5.4 pode-se observar os tempos de processamento em todas as configurações de execução. Ambas configurações que utilizam Jena como motor de inferência apresentam um tempo consideravelmente maior do que os que utilizam o motor modificado. A configuração que não expande a consulta ingressada pelo usuário apresentou um tempo bem próximo a zero para todas as consultas. Vale a pena ressaltar que a utilização do índice das transcrições incrementa o tempo de processamento em ambas configurações de motores de inferência.

As Tabelas B.1, B.2 e B.3 no Apêndice B, mostram os resultados obtidos de Precisão, Cobertura e Medida-F respectivamente, para as 50 consultas da bateria de testes.

Os documentos recuperados pelo sistema com os dois motores de inferência, o Jena e o modificado, variam por causa do limite estabelecido para considerar uma relação de hierarquia válida.

Existem seis consultas nas quais os resultados obtidos por `noexpand` têm maior precisão que as outras configurações. Uma medida de precisão 0, indica que a configuração não recuperou documentos, enquanto a precisão marcada com “-”, indica que houve documentos recuperados mas nenhum foi relevante. Em quinze das consultas, nenhuma das configurações obteve documentos relevantes, i.e. as consultas utilizam palavras que não permitem a detecção (nem com o uso da ontologia) da relevância dos documentos. Pode-se ressaltar que dessas consultas, duas (30 e 31), não têm respostas relevantes na coleção, com o que o resultado obtido nas cinco configurações foi o correto. Na maioria das consultas que puderam ser resolvidas, as configurações com expansão de consulta obtiveram precisão maior ou igual do que `noexpand`.

Analisando a Tabela B.2, pode-se observar que a cobertura das configurações com expansão é bem maior do que a da `noexpand`, e ela só apresenta maior valor em três consultas (3, 4 e 21), em duas das quais a expansão da consulta não conseguiu trazer nenhum documento relevante. Pode-se observar também que na maioria dos casos em que há diferença entre a cobertura de `ed-palch` (`jena-palch`) e `ed-ambos-05` (`jena-ambos-05`), a última tem uma maior cobertura, o que

Figura 5.4: Tempo de Processamento para as consultas de teste.

implica que a utilização do índice baseado nas transcrições contribui na melhora da cobertura do sistema.

A Medida-F permite analisar o desempenho do sistema em termos de precisão e cobertura ao mesmo tempo, e pode ser vista na Tabela B.3. Pode-se observar que das oito consultas nas quais noexpand obteve maior precisão, só seis tem maior Medida-F, pois a cobertura apresentada por essa configuração foi menor. Em média, em termos de Medida-F, as configurações jena-ambos-05 e ed-ambos-05 apresentaram melhor desempenho das cinco configurações.

Em termos de Medida-F os valores obtidos são baixos para um sistema de recuperação de informação. Isso é devido fundamentalmente ao fato do tamanho da coleção ser bastante reduzido. O conjunto de vídeos utilizado no teste ainda é pequeno. Isso faz com que um único documento irrelevante recuperado ou, um único documento relevante não recuperado, afete consideravelmente a eficiência do sistema.

Em vista da variabilidade dos resultados em função das consultas, se decidiu fazer uma análise detalhada de algumas consultas, que será apresentada na próxima Seção.

5.4.2 Testes por consulta

Nesta Seção apresentaremos a análise detalhada para quatro das cinquenta consultas utilizadas na bateria de testes. As consultas selecionadas foram:

1. Não entendi seu trabalho. Explique (Consulta 9)
2. Como você vende sua obra? (Consulta 13)
3. Por que você não dá então alguma coisa útil ao povo que está passando necessidade? (Consulta 25)
4. E o povo que participou da sua obra não ganha nada? (Consulta 44)

As consultas foram selecionadas considerando duas em que nenhuma configuração recuperou documentos relevantes (9 e 25), uma em que a não expansão da consulta funcionou melhor (44), e uma (13) em que *ed-palch* trouxe melhor precisão, mas menor cobertura do que a configuração *ed-ambos-05*. Por motivos didáticos, só serão utilizadas as configurações sem expansão e as que utilizam o motor de inferência modificado.

