

**Universidade Federal de Santa Catarina**  
**Programa de Pós-Graduação em**  
**Engenharia e Gestão do Conhecimento**

Raphael Winckler de Bettio

Interrelação das Técnicas *Term Extraction* e *Query Expansion*  
aplicadas na Recuperação de Documentos Textuais

Tese

Florianópolis  
2007

Raphael Winckler de Bettio

Interrelação das Técnicas *Term Extraction* e *Query Expansion*  
aplicadas na Recuperação de Documentos Textuais

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de Doutor em Engenharia e Gestão do Conhecimento

Orientador: Prof. Alejandro Martins Rodriguez, Dr.

Florianópolis  
2007

Raphael Winckler de Bettio

Interrelação das Técnicas *Term Extraction* e *Query Expansion*  
aplicadas na Recuperação de Documentos Textuais

Esta tese foi julgada e aprovada para a obtenção do grau de **Doutor em Engenharia e Gestão do Conhecimento** no **Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento** da Universidade Federal de Santa Catarina.

Florianópolis, 25 de outubro de 2007.

Prof. Roberto Carlos dos Santos Pacheco, Dr.  
Coordenador do Programa

BANCA EXAMINADORA

Prof. Alejandro Martins Rodriguez, Dr. Universidade Federal de Santa Catarina Orientador	Prof. Álvaro José Periotto, Dr. Universidade Estadual de Maringá Membro Externo
Prof. Fabiano Luiz Santos Garcia, Dr. Universidade Federal de Santa Catarina Co-orientador	Prof. Marcos Antonio G. Brasileiro, Dr. Universidade Federal da Paraíba Membro Externo
Prof. Andréa da Silva Miranda, Dra. Universidade Federal de Santa Catarina Moderadora	

**À Fernanda, pelo seu amor, dedicação e paciência.**

## ***Agradecimentos***

Ao Prof. Alejandro Martins pela amizade e pelo bom coração

...

Ao Prof. Fabiano Garcia pela colaboração na orientação e amizade

...

Ao Fábio dos Anjos pela importante participação nesta pesquisa

...

Aos membros da banca, Prof. Álvaro Periotto e Prof. Marcos Brasileiro, que se dedicaram à leitura deste trabalho, trazendo suas contribuições

...

Aos meus pais, Vânio e Edi por todo o carinho, dedicação e amor

...

A todos os meus amigos verdadeiros

...

***“Existe algo mais importante que o  
talento: chama-se determinação”  
Ory Rodrigues***

## Resumo

**BETTIO, Raphael Winckler de.** INTERRELAÇÃO DAS TÉCNICAS TERM EXTRACTION E QUERY EXPANSION APLICADAS NA RECUPERAÇÃO DE DOCUMENTOS TEXTUAIS. 2007. 99 f. Tese (Doutorado em Engenharia e Gestão do Conhecimento) – Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, UFSC, Florianópolis.

Conforme Signal (2006) as pessoas reconhecem a importância do armazenamento e busca da informação e, com o advento dos computadores, tornou-se possível o armazenamento de grandes quantidades dela em bases de dados. Em consequência, catalogar a informação destas bases tornou-se imprescindível. Nesse contexto, o campo da Recuperação da Informação, surgiu na década de 50, com a finalidade de promover a construção de ferramentas computacionais que permitissem aos usuários utilizar de maneira mais eficiente essas bases de dados. O principal objetivo da presente pesquisa é desenvolver um Modelo Computacional que possibilite a recuperação de documentos textuais ordenados pela similaridade semântica, baseado na intersecção das técnicas de *Term Extration* e *Query Expansion*.

Palavras-chave: *Term Extration*, *Query Expansion*, Busca Textual, Ontologias, Semântica.

## **Abstract**

**BETTIO, Raphael Winckler de. INTERRELAÇÃO DAS TÉCNICAS TERM EXTRACTION E QUERY EXPANSION APLICADAS NA RECUPERAÇÃO DE DOCUMENTOS TEXTUAIS. 2007. 99 f. Tese (Doutorado em Engenharia e Gestão do Conhecimento) – Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, UFSC, Florianópolis.**

As Sigal (2006) the people recognize the importance of the storage and search of the information and, with the advent of the computers, the storage of great amounts of it in databases became possible. In consequence, to catalogue the information of these bases became essential. In this context, the field of the Information Recovery appeared in the decade of 50, with the purpose to promote the construction of computational tools that allow the use of these databases in more efficient way. The main objective of the present research is to develop a Computational Model that makes possible textual documents recovery by the similarity semantics, based on the intersection of Term Extration and Query Expansion techniques.

Palavras-chave: Term Extration, Query Expansion, Textual Search, Ontology, Semantic.



## SUMÁRIO

LISTA DE FIGURAS .....	9
LISTA DE QUADROS .....	10
LISTA DE TABELAS .....	11
1 INTRODUÇÃO .....	12
1.1 Contextualização da Pesquisa.....	12
1.2 Problema da Pesquisa.....	14
1.4 Objetivos da Pesquisa .....	16
1.4.1 Objetivo Geral .....	16
1.4.2 Objetivos Específicos .....	16
1.5 Metodologia da Pesquisa.....	17
1.5.1 Hipótese da Pesquisa .....	17
1.5.2 Classificação da Pesquisa .....	17
1.5.3 Validação da Pesquisa.....	18
1.5.4 Organização do Trabalho.....	19
2 FUNDAMENTAÇÃO TEÓRICA.....	21
2.1 Ontologias.....	21
2.2 <i>Inverse Document Frequency</i> .....	31
2.3 <i>StopWords</i> .....	35
2.4 <i>Stemming</i> .....	36
2.5 <i>Term Extraction</i> .....	40
2.6 <i>Query Expansion</i> .....	42
2.7 <i>Text Retrieval Conference (TREC)</i> .....	45
3 DESENVOLVIMENTO DO MODELO COMPUTACIONAL.....	58
3.1 Técnicas Utilizadas para Construção do Modelo Computacional .....	58
3.2 Integração das Técnicas para Construção do Modelo Computacional.....	59
3.3 Etapas do Modelo Computacional .....	60
3.3.1 Criação do Vetor Inicial .....	64
3.3.2 Criação do Vetor Expandido .....	65
3.3.3 Preenchimento do Vetor de Termos com suas Relevâncias.....	67
3.3.4 Preenchimento do Vetor de Termos com suas Relevâncias Cruzadas.....	68
3.3.5 Criação do Vetor de Corte.....	70
3.3.6 Criando a <i>Query</i> .....	73
4 VALIDAÇÃO DO MODELO E IMPLEMENTAÇÃO DO PROTÓTIPO .....	75
4.1 Indexação de Documentos Textuais.....	75
4.2 Manipulação de Ontologias .....	76
4.3 Validação do Modelo .....	77
4.4 Protótipo .....	79
4.4.1 Indexação de Documentos.....	79
4.4.2 Manipulação da Ontologia.....	81
4.4.3 Execução da Busca.....	83
4.5 Documentos Utilizados e Ontologia Criada .....	85
4.5.1 Seleção de Documentos .....	85
4.5.2 Criação da Ontologia.....	87
4.6 <i>Recall e Precision</i> .....	90
5 CONSIDERAÇÕES FINAIS .....	92
5.1 Conclusões .....	92
5.2 Recomendações para Futuros Trabalhos.....	95
REFERÊNCIAS BIBLIOGRÁFICAS .....	97

## LISTA DE FIGURAS

Figura 1: Representação Gráfica da Ontologia .....	28
Figura 2: Representação Utilizando OWL .....	29
Figura 3: Representação Gráfica das Instâncias e Relações.....	31
Figura 4: Fórmula do IDF .....	32
Figura 5: Representação Gráfica da Curva do IDF .....	32
Figura 6: Etapas do Algoritmo de Orenço/Stemming para Língua Portuguesa.....	37
Figura 7: Exemplo de Regra do Algoritmo de Stemming da Língua Portuguesa .....	38
Figura 8: Fórmula C-Value .....	41
Figura 9: Interrelação das Tecnologias .....	58
Figura 10: Entrada e Saída do Modelo Computacional.....	59
Figura 11: Distribuição das Técnicas .....	61
Figura 12: Estrutura da Ontologia .....	62
Figura 13: Representação Matemática das Inferências .....	63
Figura 14: Termos e Relações Semânticas.....	64
Figura 15: Cálculo da Relevância Cruzada .....	69
Figura 16: Número de Termos Seleccionados por Grupo .....	72
Figura 17: <i>Query</i> Gerada .....	74
Figura 18: Representação Hiperbólica de Ontologia.....	77
Figura 19: Fluxo do Processo de Validação do Modelo .....	78
Figura 20: Exemplo de Cálculo de <i>Recall</i> e <i>Precision</i> .....	79
Figura 21: XML Específico para Indexação.....	80
Figura 22: Módulo de Indexação .....	81
Figura 23: Módulo de Manipulação da Ontologia .....	82
Figura 24: Execução das Buscas .....	83
Figura 25: Valores Resultantes do Algoritmo .....	84
Figura 26: Representação Visual do Vetor Final .....	85
Figura 27: Exemplo de <i>Query</i> Modificada .....	90

## LISTA DE QUADROS

Quadro 1: Linguagens para Representação de Ontologias .....	25
Quadro 2: Resultado das Inferências .....	30
Quadro 3: Lista de Exemplos de <i>StopWords</i> .....	35
Quadro 4: Palavras <i>Stemmizadas</i> .....	39
Quadro 5: Artigos, Autores e Universidades Relevantes da TREC.....	53
Quadro 6: Artigos e Tecnologias Utilizadas .....	56
Quadro 7: Portais com Documentos Escolhidos .....	86
Quadro 8: Documentos Escolhidos .....	86

## LISTA DE TABELAS

Tabela 1: Representação Numérica da Curva do IDF.....	33
Tabela 2: Termos e Respectivos Pesos IDF .....	34
Tabela 3: Pesos Associados às Relações Semânticas dos Termos .....	62
Tabela 4: Resultado da Primeira Fase do Modelo .....	65
Tabela 5: Termos Inferidos e Pesos Semânticos .....	67
Tabela 6: Termos e Relevâncias Ontológica e IDF .....	68
Tabela 7: Vetor de Termos de Relevâncias Cruzadas .....	69
Tabela 8: Termos e Relevâncias Cruzadas .....	70
Tabela 9: Termos e Pesos Finais.....	72
Tabela 10: Grupos e Somatórios.....	73
Tabela 11: Vetor de Termos Atual.....	73
Tabela 12: Termos e Relações Explícitas .....	89
Tabela 13: Resultados de Recuperação da <i>Query</i> Original .....	91
Tabela 14: Resultados de Recuperação da <i>Query</i> Modificada .....	91
Tabela 15: Resultados de <i>Recall</i> e <i>Precision</i> .....	91

# 1 INTRODUÇÃO

## 1.1 Contextualização da Pesquisa

“Durante toda a história da humanidade, a cada novo paradigma que aparece voltamos a zero em termos dos padrões e regras até então utilizados. Neste processo evolutivo, passamos da Sociedade Agrícola para a Sociedade Industrial e da Sociedade da Informação e para a Era do Conhecimento, a qual está baseada no conhecimento e em valores intangíveis que este conhecimento poderá trazer de retorno às organizações” (RODRIGUEZ y RODRIGUEZ, 2001, p. 05).

Muitos teóricos atestam que o desenvolvimento da Sociedade Industrial está em seu fim e que um novo contexto sócio-econômico surge: a Sociedade do Conhecimento (DRUKER *apud* PONCHIROLLI, 2000).

Essa nova sociedade está caracterizada, principalmente, por um período de rápidas mudanças tecnológicas, econômicas e sociais. Segundo Crawford (1994), os próximos anos nos reservam um período em que empresas seculares desaparecerão em um ano, um período onde países em que ninguém acreditava começarão a emergir como novas forças mundiais. Essas mudanças vêm surgindo em função de uma profunda transformação na economia mundial. Enquanto países de terceiro mundo passam pelo processo de industrialização, as economias desenvolvidas são rapidamente transformadas em economias pós-industriais baseadas em conhecimento.

Como parte integrante desse processo e com a finalidade de amparar a nova sociedade em seu desenvolvimento surgiu a Engenharia do Conhecimento. A Engenharia do Conhecimento tem como principal objetivo pesquisar acerca do conhecimento em todos os seus aspectos.

A forma de conhecimento que será abordado nesta pesquisa é o manifestado através da escrita.

Conforme Signal (2006) as pessoas sabem sobre a importância do armazenamento e busca de informação. Com o advento dos computadores, tornou-se possível o armazenamento de grandes quantidades de informação em bases de dados e, em conseqüência, catalogar a informação dessas bases tornou-se imprescindível.

Nesse contexto, o campo da Recuperação da Informação surgiu na década de 50 e vem sendo aprimorado desde então, a partir da criação de modelos computacionais capazes de tornar essa atividade possível. Nesta mesma época, diversas técnicas foram criadas, entre elas pode-se citar como relevantes: as criadas por H. P. Luhn em 1957 e os sistemas SMART criados por Gerard Salton (SIGNAL, 2006).

As técnicas continuaram a evoluir até que, em 1992, foi criada a *Text Retrieval Conference* – TREC – que consiste numa série de conferências com o objetivo de discutir e avaliar as técnicas de Recuperação da Informação.

Diversos modelos modernos como por exemplo os criados por Korfhage (1992), Gallant (1992) e Nelson (1992) foram implementados e apresentados na TREC, baseados nas mais diversas técnicas, tais como: *Estatística, Query Expansion, Term Extration, Neural Networks, Genetic Algorithms* entre outras. Os modelos que têm como entrada uma pequena quantidade de termos informados pelo usuário, como em ferramentas de Recuperação da Informação disponíveis na Internet – Google, Yahoo, MSN – são largamente utilizados e conhecidos como “máquinas de busca”.

As pesquisas desenvolvidas são direcionadas a usuários finais e também aos pesquisadores. Em geral, as ferramentas direcionadas aos usuários têm por objetivo buscar textos na Internet, na rede interna das instituições ou em seus computadores

personais. Já as ferramentas direcionadas aos pesquisadores têm por objetivo facilitar a implementação e validação de novos modelos computacionais.

Analisando as publicações da TREC e percebendo a quantidade de novas técnicas apresentadas a cada ano, é possível constatar que existem lacunas a serem preenchidas, principalmente, a partir da implementação de modelos computacionais que acrescentem outra visão em uma área que se encontra em pleno crescimento.

## 1.2 Problema da Pesquisa

De acordo com Almeida (2003), o aumento exponencial dos dados disponíveis tem conferido importância significativa às técnicas de organização da informação. Essas técnicas fazem parte de um corpo de disciplinas que busca melhorias no tratamento de dados, atuando na sua seleção, processamento, recuperação e disseminação.

Analisando as publicações da TREC, pode-se afirmar que existem duas grandes áreas de pesquisa relevantes para a presente pesquisa, a saber: a busca de documentos através de palavras-chave (um conjunto destas palavras é denominado *Query*) e a extração de palavras que melhor representam um determinado documento (*Term Extraction*).

Uma subárea bastante explorada nas pesquisas publicadas na TREC é a *Query Expansion*, onde os pesquisadores têm por objetivo modificar a *Query* formulada pelo usuário para melhorar a eficiência da busca. Estas pesquisas apresentam diversas formas de expandir a *Query* como em Keefer (1994), Efthimiadis (1993), Zhai (1996), entre outros, sendo que a utilização de Ontologias

(modelo matemático que permite a explicitação dos conceitos e relações dos mesmos em uma determinada área do conhecimento) vem se tornando bastante comum. Um dos motivos da utilização das Ontologias é permitir que a similaridade dos documentos seja semântica e não apenas estatística.

Entretanto, a quantidade de artigos publicados na TREC referentes à utilização da técnica *Query Expansion* em conjunto com a *Term Extraction* é reduzida em relação ao número total de pesquisas apresentadas, deixando, assim, um espaço vazio no que se refere à área de Recuperação da Informação. A união da técnica *Query Expansion* com as Ontologias possibilita a busca semântica de textos, no entanto, as pesquisas apresentadas utilizam como entrada de dados um conjunto de palavras determinadas pelo usuário.

A utilização dessas técnicas em conjunto com a *Term Extraction* possibilitará que o usuário opte por usar como entrada de dados um texto, o que representa um ganho significativo no que diz respeito à Recuperação da Informação, indo além do uso de palavras-chave como nos sistemas de recuperação mais conhecidos.

Com o intuito de validar o modelo computacional que será desenvolvido nesta pesquisa, um software será criado. Estabeleceu-se que no desenvolvimento deste software serão utilizadas ferramentas *Open Source*, pois o uso de ferramentas computacionais distribuídas sob esse conceito, permite que o software criado para esta tese seja focado no modelo computacional principal, que será apresentado nesta pesquisa, e não nas funcionalidades marginais, sendo estas supridas por ferramentas já existentes. Considera-se funcionalidades marginais a indexação de documentos e a manipulação de Ontologias.

De acordo com o *Working Group on Libre Software* (WGLS, 2000) não é fácil definir o termo *Open Source* em poucas palavras, pois existem muitas categorias e



variantes deste conceito. As principais características são vinculadas às permissões que os usuários têm. Entre estas características pode-se citar, por exemplo, usar o software como desejarem, para o que desejarem, no número de computadores que desejarem e em qualquer situação técnica que desejarem. Deste modo, as ferramentas liberadas sobre o conceito de Software Livre poderão ser utilizadas na fase de prototipação e validação do modelo computacional.

## 1.4 Objetivos da Pesquisa

### 1.4.1 Objetivo Geral

Esta tese tem como objetivo geral desenvolver um Modelo Computacional que possibilite a recuperação de documentos textuais ordenados pela similaridade semântica em relação a um documento base determinado pelo usuário.

### 1.4.2 Objetivos Específicos

Para se alcançar o objetivo geral desta tese, estabeleceu-se os seguintes objetivos específicos:

- Definição de uma Ontologia que possibilite o mapeamento de relações entre termos de um determinado domínio;
- Implementação de um modelo computacional empregando a técnica de *Query Expansion* através da Ontologia criada;
- Incorporação da técnica de *Term Extraction* ao modelo criado;

- Implementação de um software baseado no modelo computacional criado, com a finalidade de validação do mesmo, utilizando como base em ferramentas *Open Source* já existentes.

## 1.5 Metodologia da Pesquisa

### 1.5.1 Hipótese da Pesquisa

Para se atingir o objetivo principal desta pesquisa toma-se como hipótese a criação de um modelo computacional associando-se as técnicas de *Query Expansion* e *Term Extration*, que torne viável a busca de textos semanticamente similares e que possa ser implementado, utilizando-se como base ferramentas computacionais já existentes e liberadas sob a licença *Open Source*.

### 1.5.2 Classificação da Pesquisa

Para Gil (*apud* Silva e Menezes, 2001, p.19) a pesquisa tem um caráter pragmático, é um “processo formal e sistemático de desenvolvimento do método científico. O objetivo fundamental da pesquisa é descobrir respostas para problemas, mediante o emprego de procedimentos científicos”.

Seguindo o mesmo raciocínio, Silva e Menezes complementam:

“Pesquisa é um conjunto de ações, propostas para encontrar a solução para um problema, que têm por base procedimentos racionais e sistemáticos. A pesquisa é realizada quando se tem um problema e não se têm informações para solucioná-lo” (SILVA e MENEZES, 2001, p.20).

Quanto à classificação desta pesquisa, do ponto de vista de sua natureza, pode ser considerada uma pesquisa aplicada, pois tem o objetivo de promover conhecimento para aplicações práticas, dirigidas à solução de um problema específico, envolvendo verdades e interesses locais.

Já no que diz respeito à forma de abordagem do problema a ser estudado, esta pesquisa está classificada como qualitativa, uma vez que considera a relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito, que não pode ser traduzido em números.

Levando-se em consideração seus objetivos, esta pesquisa pode ser considerada uma pesquisa exploratória, porque visa proporcionar maior familiaridade com o problema, com vistas a torná-lo explícito ou a construir hipóteses.

Acerca dos procedimentos técnicos a serem utilizados nesta pesquisa, ela está classificada como um Pesquisa Experimental, já que envolve um estudo profundo e exaustivo dos conceitos que envolvem as Máquinas de Busca, de maneira que se permita o seu amplo e detalhado conhecimento.

### 1.5.3 Validação da Pesquisa

Segundo Singhal (2006), a avaliação das técnicas de busca é uma das áreas de pesquisa no campo da Recuperação da Informação, já que o desenvolvimento desta área está diretamente ligado à criação de novas idéias e à avaliação dos efeitos destas idéias, especialmente devido à natureza experimental deste campo.

Os testes de Cranfield, conduzidos nos anos 60, designaram uma série de características que foram debatidas durante anos e os resultados desses debates foram o estabelecimento de duas características que são aceitas pela comunidade científica como medidas de eficiência da busca (SIGHAL, 2006):

- *Recall*: é a proporção de documentos relevantes recuperados pelo sistema em relação a todos os documentos da base;
- *Precision*: é a proporção de documentos relevantes recuperados pelo sistema em relação aos documentos recuperados.

A análise dos resultados destes testes resultou na concepção que um bom sistema de Recuperação da Informação deve recuperar o máximo possível de documentos relevantes e trazer entre esses documentos o mínimo possível de documentos não relevantes. Infelizmente, durante anos de pesquisa, aparentemente essas características são contraditórias, ou seja, quanto maior o número de documentos recuperados pelos sistemas, maior o número de documentos não relevantes são encontrados.

Assim sendo, foi definido que estas características deverão estar contempladas no que tange a análise dos resultados encontrados utilizando-se o modelo criado para esta pesquisa.

#### 1.5.4 Organização do Trabalho

Esta Tese está estruturada em cinco capítulos, a saber:

- Capítulo I – contém a contextualização, a originalidade e a relevância da pesquisa. Também trata do problema a ser resolvido, objetivos geral e específicos, hipóteses e metodologia da pesquisa;

- Capítulo II – expõe os principais referencias teóricos utilizados como base para o desenvolvimento do modelo;
- Capítulo III – apresenta um Modelo Computacional com a finalidade de se alcançar o objetivo geral da pesquisa.
- Capítulo IV – apresenta os resultados encontrados a partir da implementação de um protótipo do Modelo Computacional;
- Capítulo V – expõe as conclusões desta tese e recomendações para futuros trabalhos a serem desenvolvidos nesta linha de pesquisa.

## 2 FUNDAMENTAÇÃO TEÓRICA

No desenvolvimento do modelo computacional apresentado no Capítulo 3 desta Tese será necessário o conhecimento teórico das técnicas abaixo relacionadas.

*Ontologias*: proporciona o incremento da eficiência do modelo através das relações semânticas dos termos.

*Inverse Document Frequency*: técnica largamente utilizada, criada com o objetivo de verificar a relevância entre termos de bases de dados textuais.

*StopWords*: técnica que visa remover termos pouco significativos para melhorar o poder de processamento dos algoritmos.

*Stemming*: utiliza como base conhecimentos da área lingüística e tem como principal finalidade tornar possível aos algoritmos reconhecer a semelhança entre palavras.