Foram feitas as curvas de precisão e cobertura proposta na análise de média de precisão de n pontos, e a correlação de número de documentos recuperados com a Medida-F.

A média de precisão de n pontos é obtida pela execução de um sistema de recuperação trazendo uma quantidade diferente de documentos relevantes cada vez⁴, e calculando a cobertura e precisão para cada quantidade de documentos até chegar em uma cobertura de 100%. Com n desses pontos, desenha-se uma curva para obter a média de precisão de n pontos. No nosso caso, devido ao tamanho da coleção, o número de pontos de cobertura obtidos é reduzido.

Nas Figuras 5.5 e 5.6 são apresentados os resultados para as consultas 9 e 25 respectivamente. Em ambos, sob a execução parametrizada apresentada na Seção anterior, nenhuma das configurações recuperou documentos relevantes. A configuração *noexpand* não trouxe nenhum documento, por isso não apresenta pontos nos gráficos. Isso, é devido a dois fatores:

- à presença de palavras muito genéricas, como “*trabalho*” no caso da consulta 9 e;
- à presença de palavras irrelevantes (mesmo desconsiderando as palavras de parada), como “*alguma*” e “*coisa*”, e a que as palavras existentes são muito genéricas como no caso anterior.

Para comprovar isso, foi re-elaborada a consulta 25 como “*Por que você não dá algo para as pessoas que necessitam?*”, sem a palavra “*coisa*”, e com uma menor quantidade de palavras no total. Com isso, como pode-se observar na Figura 5.7(a), se consegue uma maior cobertura, mesmo

⁴Isto pode ser obtido mudando o limite estabelecido como mínima similaridade para aceitar um documento como relevante.

Figura 5.5: Resultados para a consulta "Não entendi seu trabalho. Explique".

(a) Média de Precisão de n pontos. (b) Relação entre documentos recuperados e Medida-F.

Figura 5.6: Resultados para a consulta "Por que você não dá então alguma coisa útil ao povo que está passando necessidade?".

(a) Média de Precisão de n pontos. (b) Relação entre documentos recuperados e Medida-F.

diminuindo a precisão para ambas as configurações com expansão de consulta. Além disso, na Figura 5.7(b) pode-se ver que a Medida-F é mantida com um menor número de documentos do que com a consulta original.

Figura 5.7: Resultados para a consulta “Por que você não dá algo para as pessoas que necessitam?”.

(a) Média de Precisão de n pontos. (b) Relação entre documentos recuperados e Medida-F.

Na Figura 5.8, é apresentada uma consulta em que noexpand apresentou melhores resultados do que as outras configurações. Na parte (a) pode-se observar que a precisão de noexpand é bem maior do que as outras configurações, principalmente da ed-ambos-05, e que o ganho em cobertura obtido pela configuração ed-palch não foi nenhum.

Nesse caso, a expansão da consulta não ajudou, pois os termos que permitiram a recuperação dos documentos relevantes já estavam inclusos na consulta: “participou” e “obra”. Alguns dos termos adicionados na consulta que trazem documentos irrelevantes são “cultura” e “costumes”, ambos relacionados com “povo”, mas que sob este contexto, não tem muita relação. Neste contexto, “povo” se refere a um conjunto de pessoas.

Figura 5.8: Resultados para a consulta “E o povo que participou da sua obra não ganha nada?”.

(a) Média de Precisão de n pontos. (b) Relação entre documentos recuperados e Medida-F.

A precisão pode ser melhorada relacionando o conceito “*peçoas*” com “*povo*” fortemente, pois realmente existem contextos em que esses dois conceitos estão fortemente relacionados. Após fazer isso, a precisão da consulta expandida melhorou, mas o número de documentos recuperados diminuiu e a cobertura máxima obtida diminuiu também.

Finalmente foi analisada uma consulta em que a cobertura e a precisão foram melhores quando se utilizou o mecanismo de expansão de consulta. A Figura 5.9(a) permite perceber que a precisão das configurações se manteve alta, e a cobertura também para as configurações sem expansão. Na Figura 5.9(b), pode-se observar que a máxima Medida-F, i.e. combinação de precisão e cobertura, foi obtida por ed-ambos-05 ao recuperar cinco documentos, mas que foi mantida alta até o total de documentos recuperados.

Neste exemplo, a consulta foi expandida com termos que permitiram recuperar com alta similaridade os documentos relevantes. Esses termos foram “*loja*”, “*mercado*” e “*galéria*” relacionados a “*venda*” na ontologia.

Figura 5.9: Resultados para a consulta “Como você vende sua obra?”.

(a) Média de Precisão de n pontos. (b) Relação entre documentos recuperados e Medida-F.

5.5 Considerações finais

Neste capítulo, foi apresentado o sistema implementado em execução, com o detalhe do processamento de uma consulta para a recuperação dos vídeos relevantes para ela. Foi descrita a estrutura de dados que permitiu a melhora, em termos de tempo de processamento, no mecanismo de expansão de consulta. Foram descritas algumas características interessantes detectadas no processo iterativo de adaptação da ontologia e do conjunto de palavras-chave ao sistema desenvolvido. Após isso, foram apresentados os testes do sistema, com uma bateria de cinquenta consultas, elaboradas pela especialista do domínio, a serem executados em diferentes configurações.

Duas análises foram efetuadas. A primeira consistiu na análise dos resultados globais de tempo, precisão, cobertura e Medida-F. Com a análise de tempo de processamento verificou-se a importância da melhora no mecanismo de expansão, pois a diferença foi significativa entre as configurações usando Jena, e as equivalentes com a modificação proposta. Em termos de precisão e cobertura, a expansão de consultas mostrou melhorar ambas as medidas, sendo mais significativa a diferença em cobertura. Isso é lógico pois o conjunto de termos para a recuperação aumenta.

Como resumo de ambas medidas, em média, a configuração com expansão de consulta e que utiliza tanto o índice de palavras-chave quanto o das transcrições teve o melhor desempenho em termos de Medida-F.

Na segunda análise, selecionaram-se quatro consultas da bateria, com características específicas. Para as duas primeiras, os documentos relevantes não foram recuperados com alta similaridade, ocasionando que nos testes globais, a precisão e cobertura fossem zero. Observou-se que isso é ocasionado pelo fato das consultas utilizarem palavras genéricas e muitas palavras irrelevantes. Para comprovar isso, modificou-se uma das consultas, reduzindo o número de termos dela e utilizando palavras menos genéricas. Na consulta modificada foram obtidas melhoras significativas em termos de precisão e cobertura, reduzindo o número de documentos irrelevantes recuperados.

Foi detectado que quando uma consulta utiliza uma palavra em um sentido que não foi cadastrado na ontologia o desempenho das configurações com expansão de consulta é baixo. Por exemplo: “*povo*” no sentido de “*pessoas*”, na terceira consulta avaliada. Com o cadastro da relação entre as palavras, o desempenho dessas configurações melhorou consideravelmente. Na quarta consulta, foi possível corroborar que a ontologia influenciou na escolha dos documentos relevantes, pois a configuração sem expansão teve um desempenho bastante menor.

Conclusões e Trabalhos Futuros

Após a descrição da proposta e dos resultados obtidos, neste Capítulo resume-se as conclusões obtidas ao longo do desenvolvimento do trabalho assim como as contribuições feitas. Finalmente, são apresentados alguns trabalhos futuros a serem desenvolvidos com base nesta pesquisa.

6.1 Conclusões

A utilização de uma ontologia do domínio de arte contemporânea, em um sistema de recuperação de trechos de vídeos sobre esse tópico, permite melhorar a eficiência do mesmo em termos de precisão e cobertura.

Este trabalho é um sistema de recuperação de informação (RI) que utiliza ontologias mas não utiliza o total do poder de expressão que elas oferecem. Isso faz com que se possa aproveitar as vantagens do uso de ontologias sem ter de utilizar os mecanismos de inferência para as mesmas como o Jena. Assim, podem-se criar mecanismos simplificados, adaptados para o problema que tenta-se resolver obtendo melhores resultados em termos de tempo de processamento.