*Term Extration*: possibilita a seleção de termos que melhor representam um determinado documento em uma base de dados.

*Query Expansion*: tem por objetivo melhorar as *Querys* (conjunto de palavras-chaves) informadas pelos usuários no momento da busca.

### 2.1 Ontologias

Conforme Gruber (2005) o termo Ontologia tem gerado uma série de controvérsias em discussões sobre Inteligência Artificial – IA (conceitos desta área do conhecimento são largamente utilizadas na Eng. Do Conhecimento). Ao longo da história da filosofia, o termo Ontologia refere-se ao sujeito da existência. Já no

contexto da IA, onde o que "existe" é aquilo que pode ser "representado", mais especificadamente dentro do contexto de compartilhamento do conhecimento, o termo Ontologia significa especificação de conceitos. Portanto, uma Ontologia é uma especificação formal sobre conceitos e relações que existem em um agente ou em uma comunidade de agentes.

Sob essa ótica, é possível afirmar que seu uso é extenso. Entretanto, em função do contexto deste trabalho, sua importância será simplificada e tratada sobre três aspectos apresentados por Lima (2005):

- Identificação de contexto: quando dois agentes de software trocam informações sobre **braço** é preciso assegurar em que contexto este termo está sendo referenciado. Isto é, um agente pode se referir ao termo **braço** no contexto da medicina, por exemplo; logo, **braço** é um membro do corpo humano. O outro agente pode se referir ao termo **braço** no contexto de móveis; logo, ele está se referindo a um **braço** de sofá, por exemplo.

- Fornecimento de definições compartilhadas: se uma aplicação X possui uma Ontologia que define uma loja que vende **carros** e uma aplicação Y possui outra Ontologia que define uma loja que vende **veículos**, logo se percebe o problema caso ambas queiram intercambiar informações. Este problema é bastante natural para o ser humano e difícil para uma máquina. Este tipo de confusão pode ser resolvido se as Ontologias proverem relações de equivalência, ou seja, se uma ou as duas Ontologias possuírem informações dizendo que o **carro** da aplicação X é equivalente ao **veículo** da aplicação Y.

- Reuso de Ontologias: se uma determinada pessoa X já tem construída uma Ontologia (na área da medicina, por exemplo) que define um conjunto de termos que outra pessoa Y também necessita, então não há porque a pessoa Y criar

outra Ontologia, isto é, refazer o trabalho que já foi feito. Ela pode simplesmente fazer uso da Ontologia já criada.

Para que a formalização do conhecimento possa ser feita e, assim, a construção de softwares com as características anteriormente citadas possa ser realizada, é necessária a criação de uma linguagem formal para representação das Ontologias. De acordo com Almeida (2003) diversas delas já foram criadas e estão descritas no quadro a seguir (Quadro 1).

Nome	Descrição
Frame Logic <a href="http://flora.sourceforge.net/aboutFlogic.php">http://flora.sourceforge.net/aboutFlogic.php</a>	Linguagem formal que expressa conhecimento por meio de um vocabulário de termos (constantes semânticas, variáveis, número, seqüências de caracteres, etc) os quais são combinados em expressões, sentenças e, finalmente, bases de conhecimento.
FLOGIC <a href="http://www.cs.umbc.edu/771/papers/flogic.pdf">http://www.cs.umbc.edu/771/papers/flogic.pdf</a>	Integra <i>frames</i> e lógica de primeira ordem. Trata-se de uma forma declarativa dos aspectos estruturais das linguagens baseadas em <i>frames</i> e orientadas a objeto (identificação de objetos, herança, tipos polimórficos, métodos de consulta, encapsulamento). Permite a representação de conceitos, taxonomias, relações binárias, funções, instâncias, axiomas e regras.
LOOM <a href="http://www.isi.edu/isd/LOOM/LOOM-HOME.html">http://www.isi.edu/isd/LOOM/LOOM-HOME.html</a>	Descendente da família KL-ONE (Knowledge Language One), é baseada em lógica descritiva e regras de produção. Permite a representação de conceitos, taxonomias, relações n-árias, funções, axiomas e regras de produção.
CARIN	Trata-se de uma combinação de Datalog (linguagem baseada em regras) e lógica descritiva ALN. Uma Ontologia CARIN é construída por dois



	componentes terminológicos: um conjunto de conceitos com declarações de inclusão e um conjunto de regras que usam conceitos.
GRAIL	É uma linguagem que especifica uma Ontologia de domínio médico (Galen). É uma linguagem baseada em lógica descritiva, terminologicamente limitada, que permite a construção de hierarquias de primitivas e axiomas de inclusão de conceitos.
OntoLíngua <a href="http://www.ksl.stanford.edu/software/ontolingua/">http://www.ksl.stanford.edu/software/ontolingua/</a>	Combina paradigmas de linguagens baseadas em <i>frames</i> e lógica de primeira ordem. Permite a representação de conceitos, taxonomias de conceitos, relações n-árias, funções, axiomas, instâncias e procedimentos. Sua alta expressividade causa problemas na construção de mecanismos de inferência.
OCML <a href="http://kmi.open.ac.uk/projects/ocml/">http://kmi.open.ac.uk/projects/ocml/</a>	Permite a especificação de funções, relações e classes, instâncias e regras. Utilizada em aplicações do gerenciamento do conhecimento, desenvolvimento de Ontologias, comércio eletrônico e sistemas baseados em conhecimento. Aplicada em medicina, ciências sociais, memória corporativa, engenharia, portais WEB, etc.
OML (Ontology Markup Language) <a href="http://www.ontologos.org/OML/">http://www.ontologos.org/OML/</a>	Linguagem baseada em lógica descritiva e grafos conceituais, que permite a representação de conceitos organizados em taxonomias, relações e axiomas.
RDF (Resource Description Framework) / RDFS (RDF Schema) <a href="http://www.w3.org/RDF/">http://www.w3.org/RDF/</a>	Desenvolvido pelo W3 Consortium, tem por finalidade a representação do conhecimento por meio da idéia de redes semânticas. São linguagens que permitem a representação de conceitos, taxonomias de conceitos e relações binárias.
NKRL (Narrative Knowledge	Linguagem de representação baseada em <i>frames</i> ,

Representation Language)	especialmente desenvolvida para descrever modelos semânticos de documentos multimídia.
SHOE (Simple HTML Ontology Extensions) <a href="http://www.cs.umd.edu/projects/plus/SHOE/onts/">http://www.cs.umd.edu/projects/plus/SHOE/onts/</a>	Utiliza extensões ao HTML, adicionando marcações para inserir <i>metadados</i> em páginas WEB. As marcações podem ser utilizadas para a construção de Ontologias e para anotações em documentos WEB.
XOL <a href="http://www.ai.sri.com/pkarp/xol/">http://www.ai.sri.com/pkarp/xol/</a>	É uma linguagem que pode especificar conceitos, taxonomias e relações binárias. Não possui mecanismos de inferência e foi projetada para o intercâmbio de Ontologias no domínio da biomédica.
OIL (Ontology Interchange Language) <a href="http://www.ontoknowledge.org/oil/">http://www.ontoknowledge.org/oil/</a>	Precursor do DAML+OIL e base para uma linguagem para a WEB Semântica. Combina primitivas de modelagem das linguagens baseadas em <i>frames</i> com semântica formal e serviços de inferência da lógica descritiva. Pode verificar classificação e taxonomias de conceitos.
DAML (DARPA Agent Markup Language) + OIL: DAML+OIL <a href="http://www.daml.org/">http://www.daml.org/</a>	É uma linguagem de marcação semântica para WEB, que apresenta extensões a linguagens como DAML, RDF e RDFS, por meio de primitivas de modelagem baseadas em linguagens lógicas.
FOML (Formal Ontology Markup Language)	Trata-se de uma linguagem de marcação, baseada em XML, que conecta documentos da WEB com Ontologias formais. O objetivo é a aquisição automática de conhecimento de domínios específicos.
OWL (Ontology Web Language) <a href="http://www.w3.org/TR/owl-features">http://www.w3.org/TR/owl-features</a>	É uma linguagem criada para ser utilizadas em aplicações que precisem processar informações ao invés de apenas apresentá-las a seres humanos.

Quadro 1: Linguagens para Representação de Ontologias

A linguagem que será utilizada nesta tese é a OWL e foi escolhida por possuir uma série de características citadas a seguir. Para Lima (2005) ela é uma revisão da linguagem DAML+OIL. Essa linguagem tem facilidade para expressar significados e semânticas em relação a XML, RDF e RDF Schema, embora ela seja baseada em RDF e RDF Schema e utilize a sintaxe XML.

A OWL foi projetada para ser usada em aplicações que necessitem processar o conteúdo de informações, ao invés de somente apresentar a visualização destas informações. O W3C (World Wide Web Consortium) que é um consórcio criado para especificar e padronizar tecnologias para internet recomenda que as pessoas que queiram construir Ontologias utilizem a linguagem OWL, pois com isso espera-se tornar essa linguagem padrão.

Harmelen (2005) apresenta três sub-linguagens incrementais, que fazem parte da definição da OWL:

- OWL DL: é aplicada por usuários que precisem do máximo de expressividade, com completude (todas as conclusões são garantidas de serem computáveis) e decidibilidade (todas as computações terminarão em um tempo finito) computacional. Ela inclui todas as construções da linguagem OWL, mas estas construções somente podem ser usadas sob certas restrições. A sigla DL possui correspondência com a lógica descritiva (*Description Logics*), uma área de pesquisa que estuda um fragmento particular da lógica de primeira ordem.

- OWL Full: é aplicada por usuários que necessitem do máximo de expressividade e independência sintática de RDF, sem nenhuma garantia computacional. A OWL Full e a OWL DL suportam o mesmo conjunto de construções da linguagem OWL, embora com restrições um pouco diferentes. Enquanto a OWL DL impõe restrições sobre o uso de RDF e requer disjunção de

classes, propriedades, indivíduos e valores de dados, a OWL Full permite misturar OWL com RDF Schema e não requer a disjunção de classes, propriedades, indivíduos e valores de dados. Isto é, uma classe pode ser ao mesmo tempo uma classe e um indivíduo.

- OWL Lite: é uma sub-linguagem da OWL DL que usa somente algumas características da linguagem OWL e possui mais limitações do que OWL DL ou OWL Full.

De acordo com Lima (2005), os elementos básicos de uma Ontologia fundamentada em OWL estão a seguir relacionados e definidos:

1. Classes: provém um mecanismo de abstração para agrupar recursos com características semelhantes, ou seja, uma classe define um grupo de indivíduos que compartilham algumas propriedades.

2. Indivíduos: são instâncias das classes.

3. Propriedades: são relações binárias, que podem ser usadas para estabelecer relacionamentos entre indivíduos ou entre indivíduos e valores de dados. Estes relacionamentos permitem afirmar fatos gerais sobre membros das classes e podem também especificar fatos sobre indivíduos. As propriedades podem ter características, sendo que a OWL suporta os seguintes mecanismos: transitivo, simétrico, funcional, funcional inverso e inverso de.

Para ilustrar o emprego desses conceitos, os mecanismos representativos são:

Transitivo: se a propriedade **subordinado** é transitiva e o indivíduo **a** é subordinado de **b** e o indivíduo **b** é **subordinado** de **c**, então **a** é **subordinado** de **c**.

Simétrico: se a propriedade **casado** é simétrica e o indivíduo **a** é **casado** com o indivíduo **b**, então **b** é **casado** com **a**.

Inverso de: uma propriedade pode ser inversa de outra. Se a propriedade **empregador** é inverso de **empregado** e o indivíduo **a** é empregador de **b**, então **b** é empregado de **a**.

A seguir são apresentadas duas representações de uma Ontologia com objetivo de demonstrar, de maneira simplificada, através de um exemplo prático, como os conceitos anteriormente explicados podem ser utilizados na representação do conhecimento. O exemplo a ser explorado trata-se da organização geográfica de um Estado. A primeira representação é gráfica (Figura 1).