O domínio da arte contemporânea é bastante abstrato, e como pôde ser corroborado pela especialista do domínio, existe um problema conceitual ao ter de abstraí-lo, para expressá-lo em uma linguagem restrita como a requerida para o desenvolvimento de ontologias.

O tamanho da coleção é um fator importante no desenvolvimento e principalmente na avaliação de sistemas de RI. A variedade de consultas que podem ser feitas é muito grande, e o sistema termina em muitos casos trazendo respostas não muito relevantes pois não conta com respostas adequadas para a consulta do usuário. Além disso, a avaliação é muito sensível, sendo que um único documento irrelevante recuperado ou um único documento relevante não recuperado faz com que a precisão e a cobertura do mesmo sejam fortemente afetadas.

As ferramentas para a manipulação de ontologias, como o Protegé e o Jena, recém estão chegando em um nível de maturidade, pois ainda existe uma ampla comunidade trabalhando nelas. Durante o desenvolvimento da pesquisa, foram lançadas pelo menos duas novas versões de cada uma das ferramentas.

6.2 Contribuições do trabalho

O presente trabalho descreve a implementação de um sistema de recuperação de informação sobre um conjunto de vídeos, testado para uma aplicação específica, a entrevista com a artista brasileira Ana Teixeira.

O sistema foi projetado para ser genérico, é adaptável para ser utilizado com outros vídeos e em outros domínios. O sistema poderia ser utilizado para permitir o melhor aproveitamento de cursos ou seminários, os quais são usualmente longos em duração.

Foi comprovado que a precisão e cobertura do sistema foram melhoradas em média, para a aplicação, quando foi utilizado um mecanismo de expansão de consulta que utiliza uma ontologia do domínio.

Como parte do trabalho foi desenvolvida uma ontologia do domínio da arte contemporânea, principalmente contendo os tópicos tratados nas entrevistas. O desenvolvimento da ontologia do domínio e das entrevistas foram uma contribuição de Paula P. Braga, do Departamento de Filosofia da Universidade de São Paulo.

Conseguiu-se utilizar com sucesso um conjunto de ferramentas prontas em uma aplicação complexa. Essas ferramentas foram utilizadas tanto para manipulação de ontologias (o Jena) quanto para o processamento simples de linguagem (remoção de afixos e correção ortográfica).

6.3 Trabalhos futuros

Existem muitas aplicações do sistema em diversos domínios e requiere-se comprovar a sua facilidade de adaptação. Está-se começando a adaptação do sistema a um conjunto de palestras oferecidas por Joseph Yoder da Universidade de Illinois, USA, ministradas na Universidade de São Paulo no ano 2004 e filmadas pelo Grupo de Sistemas Distribuídos do IME-USP.

As palestras são sobre Padrões de Projeto e Orientação a Objetos, e deverá ser criada uma ontologia sobre esse domínio. Devido ao fato do domínio e o vocabulário serem mais restritos, se espera que a elaboração da ontologia seja mais simples. Uma característica adicional dessa adaptação, é que o idioma dos vídeos é inglês, e se pretende testar a utilização do sistema em inglês, e posteriormente permitir a utilização de português nas consultas.

Esse projeto será parte de um trabalho de iniciação científica do IME-USP orientado por Renata Wassermann e Fábio Kon.

É desejável minimizar o esforço de configuração do sistema para que o especialista do domínio possa utilizá-lo com facilidade. Para isso, se pretende: (1) eliminar a necessidade de fragmentação manual dos vídeos e, (2) permitir a interação com a ontologia de dentro da aplicação de configuração, para selecionar as palavras-chave diretamente da ontologia.

Sistemas de RI que utilizam técnicas de processamento de linguagem natural estão sendo pesquisados recentemente. Existem diversos fatores que poderiam ajudar em melhorar a relevância dos documentos. Um deles é a exploração do diálogo do usuário com o sistema, permitindo estabelecer melhor as necessidades de informação do usuário. Outro fator é a marcação semântica dos textos, ajudando a tratar a polissemia, que não foi tratada neste trabalho.

Finalmente, a interação com linguagem falada é uma área que ainda está em desenvolvimento, e as ferramentas existentes requerem um grande esforço de treinamento. Para facilitar a configuração seria possível capturar as transcrições automaticamente com reconhecimento de fala. Devido ao fato da coleção utilizada ser pequena, o esforço de transcrição foi reduzido. No entanto, esse esforço é proporcional ao tamanho da coleção o qual tende a crescer pela abundância da informação.

Consultas de Teste

Neste apêndice serão apresentadas as consultas utilizadas nos testes do sistema. Elas são apresentadas na Tabela a seguir.

Tabela A.1: Consultas de teste.

ID	CONSULTA
0	Quem é você?
1	O que é arte?
2	Sua obra imita as pegadinhas da tv?
3	Qual é o objeto que sobra das intervenções?
4	O que é intervenção?
5	Por que intervenção urbana é arte?
6	Como você expõe sua obra?
7	Por que você expõe sua obra se ela acontece mesmo na rua?
8	Sua obra se confunde com caridade?
9	Não entendi seu trabalho. Explique
10	As pessoas na rua aceitam comer o sonho?
11	Alguém já brigou com você por causa das intervenções na rua?
12	Não é ilegal usar barraca de camelô?
13	Como você vende essa obra?

Tabela A.1: Consultas de teste (cont.)

ID	CONSULTA
14	Você vive da venda de sua obra?
15	Como você virou artista?
16	Você só faz intervenções ou faz também pintura, escultura, etc?
17	Qual é o seu sonho?
18	Qual o sonho mais comum das pessoas?
19	Por que você resolveu fazer arte na rua?
20	Acho cruel exigir que o povo entenda sua obra
21	Você faz os participantes de bobos?
22	Você usa as pessoas na rua para auto-promover sua obra?
23	Você é artista ou assistente social?
24	Por que você acha que tem que ensinar ou dar aos outros? Isso é prepotência?
25	Por que você não dá então alguma coisa útil ao povo que está passando necessidade?
26	Quantas pessoas já trocaram sonhos com você?
27	Quem financia seu trabalho?
28	Você estudou artes?
29	Onde mais você já expôs?
30	Qual seu telefone?
31	Como posso entrar em contato com você?
32	Detestei seu trabalho?
33	Adorei seu trabalho?
34	Não entendi a obra das identidades?
35	Se você é artista, eu também sou?
36	O que é ser artista?
37	O que você gosta de ler?
38	O camelô é um artista?
39	O que você faz é artes plásticas ou teatro?

Tabela A.1: Consultas de teste (cont.)

ID	CONSULTA
40	Como nasce uma obra sua?
41	O que você pretende com seu trabalho?
42	A arte tem que ser bonita?
43	O que você faz com as impressões digitais?
44	E o povo que participou da sua obra não ganha nada?
45	As pessoas na rua gostam de seu trabalho?
46	Como os transeuntes reagem a sua abordagem na rua?
47	Com quem você estudou artes?
48	Você já expôs no exterior?
49	Quanto custa uma obra sua?

APÊNDICE B

Tabelas de Resultado dos Testes Globais

Neste apêndice serão apresentados os resultados obtidos para os testes globais detalhados no Capítulo 5. Eles são apresentados nas tabelas a seguir.