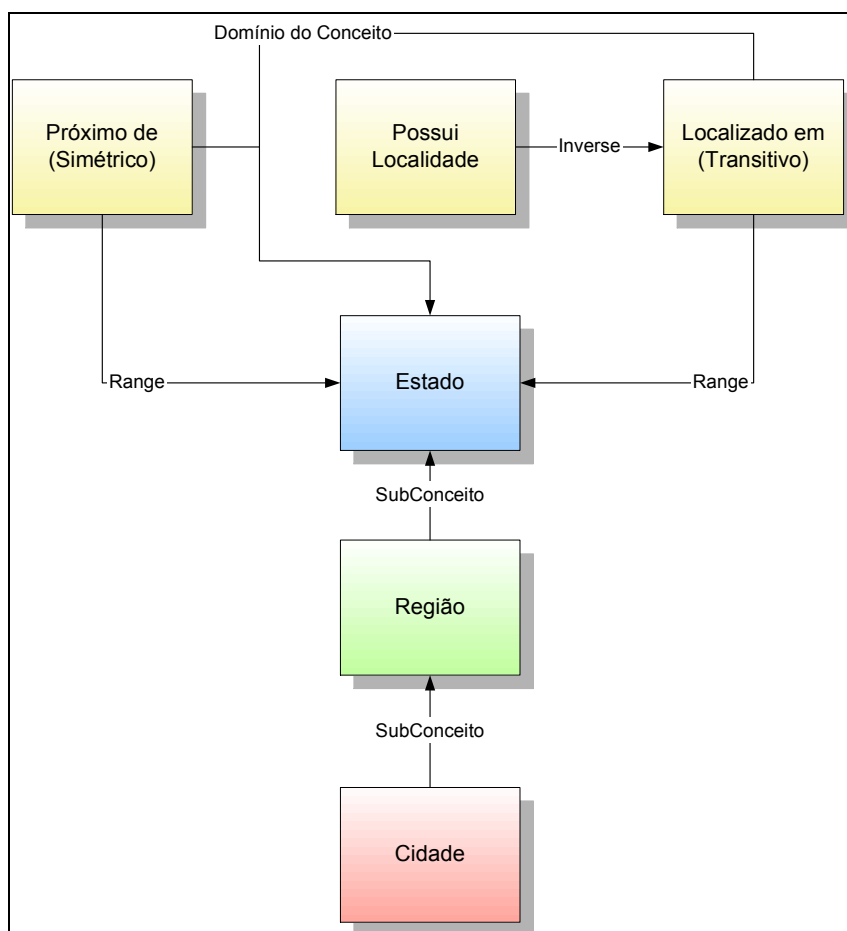


Figura 1: Representação Gráfica da Ontologia

A segunda representação (Figura 2) utiliza a linguagem OWL, a qual pode ser usada por softwares para inferir conhecimento. Essa representação é baseada na linguagem XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF      xml:base="http://geografia.com/onto#"      xmlns="http://geografia.com/onto#"
xmlns:owl="http://www.w3.org/2002/07/owl#"  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology/>
  <owl:TransitiveProperty rdf:ID="LocalizadoEm">
    <rdfs:domain rdf:resource="#Estado"/>
    <rdfs:range rdf:resource="#Estado"/>
  </owl:TransitiveProperty>
  <owl:Class rdf:ID="Regiao">
    <rdfs:subClassOf rdf:resource="#Estado"/>
  </owl:Class>
  <owl:SymmetricProperty rdf:ID="ProximoDe">
    <rdfs:domain rdf:resource="#Regiao"/>
    <rdfs:range rdf:resource="#Estado"/>
  </owl:SymmetricProperty>
  <owl:Class rdf:ID="Estado"/>
  <owl:ObjectProperty rdf:ID="PossuiLocalidade">
    <rdfs:domain rdf:resource="#Regiao"/>
    <owl:inverseOf rdf:resource="#LocalizadoEm"/>
  </owl:ObjectProperty>
  <owl:Class rdf:ID="Cidade">
    <rdfs:subClassOf rdf:resource="#Regiao"/>
  </owl:Class>
</rdf:RDF>
```

Figura 2: Representação Utilizando OWL

Seguindo o modelo ontológico anterior, é possível a criação das instâncias e suas propriedades:

- Santa Catarina é um Estado;
- Santa Catarina possui duas Regiões, sendo elas: Litoral e Interior;

- Xanxerê é uma cidade e está localizada na região Interior;
- Florianópolis é uma cidade e está localizada na região Litoral.
- Chapecó é uma cidade e está localizada na região Interior e é próxima de Xanxerê.

Utilizando a capacidade de inferência da OWL é possível deduzir conhecimentos implícitos, sendo eles apresentados no Quadro 2.

<b>Inferência</b>	<b>Resultado</b>
Cidades em Santa Catarina	Chapecó, Florianópolis e Xanxerê
Cidades próximas a Xanxerê	Chapecó
Localização de Florianópolis	Litoral, Santa Catarina

Quadro 2: Resultado das Inferências

A Figura 3 é uma representação gráfica das instâncias, propriedades e possíveis inferências acima apresentadas, as cores das instâncias devem ser utilizadas como referencial para identificação de sua classe.

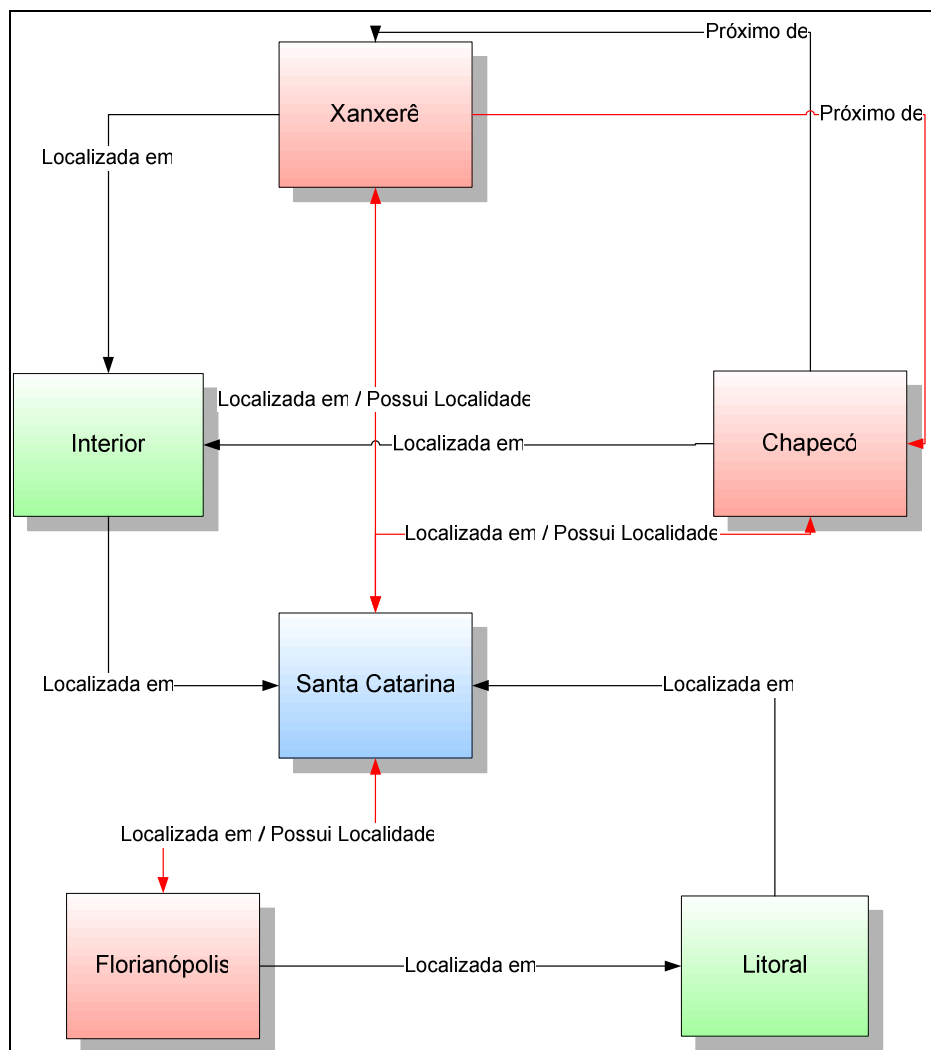


Figura 3: Representação Gráfica das Instâncias e Relações

## 2.2 Inverse Document Frequency

Segundo Robertson (2004), em 1972, Karen Sparck Jones publicou no *Journal of Documentation* um artigo intitulado “*A statistical interpretation of term specificity and its application in retrieval*”. O artigo apresentou um modelo matemático baseado na contagem de número de vezes que um determinado termo aparece em uma determinada coleção de documentos, este modelo veio a ser conhecido como *Inverse Document Frequency* ou IDF e definiu uma fórmula para representar a sua importância.



A idéia baseava-se no pressuposto que um termo que aparece em muitos documentos não é um termo que representa bem um determinado documento, e a medida proposta era uma implementação heurística desse conhecimento.

A medida proposta por Sparck Jones, atribuindo um peso ao termo é essencialmente:

$$\text{IDF}(t_i) = \log\left(\frac{N}{n_i}\right)$$

Figura 4: Fórmula do IDF

A fórmula apresentada na Figura 4 assume que  $N$  é o número de documentos em uma coleção, e o termo  $t_i$  ocorre em  $n_i$  documentos desta base. Salienta-se que “termo” pode ser considerado uma palavra, uma frase ou uma *word stemming*.

O objetivo da utilização do logaritmo é garantir que, conforme a frequência de um termo aumenta, sua importância em relação as frequências menores seja atenuada (Figura 5).

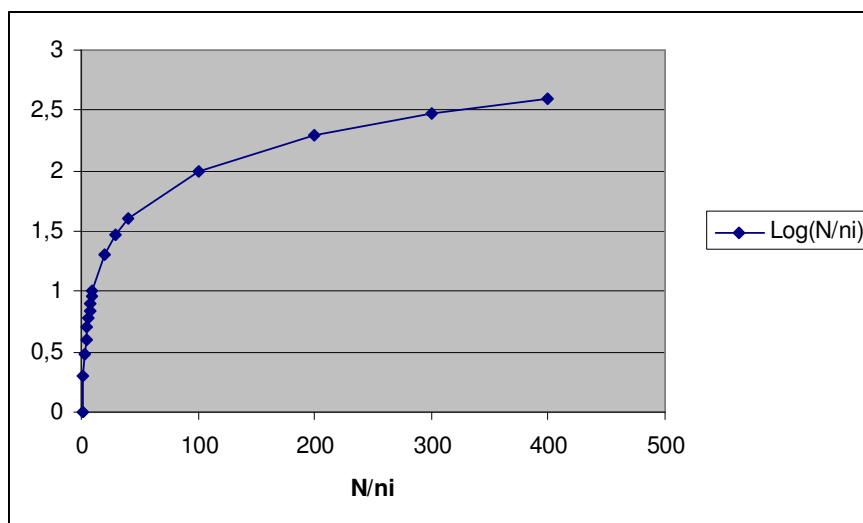


Figura 5: Representação Gráfica da Curva do IDF

De acordo com a tabela a seguir (Tabela 1), um termo com relevância 200 (sem aplicar logaritmo) e um termo com relevância 300 (sem aplicar logaritmo) têm praticamente o mesmo IDF (aplicando-se logaritmo) e pressupõe-se que, depois de um determinado número de ocorrências, o termo perca sua relevância, e a curva que representa essa perda é a logarítmica.

N/ni	Log(N/ni)
1	0,0000000
2	0,3010300
3	0,4771213
4	0,6020600
5	0,6989700
6	0,7781513
7	0,8450980
8	0,9030900
9	0,9542425
10	1,0000000
20	1,3010300
30	1,4771213
40	1,6020600
100	2,0000000
200	2,3010300
300	2,4771213
400	2,6020600

Tabela 1: Representação Numérica da Curva do IDF

Para uma melhor explicação sobre o funcionamento da medida de Sparck, a seguir estão relacionados trechos retirados da Enciclopédia Digital Wikipédia.

Trecho 1: A **Copa do Mundo**, ou Campeonato do Mundo de **Futebol** é um torneio de **futebol** masculino, realizado a cada quatro anos pela FIFA. Começou em 1930, com a vitória da seleção do Uruguai. No primeiro mundial, não havia torneio eliminatório, e os países foram convidados para o torneio. A Itália sagrou-se bicampeã em 1934 e 1938. Nos anos de 1942 e 1946, a Copa não ocorreu devido à Segunda Guerra Mundial. Em 1950, o mundial foi realizado no **Brasil**, que chegou como favorito à sua primeira final de Copa, mas a Celeste Olímpica uruguaia estragou a festa de 200 mil pessoas presentes no Maracanã, então o maior estádio do mundo, vencendo o jogo por 2x1, quando o empate teria sido suficiente ao **Brasil** para conquistar o título. O episódio ficou conhecido como Maracanazo.

Trecho 2: **O Brasil** possui a seleção com mais títulos mundiais, **o** único país pentacampeão e **o** único a ter vencido o torneio fora do seu continente. É também **o** único país a participar de todas as Copas. Seguem-se as seleções tricampeãs da Alemanha e da Itália, as bicampeãs da Argentina e do Uruguai e, por fim, as seleções da Inglaterra e da França, com um único título.

Trecho 3: A **Copa do Mundo** é o segundo maior evento esportivo do mundo, ficando atrás apenas dos Jogos Olímpicos de Verão. É realizada a cada quatro anos, tendo sido sediada pela última vez, em 2002, no Japão e na Coréia do Sul, com **o Brasil** como campeão. A próxima, em 2006, será na Alemanha.

A Tabela 2 apresenta uma contagem do número de ocorrência de determinados termos nos trechos selecionados (Trecho 1, Trecho 2 e Trecho 3). Para este exemplo, uma palavra ou um conjunto de palavras é considerado um termo. Na segunda coluna é apresentado o cálculo do IDF simplificado ( $N/n_i$ ), sem a utilização do *log*, e na terceira coluna o *log* é utilizado na operação.

Termo	quantidade	$N/n_i$	$\log(N/n_i)$ - IDF
Futebol	1	3	0,477121255
Copa do Mundo	2	1,5	0,176091259
Brasil	4	0,75	-0,124938737
o	12	0,25	-0,60206

Tabela 2: Termos e Respective Pesos IDF

Aplicando este conceito ao caso em análise, é possível afirmar que o termo **futebol** representa melhor o Trecho 1 do que o termo **o**, pois aparece apenas neste documento. Os valores calculados a partir da fórmula base representam esta afirmação. Já o termo **Copa do Mundo** é mais útil na classificação dos documentos que o termo **o**, pois aparece em apenas dois dos trechos: Trecho 1 e Trecho 3.

Apesar de simples, as idéias de Spark são úteis e largamente utilizadas em conjunto com outras técnicas no que se refere a busca de documentos.

### 2.3 StopWords

Apenas uma pequena parte das palavras contidas em um texto reflete a informação contida no mesmo. Analisando a língua inglesa é possível afirmar que palavras como *it*, *and* e *to* podem ser encontradas em praticamente qualquer sentença. Portanto, são termos que são extremamente pobres no que se refere a busca por documentos. Entretanto, representam a maioria dos termos dos documentos, estas palavras são conhecidas como *StopWords* (RACHEL, 2004).

A remoção de *StopWords* é uma tarefa existente em praticamente todos os sistemas de recuperação e informação textual. Uma lista de palavras consideradas *StopWords* pode ser construída analisando os textos utilizados como base para a busca ou fazendo uma análise do idioma utilizado. A título de exemplo, relaciona-se uma lista de *StopWords* (Quadro 3).

de	é	As	nos	eu	depois	eles
a	com	Dos	já	também	sem	estão
o	não	como	está	só	mesmo	você
que	uma	mas	seu	pelo	aos	tinha
e	os	Foi	sua	pela	ter	foram
do	no	Ao	ou	até	seus	essa
da	se	Ele	ser	isso	quem	num
em	na	das	quando	ela	nas	nem
um	por	tem	muito	entre	me	suas
para	mais	à	há	era	esse	meu
qual	essas	tu	minhas	nossa	estes	isto
será	esses	te	teu	nossos	estas	aquilo
nós	pelas	vocês	tua	nossas	aquele	havia
tenho	este	vos	teus	dela	aquela	seja
lhe	fosse	lhes	tuas	delas	aqueles	pelos
deles	dele	meus	nosso	esta	aquelas	elas
numa	têm	minha	às			

Quadro 3: Lista de Exemplos de *StopWords*

## 2.4 Stemming

Conforme Dennis (2000), no contexto da Recuperação da Informação, *Stemming* refere-se ao processo de remoção dos prefixos e sufixos das palavras. *Stemming* é usado para reconhecer os padrões de formação das palavras com a finalidade de recuperar a informação. Como um simples exemplo, considere a busca por um documento intitulado "Como Escrever". Se o usuário digitar "Escrevendo" o sistema não conseguirá encontrar nenhum documento, no entanto, se a entrada de dados for *stemmizada*, Escrever tornar-se-á "Escrev" – denominado *stem* – e o documento "Como Escrever" será apresentado ao usuário.

Diversos algoritmos foram desenvolvidos para este fim, cita-se Paice/Husk, Lovins, Dawson, Krovetz. Porém, o algoritmo mais comumente utilizado é o algoritmo Porter, escrito em 1980 e publicado no artigo "*An algorithm for suffix stripping*".

Para Porter, uma versão modificada do algoritmo *Porter* denominado *Porter2* ou *SnowBall*, trata-se de uma versão melhorada do algoritmo original e deve ser utilizada para se obter melhores resultados.

O algoritmo *SnowBall* foi modificado para funcionar em diversos idiomas inclusive o Português. Entretanto, algoritmos desenvolvidos especificadamente para um idioma tendem a apresentar melhores resultados, como é o caso do algoritmo publicado por Orengo (2001) e que será utilizado nesta pesquisa por apresentar melhores resultados que o algoritmo Porter para a Língua Portuguesa.

Segundo Orengo (2001), os resultados apresentados pelo algoritmo desenvolvido especificadamente para a língua portuguesa apresentaram um menor valor referente ao erro de *understemming* (redução de palavras com significados

iguais para *stemmings* diferentes) e *overstemming* (redução de palavras com significados diferentes para o mesmo *stemming*). Assim, conclui-se que o algoritmo de Oregon é mais eficiente que o algoritmo de Porter.

O algoritmo de Oregon é composto por 8 (oito) etapas, que estão apresentadas na Figura 6.

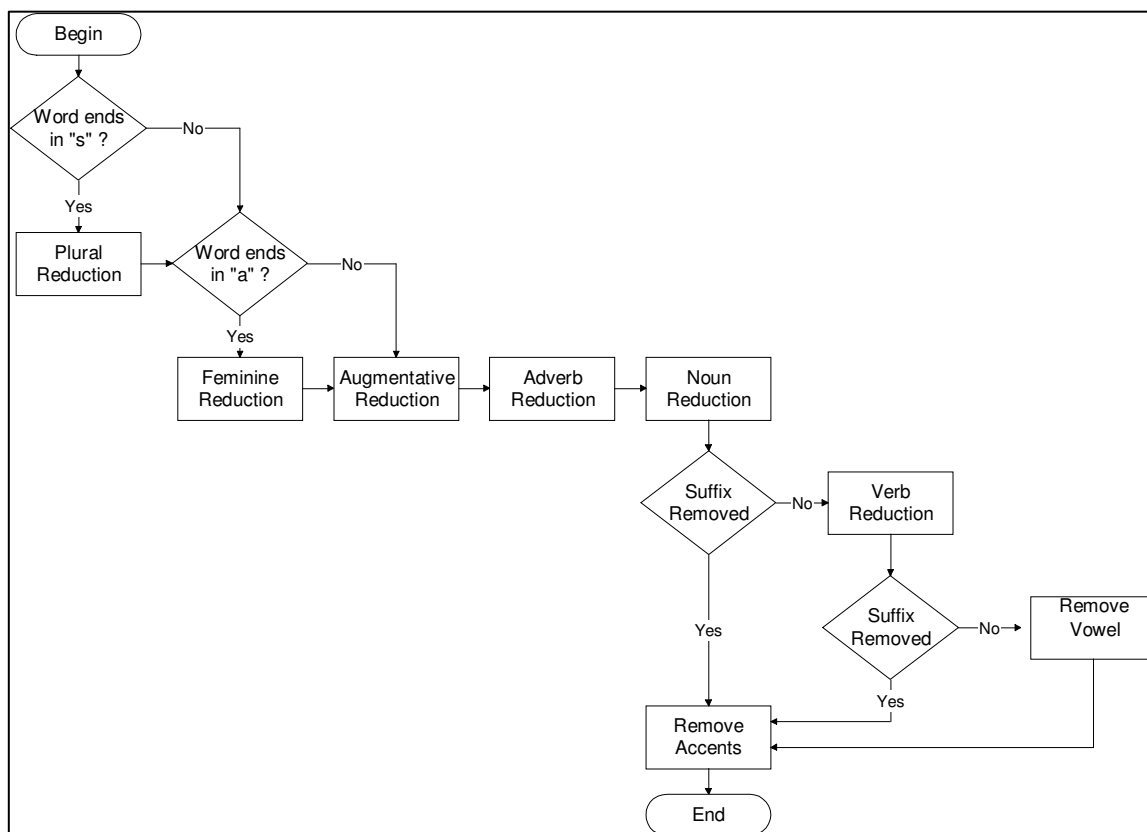


Figura 6: Etapas do Algoritmo de Orengo/Stemming para Língua Portuguesa

Cada etapa do algoritmo é composta por um grupo de regras e é executada conforme o fluxo mostrado da Figura 6, sendo que apenas uma regra pode ser executada por vez. O algoritmo de Orengo possui 199 regras e cada regra é composta por 4 fatores, a saber:

- O sufixo a ser removido;
- O tamanho mínimo do *stem*: este fator impede que determinados sufixos sejam removidos;

- Um sufixo que deve substituir o sufixo atual da palavra;
- Uma lista de excessões, palavras que apesar de cumprirem as regras não devem ser afetadas.

Para melhor compreensão do mecanismo das regras, segue o exemplo (Figura 7):

"inho", 3, "", {"caminho", "carinho", "cominho", "golfinho", "padrinho", "sobrinho", "vizinho"}

Figura 7: Exemplo de Regra do Algoritmo de Stemming da Língua Portuguesa

Onde “inho” é o sufixo a ser analisado, 3 é o tamanho mínimo do *stem*, o que previne que palavras como “linho” sejam *stemmizadas*, e uma lista de excessões onde esta regra não pode ser aplicada (carinho, caminho, etc.).

As etapas do algoritmo estão descritas a seguir:

- Redução do Plural: com raras excessões, a remoção do plural na língua portuguesa consiste na remoção da letra “s”;
- Redução do Feminino: todos os Substantivos e Adjetivos na língua portuguesa possuem uma versão masculina. Esta etapa consiste em transformar a forma feminina na forma correspondente masculina;
- Redução dos Advérbios: esta etapa consiste em analisar palavras finalizadas em “mente”, como nem todas as palavras terminadas neste sufixo representam advérbios, existe uma lista de excessões;
- Aumentativo/Diminutivo: a língua portuguesa apresenta uma variação muito grande de sufixos utilizados nestas formas, entretanto, apenas os mais comuns são utilizados para evitar o *overstemming*;

- Redução dos Substantivos: esta etapa testa as palavras, procurando por 61 sufixos utilizados em substantivos, se este sufixo é removido, as etapas 6 e 7 são ignoradas;
- Redução dos Verbos: a língua portuguesa é muito rica em termos de formas verbais, enquanto a língua inglesa possui apenas quatro variações, a língua portuguesa contém cinquenta diferentes formas;
- Remoção de Vogais: esta etapa consiste em remover as letras “a” e/ou “o” no final das palavras que não tenham sido *stemmizadas* pelos passos 5 e 6;
- Remoção de Acentos: a remoção de acentos é importante, já que existem palavras em que as mesmas regras se aplicam a versões acentuadas e não acentuadas (por exemplo, psicólogo e psicologia).

Ainda segundo Orengo (2001), os fatores que tornam o *stemming* da língua portuguesa um processo complexo são: a quantidade de excessões nas regras; a quantidade de palavras com mais de um significado; a quantidade de verbos irregulares; a quantidade de palavras onde a raiz da mesma é alterada; e a dificuldade em reconhecer nomes próprios.

O Quadro 4 apresenta alguns exemplos de palavras *stemmizadas*.

Termo	Termo após Alg
bobalhões	Bobalhõ
bocadinho	Bocadinh
quintuplicou	Quintuplic
quimioterápicos	Quimioteráp
quilométricas	Quilométr
bocaiúva	Bocaiúv
quiosque	Quiosqu

Quadro 4: Palavras *Stemmizadas*



## 2.5 Term Extraction

A Extração de Termos (*Term Extraction*) é um importante problema a ser estudado, no que diz respeito ao processamento de linguagem natural. Seu objetivo é a extração de coleções de palavras que representem o significado de um texto sendo que a base semântica de um texto pode ser representada por estes termos. As técnicas de *Term Extraction* podem ser aplicadas em ferramentas como: máquinas de tradução, ferramentas de indexação de documentos, construtoras de bases de conhecimento e sistemas de Recuperação da Informação (PANTEL, 2006).

Segundo Milios (2006), o estado da arte dessas técnicas é figurado, atualmente, pelos algoritmos C-value/NC-value desenvolvidos por Frantziy, Ananiadouy e Mimaz publicado no *International Journal on Digital Libraries Manuscript*.

Durante a evolução da técnica modelos com diferentes abordagens como as criadas por Dagan e Church, Daille, Justeson e Katz, Enguehard e Pantera utilizavam apenas informação estatística. A técnica C-value/NC-value apresenta uma nova visão sobre o tema, combinando técnicas estatísticas com técnicas lingüísticas (FRANTZIY, 2000).

C-value é uma técnica de extração estatística eficiente. A técnica NC-value foi desenvolvida para incorporar informação contextual às informações já encontradas através da C-value. Seu algoritmo utiliza-se de informações estatísticas e lingüísticas.

Basicamente, o método C-value tem como entrada um texto e como saída uma lista de termos candidatos ordenados pelo valor C-value, também denominado

*termhood*. A lista resultante da técnica deve ser analisada por especialista de domínio (domínio do texto utilizado). Não existe a necessidade de analisar todos os termos, no entanto, a eficiência do algoritmo está diretamente ligada à quantidade de termos analisados pelo especialista (indivíduo que possui conhecimento acerca dos termos técnicos utilizados em um determinado domínio do conhecimento).

A parte lingüística do algoritmo está baseada em uma lista de *StopWords* e um filtro lingüístico que analisa o tipo (verbo, pronome, etc.) dos termos a serem extraídos. Salienta-se que um termo pode ser composto de uma ou mais palavras. A parte lingüística consiste em três etapas:

1. Marcar o texto com os tipos de termos;
2. Aplicar um filtro lingüístico que remove os tipos de termos indesejáveis;
3. Excluir os termos pertencentes a uma *StopList* também conhecida por *StopWords*.

A parte estatística consiste na análise de quatro informações relativas a cada termo e está representada na fórmula a seguir (Figura 8).

1. A freqüência que o termo aparece no documento;
2. A freqüência que o termo aparece em conjunto com outro termo do documento;
3. O número de termos a serem selecionados;
4. A quantidade de caracteres do termo.

$$C\text{-value}(a) = \begin{cases} \log_2 |a| \cdot f(a) & a \text{ is not nested,} \\ \log_2 |a| \left( f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) \right) & \text{otherwise} \end{cases}$$

Figura 8: Fórmula C-Value

De acordo com Frantziy (2000), o uso de informações estatísticas que vão além da simples frequência do termo da extração, em conjunto com o uso de informações lingüísticas, aumenta significativamente a eficiência.

Salienta-se que para está Tese, a técnica de Term Extration utilizada baseia-se em técnicas estatísticas, pois optou-se por utilizar a incorporação de técnicas lingüísticas associadas a ontologia.