Consulta	noexp	palch		ambos	
		ed	jena	ed	jena
0	–	–	–	–	–
1	1	0,333	1	0,2	0,333
2	0,5	0,5	0,5	0,5	0,5
3	0,2	0	0	0	0
4	0,2	0	0	0	0
5	0	0	0	0	0
6	1	1	1	0,5	0,333
7	0,4	0,4	0,4	0,4	0,4
8	–	0,2	0,2	0,2	0,2
9	–	0	0	0	0
10	0,4	0,4	0,4	0,4	0,4
11	0,2	0,4	0	0,4	0
12	0,25	0,2	0,2	0,2	0,2
13	1	0,6	0,8	0,8	0,8
14	0,5	0,5	0,6	0,6	0,6
15	0	0,2	0	0	0
16	0,2	0,4	0	0,4	0,2
17	0	0	0	0	0
18	0,25	0,2	0,2	0,2	0,2
19	0	0	0	0	0
20	1	1	0,333	0,25	0,25
21	0,4	0,2	0,333	0	0
22	0,2	0,2	0,2	0,2	0,2
23	0,2	0,2	0,2	0,4	0,4
24	–	0,167	0,6	0,2	0,6
25	0	0	0	0	0
26	0,2	0,2	0,2	0,4	0,4
27	–	0,2	0,2	0,2	0,4
28	0	0	0	0,2	0,25
29	0	0	0	0,333	0,25
30	–	–	–	0	0
31	–	–	–	0	0
32	–	0	0,25	0	0,2
33	–	0	0,25	0	0,2
34	0,5	0,5	0,5	0,2	0,2
35	0	0,2	0	0	0
36	0,25	0,4	0,2	0,2	0,2
37	–	–	–	0	0
38	0	0	0	0	0
39	0	0,2	0,5	0,4	0,6
40	–	0	–	0	0
41	–	0,4	0	0,4	0
42	0	0	0	0,2	0,333
43	–	0	0	0	0
44	0,667	0,6	0,667	0,2	0,4
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0,2	0,25
48	0,667	0,667	0,667	0,6	0,5
49	–	–	–	0	0

Tabela B.1: Precisão para as consultas de teste.

Consulta	noexp	palch		ambos	
		ed	jena	ed	jena
0	0	0	0	0	0
1	1	1	1	1	1
2	0,333	0,333	0,333	0,667	0,333
3	0,5	0	0	0	0
4	1	0	0	0	0
5	0	0	0	0	0
6	0,2	0,2	0,2	0,4	0,2
7	0,5	0,5	0,5	0,5	0,5
8	0	0,333	0,333	0,333	0,333
9	0	0	0	0	0
10	1	1	1	1	1
11	0,5	1	0	1	0
12	1	1	1	1	1
13	0,333	0,5	0,667	0,667	0,667
14	0,25	0,5	0,75	0,75	0,75
15	0	0,333	0	0	0
16	0,167	0,333	0	0,333	0,167
17	0	0	0	0	0
18	0,5	0,5	0,5	0,5	0,5
19	0	0	0	0	0
20	0,125	0,125	0,125	0,125	0,125
21	0,4	0,2	0,4	0	0
22	0,167	0,167	0,167	0,167	0,167
23	0,25	0,25	0,25	0,25	0,5
24	0	0,143	0,429	0,143	0,429

Consulta	noexp	palch		ambos	
		ed	jena	ed	jena
25	0	0	0	0	0
26	0,5	0,5	0,5	1	1
27	0	0,2	0,2	0,2	0,4
28	0	0	0	0,2	0,2
29	0	0	0	0,5	0,5
30	–	–	–	–	–
31	–	–	–	–	–
32	0	0	0,333	0	0,333
33	0	0	0,333	0	0,333
34	0,5	0,5	0,5	0,5	0,5
35	0	0,5	0	0	0
36	0,143	0,286	0,143	0,143	0,143
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0,167	0,167	0,333	0,5
40	0	0	0	0	0
41	0	0,25	0	0,25	0
42	0	0	0	0,25	0,25
43	0	0	0	0	0
44	0,286	0,429	0,571	0,143	0,286
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0,333	0,333
48	0,667	0,667	0,667	1	0,667
49	0	0	0	0	0

Tabela B.2: Cobertura para as consultas de teste.

Consulta	noexp	palch		ambos	
		ed	jena	ed	jena
0	–	–	–	–	–
1	1	0,5	1	0,333	0,5
2	0,4	0,4	0,4	0,571	0,4
3	0,286	–	–	–	–
4	0,333	–	–	–	–
5	–	–	–	–	–
6	0,333	0,333	0,333	0,444	0,25
7	0,444	0,444	0,444	0,444	0,444
8	–	0,25	0,25	0,25	0,25
9	–	–	–	–	–
10	0,571	0,571	0,571	0,571	0,571
11	0,286	0,571	–	0,571	–
12	0,4	0,333	0,333	0,333	0,333
13	0,5	0,545	0,727	0,727	0,727
14	0,333	0,5	0,667	0,667	0,667
15	–	0,25	–	–	–
16	0,182	0,364	–	0,364	0,182
17	–	–	–	–	–
18	0,333	0,286	0,286	0,286	0,286
19	–	–	–	–	–
20	0,222	0,222	0,222	0,167	0,167
21	0,4	0,2	0,364	–	–
22	0,182	0,182	0,182	0,182	0,182
23	0,222	0,222	0,222	0,444	0,444
24	–	0,154	0,5	0,167	0,5