## 2.6 *Query Expansion*

Conforme Billerbeck (2006) as máquinas de busca são os principais mecanismos utilizados para procurar documentos na Internet. Essas ferramentas utilizam mecanismos de Recuperação da Informação para comparar *queries*, expressas em uma série de palavras, com os documentos e julgar quais deles são melhores para responder a uma determinada pergunta do usuário.

Quando as *queries* são bem formuladas, consistindo em palavras-chave de um tópico específico, as quais juntas demonstram a informação necessária com um nível baixo de ambigüidade, as máquinas de busca conseguem obter documentos que refletem as palavras. Todavia, a maioria das *queries* não são bem formuladas, elas são ambíguas, não precisas o suficiente, ou usam os termos específicos para um determinado contexto. Em geral, as *queries* imputadas nas máquinas de busca é formada por duas a três palavras. Esse tipo de entrada acaba trazendo como resultado uma quantidade grande de documentos, o que dificulta a análise.

Uma variedade de técnicas para aumentar a eficiência desses mecanismos é usada. Uma delas é a *Query Expansion*. É possível afirmar que existem dois grupos

básicos de técnicas utilizadas para expansão de *queries* (Grootjen, 2004), abaixo descritas:

**User FeedBack Relevance:** esta é provavelmente a técnica mais comum de reformulação de *queries*. Esta técnica requisita que o usuário atribua relevância a um conjunto de documentos trazidos através de uma busca inicial. Experimentos recentes têm demonstrado uma melhora significativa no resultado das buscas, trabalhando em bases com poucos documentos. O modelo matemático é simples de implementar, a única dificuldade com a técnica é persuadir o usuário a atribuir relevância a documentos, o que é um trabalho tedioso.

**Global Query Expansion:** trata-se de adicionar palavras (sinônimos ou palavras relacionadas) à *Query* original. Para fazer isso, utiliza-se um *thesaurus* ou outro tipo de fonte de dados. *Thesaurus* são freqüentemente utilizados em sistemas de Recuperação da Informação como um mecanismo para reconhecer expressões sinônimas e entidades lingüísticas que são semanticamente similares, mas superficialmente distintas. Diferente da técnica de relevância através de *feedback*, não é necessário analisar a base dos textos.

Segundo Mandala (2007), técnicas de expansão de *queries* utilizando-se *thesaurus* são alvos de pesquisas por quatro décadas e uma quantidade enorme de métodos foi desenvolvida. Segundo o mesmo autor, os vários métodos podem ser enquadrados em três grupos básicos: *Hand-crafed thesaurus based*, *Co-occurrence-based automatically constructed thesaurus based* e *Head-modi er-based automatically constructed thesaurus based*.

A *Query Expansion* baseada em *Hand-Crafed Thesaurus* somente tem sucesso se o domínio do *thesaurus* é o mesmo domínio das bases textuais. De acordo com os experimentos da *Text Retrieval Conference* – TREC – o uso de

*thesaurus* genéricos não tem tido muito êxito. Já os modelos que utilizam *thesaurus* construídos automaticamente (são construídos baseados em uma coleção de textos, sem intervenção humana) têm obtido pequenas taxas de eficiência, em torno de 20%. (Mandala, 2007).

Analisando os *proceedings* (TREC) é possível assegurar que pesquisas na área de expansão de *queries* estão longe de se esgotar e diversas técnicas que utilizam cruzamento de métodos já criados foram especificadas com o objetivo de promover melhoria na eficiência das buscas, como exemplo citam-se algumas pesquisas desenvolvidas, que utilizam *Query Expansion*:

A) UMass Robust 2005. Using Mixtures of Relevance Models for Query Expansion: utiliza como base as técnicas de aproximação de termos e também técnicas de pseudo relevância através de *feedback*.

B) Symbol-Based Query Expansion Experiments: estuda a eficiência de algoritmos de expansão de *Queries* baseados no *feedback* dos usuários.

C) The Effects of Primary Keys, Bigram Phrases and Query Expansion on Retrieval Performance: procura expandir as *Queries* através da análise estatística dos termos contidos na base textual utilizada.

D) Concept-Based Query Expansion and Bayes Classification: utiliza uma máquina de indexação de conceitos denominada Collexis para expandir as *queries*.

Como um dos propósitos desta tese é favorecer a busca de documentos textuais levando-se em consideração a semântica do conteúdo dos textos, o uso de expansão de *queries* utilizando-se termos relacionados armazenados na forma de *thesaurus* ou Ontologias foi considerado essencial. A abordagem prática da utilização desse modelo será apresentada no Capítulo 3.

Destaca-se que as técnicas que utilizam a relação entre documentos (*links*) para expandir *queries* (largamente utilizadas na pesquisa de documentos na Internet) não foram analisadas, visto que a finalidade desta pesquisa é utilizar bases textuais sem *hiperlinks* entre os documentos.

## 2.7 Text Retrieval Conference (TREC)

Conforme dito anteriormente, a *Text Retrieval Conference* – TREC – co-patrocinada pelo Instituto Nacional de Padrões e Tecnologia – NIST – e pelo Departamento de Defesa dos E.U.A. se iniciou em 1992, como parte do programa TIPSTER Text. Seu propósito é dar suporte às pesquisas da comunidade de Recuperação da Informação, proporcionando a infra-estrutura necessária para avaliação de metodologias e tendo como objetivos básicos os seguintes (TREC, 2006):

- Encorajar a pesquisa sobre Recuperação da Informação em grandes bases de dados;
- Melhorar a comunicação entre a indústria, a academia e o governo, criando um fórum aberto para troca de informações sobre pesquisas;
- Aumentar a velocidade de transferência de tecnologia entre laboratórios de pesquisa e produtos comerciais, demonstrando melhoras substanciais na Recuperação da Informação em problemas do mundo real;
- Aperfeiçoar a disponibilidade de técnicas de avaliação para serem utilizadas pela academia e indústria, incluindo aqui o desenvolvimento de novas técnicas, que possam ser aplicadas aos sistemas atuais.

A TREC é coordenada por um comitê composto de representantes do governo, da indústria e da academia. Todos os anos o NIST providencia um grupo de documentos e questões. Os participantes executam seus softwares e entregam ao NIST uma lista de documentos recuperados. O NIST é responsável por analisar os resultados individuais e julgá-los. O ciclo termina em um *Workshop*, onde os participantes compartilham suas experiências.

Cada *Workshop* da TREC consiste em um grupo de *tracks*, que são determinadas áreas de estudo, em que tarefas de Recuperação da Informação são definidas. Novas tarefas são incluídas quando ocorrem necessidades comerciais e acadêmicas a suprir. A seguir, citam-se as *tracks* já estudadas:

- Blog Track: tem o objetivo de explorar informações textuais disponibilizadas sobre a forma de *blogs*;
- Enterprise Track: seu propósito é satisfazer os usuários que têm por finalidade procurar informações dentro das organizações;
- Genomics Track: tem como objetivo estudar tarefas de recuperação em domínios específicos, mais particularmente na área de genomas;
- Legal Track: o propósito desta *track* é desenvolver tecnologia que auxilie advogados a descobrir informações em documentos digitais da área de direito;
- Spam Track: tem por finalidade propor novas linhas de ação, no que diz respeito ao filtro de *e-mails* de *spam*;
- Terabyte Track: estuda como avaliar a qualidade de ferramentas de recuperação em bases de dados volumosas;

- Cross-Language Track: investiga a possibilidade de se procurar em bases textuais, onde o idioma é diferente do utilizado para definir o que se está buscando;
- Filtering Track: nesta tarefa, a base textual é conhecida (estável);
- Hard Track: a principal característica desta *track* é a recuperação de documentos com alto grau de exatidão;
- Interactive Track: estuda a recuperação da informação, onde os algoritmos usam a interação do usuário no processo;
- Novelty Track: investiga sistemas com a habilidade de descobrir informações;
- Robust Retrieval Track: inclui a recuperação tradicional de documentos;
- Video Track: estuda a segmentação automática, indexação e recuperação de vídeos digitais;
- Web Track: tem por finalidade a procura de documentos na Internet.

As técnicas apresentadas anteriormente (Ontologias, IDF, *StopWords*, *Stemming*, *Term Extration* e *Query Expansion*) são ferramentas básicas para a construção dos modelos computacionais utilizados na TREC.

O Quadro 5 apresenta um levantamento realizado na base de dados da TREC, contendo artigos que utilizam em seu desenvolvimento as mesmas técnicas usadas nesta pesquisa. Embora não seja a finalidade desta tese utilizar as técnicas de indexação e interação com o usuário, as mesmas são explicitadas com o intuito de demonstrar sua relevância.



Os artigos que foram avaliados pertencem a todas as *tracks*, contudo, estão relacionados apenas às que apresentam características técnicas importantes para o contexto desta pesquisa.

	<b>Título</b>	<b>Autores</b>	<b>Instituição</b>	<b>Ano</b>
1	Query Improvement in Information Retrieval Using Genetic Algorithms	J. Yang, R. Korfhage, B. Rasmussen	University of Pittsburgh	1992
2	TIPSTER Panel—HNC's MatchPlus System	S. Gallant, R. Recht-Nielson, W. Caid, K. Qing, J. Carleton, D. Sudbeck	HNC, Inc.	1992
3	Site Report for the Text REtrieval Conference	P. Nelson	ConQuest Software, Inc.	1992
4	Vector Expansion in a Large Collection	E. Voorhees, Y-W. Hou	Siemens Corporate Research, Inc.	1992
5	Okapi at TREC-2	S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gattford	City University, London	1993
6	Recent Developments in Natural Language Text Retrieval	T. Strzalkowski, J. Carballo	New York University	1993
7	Design and Evaluation of the CLARIT-TREC-2 System	D. Evans, R. Lefferts	Carnegie Mellon University and CLARIT Corporation	1993
8	On Expanding Query Vectors with Lexically Related Words	E. Voorhees	Siemens Corporate Research, Inc.	1993
9	UCLA-Okapi at TREC-2: Query Expansion Experiments	E. Efthimiadis, P. Biron	University of California, Los Angeles	1993
10	Incorporating Semantics Within a Connectionist Model and a Vector Processing Model	R. Boyd, J. Driscoll	University of Central Florida	1993
11	Document Retrieval and Routing Using the INQUERY System	J. Broglio, J. P. Callan, W. B. Croft, D. W. Nachbar	University of Massachusetts	1994
12	Natural Language Information Retrieval	T. Strzalkowski, J. P. Carballo, M. Marinescu	New York University	1994

13	Comparison of Fragmentation Schemes for Document Retrieval	R. Wilkinson, J. Zobel	CITRI, Royal Melbourne Institute of Technology	1994
14	Okapi at TREC-3	S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford	City University	1994
15	Query Expansion/Reduction and its Impact on Retrieval Effectiveness	X. A. Lu, R. B. Keefer	Data Central, Inc.	1994
16	Searching For Meaning With The Help Of A PADRE	D. Hawking, P. Thistlewaite	Australian National University	1994
17	Research in Automatic Profile Creation and Relevance Ranking with LMDS	J. Yochum	Logicon, Inc.	1994
18	The FDF Query Generation Workbench	K-I. Yu, P. Scheibe, F. Nordby	Paracel, Inc.	1994
19	Logistic Regression at TREC4: Probabilistic Retrieval from Full Text Document Collections	Fredric C. Gey, Aitao Chen, Jianzhang He and Jason Meggs	University of California, Berkeley	1995
20	Okapi at TREC-4	S. E. Robertson, S. Walker, M. M. Beaulieu, M. Gatford, A. Payne	City University, London	1995
21	TREC-4 Ad-Hoc, Routing Retrieval and Filtering Experiments using PIRCS	K.L. Kwok and L. Grunfeld	Queens College, CUNY	1995
22	TREC-4 Experiments at Dublin City University: Thresholding Posting Lists, Query Expansion with WordNet and POS Tagging of Spanish	A. F. Smeaton, F. Kelledy and R. O'Donnell	Dublin City University	1995
23	INQUERY at TREC-5	J. Allan, J. Callan, B. Croft, L. Ballesteros, J. Broglio, J. Xu, H. Shu	University of Massachusetts, Amherst	1996
24	Term importance, Boolean conjunct training, negative terms, and foreign language retrieval: probabilistic algorithms at TREC-5	F. C. Gey, A. Chen, J. He, L. Xu, and J. Meggs	University of California, Berkeley	1996
25	Berkeley Chinese Information Retrieval at TREC-5: Technical	J. He, J. Xu, A. Chen, J. Meggs, F.C. Gey	University of California, Berkeley	1996

	Report			
26	OCR Correction and Query Expansion for Retrieval on OCR Data—CLARIT TREC-5 Confusion Track Report	C. Zhai, X. Tong	Carnegie Mellon University	1996
27	ANU/ACSys TREC-5 Experiments	D. Hawking, P. Thistlewaite, P. Bailey	Australian National University	1996
28	MDS TREC6 Report	M. Fuller, M. Kaszkiel, C.L. Ng, P. Vines, R. Wilkinson, J. Zobel	RMIT	1997
29	Natural Language Information Retrieval TREC-6 Report	T. Strzalkowski, F. Lin, J. Perez-Carballo	GE Corporate Research & Development, Rutgers University	1997
30	Conceptual Indexing Using Thematic Representation of Texts	B.V. Dobrov, N.V. Loukachevitch, T.N. Yudina	Center for Information Research, Russia	1997
31	CSIRO Routing and Ad-Hoc Experiments at TREC-6	A. Kosmynin	CSIRO	1997
32	TREC-6 Ad-Hoc Retrieval	M. Franz, S. Roukos	IBM T.J. Watson Research Center	1997
33	Query Term Expansion based on Paragraphs of the Relevant Documents	K. Ishikawa, K. Satoh, A. Okumura	C&C Media Research Labs. NEC Corporation	1997
34	Ad hoc and Multilingual Information Retrieval at IBM	M. Franz, J.S. McCarley, S. Roukos	IBM T.J. Watson Research Center	1998
35	INQUERY and TREC-7	J. Allan, J. Callan, M. Sanderson, J. Xu, S. Wegmann	University of Massachusetts, Dragon Systems, Inc.	1998
36	Natural Language Information Retrieval: TREC-7 Report	T. Strzalkowski, G. Stein, G. Bowden Wise, J. Perez-Carballo, P. Tapananinen, T. Jarvinen, A. Voutilainen, J. Karlgren	GE Research & Development, Rutgers University, University of Helsinki, University of Helsinki	1998
37	Twenty-One at TREC7: Ad-hoc and Cross-Language track	D. Hiemstra, W. Kraaij	University of Twente, CTIT, TNO-TPD	1998
38	ACSys TREC-7 Experiments	D. Hawking, N. Craswell, P.	CSIRO Mathematics and Information Sciences,	1998

		Thistlewaite	Australian National University	
39	Information term selection for automatic query expansion	C. Carpineto, G. Romano, R. De Mori	Fondazione Ugo Bordoni, Rome, University of Avignon	1998
40	Document Retrieval Using The MPS Information Server	F. Schiettecatte	FS Consulting, Inc	1998
41	Fujitsu Laboratories TREC7 Report	I. Namba, N. Igata, H. Horai, K. Nitta, K. Matsui	Fujitsu Laboratories Ltd.	1998
42	TREC-7 Experiments: Query Expansion Method Based on Word Contribution	K. Hoashi, K. Matsumoto, N. Inoue, K. Hashimoto	KDD R&D Laboratories, Inc.	1998
43	Query Expansion and Classification of Retrieved Documents	C. de Loupy, P. Bellot, M. El-Bèze, P.-F. Marteau	Laboratoire d'Infomatique d'Avignon (LIA), Bertin & Cie	1998
44	Ad Hoc Retrieval Experiments Using WordNet and Automatically Constructed Thesauri	R. Mandala, T. Tokunaga, H. Tanaka, A. Okumura, K. Satoh	NEC Corporation and Tokyo Institute of Technology	1998
45	NTT DATA at TREC-7: system approach for ad-hoc and filtering	H. Nakajima, T. Takaki, T. Hirao, A. Kitauchi	NTT DATA Corporation	1998
46	A Two-Stage Retrieval Model for the TREC-7 Ad Hoc Task	D.-H. Shin, B.-T. Zhang	Seoul National University	1998
47	The Weaver System for Document Retrieval	A. Berger, J. Lafferty	Carnegie Mellon University	1999
48	Fujitsu Laboratories TREC8 Report - Ad hoc, Small Web, and Large Web Track	I. Namba, N. Igata	Fujitsu Laboratories Ltd.	1999
49	Twenty-One at TREC-8: using Language Technology for Information Retrieval	W. Kraaij, R. Pohlmann, D. Hiemstra	TNO-TPD, University of Twente, CTIT	1999
50	TREC-8 Automatic Ad-Hoc Experiments at Fondazione Ugo Bordoni	C. Carpineto, G. Romano	Fondazione Ugo Bordoni	1999
51	Natural Language Information Retrieval: TREC-8 Report	T. Strzalkowski, J. Perez-Carballo, J. Karlgren, A. Hulth, P. Tapanainen, T. Lahtinen	GE Research & Development, Rutgers University, Swedish Institute of Computer Science, Stockholm	1999

			University, Conexor OY, Helsinki	
52	Structuring and expanding queries in the probabilistic model	O. Yasushi, M. Hiroko, N. Masumi, H. Sakiko	RICOH Co., Ltd.	1999
53	TREC-8 Experiments at SUNY Buffalo	B. Han, R. Nagarajan, R. Srihari, M. Srikanth	State University of New York at Buffalo	1999
54	Structuring and Expanding Queries in the Probabilistic Model	Y. Ogawa, H. Mano, M. Narita, S. Honma	RICOH Co., Ltd	2000
55	Question Answering in Webclopedia	E. Hovy, L. Gerber, U. Hermjakob, M. Junk, C-Y Lin	University of Southern California	2000
56	The LIMSI SDR System for TREC-9	J.-L. Gauvain, L. Lamel, C. Barras, G. Adda, Y. de Kercardio	LIMSI-CNRS	2000
57	Mercure at trec9: Web and Filtering tasks	M. Abchiche, M. Boughanem, T. Dkaki, J. Mothe, C. Soule Dupuy, M. Tmar	IRIT-SIG	2000
58	Question Answering Considering Semantic Categories and Co-Occurrence Density	S-M Kim, D-H Baek, S-B Kim, H-C Rim	Korea University	2000
59	Question Answering, Relevance Feedback and Summarisation	N. Alexander, C. Brown, J. Jose, I. Ruthven, A. Tombros	University of Glasgow	2000
60	TREC-10 Experiments at CAS-ICT: Filtering, Web e QA	B. Wang, H. Xu, Z. Yang, Y. Liu, X. Cheng, D. Bu, S. Bai	Chinese Academy of Sciences	2001
61	IBM's Statistical Question Answering System	A. Ittycheriah, M. Franz, S. Roukos	IBM T.J. Watson Research Center	2001
62	TREC-10 Web Track Experiments at MSRA	J. Gao, S. Walker, S. Robertson, G. Cao, H. He, M. Zhang, J-Y Nie	Microsoft Research, Tianjin Univ, Tsinghua Univ, Université de Montréal	2001
63	RICOH at TREC-10: Web Track Ad-hoc Task	H. Itoh, H. Mano, Y. Ogawa	RICOH Co., Ltd	2001
64	Aggressive Morphology and Lexical Relations for Query Expansion	W.A. Woods, S. Green, P. Martin, A. Houston	Sun Microsystems Labs	2001
65	Using Hierarchical Clustering and Summarisation Approaches for	R. Osdin, I. Ounis, R.W. White	University of Glasgow	2002

	Web Retrieval: Glasgow at the TREC 2002 Interactive Track			
66	Augmenting and Limiting Search Queries	E.G. Toms, L. Freund, C. Li	University of Toronto	2002
67	Concept Extraction and Synonymy Management for Biomedical Information Retrieval	C. Crangle, A. Zbyslaw, M. Cherry, E. Hong	ConverSpeech LLC, Stanford University	2004
68	Identifying Relevant Full-Text Articles for GO Annotation Without MeSH Terms	C. Lee, W.-J. Hou, H.-H. Chen	National Taiwan University	2004
69	Structural Term Extraction for Expansion of Template-Based Genomic Queries	F. Camous, S. Blott, C. Gurrin, G.J.F. Jones, A.F. Smeaton	Dublin City University	2005
70	Biomedical Document Triage: Automatic Classification Exploiting Category Specific Knowledge	L.V. Subramaniam, S. Mukherjea, D. Punjani	IBM India Research Lab, Indian Institute of Technology	2005
71	Retrieval of Biomedical Documents by Prioritizing Key Phrases	K.H.-Y. Lin, W.-J. Hou, H.-H. Chen	National Taiwan University	2005
72	Identifying Relevant Full-Text Articles for Database Curation	C. Lee, W.-J. Hou, H.-H. Chen	National Taiwan University	2005

Quadro 5: Artigos, Autores e Universidades Relevantes da TREC

O Quadro 6 apresenta a intersecção das técnicas utilizadas nos artigos apresentados no Quadro 5 e as técnicas usadas nesta pesquisa. A primeira coluna equivale à primeira coluna do quadro anterior, as colunas seguintes referem-se às técnicas presentes em cada artigo analisado e a última coluna traz observações acerca dos artigos estudados.

Artigo	Query Expansion	Term Extration	IDF	Stemming	StopWords	Ontology/Tessaurus	User Feedback	Index	Observações
1	x			x	x		x		Utiliza técnicas de Algoritmos Genéticos
2	x				x	x	x	x	Utiliza Redes Neurais
3	x				x	x		x	Processamento de Linguagem Natural
4	x		x						
5	x					x			Combinação de Querys utilizando redes neurais
6	x	x		x				x	Term Extration é usado apenas na Indexação, não na expansão de query
7	x	x				x		x	Usa o termo Query Augmentation e um tipo de FeedBack Automático. Term Extration é utilizado para indexação
8	x					x			Apresenta um modelo de expansão de query manual utilizando-se o WordNet
9	x		x		x	x		x	
10						x		x	Utiliza um thesaurus genérico denominado Roget's Thesaurus o qual disponibiliza pesos a termos correlatos
11	x							x	Expande as queries utilizando-se análise linguística em conjunto com a base de textos indexada
12	x	x				x		x	Criação de Ontologias Automáticas, O Term Extration é utilizado na indexação. Constroi Querys através de frases (Linguagem Natural)
13		x	x	x				x	Selecionam partes do texto para indexar o que definem como Fragmentos
14	x		x					x	Utiliza FeedBack do usuário para expandir as queries
15	x						x		Compara métodos de expansão de queries (automáticos, manuais e mistos)
16		x	x						Utiliza técnicas estatísticas para o que chama de Generation of Automatic Queries, não utiliza Ontologias ou Tessaurus
17		x	x	x	x				
18		x	x	x	x			x	Query generation utilizando-se métodos estatísticos
19	x			x	x		x	x	Query expansion utilizando-se métodos estatísticos e feedback do usuário
20	x			x	x				Query expansion utilizando-se métodos estatísticos e feedback do usuário
21	x						x	x	Query expansion através de feedback do usuário
22	x					x			Query expansion utilizando-se Ontologia, entretando os pesos não levam em conta a relação semântica, apenas a distância
23	x	x							Faz Query expansion utilizando-se a base textual como referência, faz referencia a uma pesquisa que utiliza tessaurus em conjunto.
24	x								Query expansion utilizando-se métodos estatísticos
25	x						x		Query expansion manual em chinês
26	x								Query expansion utilizando-se distância de edição para corrigir textos OCR
27	x								
28	x		x	x	x				Query expansion utilizando-se métodos estatísticos
29	x	x							Faz referencia (sem modelo matemático) ao uso de query expansion em conjunto com term extration, entretando a query expansion é feita utilizando-se métodos estatísticos

30						x		x	Modelo de indexação utilizando-se thesaurus
31		x							Apresenta um modelo de redução de documento
32	x								Modelo de expansão de queries utilizando bigramas
33		x							Apresenta um modelo para criação de queries baseadas em parágrafos escritos em linguagem natural
34	x		x		x				Query Expansion utilizando-se métodos estatísticos e bigramas
35	x								Query Expansion utilizando-se métodos estatísticos
36	x	x					x		Query Expansion utilizando-se métodos estatísticos em conjunto com interação com o usuário
37	x								Query expansion utilizando-se lógica fuzzy
38		x	x						Term extration em textos escritos em linguagem natural com o objetivo de construir uma query
39	x								Query expansion utilizando-se a teoria da entropia relativa ou Kullback-Lieber Distance
40		x							Term extration em textos escritos em linguagem natural com o objetivo de construir uma query
41	x	x							Utiliza técnicas de Term Extration e Query Expansion em conjunto, entretanto a Query Expansion é construída apenas utilizando-se sinônimos, não apresenta modelo matemático detalhado para comparação
42	x		x						Utiliza técnicas estatísticas para expandir a query e introduz um novo conceito Word Contribution
43	x			x					Apresenta uma comparação de técnicas de query expansion
44	x						x		Técnica de Query Expansion utilizando-se thesaurus construídos automaticamente
45	x	x							Utiliza Query Expansion e Term Extration em conjunto, no entanto as técnicas aplicadas a Query Expansion são restritamente Estatísticas
46	x	x							Utiliza query Expansion e Term Extration em conjunto, e um thesaurus genérico com Synonimos e Hyperonimos
47		x							Apresenta o conceito de Document-Query Translation
48	x			x	x				
49	x						x		
50	x								Query Expansion utilizando-se métodos estatísticos
51	x								Query Expansion utilizando-se métodos estatísticos
52	x	x			x				Term Extration e Query expansion, query expansion utiliza expansão morfológica, ou seja, um tipo de stemming ao contrário
53	x						x		Query expansion através de thesaurus genéricos (WordNet) e Phrase Extration (Linguagem Natural)
54	x								Query Expansion utilizando-se métodos estatísticos
55	x					x			Utiliza WordNet Thesaurus Genérico e não utiliza pesos para as relações semânticas
56	x								Recuperação de documentos médicos
57	x								Query Modification através de feedback do usuário
58		x							Faz extração de termos com análise linguística para montar uma query que represente uma pergunta
59	x	x							Faz sumarização de perguntas e expansão de query através de técnicas estatísticas e feedback do usuário
60		x			x				Construção de queries automáticas
61	x		x						Utiliza IDF para expandir queries
62	x				x				Expansão de queries através de técnicas estatísticas
63	x	x			x				Query Expansion através de feedback do usuário e técnicas estatísticas
64	x	x							Utiliza Query expansion e Term Extration para responder questões, mas não apresenta o modelo matemático detalhado



65		x							
66		x							Metodos estatísticos, voltado para web
67	x	x							Utiliza Query expansion e Term Extration entretando a query expansion utiliza apenas sinonimos
68	x								Utiliza uma Ontologia para procurar artigos que retratem informações sobre genes contidos na Ontologia
69	x	x							Utiliza métodos estatísticos para a expansão, utilizado na área médica
70						x			Utiliza uma Ontologia para classificar documentos de uma base médica
71	x					x			Utiliza Query Expansion e Ontologias para classificar documentos médicos
72		x							Utiliza técnicas de Term Extration para classificar documentos médicos

Quadro 6: Artigos e Tecnologias Utilizadas

Analisando as pesquisas citadas no Quadro 6, é possível concluir que o número de publicações em que as técnicas de *Query Expansion* e *Term Extration* são utilizadas em conjunto em relação ao total de artigos publicados é reduzido. Excluindo-se as pesquisas onde o *Term Extration* é utilizado para indexação de documentos (o que não representa o foco desta tese) pode-se agrupar as pesquisas em dois grupos, a saber:

1. Pesquisas onde a técnica *Query Expansion* é utilizada a partir de técnicas estatísticas. Este tipo de técnica permite melhorar a eficiência da busca, entretanto, não permite a incorporação semântica da busca;
2. Grupo onde as pesquisas procuram incorporar semântica e, portanto, deverão servir como referência para a presente pesquisa. Neste grupo um tipo de estrutura linguística (dicionário, *thesaurus* ou ontologia) deve ser incorporada, sendo que fazem parte deste grupo os artigos 41, 46 e 67 que tem como foco objetivos semelhantes aos desta pesquisa.

Deve-se ressaltar que as estruturas lingüísticas usadas nessas pesquisas restringem-se à utilização de sinônimos e hiperônimos, distinguindo-se deste trabalho, que utiliza uma Ontologia para obter uma estrutura linguística capaz de

representar maior quantidade de relações entre os conceitos, tendo como finalidade incrementar a capacidade semântica do modelo.

### 3 DESENVOLVIMENTO DO MODELO COMPUTACIONAL

#### 3.1 Técnicas Utilizadas para Construção do Modelo Computacional

De acordo com levantamento bibliográfico apresentado no capítulo anterior, as técnicas *Query Expansion*, *Inverse Document Frequency*, *Term Extration*, *Ontology Models*, *Stemming* e *StopWords* são largamente utilizadas na recuperação de documentos textuais. Conforme apresentado na fundamentação teórica desta tese, as técnicas acima citadas são utilizadas em conjunto com o objetivo de desenvolver modelos computacionais.

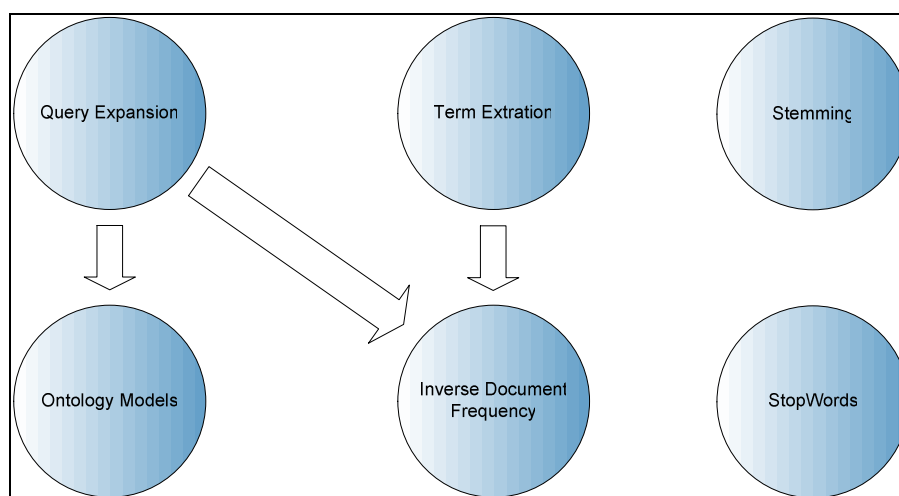


Figura 9: Interrelação das Tecnologias

Exemplos de pesquisas onde as técnicas acima são utilizadas em conjunto são encontrados na literatura e estão apresentadas no Quadro 6 e sua interrelação na Figura 9.

O Modelo Computacional que será utilizado para validar esta tese segue a mesma linha de raciocínio, no entanto, utiliza as seis técnicas em conjunto, apresentando como diferencial um maior nível de agregação das mesmas.

### 3.2 Integração das Técnicas para Construção do Modelo Computacional

O objetivo final do modelo computacional (item 3, Figura 10) a ser desenvolvido nesta tese é a criação de um vetor (considera-se, nesta pesquisa, vetor como sendo uma lista de determinados itens, no caso, termos) de termos e suas respectivas relevâncias (item 4, Figura 10), levando-se em consideração um texto de entrada (item 1, Figura 10), uma Ontologia (item 5, Figura 10) e os demais documentos contidos em uma base de informações textuais (item 2, Figura 10). Esse vetor de termos permite a criação de uma *Query* (conjunto de termos agrupados pelos operadores lógicos *or* e *and*), que poderá ser utilizada em qualquer ferramenta de busca que leve em consideração a relevância dos termos da *Query*.

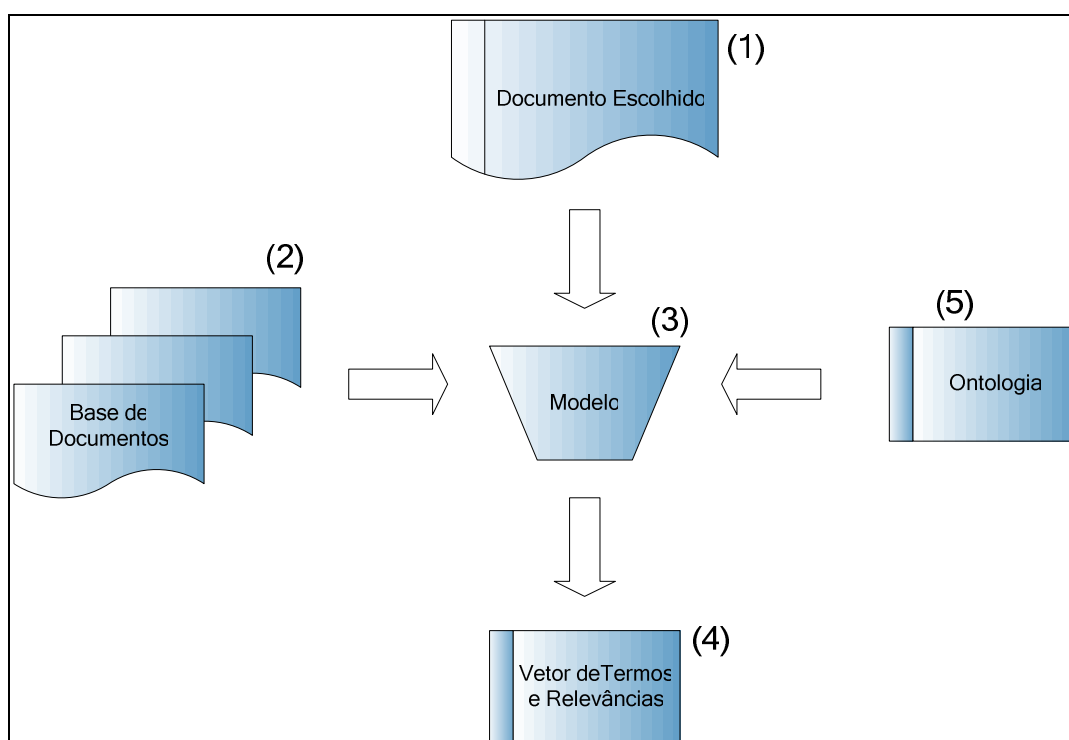


Figura 10: Entrada e Saída do Modelo Computacional

O Modelo computacional desta pesquisa foi dividido em seis etapas distintas são necessárias para a criação do **Vetor de Termos**, estas etapas serão detalhadas no tópico 3.3:

1. Criação do **Vetor de Termos** inicial baseado no **Documento Escolhido**.
2. Expansão do **Vetor de Termos** inicial baseando-se em uma **Ontologia**.
3. Preenchimento do **Vetor de Termos** com suas respectivas **Relevâncias**, levando-se em consideração os demais documentos.
4. Preenchimento do **Vetor de Termos** com as **Relevâncias Cruzadas**.
5. Criação do **Vetor de Corte**.
6. Criação da *Query*.

### 3.3 Etapas do Modelo Computacional

Neste tópico são descritas em detalhes as seis etapas citadas anteriormente, sendo que a Figura 11 mostra a distribuição das técnicas nas referidas etapas. Vale ressaltar que as etapas 4 e 5 não possuem técnicas apresentadas na fundamentação teórica, pois tratam-se de técnicas criadas especificamente para este modelo.

Salienta-se que a presente pesquisa baseou-se na hipótese de que um modelo computacional que utilize as técnicas de *Term Extration* e *Query Expansion*, em conjunto, e com um modelo de integração destas técnicas que o diferencie de outras pesquisas na área (analisadas no tópico 2.7 deste trabalho), tornaria possível

a recuperação de documentos textuais, fazendo uma análise semântica dos mesmos.

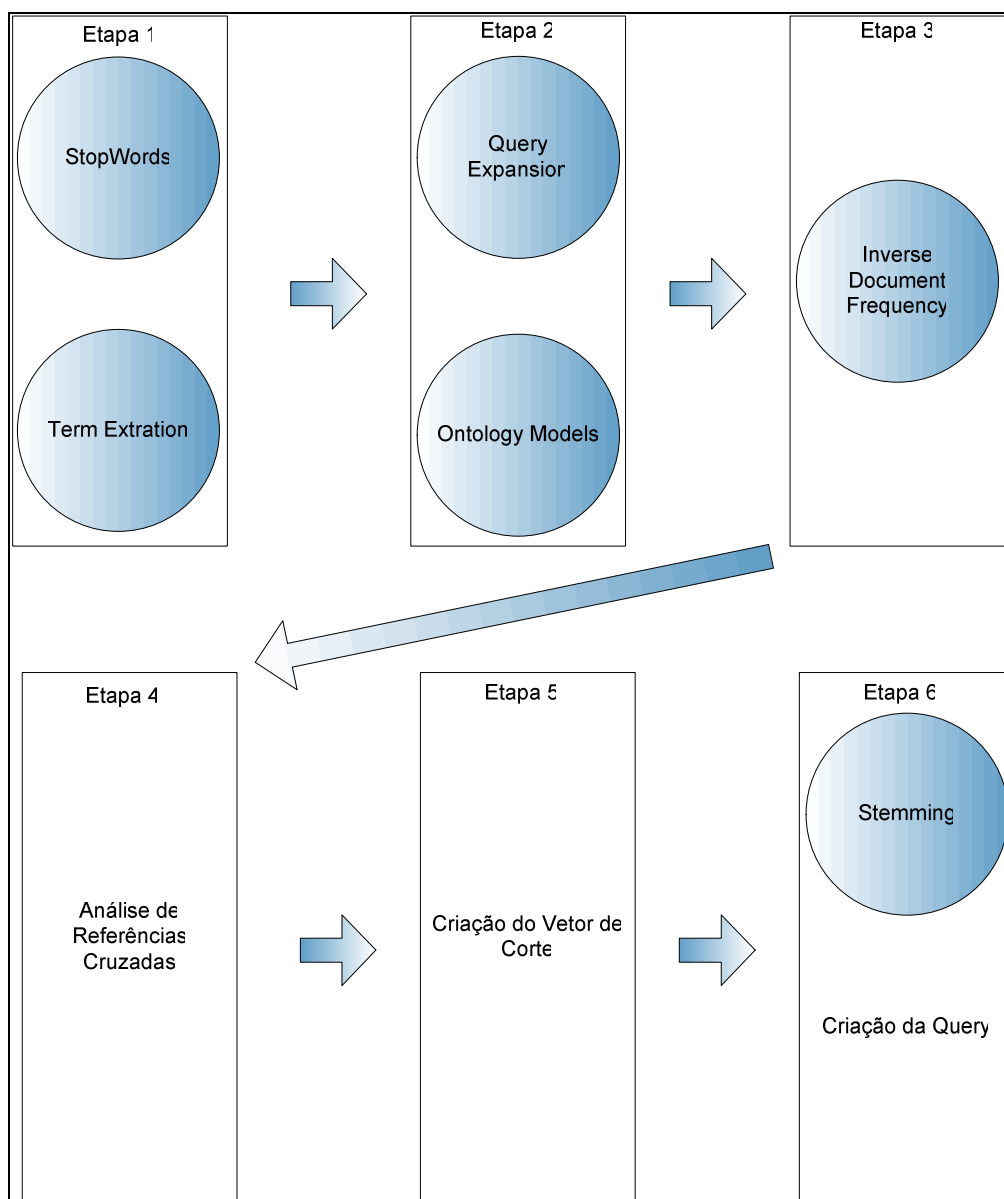


Figura 11: Distribuição das Técnicas

As duas primeiras etapas usam como base uma Ontologia que representa a ligação semântica dos termos utilizados nos documentos que compõem a base de informação utilizada. A Ontologia tem por finalidade possibilitar uma interpretação contextual do texto, exibindo em sua estrutura as relações semânticas mais comuns encontradas nas linguagens. Os valores que representam as relações semânticas utilizadas foram baseadas no algoritmo H-MATCH (Castelano) com uma

modificação: o algoritmo original apresenta a relação **sinônimo** com peso semântico um (1.0) e nesta relação, o peso definido foi zero ponto nove (0.9), com o intuito de representar uma ligação forte, próxima a um (1.0), que é o peso utilizado pelas palavras originais do texto origem, também foram adicionados as relações Tem e Agrupa.

A Tabela 3 apresenta os pesos associados às relações semânticas utilizadas na Ontologia, o uso dos pesos semânticos será detalhado no tópico 3.3.2.

Relação	Peso
Sinônimo	0.9
Tipo de	0.8
Associado	0.3
Parte de	0.7
Agrupa	0.8
Tem	0.7

Tabela 3: Pesos Associados às Relações Semânticas dos Termos

A Figura 12 representa o modelo estrutural da Ontologia, seguindo os princípios anteriormente citados.

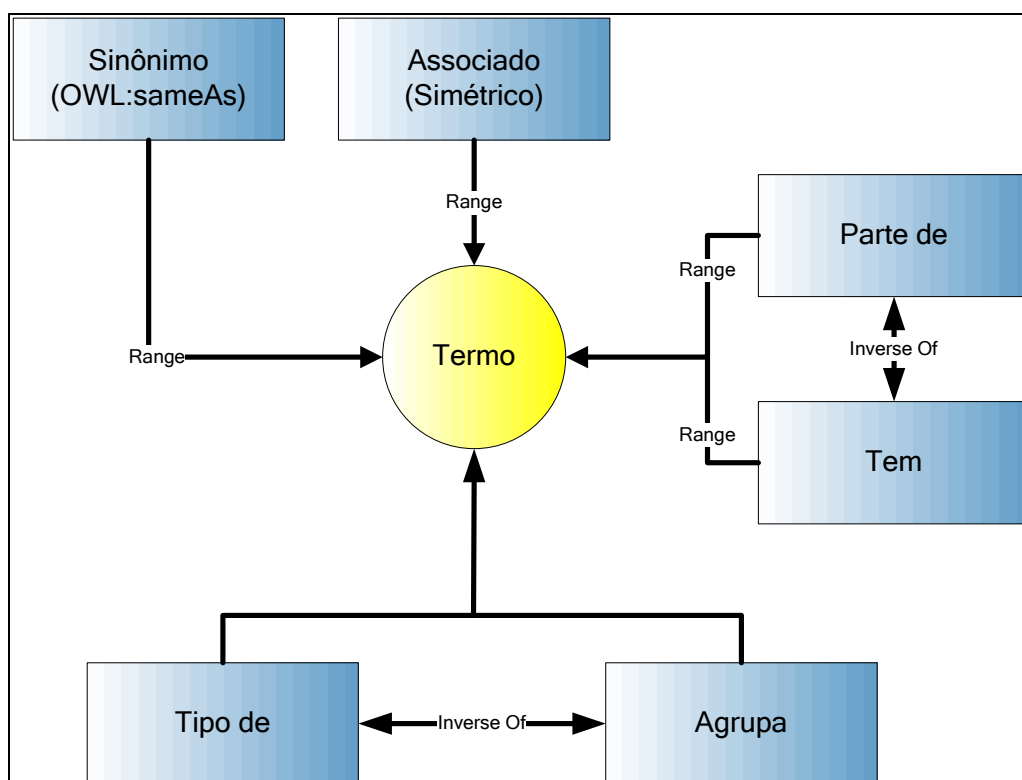


Figura 12: Estrutura da Ontologia

O modelo estrutural da Ontologia criado permite que as seguintes inferências possam ser verificadas (Figura 13):

<p>1 - (?a :s ?b) -&gt; (?b :s ?a)</p> <p>2 - (?a :s ?b) (?b :s ?c) -&gt; (?c :s ?a)</p> <p>3 - (?a :c ?b) (?b :c ?a)</p> <p>4 - (?a :p ?b) (?b :s ?c) -&gt; (?a :p ?c)</p> <p>5 - (?a :c ?b) (?b :s ?c) -&gt; (?a :c ?c)</p> <p>6 - (?a :t ?b) (?b :s ?c) -&gt; (?a :t ?c)</p> <p>7 - (?a :p ?b) -&gt; (?b :t ?a)</p> <p>8 - (?a :t ?b) -&gt; (?b :g ?a)</p>
---

Figura 13: Representação Matemática das Inferências

1. Se o termo **a** é sinônimo do termo **b**, então o termo **b** é sinônimo do termo **a**.
2. Se o termo **a** é sinônimo do termo **b** e o termo **b** é sinônimo do termo **c**, então o termo **c** é sinônimo do termo **a**.
3. Se o termo **a** é conexo do termo **b**, então o termo **b** é conexo do termo **a**.
4. Se o termo **a** é parte do termo **b** e o termo **b** é sinônimo do termo **c**, então o termo **a** é parte do termo **c**.
5. Se o termo **a** é conexo do termo **b** e o termo **b** é sinônimo do termo **c**, então o termo **a** é conexo do termo **c**.
6. Se o termo **a** é tipo de do termo **b** e o termo **b** é sinônimo do termo **c**, então o termo **a** é tipo de do termo **c**.
7. Se o termo **a** é parte do termo **b**, então o termo **b** tem o termo **a**.
8. Se o termo **a** é um tipo de do termo **b**, então o termo **b** agrupa o termo **a**.



A Figura 14 apresenta os termos **Carro**, **Automóvel**, **Pálio**, **Aparelho de Som**, **Roubc**, **Furto** e **FIAT** e suas relações semânticas estruturadas na forma da Ontologia citada. Os termos apresentados serão utilizados como exemplo nas etapas subseqüentes.

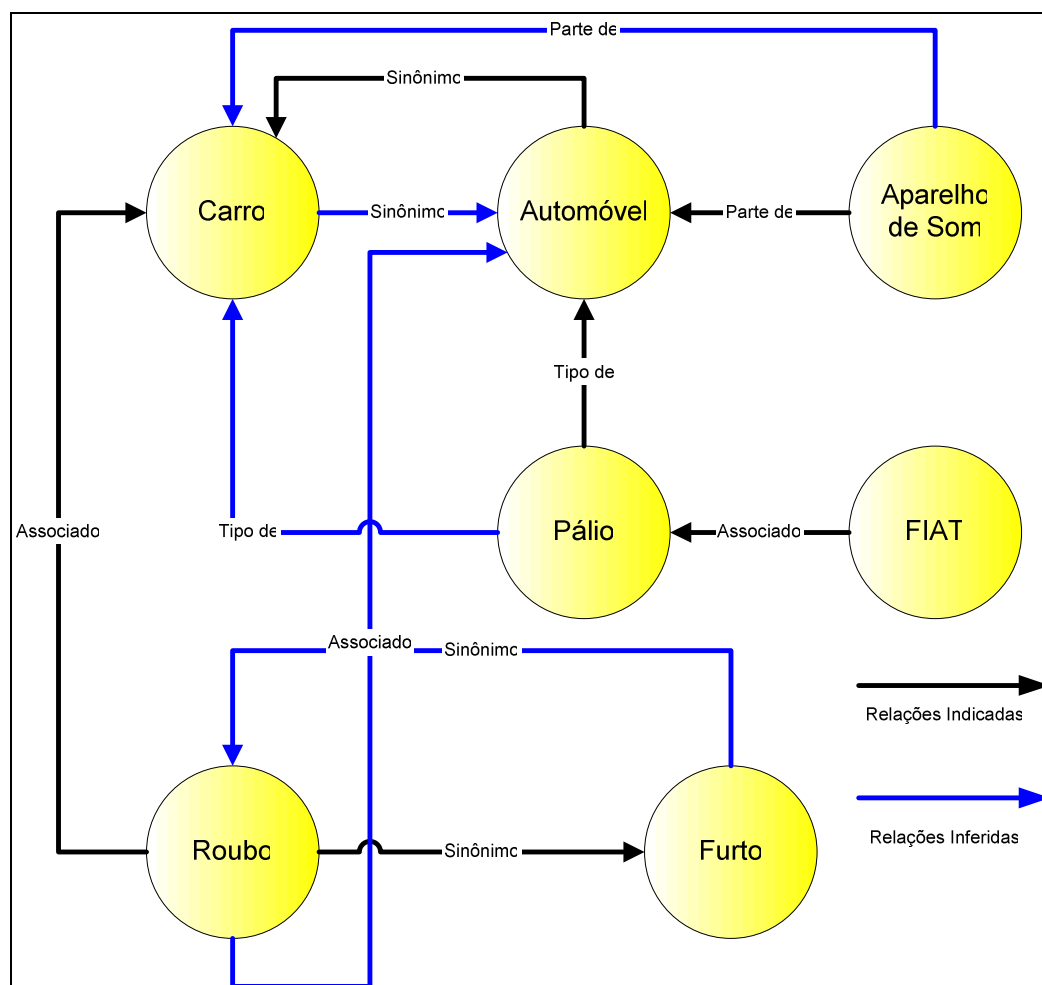


Figura 14: Termos e Relações Semânticas

### 3.3.1 Criação do Vetor Inicial

O Vetor Inicial é criado a partir do documento escolhido para servir como base para a busca. Para construção do vetor são necessárias duas etapas, a saber:

a) o texto é analisado e todos os termos compostos constantes na Ontologia selecionada são removidos do texto original e inseridos no vetor;

b) o texto restante passa pelo processo de remoção das *StopWords* (são palavras comumente encontradas e que não representam o sentido da frase, descritas em detalhes na fundamentação teoria), sendo que os termos restantes são inseridos no vetor.

Salienta-se que todos os termos são inseridos no vetor com peso 1 (um). Esse peso indica que as palavras são originais. As etapas seguintes usarão pesos que podem variar de 0 (zero) à 1 (um), dependendo do tipo de termo inserido e qual seu objetivo.

Tendo o texto como entrada: **“O furto do aparelho de som de um pálio foi registrado por Raphael”**, a saída da etapa seria o vetor representado na Tabela 4.

Termo	Peso Inicial
Aparelho de Som	1.0
Raphael	1.0
Furto	1.0
Pálio	1.0
Registrado	1.0

Tabela 4: Resultado da Primeira Fase do Modelo

### 3.3.2 Criação do Vetor Expandido

O Vetor Expandido é criado baseando-se no resultado da inferência dos termos do vetor de entrada da etapa anterior sobre a Ontologia. Para construção deste vetor são necessárias três etapas:

- a) criação de um vetor de termos correlatos para cada termo de entrada;
- b) inserção do termo original nesse segundo vetor, com o peso relativo à Ontologia 1.0 (sinônimo);
- c) se o termo for um termo composto constante na Ontologia, o mesmo deverá ser dividido em termos não compostos os quais serão inseridos no segundo