Consulta	noexp	palch		ambos	
		ed	jena	ed	jena
25	–	–	–	–	–
26	0,286	0,286	0,286	0,571	0,571
27	–	0,2	0,2	0,2	0,4
28	–	–	–	0,2	0,222
29	–	–	–	0,4	0,333
30	–	–	–	–	–
31	–	–	–	–	–
32	–	–	0,286	–	0,25
33	–	–	0,286	–	0,25
34	0,5	0,5	0,5	0,286	0,286
35	–	0,286	–	–	–
36	0,182	0,333	0,167	0,167	0,167
37	–	–	–	–	–
38	–	–	–	–	–
39	–	0,182	0,25	0,364	0,545
40	–	–	–	–	–
41	–	0,308	–	0,308	–
42	–	–	–	0,222	0,286
43	–	–	–	–	–
44	0,4	0,5	0,615	0,167	0,333
45	–	–	–	–	–
46	–	–	–	–	–
47	–	–	–	0,25	0,286
48	0,667	0,667	0,667	0,75	0,571
49	–	–	–	–	–

Tabela B.3: Medida-F para as consultas de teste.

Referências Bibliográficas

- [1] J. Aslam and M. Frost. An information-theoretic measure for document similarity. In *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 449–450, Toronto, Canada, 2003. ACM Press, 2003.
- [2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman, 1999.
- [3] W. Bailer, H. Mayer, H. Neuschmied, W. Haas, M. Lux, and W. Klieber. Content-based video retrieval and summarization using MPEG-7. In *Proceedings of the Internet Imaging V*, pages 1–12, San Jose, CA, USA, Janeiro 2004.
- [4] B. Bassett and Histor Systems. Conversation with Jacques Lipchitz: A breakthrough in interactivity, 2001. <http://www.conversationwithjacqueslipchitz.org/>.
- [5] M. G. Brown, J. T. Foote, Gareth J. F. Jones, K. Sparck-Jones, and S. J. Young. Automatic content-based retrieval of broadcast news. In *Proceedings of the 3rd ACM Multimedia Conference*, pages 35–43, San Francisco, USA, Novembro 1995.
- [6] World Wide Web Consortium. RDF Resource Definition Framework, 2004. <http://www.w3.org/RDF/>.
- [7] O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Methodologies, tools and language

- ges for building ontologies. where is their meeting point? *Data & Knowledge Engineering*, 46(1):41–64, Julho 2003.
- [8] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and O. Vicente. WebODE: An integrated workbench for ontology representation, reasoning, and exchange. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*, volume 2473 of *Lecture Notes in Computer Science*, pages 138–153, Siguenza, Espanha, Outubro 2002. Springer–Verlag, 2002.
 - [9] O. Corcho and A. Gómez-Pérez. A roadmap to ontology specification languages. In R. Dieng and O. Corby, editors, *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management Methods, Models and Tools*, volume 1937 of *Lecture Notes in Computer Science*, pages 80–97, França, Outubro 2000. Springer–Verlag, 2000.
 - [10] F. Crestani, M. Lalmas, C. Van Rijsbergen, and I. Campbell. Is this document relevant?: probably: a survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, 1998.
 - [11] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
 - [12] R. Johnson E. Gamma, R. Helm and R. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Adisson Wesley, 1995.
 - [13] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, and S. Tu. The evolution of Protégé–2000: An environment for knowledge–based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.
 - [14] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.
 - [15] N. Guarino, C. Masolo, and G. Vetere. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, Maio 1999.