vetor com o peso relativo à Ontologia 1.0 (sinônimo). A lista de termos resultados da decomposição do termo composto deve passar pelo processo de *stopwords* para evitar que termos não significativos sejam inseridos no vetor. Estes termos inseridos, apesar de serem termos resultantes do termo original, devem possuir peso inicial 0.5, pois o objetivo deste valor é diminuir o valor semântico do termo, já que os termos restantes da decomposição podem não representar o termo original.

O exemplo a seguir toma como entrada os dados da Tabela 4 (saída da etapa anterior do algoritmo) para exemplificar as três etapas desta fase.

Na primeira (a) e segunda (b) etapas cada termo do vetor é inferido utilizando-se a Ontologia e os termos inferidos são inseridos em um vetor, levando-se em consideração os pesos semânticos apresentados na Tabela 3.

O termo “Aparelho de Som” após ser inferido apresenta o resultado: **Parte de Automóvel e Parte de Carro**.

Seguindo o mesmo exemplo, deve-se inferir todos os termos tendo como resultado final a Tabela 5. Destaca-se que na terceira etapa (c), o termo “Aparelho de Som” é localizado e como se trata de um termo composto constante na Ontologia, deve ser decomposto e inserido no vetor, seguindo as especificações desta etapa.

Termo Original	Termo Inferido	Inicial	Ontologia
Aparelho de Som	Aparelho de Som	1,000	1,000
	Automóvel	1,000	0,700
	Carro	1,000	0,700
	Aparelho	0,500	1,000
	Som	0,500	1,000
Raphael	Raphael	1,000	1,000
Furto	Furto	1,000	1,000
	Roubo	1,000	0,900
	Carro	1,000	0,300
	Automóvel	1,000	0,300

Pálio	Pálio	1,000	1,000
	Carro	1,000	0,800
	Automóvel	1,000	0,800
	FIAT	1,000	0,300
Registrado	Registrado	1,000	1,000

Tabela 5: Termos Inferidos e Pesos Semânticos

### 3.3.3 Preenchimento do Vetor de Termos com suas Relevâncias

Esta etapa se baseia na premissa da frequência inversa para verificar o potencial representativo de um termo perante uma base textual. Mais detalhes sobre a frequência inversa (IDF) podem ser encontrados na fundamentação teórica.

Para concluir esta etapa do modelo, todos os termos do vetor de entrada da etapa anterior (Tabela 5) deverão ter sua frequência de aparecimento na base textual analisada, seguindo a fórmula do IDF (Figura 4). A fórmula do IDF pode ser textualmente explicada como: *“A Relevância do Termo é igual ao Logaritmo do Número de Documentos dividido pelo Número de Documentos contém o termo”*.

Supondo-se que um determinado termo apareça em 10 documentos de uma base de 100 documentos, então sua relevância é representada pelo logaritmo de  $100/10$ , e o resultado seria um. A finalidade da utilização do logaritmo é garantir que, conforme a frequência de um termo aumenta sua importância em relação às frequências menores seja atenuada.

Após efetivar esta etapa para todos os termos do vetor de saída da etapa anterior o resultado deverá ser um vetor contendo os pesos das etapas anteriores e também os desta etapa. Os valores de relevância apresentados na tabela a seguir servem meramente para fins de demonstração do modelo (Tabela 6).

Termo Original	Termo Inferido	Inicial	Ontologia	Relevância
Aparelho de Som	Aparelho de Som	1,000	1,000	1,600
	Automóvel	1,000	0,700	1,780
	Carro	1,000	0,700	1,340
	Aparelho	0,500	1,000	1,020
	Som	0,500	1,000	1,230
Raphael	Raphael	1,000	1,000	1,940
Furto	Furto	1,000	1,000	1,000
	Roubo	1,000	0,900	0,300
	Carro	1,000	0,300	1,340
	Automóvel	1,000	0,300	1,780
Pálio	Pálio	1,000	1,000	1,300
	Carro	1,000	0,800	1,340
	Automóvel	1,000	0,800	1,780
	FIAT	1,000	0,300	1,040
Registrado	Registrado	1,000	1,000	1,000

Tabela 6: Termos e Relevâncias Ontológica e IDF

### 3.3.4 Preenchimento do Vetor de Termos com suas Relevâncias Cruzadas

O objetivo desta etapa é garantir que os termos que possuam relações semânticas fortes entre si recebam um incremento em seus respectivos pesos, assim, aumentando as chances desses termos serem selecionados como base para a pesquisa durante a etapa seguinte.

Partiu-se da premissa que se um mesmo termo aparece como resultado da inferência de dois ou mais termos, o mesmo deve possuir uma relação semântica mais forte que um termo que aparece na inferência de apenas um termo.

A Tabela 7 representa o Vetor de Termos atual. Toma-se como exemplo o termo “Automóvel” e o termo “FIAT”. Observada a representação do vetor, é possível verificar que o termo “Automóvel” aparece como termo inferido a partir dos termos “Furto”, “Pálio” e “Aparelho de Som” e o termo “FIAT” aparece como inferência apenas do termo “Pálio”. Portanto, o termo “Automóvel” possui uma maior relação semântica com os termos bases da pesquisa e deve receber um incremento em seu peso final.

Aparelho de Som	Raphael	Furto	Pálio	Registrado
Aparelho de Som	Raphael	Furto	Pálio	Registrado
Automóvel		Roubo	Carro	
Carro		Carro	Automóvel	
Aparelho		Automóvel	FIAT	
Som				

Tabela 7: Vetor de Termos de Relevâncias Cruzadas

O peso de relevância cruzada (Figura 15) é obtido a partir da média aritmética da multiplicação dos pesos iniciais, ontologia e relevância de cada termo inferido repetido.

Toma-se, por exemplo, o termo “Automóvel” inferido a partir do termo “Aparelho de Som”, este termo é encontrado na inferência do termo “Furto” e do termo “Pálio”. Então, o peso de relevância cruzada é encontrado a partir da seguinte fórmula:

*AF = termo Automóvel inferido a partir do termo Furto*

*AP = termo Automóvel inferido a partir do termo Pálio*

*Pi = Peso Inicial*

*Po = Peso da Ontologia*

*Pr = Peso da Relevância*

*Prc = Peso de Relevância Cruzada*

$$Prc = ( ( PiAF * PoAF * PrAF ) + ( PiAP * PoAP * PrAP ) ) / 2$$

$$Prc = ( ( 1.0 * 0.3 * 1.780 ) + ( 1.0 * 0.8 * 1.780 ) )$$

$$Prc = 1.958$$

Figura 15: Cálculo da Relevância Cruzada

O mesmo cálculo deve ser feito para todos os termos originais e inferidos. O resultado será um vetor contendo os seguintes pesos: Inicial, Ontologia, Relevância e Relevância Cruzada, que serão utilizados na etapa posterior (Tabela 8).

Termo Original	Termo Inferido	Inicial	Ontologia	Relevância	R. Cruzada
Aparelho de Som	Aparelho de Som	1,000	1,000	1,200	1,000
	Automóvel	1,000	0,700	1,780	1,958
	Carro	1,000	0,700	1,230	1,762
	Aparelho	0,500	1,000	1,020	1,000
	Som	0,500	1,000	1,230	1,000
Raphael	Raphael	1,000	1,000	1,940	1,000
Furto	Furto	1,000	1,000	1,000	1,000
	Roubo	1,000	0,900	0,300	1,000
	Carro	1,000	0,300	1,080	2,031
	Automóvel	1,000	0,300	1,780	2,017
Pálio	Pálio	1,000	1,000	1,300	1,000
	Carro	1,000	0,800	1,500	1,593
	Automóvel	1,000	0,800	1,780	1,517
	FIAT	1,000	0,300	1,040	1,000
Registro	Registro	1,000	1,000	1,000	1,000

Tabela 8: Termos e Relevâncias Cruzadas

### 3.3.5 Criação do Vetor de Corte

O modelo aplicado até o presente momento tem como resultado um vetor de termos expandido, tendo por base um arquivo textual. Este vetor contém os termos encontrados no documento, os termos inferidos a partir da Ontologia e também os pesos de relevância encontrados a partir das etapas anteriores. Esta etapa visa reduzir o vetor criado para um número de termos que represente o conteúdo do texto. Este vetor é denominado Vetor de Corte.

Inicialmente, uma constante MNT (Máximo Número de Termos) deve ser definida. Essa constante deve conter o número máximo de termos que a *Query* final deve possuir. O número deve ser definido levando-se em consideração a etapa de execução da *Query*.

Como a finalidade do modelo proposto é a criação da *Query* e não a execução da mesma, essa constante dependerá exclusivamente da maneira como a *Query* será executada, sendo, assim, irrelevante no que se refere à construção do modelo computacional. Mais detalhes sobre essa constante são apresentados no capítulo seguinte, que abrange a construção do protótipo do software e resultados para validação do modelo.

Os termos que estarão contidos no vetor de corte são definidos a partir de uma seleção que leva em consideração a constante MNT e duas variáveis, sendo elas a somatória dos pesos finais do termo original e seus termos inferidos e a somatória de todos os termos.

A Tabela 9 apresenta os termos “Aparelho de Som”, “Raphael”, “Furto”, “Pálio” e “Registro” com seus respectivos termos inferidos e pesos finais. A variável NTSG (Número de Termos Seleccionados por Grupo, Figura 16) define o número de termos selecionados em cada grupo. Um grupo é representado pelo termo original e seus termos inferidos. Para calcular o NTSG é necessário, primeiramente, calcular o somatório dos pesos de cada termo do grupo e, posteriormente, o somatório dos pesos de todos os grupos (SPT). Os pesos finais dos termos são dados pela multiplicação dos pesos parciais, a saber: peso inicial, peso da ontologia, peso da relevância e peso de relevância cruzada.

Termo Original	Termo Inferido	Inicial	Ontologia	Relevância	R. Cruzada	Peso Final
Aparelho de Som	Aparelho de Som	1,000	1,000	1,600	1,000	1,600
	Automóvel	1,000	0,700	1,780	1,958	2,439
	Carro	1,000	0,700	1,340	1,737	1,629
	Aparelho	0,500	1,000	1,020	1,000	0,510
	Som	0,500	1,000	1,230	1,000	0,615
Raphael	Raphael	1,000	1,000	1,940	1,000	1,940
Furto	Furto	1,000	1,000	1,000	1,000	1,000
	Roubo	1,000	0,900	0,300	1,000	0,270
	Carro	1,000	0,300	1,340	2,005	0,806
	Automóvel	1,000	0,300	1,780	2,335	1,247



Pálio	Pálio	1,000	1,000	1,300	1,000	1,300
	Carro	1,000	0,800	1,340	1,670	1,790
	Automóvel	1,000	0,800	1,780	1,890	2,691
	FIAT	1,000	0,300	1,040	1,000	0,312
Registro	Registro	1,000	1,000	1,000	1,000	1,000

Tabela 9: Termos e Pesos Finais

As fórmulas a seguir (Figura 16) representam as etapas necessárias para chegar ao NTSG de cada grupo:

*Pi = Peso Inicial*

*Po = Peso da Ontologia*

*Pr = Peso da Relevância*

*Prc = Peso de Relevância Cruzada*

PF = Peso Final do Termo

NTSG = Número de Termos Seleccionados no Grupo

SPG = Somatório dos Pesos do Grupo

MNT = Máximo Número de Termos

SPT = Somatório dos Pesos de Todos os Grupos

$$\begin{aligned}
 PF &= ( Pi * Po * Pr * Prc ) \\
 SPG &= \sum (PF) \\
 SPT &= \sum (SPG) \\
 NTSG &= MNT * SPG / SPT
 \end{aligned}$$

Figura 16: Número de Termos Seleccionados por Grupo

Como o NTSG representa o número de termos que serão seleccionados em um determinado grupo, deverá obrigatoriamente ser um número inteiro. Este número deverá ser arredondado para o valor inferior mais próximo, caso seja superior a um e ser igualado a um caso o NTSG seja inferior a um.

Esse procedimento de arredondamento visa garantir que o termo original não seja excluído do vetor final, mesmo que sua relevância seja inferior a outros termos (Tabela 10), pois mesmo contendo um valor semântico baixo, considera-se que o termo original possui importância representativa no texto.

Grupo	SPG	NTSG	Arredondamento
Aparelho de Som	6,793	2,837807	2
Raphael	1,940	0,810444	1
Furto	3,323	1,388198	1
Pálio	6,094	2,545796	2
Registro	1,000	0,417755	1
<b>SPT</b>	<b>19,150</b>		

Tabela 10: Grupos e Somatórios

A título de exemplo, definiu-se a constante  $MNT = 8$ , portanto os termos representados na Tabela 11 em cinza deverão ser removidos do vetor final.

Aparelho de Som	Raphael	Furto	Pálio	Registro
Aparelho de Som	Raphael	Furto	Pálio	Registro
Automóvel		Roubo	Carro	
Carro		Carro	Automóvel	
Aparelho		Automóvel	FIAT	
Som				

Tabela 11: Vetor de Termos Atual

### 3.3.6 Criando a *Query*

O objetivo desta etapa é a construção da *Query* de termos que representam o documento de entrada, *Query* a qual é o resultado do modelo computacional. Essa *Query* deve ser construída a partir dos termos resultantes da etapa anterior, utilizando-se os operadores lógicos AND (e) e OR (ou).

Será necessário aplicar aos termos a técnica denominada *Stemming* (Capítulo 2) para remover os sufixos de linguagem.

Os pesos finais dos termos também serão utilizados. A construção da *Query* baseia-se em duas premissas:

a) os termos de um mesmo grupo possuem relações semânticas e a força da mesma representada pelo peso final do termo, sendo assim qualquer um dos termos do grupo representa o termo original com mais ou menos força; e

b) pelo menos um termo de cada grupo deve constar nos documentos para que o documento tenha alguma semelhança.

Para construir uma *Query* que represente as premissas citadas, deve-se agrupar os termos de cada grupo separando-os pelo operador lógico OR (ou) e os grupos devem ser separados pelo operador AND (e). A *Query* também deverá conter o peso final de cada termo (Figura 17).

***(aparelh som^1.6 OR automovel^2,439) AND (raphael^1.940)  
AND (furt^1.0) AND (pali^1.3 OR carr^1.790) AND (registr^1.0)***

Figura 17: *Query* Gerada

Como assinalado no início deste capítulo, o modelo computacional concebido iniciou com um documento selecionado de uma base textual e finalizou com a construção de uma *Query*, que representa a informação deste documento, levando-se em consideração os outros documentos. No próximo capítulo será apresentado o protótipo criado para validar esse modelo e também os resultados alcançados.

## 4 VALIDAÇÃO DO MODELO E IMPLEMENTAÇÃO DO PROTÓTIPO

Para possibilitar a validação do Modelo Computacional criado foi necessária a construção de um protótipo de software com as seguintes características:

1. Capacidade de indexação de documentos textuais;
2. Capacidade de criação e manipulação de Ontologias no padrão OWL;
3. Capacidade de executar as buscas seguindo o modelo desta pesquisa.

Conforme citado no capítulo introdutório, a utilização de ferramentas que sigam as definições *Open Source* foram usadas para implementar as funcionalidades marginais, sendo estas as citadas como características 1 e 2. A linguagem de programação Java foi escolhida, porque além de ser uma linguagem amplamente utilizada na comunidade científica também possui bibliotecas disponíveis com as funcionalidades necessárias. A característica 3 foi implementada utilizando-se a linguagem selecionada e reflete as fases apresentadas no Capítulo 3.

### 4.1 Indexação de Documentos Textuais

A indexação de documentos trata-se de uma fase importante, já que ela está diretamente ligada à capacidade de busca do protótipo. Para esta fase foi selecionada a biblioteca denominada Lucene, mantida pela *Apache Foundation*.

“Apache Lucene é uma máquina de busca textual completa de alta-performance escrita inteiramente na linguagem Java. Trata-se de uma biblioteca versátil para ser utilizada em qualquer aplicação que tenha como requisitos buscas em bases de dados textuais” Apache (2005).

A biblioteca Lucene (APACHE, 2006) foi selecionada por possuir as características abaixo citadas:

- Capacidade de fazer buscas tendo como entrada um conjunto de termos e seus respectivos pesos organizados em blocos lógicos;
- Trata-se de uma biblioteca *Open Source* disponível na linguagem Java.

#### 4.2 Manipulação de Ontologias

Para esta fase da pesquisa foram selecionadas duas bibliotecas computacionais, sendo elas:

Jena: é uma biblioteca computacional para construção de aplicações semânticas construída pela equipe de desenvolvimento da Hewlett-Packard. Esta biblioteca será utilizada para o armazenamento da Ontologia e inferências necessárias no modelo computacional. Ela foi selecionada porque possibilita o armazenamento das Ontologias, utilizando o formato OWL e é distribuída no formato *Open Source* (JENA, 2007).

HyperGraph: trata-se de uma biblioteca que possibilita a construção de árvores hiperbólicas, construída em Java e liberada como *Open Source*. Será utilizada para possibilitar uma visão gráfica da Ontologia (HYPERGRAPH, 2007).

A Figura 18 mostra uma representação hiperbólica de uma Ontologia, que foi retirada do protótipo.

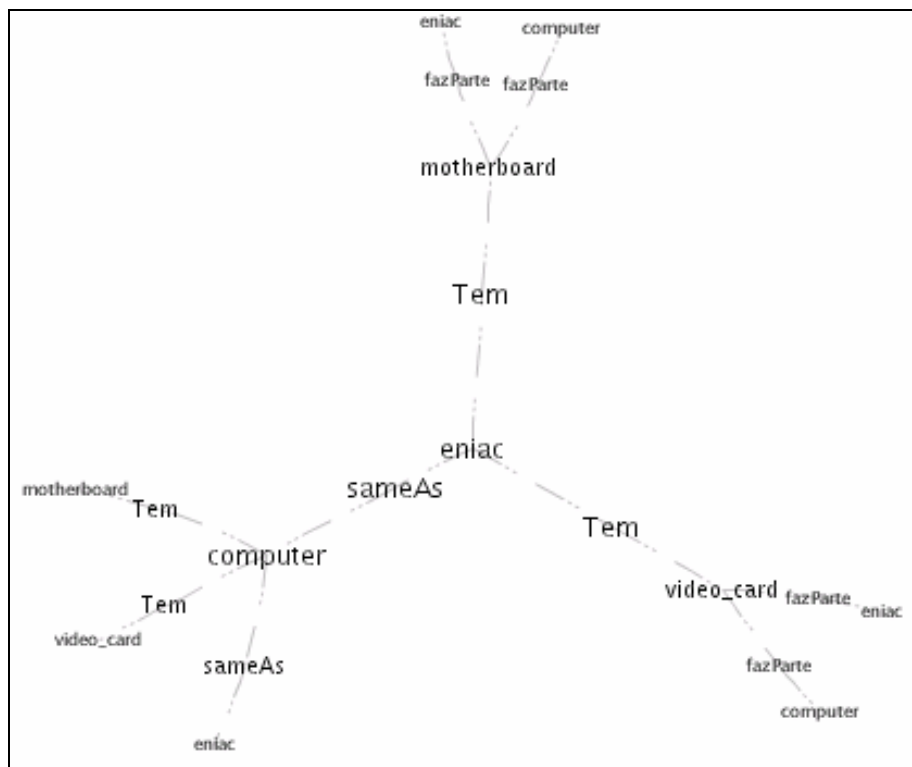


Figura 18: Representação Hiperbólica de Ontologia

### 4.3 Validação do Modelo

A finalidade principal da validação é chegar aos percentuais de *recall* e *precision* do modelo criado, seguindo a especificação do capítulo inicial. Os valores encontrados serão comparados com os valores das técnicas apresentadas TREC 2005. A Figura 19 apresenta o fluxo do processo de validação do modelo.

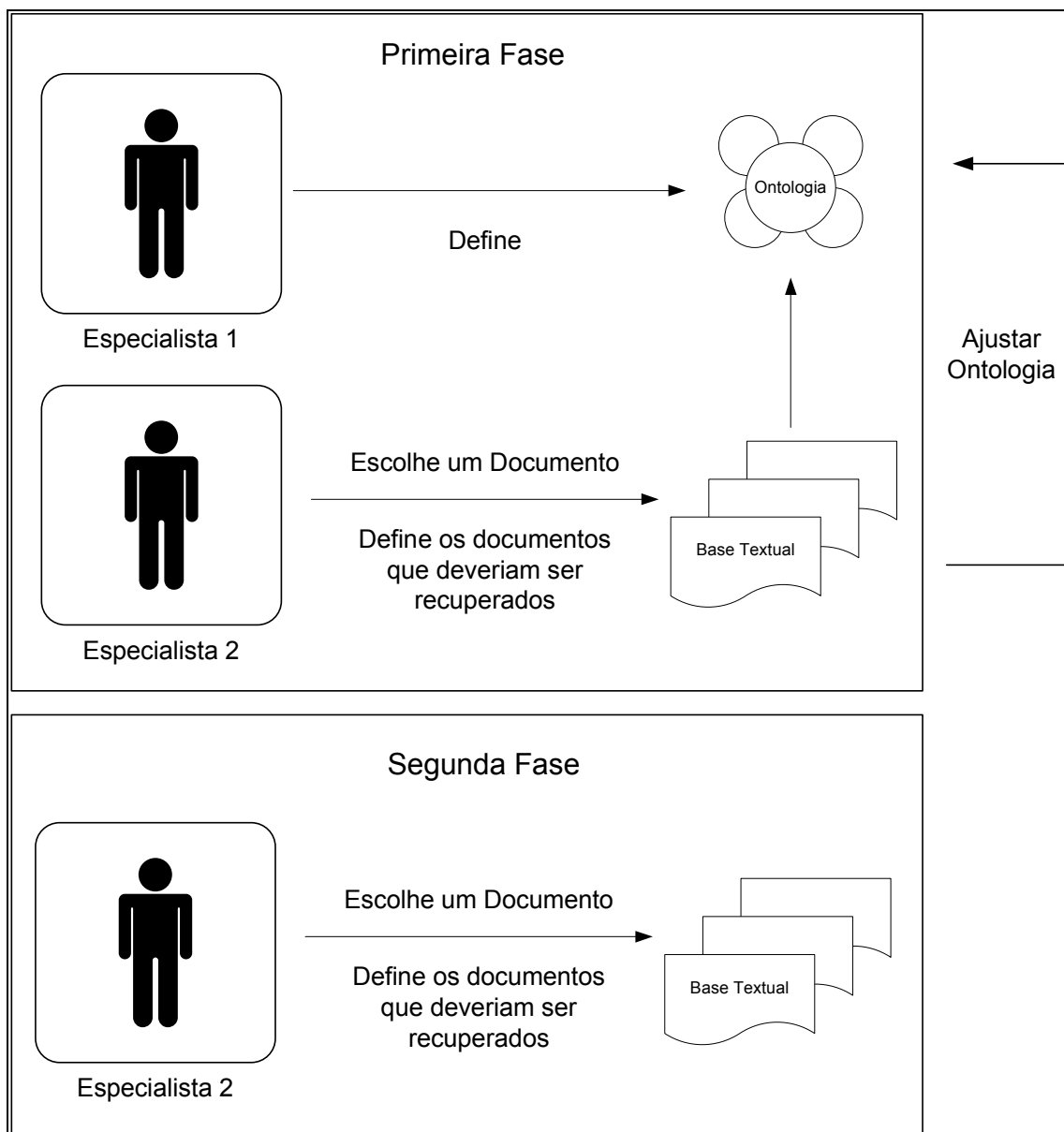


Figura 19: Fluxo do Processo de Validação do Modelo

Como mostra a Figura 19, o processo de validação passou por duas fases, sendo que a primeira tem o objetivo de ajustar a Ontologia. Esta etapa é necessária, pois a eficiência do modelo depende diretamente da construção da Ontologia.

Para a segunda fase será necessário que o especialista selecione dez conjuntos de testes para que se possa efetivamente chegar a uma média dos valores de *recall* e *precision*, sendo que para chegar a estes valores deve-se fazer uma regra de três simples, utilizando-se o número de documentos que deveriam ter

sido recuperados e os documentos que não deveriam ter sido recuperados como no exemplo a seguir (Figura 20).

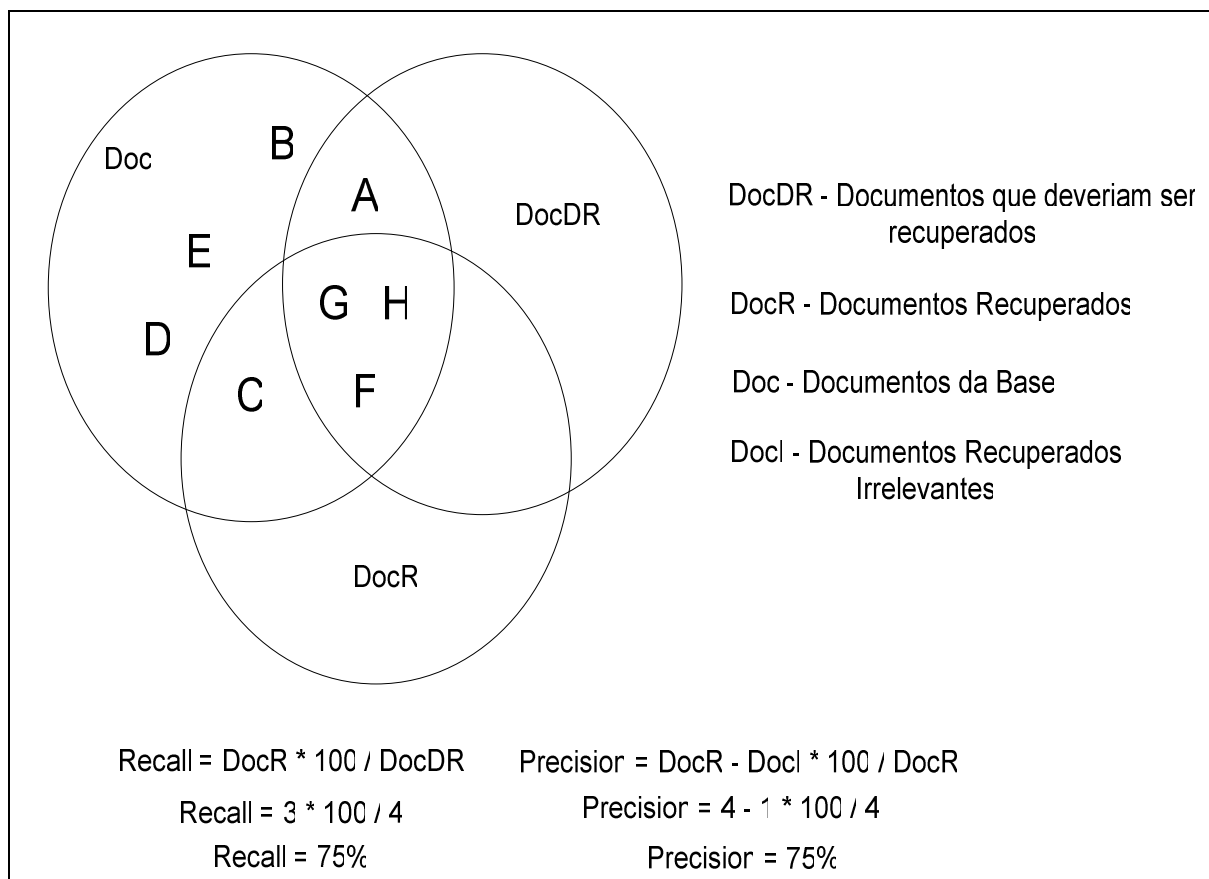


Figura 20: Exemplo de Cálculo de *Recall* e *Precision*

#### 4.4 Protótipo

O protótipo possui três grandes componentes cada um responsável por uma das características necessárias citadas no início deste capítulo, sendo eles Indexador de Documentos, Manipulador de Ontologias e Executor da Busca.

##### 4.4.1 Indexação de Documentos



Esta ferramenta possibilita que o usuário indique o endereço na Internet de um arquivo padrão RSS – padrão de documento XML, utilizado para indexar notícias na Internet – ou um arquivo local seguindo um padrão XML específico (Figura 21).

```
<list>
  <feed.Data>
    <file>Applet1.txt</file>
    <title>O que são os Applets de Java</title>
    <link>http://www.devmedia.com.br/...</link>
  </feed.Data>

  <feed.Data>
    <file>JSF1.txt</file>
    <title>JavaServer Faces: A mais nova tecnologia ...</title>
    <link>http://www.guj.com.br/...</link>
  </feed.Data>

  ...

</list>
```

Figura 21: XML Específico para Indexação

O usuário deve digitar o endereço do arquivo RSS ou do arquivo contendo o padrão XML acima demonstrado no campo *feed*. Caso opte pela utilização do arquivo XML o *checkbox text* deve estar selecionado. O próximo passo é a seleção de um diretório onde os arquivos indexados serão armazenados. Caso este diretório possua arquivos anteriormente indexados a ferramenta irá adicionar os novos arquivos ao *index* antigo. Depois de ter finalizado as etapas anteriores basta o usuário clicar no botão *start* e a indexação será efetivada. A ferramenta mostra o percentual de indexação concluído e um pequeno relatório contendo os títulos dos textos indexados (Figura 22).

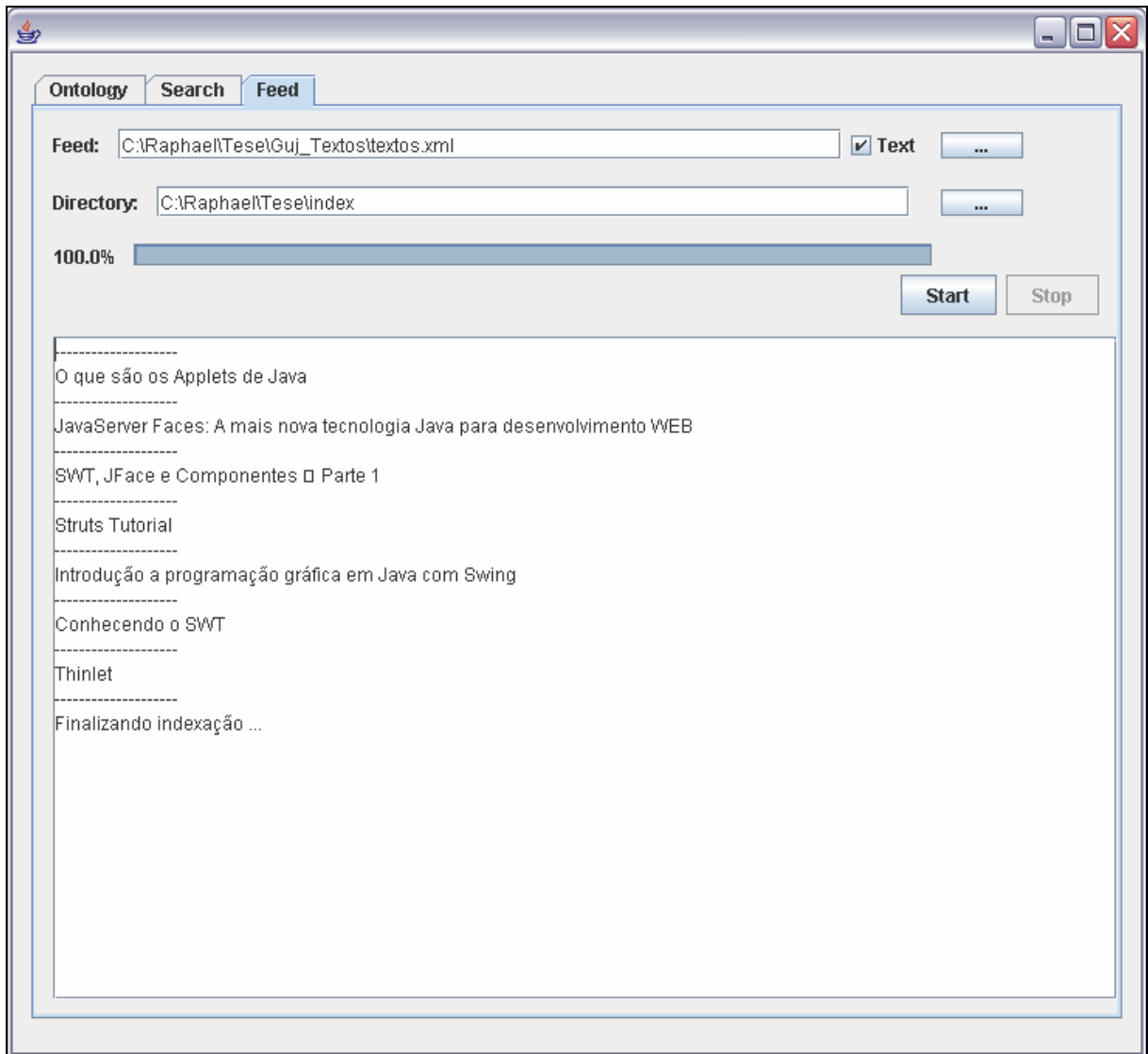


Figura 22: Módulo de Indexação

#### 4.4.2 Manipulação da Ontologia

O módulo de manipulação de Ontologia foi dividido em três partes:

1. Lista de Termos (**Terms**): os mesmos podem ser incluídos utilizando-se o botão **New** e removidos utilizando-se o botão **Delete**;
2. Relações Explícitas (**Relations**): lista de relações explicitamente definidas. Para incluir uma relação o usuário deve clicar no termo escolhido e depois no botão **New**. Para excluir deve utilizar o botão **Delete**;

3. Relações Inferidas ou Implícitas (**Inferente**): lista de relações construídas a partir da inferência da Ontologia.

Este módulo também possui quatro botões gerais que são usados para: armazenar a Ontologia no padrão OWL (**Save**), carregar uma Ontologia armazenada no padrão OWL (**Load**), criar uma nova Ontologia (**New**) e visualizar a árvore hiperbólica demonstrada no tópico 4.2 deste capítulo (**Hiperbolic**) (Figura 23).

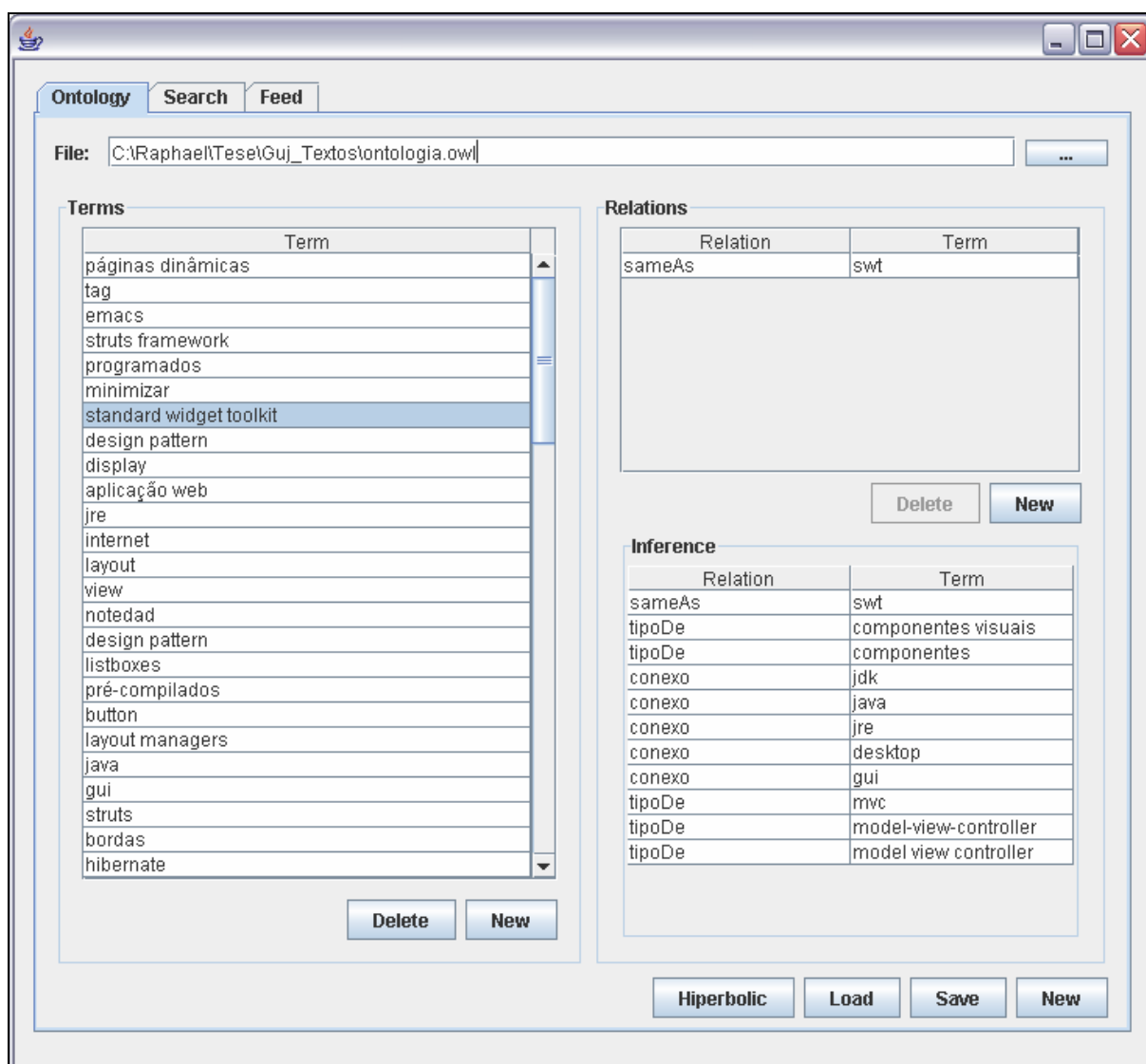


Figura 23: Módulo de Manipulação da Ontologia

#### 4.4.3 Execução da Busca

Este módulo é utilizado para executar as buscas e visualizar os resultados do algoritmo. O usuário deve indicar o diretório onde os documentos estão indexados, indicar o texto que será usado como base para a busca e clicar no botão **Search**. Os resultados serão apresentados abaixo, em uma tabela, com o índice de semelhança (Figura 24).

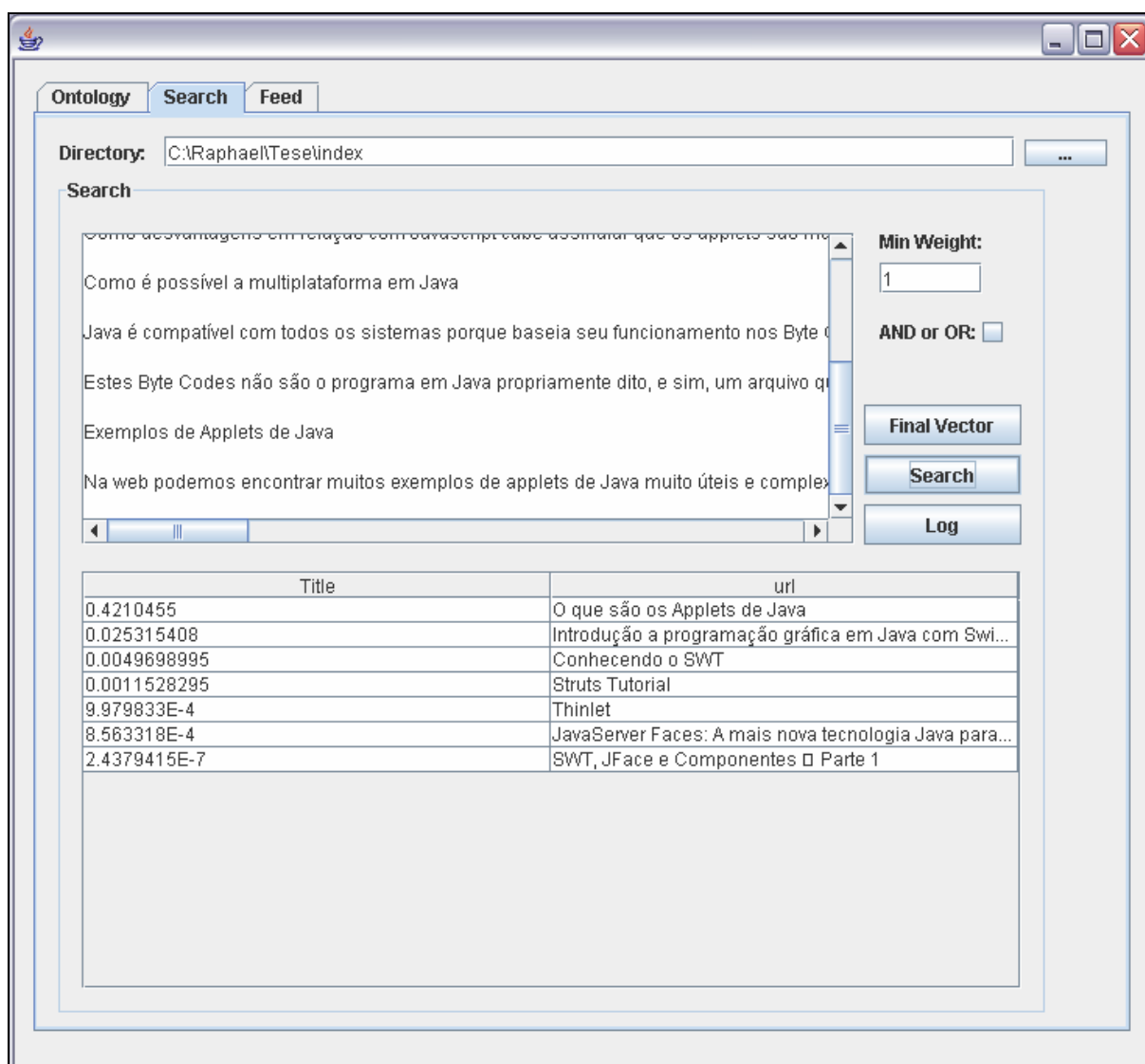
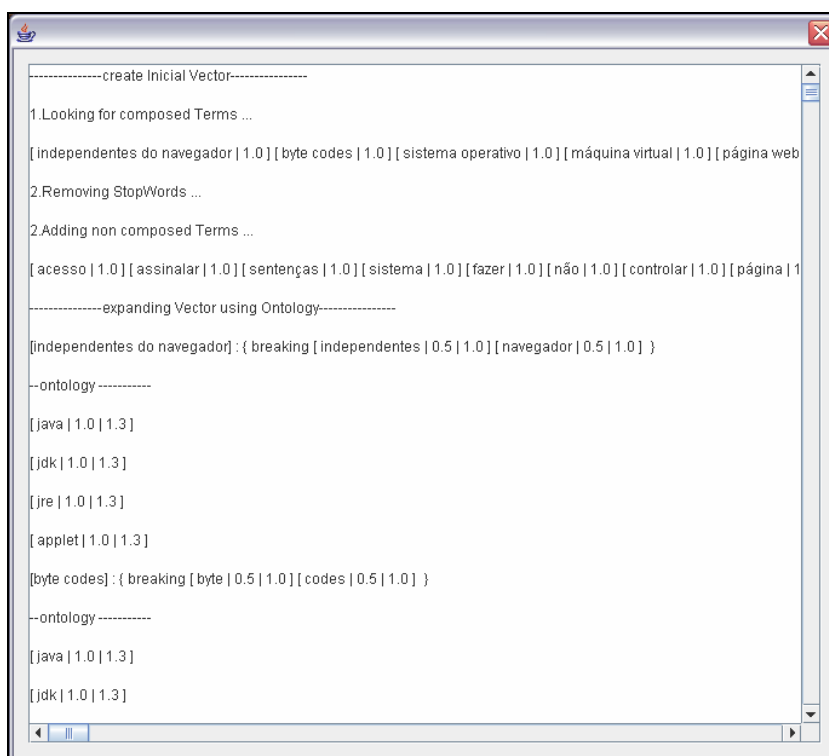


Figura 24: Execução das Buscas

A ferramenta ainda possibilita duas configurações utilizadas para mudar o parâmetros básicos da busca: **Min Weight**; e, se o sistema deve utilizar o conector **OR** ou **AND**. Estas configurações serão mais bem detalhadas no tópico 4.6.

Ainda são apresentadas duas telas que foram utilizadas para analisar os resultados do algoritmo:

- **Log**: esta tela apresenta um texto contendo os valores calculados pelo algoritmo em todas as etapas, semelhante aos valores apresentados nas tabelas 4, 5, 6, 7, 8 e 9 (Figura 25);
- **Final Vector**: esta tela apresenta o vetor final que é utilizado para construção da *Query*. Apresenta na cor azul os itens que serão utilizados e em vermelho os itens que foram removidos na etapa de criação do vetor de corte e que representa a Tabela 11 (Figura 26).