- [16] V. Haarslev and R. Möller. Description of the Racer system and its applications. In *Proceedings of the International Workshop on Description Logics DL'2001*, pages 132–141, Stanford, USA, Agosto 2001.
- [17] D. Harman. Ranking algorithms. In W. Frakes and R. Baeza-Yates, editors, *Information Retrieval*, chapter 14, pages 363–392. Prentice Hall, 1992.
- [18] P. Jackson and I. Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization*. John Benjamins Publishing Co, 2002.
- [19] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, pages 19–33, Taiwan, 1997.
- [20] V. Kashyap. Design and creation of ontologies for environmental information retrieval. In *Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management*, Banff, Canada, October 1999.
- [21] L. Khan. *Ontology-based Information Selection*. PhD thesis, Department of Computer Science, University of Southern California, 2000.
- [22] G. Kline. High-tech sculptor has the answers. *The News-Gazette Online*, Outubro 2001. Publicado na WWW no 4 de Outubro de 2001: <http://www.news-gazette.com/story.cfm?Number=10249>.
- [23] H. Knublauch, M. Musen, and A. Rector. Editing description logics ontologies with the Protégé OWL plugin. In *International Workshop on Description Logics*, Whistler, BC, Canada, 2004.
- [24] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, San Francisco, USA, 1998. Morgan Kaufmann Publishers Inc., 1998.
- [25] M. Mauldin. *Conceptual Information Retrieval A case study in adaptive partial parsing*. Kluwer Academic Publishers, 1991.

- [26] D. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. Technical report, World Wide Web Consortium, 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [27] G. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
- [28] N. Noy and D. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001.
- [29] V. Orenço and C. Huyck. A stemming algorithm for the Portuguese language. In *Proceedings of the 8th International Symposium on String Processing and Information Retrieval(SPIRE) 2001*, pages 186–193, 2001. O algoritmo está disponível em <http://www.cs.mdx.ac.uk/research/PhDArea/rslp/RSLP.htm>.
- [30] C. Paz-Trillo, R. Wassermann, and P. Braga. Using ontologies to retrieve video information. In F. Freitas, H. Stickschmidt, and R. Volz, editors, *Proceedings of the Workshop on Ontologies and their Applications*, pages 55–66, São Luís Maranhão, Brazil, 2004.
- [31] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, Julho 1980. Diversas implementações do algoritmo estão disponíveis em: <http://www.tartarus.org/~martin/PorterStemmer/>.
- [32] R. Rada, H.Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, Janeiro-Fevereiro 1989.
- [33] R. Richardson, A. Smeaton, and J. Murphy. Using WordNet as a knowledge base for measuring semantic similarity between words. Technical Report CA-1294, Dublin University - School of Computer Applications, Dublin, Ireland, 1994.
- [34] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

- [35] M. Smith, C. Welthy, and D. McGuiness. OWL Web Ontology Language Guide. Technical report, World Wide Web Consortium, 2004. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- [36] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. Ontoedit: Collaborative ontology development for the semantic web. In I. Horrocks and J. Hendler, editors, *Proceedings of the First International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 221–235, Sardinia, Italia, June 2002. Springer-Verlag, 2002.
- [37] H. Turtle. Natural language vs. boolean query evaluation: a comparison of retrieval performance. In *Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 212–220, Dublin, Ireland, 1994. Springer-Verlag, 1994.
- [38] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.
- [39] E. Voorhes. Using wordnet for text retrieval. In C. Fellbaum, editor, *WordNet, an Electronic Lexical Database*, chapter 12, pages 285–303. The MIT Press, 1998.
- [40] M. Worring, A. Bagdanov, J. v. Gemert, J-M. Geusebroek, M. Hoang, A.Th. Schreiber, C.G.M. Snoek, J. Vendrig, J. Wielemaker, and A.W.M. Smeulders. Interactive indexing and retrieval of multimedia content. In *Proceedings of the 29th Conference on Current Trends in Theory and Practice of Informatics*, volume 2540 of *Lecture Notes in Computer Science*, pages 135–148, Milovy, Czech Republic, 2002. Springer-Verlag, 2002.