```
-----create Inicial Vector-----
1.Looking for composed Terms ...
[independentes do navegador | 1.0][byte codes | 1.0][sistema operativo | 1.0][máquina virtual | 1.0][página web
2.Removing StopWords ...
2.Adding non composed Terms ...
[acesso | 1.0][assinalar | 1.0][sentenças | 1.0][sistema | 1.0][fazer | 1.0][não | 1.0][controlar | 1.0][página | 1
-----expanding Vector using Ontology-----
[independentes do navegador]:{ breaking [independentes | 0.5 | 1.0][navegador | 0.5 | 1.0] }
--ontology-----
[ java | 1.0 | 1.3 ]
[ jdk | 1.0 | 1.3 ]
[ jre | 1.0 | 1.3 ]
[ applet | 1.0 | 1.3 ]
[byte codes]:{ breaking [byte | 0.5 | 1.0][codes | 0.5 | 1.0] }
--ontology-----
[ java | 1.0 | 1.3 ]
[ jdk | 1.0 | 1.3 ]
```

Figura 25: Valores Resultantes do Algoritmo

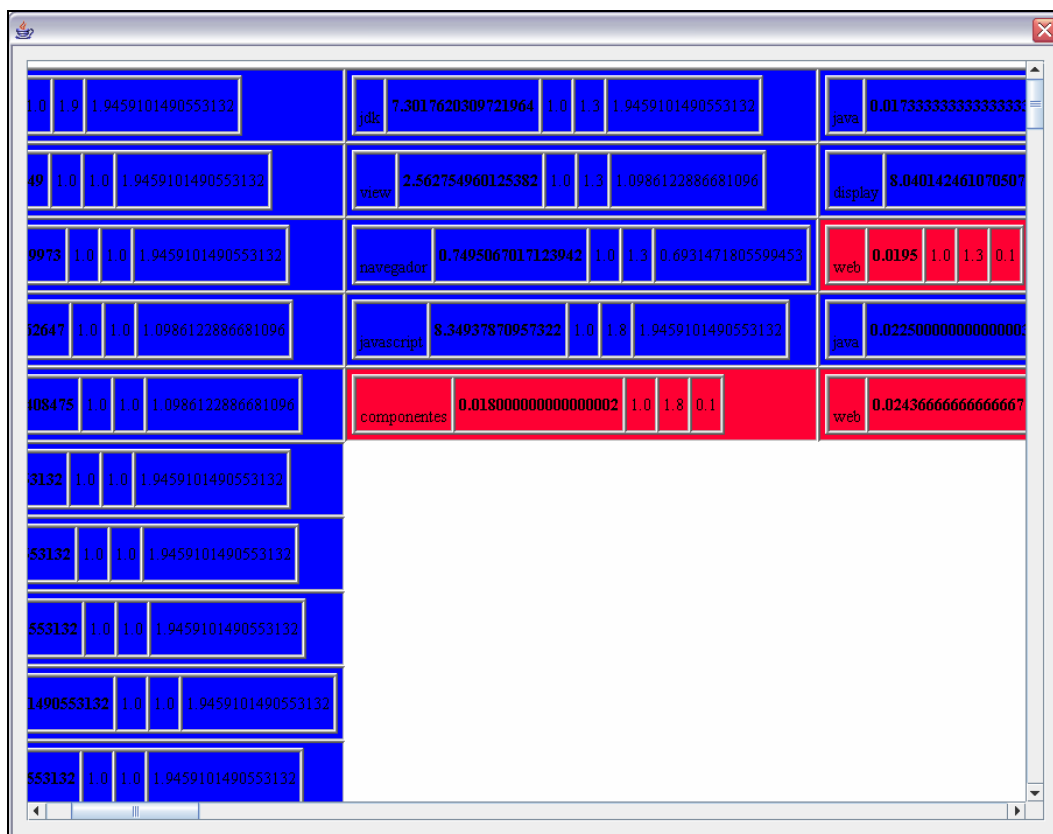


Figura 26: Representação Visual do Vetor Final

#### 4.5 Documentos Utilizados e Ontologia Criada

Esta etapa pode ser dividida em três outras: seleção dos documentos que serão utilizados para analisar o *recall* e *precision* da aplicação; construção da Ontologia utilizando-se os termos técnicos contidos nestes documentos; e, por fim, a seleção de documentos que serão indexados apenas para trabalhar como ruído no momento da recuperação.

##### 4.5.1 Seleção de Documentos

Em um primeiro momento foi selecionada uma subárea do conhecimento. Esta subárea foi “**Tecnologias Utilizadas para Construção de Interfaces com o**

**Usuário na Linguagem Java**”, sendo ela largamente estudada na Ciência da Computação. Existe a necessidade da seleção de uma subárea específica, pois os termos técnicos que representam esses documentos dentro da área selecionada devem pertencer ao rol de termos conhecidos pelos especialistas, para que os mesmos possam efetuar a construção da Ontologia (Figura 19).

O Especialista 2 selecionou um grupo de 7 (sete) documentos em portais conhecidos por disponibilizarem artigos técnicos na subárea selecionada. Os portais usados e os documentos selecionados estão citados nos Quadros 7 e 8. Os documentos que serão usados como ruído, em um total de 40 (quarenta), foram selecionados no portal **DevMedia** nas áreas **Java Desktop** e **Java Web**.

Portal	Endereço na Internet
DevMedia	<a href="http://www.devmedia.com.br/">http://www.devmedia.com.br/</a>
Grupo de Usuários Java	<a href="http://www.guj.com.br/">http://www.guj.com.br/</a>
Portal Java	<a href="http://www.portaljava.com.br">http://www.portaljava.com.br</a>
iMasters	<a href="http://www.imasters.com.br/">http://www.imasters.com.br/</a>

Quadro 7: Portais com Documentos Escolhidos

Índice	Título	Autor
1	O que são os Applets de Java	Miguel Angel Alvarez
2	JavaServer Faces: A mais nova tecnologia Java para desenvolvimento WEB	Talita Pitanga
3	SWT, JFace e Componentes - Parte 1	Maurício Linhares de Aragão Junior
4	Struts Tutorial	Wellington B. Souza
5	Introdução a programação gráfica em Java com Swing	Rafael Steil
6	Conhecendo o SWT	Maurício Linhares de Aragão Junior
7	Thinlet	Luiz Rafael Fernandes

Quadro 8: Documentos Escolhidos

## 4.5.2 Criação da Ontologia

A Tabela 12 apresenta a Ontologia construída pelo Especialista 1 baseando-se nos termos técnicos selecionados dos documentos selecionados pelo Especialista 2 (Quadro 8).

Termo	Tipo de Relação	Termo
páginas dinâmicas		
tag		
emacs		
struts framework		
programados	conexo	java
minimizar	tipoDe	eventos
standard widget toolkit	sameAs	swt
design pattern	agrupa	mvc
display	sameAs	view
aplicação web		
jre		
internet		
layout		
view		
notedad		
design pattern		
listboxes	tipoDe	componentes visuais
pré-compilados		
button		
layout managers		
java	conexo	máquina virtual
	conexo	byte codes
	conexo	pré-compilados
	conexo	so
	sameAs	jre
	sameAs	jdk
gui		
struts	tipoDe	model view controller
	sameAs	struts framework
hibernate		
servlets	tipoDe	camada de negócio
	conexo	java
independentes do navegador	conexo	java
componentes visuais		
listas		
common gateway interface		
ide	agrupa	emacs
	agrupa	notedad
	agrupa	eclipse
	conexo	programados
camada de negócio		



eclipse		
desktop		
thinlet	tipoDe	mvc
	conexo	view
	tipoDe	componentes visuais
model view controller		
caixas de texto	tipoDe	componentes visuais
apis		
mvc	agrupa	struts framework
	conexo	gui
	sameAs	model view controller
	sameAs	model-view-controller
webwork	tipoDe	mvc
applet	tipoDe	interface gráfica
	tipoDe	gui
	conexo	navegador
	conexo	java
	conexo	página web
	tipoDe	visualização
	conexo	web
	conexo	aplicações para web independentes do navegador
	conexo	
	agrupa	componentes visuais
so		
byte codes		
maximizar	tipoDe	eventos
servidor	conexo	web
sistema operativo	sameAs	so
swt	conexo	desktop
	conexo	gui
	tipoDe	componentes visuais
	tipoDe	mvc
	conexo	jre
jdk		
aplicações para web		
linguagem de programação	agrupa	java
	agrupa	cgi
jsp	conexo	view
	conexo	java
	sameAs	java server pages
labels		
java server pages		
camada de apresentação		
frames	conexo	view
gráficas		
janelas		
model	sameAs	modelo
swing	tipoDe	componentes visuais
	conexo	gráficas
	tipoDe	mvc
	conexo	java
controle	conexo	mvc

componentes	sameAs	componentes visuais
visualização		
cgi	sameAs	common gateway interface
botões	tipoDe	componentes visuais
	sameAs	button
awt		
navegador	conexo	view
	conexo	visualização
	conexo	web
	conexo	internet
	conexo	interface
	conexo	html
	conexo	interface gráfica
model-view-controller		
modelo visualização controle		
menus	tipoDe	componentes visuais
formulários	fazParte	página web
	tipoDe	componentes visuais
modelo	conexo	mvc
interface		
máquina virtual		
jsf	tipoDe	mvc
	sameAs	javaserver faces
web		
html	agrupa	tag
comboboxes	tipoDe	componentes visuais
eventos	conexo	mvc
	conexo	páginas dinâmicas
página web	conexo	aplicações para web
	sameAs	web
	conexo	javascript
	conexo	gráficas
	sameAs	páginas dinâmicas
	conexo	janelas
	tipoDe	interface
	tipoDe	view
	conexo	model view controller
	agrupa	html
	conexo	navegador
	sameAs	camada de apresentação
	agrupa	frames
	tipoDe	visualização
	linguagem	agrupa
agrupa		java
interface gráfica		
javascript	conexo	navegador
design pattern mvc		
xml	agrupa	tag
jface	tipoDe	componentes visuais
javaserver faces		
bordas	tipoDe	componentes visuais

Tabela 12: Termos e Relações Explícitas

#### 4.6 Recall e Precision

Este tópico tem por finalidade apresentar os resultados da execução do modelo e procedimentos que foram utilizados para calcular os percentuais de *recall* e *precision* do modelo apresentado. Após um teste preliminar, percebeu-se que a construção da *Query*, como definido no modelo original (Figura 17), retornou resultados pouco expressivos e decidiu-se modificar a construção da *Query* (Figura 27).

```
aparelh som^1.6 OR automovel^2,439 OR raphael^1.940 OR  
furt^1.0 OR pali^1.3 OR carr^1.790 OR registr^1.0
```

Figura 27: Exemplo de *Query* Modificada

Diferentemente da *Query* original, a modificada retorna uma quantidade alta de documentos, já que retira a obrigatoriedade da existência dos termos.

Os testes foram executados considerando-se o modelo original e o modelo modificado, sendo que os resultados estão representados a seguir. As fórmulas para calcular o *recall* e *precision* estão apresentadas na Figura 20 e os resultados estão apresentados na Tabela 15.

Os valores utilizados para cálculo (DocDR, DocR, DocI) foram conseguidos executando-se 14 buscas, sendo 7 para cada tipo de *Query*, cada uma com um dos documentos pré-selecionados pelo especialista (Tabelas 13 e 14).

Índice	DocR	DocI
1	1	0
2	1	0
3	2	0
4	1	0
5	1	0
6	1	0
7	1	0
<b>Média</b>	<b>1,14</b>	<b>0</b>

Tabela 13: Resultados de Recuperação da *Query* Original

Índice	DocR	DocI
1	4	6
2	5	5
3	5	5
4	4	6
5	5	5
6	7	3
7	6	4
<b>Média</b>	<b>5,14</b>	<b>4,86</b>

Tabela 14: Resultados de Recuperação da *Query* Modificada

		DocDR	DocR	DocI
		7	1,14	0
<b>Query Original</b>	<b>Recall</b>			<b>16%</b>
	<b>Precision</b>			<b>100%</b>
		7	5,14	4,86
<b>Query Modificada</b>	<b>Recall</b>			<b>73%</b>
	<b>Precision</b>			<b>5,4%</b>

Tabela 15: Resultados de *Recall* e *Precision*

## 5 CONSIDERAÇÕES FINAIS

### 5.1 Conclusões

A primeira conclusão a que se chega após o término desta pesquisa é que a relevância do tema fica confirmada a partir da quantidade de publicações sobre o assunto apresentadas e analisadas no tópico que demonstra uma matriz comparativa das técnicas publicadas na *Text Retrieval Conference*. Conforme apresentado neste mesmo tópico, o número de artigos referentes ao tema vem crescendo, desde o início da conferência em 1992, e o número de áreas de aplicação também.

Destaca-se, ainda, como representativo o número de empresas de grande porte e mundialmente conhecidas que estudam o assunto, tais como: Sun Microsystems, IBM e Microsoft. A tendência natural é que a aplicação desta pesquisa seja diversificada, atingindo diversas áreas do conhecimento e incrementando a capacidade de busca textual aos mais variados tipos de documentos.

A possibilidade de uso das técnicas *Term Extration* e *Query Expansion*, em conjunto, para o desenvolvimento de um Modelo Computacional que permita a busca de textos similares semanticamente, tida como hipótese inicial desta pesquisa, foi confirmada. O modelo apresentado permite que essas duas técnicas sejam *linkadas* e o modelo final é o resultado desta união. A utilização de ferramentas *Open Source* para suprir as funcionalidades marginais do protótipo demonstrou-se válida, já que representou um ganho no tempo de desenvolvimento.

O objetivo geral e os objetivos específicos desta tese foram alcançados, pois o objetivo geral estava diretamente ligado à hipótese da pesquisa e os específicos

foram necessários na criação do modelo e na implementação do protótipo usado na validação do mesmo e apresentado no tópico final desta pesquisa.

Salienta-se que a maioria dos modelos usados como base para esta pesquisa utilizam linguagem matemática, tornando seu entendimento mais complexo que o necessário. Nesta tese optou-se pela utilização da linguagem algorítmica, sendo este meio de formalização mais adequado ao tema da tese. Atesta-se, assim, a simplicidade do modelo criado, sendo esta característica de suma importância para a continuidade desta pesquisa.

Durante a fase de validação, confirmou-se uma propriedade do modelo que demonstra que a qualidade dos resultados está diretamente ligada à qualidade da Ontologia criada. Esta propriedade corrobora com a idéia de que existe uma fase onde o especialista pode ajustar a Ontologia para que a qualidade dos resultados seja melhorada.

A ocorrência dessa propriedade já era esperada, levando-se em consideração que a Ontologia tem por objetivo mapear os conceitos do especialista. Sendo assim, um especialista com um mapa mental de conceitos não bem definido tem menos condições de analisar textos que outro especialista que tenha mais experiência na área, e, portanto, um mapa mental mais concreto.

Os resultados corroboraram com as informações apresentadas por SIRIHAL (2005), que afirma que um bom sistema de recuperação textual deve trazer o maior número possível de documentos relevantes e o menor número de documentos não relevantes possível. Contudo, essas duas características são aparentemente contraditórias, já que as técnicas que melhoram a *recall* acabam reduzindo o *precision* e vice-versa.

De acordo com os resultados apresentados na validação do modelo, a *Query* originalmente criada apresentou bons resultados no que diz respeito à precisão, entretanto, os resultados relativos à recuperação foram pouco expressivos. Utilizando-se a *Query* modificada esses valores foram invertidos. Então, os resultados da *Query* original poderiam ser melhorados optando-se por utilizar técnicas mais avançadas de *Term Extration* como, por exemplo, a utilizada por Hawking (item 38, Quadro 5).

O uso de pontuação relativa a *links* entre documentos também poderia ser explorado, melhorando significativamente os resultados da busca como os utilizados por Toms (item 66, Quadro 5). Este tipo de informação não foi utilizada, pois os objetivos desta pesquisa estavam diretamente ligados a documentos textuais e não hipertextuais (documentos padrão *web* que contém *links* entre si). Este tipo de informação é utilizado em ferramentas que tem por objetivo principal buscar páginas na Internet como Google, Yahoo, entre outros.

A opção pela utilização da ferramenta Lucene, como base para a busca estatística, foi considerada acertada, já que a ferramenta demonstrou características as quais tornaram possível a incorporação de outros modelos, cobrindo quase que na totalidade as funcionalidades marginais necessárias para implementação do modelo concebido.

Todavia, o uso de outras ferramentas desenvolvidas com o mesmo propósito é bem vindo, já que esta ferramenta apresentou limitações no que diz respeito ao número de termos que uma *Query* pode possuir e não permitiu o cálculo do IDF de termos compostos. Como o modelo necessitava deste cálculo, foi utilizado um subterfúgio que minimizou o impacto desta limitação.

A ferramenta utilizada para construção e manutenção de Ontologias – Jena – cumpriu todos os requisitos, não apresentando nenhuma limitação no que tange o modelo apresentado nesta pesquisa. Entretanto, percebeu-se que, caso a Ontologia possua um número grande de termos, o que seria necessário para uma aplicação real, o tempo de processamento deixa a desejar, podendo tornar-se um problema em um sistema computacional em produção.

Salienta-se que o objetivo inicial era validar o modelo em áreas que não as da Computação. No entanto, depois de se obter diversas evasivas de universidades, o pesquisador decidiu validar o modelo na área de seu domínio, visto que a existência de pesquisadores nesta área interessados em validar o modelo tornou esta tarefa menos árdua.

Enfim, é possível afirmar que a construção desse tipo de ferramenta está se iniciando e que a finalidade das pesquisas é promover a construção de modelos computacionais que simulem o funcionamento de um especialista, caminhando-se, assim, para um futuro onde as ferramentas de busca textual deixem de “enxergar” os documentos não só como agregados de palavras, mas sim, como o que eles realmente são: agregados de conhecimento, desta forma, passando-se a entender o significado dos textos.

## 5.2 Recomendações para Futuros Trabalhos

As possibilidades de pesquisas advindas da conclusão deste trabalho são muitas. Sendo assim, como recomendações para futuros trabalhos a serem desenvolvidos acerca dos temas pesquisados nesta tese, têm-se:



- Modificar o modelo concebido para utilização de técnicas de *Term Extration* baseadas em Algoritmos de Linguagem Natural, com o objetivo de melhorar os indicadores de eficiência;
- Utilizar outras ferramentas para busca e indexação, que suportem as características não suportadas pela ferramenta Lucene, tais como: extração do IDF de termos compostos e número superior de termos na formação da *Query*;
- Utilizar outras ferramentas na criação e manipulação de Ontologias no formato OWL, que permitam a utilização de uma quantidade maior de termos sem aumentar a necessidade de processamento;
- Validar o modelo em outras áreas do conhecimento como o Direito, mais especificamente na busca por jurisprudências, e a Medicina, na busca por casos clínicos semelhantes.

## REFERÊNCIAS BIBLIOGRÁFICAS

- APACHE. **Apache Foundation**. <http://www.apache.org>. 2006.
- ALMEIDA, Mauricio B. BAX, Marcello P. **Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção**. IBICT – Ciência da Informação. 2003.
- BETTIO, Raphael. Bogo, Luis Henrique. Fulber Heleno. **Algoritmos de Stemming**. UFSC. Engenharia e Gestao do Conhecimento. 2004.
- BACCHINI, M. Melucci, M. **Symbol-Based Query Expasion Experiments at TREC 2005 Genomics Track**. Univesity of Padova. 2005.
- BILLERBECK, Bodo. Zobel, Justin. **Questioning Query Expansion: An Examination of ehaviour and Parameters**. School of Computer Science and Information Technology. RMTI University. Australia. 2006.
- BODO Billerdeck, Justin Zobel. **Efficient Query Expansion with Auxiliary Data Structures**. 2005.
- CASTELANO, S. Ferrara, A. Montanelli S. **H-MATH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems**. Università degli Studi di Milano. Italy. 2005.
- CRAWFORD, Richard. **Na era do capital humano**. São Paulo: Atlas, 1994.
- CUI, Hang. Wen, Ji-Rong. Nie, Jian-Yun. Ma, Wei-AYing. **Probabilistic Query Expansion Using Query Logs**. Tianjin University, Microsoft Research Asia, University of Montreal. 2005.
- DENNIS, Simon. **Stemming Algorithms for Information Retrieval and Question/Answer Systems**. Institute of Cognitive Science University of Colorado. 2000.
- E. Milios, Y. Zhang, B. He and L. Dong. **Automatic Term Extraction and Document Similarity in Special Text Corpora**. Faculty of Computer Science, Dalhousie University, Halifax, Canada B3H 1W5. 2006.
- EIJA, Airo. Kalervo Järvelin, Pirkko Saatsi. **CIRI – ONTOLOGY-BASED QUERY INTERFACE FOR TEXT RETRIEVEL**. 2005.
- FRANCOPOULO, Gil. **Pruning Texts with NLP and Expanding Queries with an Ontology: TagSearch**. [www.tagmatica.com](http://www.tagmatica.com). 2005.
- FRANTZIY, Katerina. Ananiadouy, Sophia. Mimaz, Hideki. **Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method**. Centre for Computational Linguistics, Manchester and Dept. of Information Science, University of Tokyo. 2000.

GROOTJEN F.A., van der Weide Th.P. **Conceptual query expansion**. Faculty of Science, Mathematics and Computer Science, Radboud University Nijmegen. The Netherlands. 2004.

GRUBER, T. **What is an Ontology?** <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>, 2005.

HARMELEN, F. V; MCGUINNESS, D. L. **OWL Web Ontology Language Overview**. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, 2005.

HYPERGRAPH. **HyperGraph Project**. <http://hypergraph.sourceforge.net>. 2007

JELIER, R. Schuemie, Eijk van der C, Weeber M. Mulligen van E. Schijvenaars B. Mons B. Kors J. **Concept-Based Query Expansion and Bayes Classification**. Erasmus University Medical Center Rotterdam. 2005.

JENA. **A Semantic Web Framework for Java**. <http://jena.sourceforge.net>. 2007

JINAN Fiaidhi, Sabah Mohammed, Jihad Jaam, Ahmad Hasnah. **A Standart Framework for Personalization Via Ontology-Based Query Expasion**. 2005.

LIMA, César Júnio, CARVALHO, Cedric L. Ontologia – **OWL (Web Ontology Language)** – Instituto de Informática – Universidade Federal de Goiás. 2005.

MANDALA, Rila. Tokunaga Takenobu, Tanaka Hozumi. **Combining Multiple Evidence from Different Types of Thesaurus for Query Expasion**. Department of Computer Science. Tokyo Institute of Technology. 2007.

METZER, D. Diaz F. Strohman T. Croft W. B. **UMass Robust 2005: Using Mixtures of Relevance Models for Query Expansion**. University of Massachusetts. 2005.

MONTEIRO, Manuel. **Uma História da Internet..** 2006

MADAHHAIN. Joshua O. **Fuzzy Term Expansion and Document Reweighting**. 2001.

NAKAGAWA, Hiroshi. Mori, Tatsunori. **A Simple but Powerful Automatic Term Extration Method**. 2005.

NAVIGLI, Roberto. Velardi, Paola. **An Analysis of Ontology-based Query Expansion Strategies**. Dipartimento di Infomatica. Università di Roma. 2005.

OLIVEIRA, José Manuel Godinho. **Web Intelligence**. 2004.

ORENGO, V. M. HUYCK, C. **A Stemming Algorithm for Portuguese Language**. Proceedings of Eighth Symposium of String Processing and Information Retrieval. Chile, 2001.

PANTEL, Patrick. Lin, Dekang. **A Statistical Corpus-Based Term Extractor**. Departament of Computing Science. Univesity of Alberta. Canada. 2006.

PONCHIROLI, Osmar. **O capital humano como elemento estratégico na economia da sociedade do conhecimento sob a perspectiva da teoria do agir comunicativo.** Florianópolis 2000.

PIRKOLA, A. **The Effects of Primary Keys, Bigram Phrases and Query Expansion on Retrieval Performance.** University of Tapere (UTA) 2005.

PORTER, Martin. <http://www.tartarus.org/~martin/PorterStemmer/def.txt>. 1980.

PORTER, Martin. **The Porter Stemming Algorithm.** <http://www.tartarus.org/~martin/PorterStemmer/> 2006.

RACHEL, Tsz-Wai Lo, He Ben, Ounis Iadh. **Automatically Building a Stopword List for an Information Retrieval System.** Department of Computing Science. University of Glasgow. UK. 2004.

RILA Mandala, Takenobu Tokunaga, and Hozumi Tanaka. **Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion.** Department of Computer Science Tokyo Institute of Technology.

ROBERTSON, Stephen. **Understanding Inverse Document Frequency: On Theoretical arguments for IDF.** Microsoft Research. 2004.

RODRIGUEZ y Rodriguez, Martius Vicente. **Gestão do conhecimento: reinventando a empresa para uma sociedade baseada em valores intangíveis.** Rio de Janeiro: IVPI Press, 2001.

SIGNAL Amit. **Modern Information Retrieval: A Brief Overview.** Google Inc. 2006.

SILVA, Edna Lúcia da; Menezes, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação.** 3ª Edição. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

SIRIHAL, Adriana Bogliolo, Lourenço, Cíntia de Azevedo. **Information and Knowledge: philosophical and informational aspects.** 2005.

TREC. Text REtrieval Conference. <http://trec.nist.gov/>. 2006.

W3C. <http://www.w3c.org>. 2005.

WikiPedia. [http://pt.wikipedia.org/wiki/Copa\\_do\\_Mundo](http://pt.wikipedia.org/wiki/Copa_do_Mundo). 2005.

WGLS (WORKING GROUP ON LIBRE SOFTWARE). **Free Software / Open Source: Information Society Opportunities for Europe?** Abril 2000.