

**UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE FILOSOFIA E CIÊNCIAS, CAMPUS DE MARÍLIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO**

JORGE JANAITE NETO

**Recuperação de Informação Baseada em Ontologia:
Uma proposta utilizando o Modelo Vetorial**

MARÍLIA - SP
2018

**UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE FILOSOFIA E CIÊNCIAS, CAMPUS DE MARÍLIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO**

JORGE JANAITE NETO

Recuperação de Informação Baseada em Ontologia: Uma proposta utilizando o Modelo Vetorial

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Informação da Faculdade de Filosofia e Ciências - Universidade Estadual Paulista “Júlio de Mesquita Filho” – UNESP, campus de Marília, como requisito parcial para obtenção do título de Mestre em Ciência da Informação.

ÁREA DE CONCENTRAÇÃO: Informação, Tecnologia e Conhecimento.

LINHA DE PESQUISA: Informação e Tecnologia.

ORIENTADOR: PROF. DR. EDBERTO FERNEDA

MARÍLIA – SP
2018

J33r Janaite Neto, Jorge.
Recuperação de informação baseada em ontologia:
uma proposta utilizando o modelo vetorial / Jorge Janaite
Neto. – Marília, 2018.
105 f. ; 30 cm.

Orientador: Edberto Ferneda
Dissertação (Mestrado em Ciência da Informação) –
Universidade Estadual Paulista (Unesp), Faculdade de
Filosofia e Ciências, 2018.
Bibliografia: f. 98-105

1. Recuperação da informação. 2. Ontologias
(Recuperação da informação). 3. Indexação automática. I.
Título.

CDD 005.73

Elaboração: André Sávio Craveiro Bueno
CRB 8/8211

Unesp – Faculdade de Filosofia e Ciências

JORGE JANAITE NETO

Recuperação de Informação Baseada em Ontologia: Uma proposta utilizando o Modelo Vetorial

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Informação da Faculdade de Filosofia e Ciências - Universidade Estadual Paulista “Júlio de Mesquita Filho” – UNESP, campus de Marília, como requisito parcial para obtenção do título de Mestre em Ciência da Informação.

ÁREA DE CONCENTRAÇÃO: Informação, Tecnologia e Conhecimento.

LINHA DE PESQUISA: Informação e Tecnologia.

ORIENTADOR: PROF. DR. EDBERTO FERNEDA

Prof. Dr. Edberto Ferneda (orientador)
Faculdade de Filosofia e Ciências, UNESP-Marília

Dr^a. Rachel Cristina Vesu Alves
Faculdade de Filosofia e Ciências, UNESP-Marília

Dr. Rogério Aparecido Sá Ramalho
Universidade Federal de São Carlos - UFSCar

Marília, 30 de maio de 2018

Agradecimentos

Agradeço especialmente a:

Edberto Ferneda

Giselli Hara Macedo

Rachel Cristina Vesu Alves

“Aparentemente as pessoas não gostam da verdade, mas eu gosto,
eu gosto dela porque incomoda muita gente”

Lemmy Kilmister (1945-2015†) - Motörhead

Resumo

JANAITE NETO, Jorge. **Recuperação de Informação Baseada em Ontologia: Uma proposta utilizando o Modelo Vetorial**. Marília, 2018. 103f. Dissertação (mestrado). Programa de Pós-Graduação em Ciência da Informação — Universidade Estadual Paulista (Unesp), Faculdade de Filosofia e Ciências, Marília, 2018.

A recuperação de informação ocorre por meio da comparação entre as representações dos documentos de um acervo e a representação da necessidade de informação do usuário. Um documento é recuperado quando sua representação coincidir total ou parcialmente com a representação da necessidade de informação do usuário. O processo de recuperação de informação pode ser visto como um problema linguístico no qual o conteúdo informacional dos documentos e a necessidade de informação do usuário são representados por um conjunto de termos. A eficiência do processo de recuperação de informação depende da qualidade das representações dos documentos e dos termos empregados pelo usuário para representar sua necessidade de informação. Quanto mais compatíveis forem essas representações maior será a eficiência do processo de recuperação. A partir de uma pesquisa exploratória e descritiva fundamentada em bibliografia específica, este trabalho propõe a utilização de ontologias computacionais em sistemas de recuperação de informação baseados no Modelo Espaço Vetorial. As ontologias são empregadas como estrutura terminológica externa utilizadas tanto na expansão dos termos de indexação quanto na expansão dos termos que compõe a expressão de busca. A expansão dos termos de indexação é feita logo após a extração dos termos mais representativos do documento em análise durante o processo de indexação, consistindo na adição de novos termos conceitualmente relacionados a fim de enriquecer a representação do documento. A expansão da consulta é obtida a partir da adição de novos termos relacionados aos já existentes na expressão de busca com o objetivo de melhor contextualizá-los. Nesta proposta utiliza-se apenas a estrutura terminológica e hierárquica oferecida por uma ontologia computacional OWL, sem considerar os demais tipos de relações possíveis nem as restrições lógicas que podem ser descritas, podendo esses recursos serem utilizados em trabalhos futuros na tentativa de melhorar ainda mais a eficiência do processo de recuperação. A proposta apresentada neste estudo pode ser implementada e futuramente tornar-se um sistema de recuperação de informação totalmente operacional.

Palavras-chave: recuperação de informação; ontologia; indexação automática; expansão de consulta; OWL; OWL2.

Abstract

JANAITE NETO, Jorge. **Ontology based Information Retrieval: A proposal using the Vector Space Model**. Marília, 2018. 103f. Dissertation (master). Post-Graduate Program in Information Science — São Paulo State University (Unesp), School of Philosophy and Sciences, Marília, 2018.

The information retrieval occurs by means of match between the representations of documents from a collection and the representation of user information's needs. A document is retrieved when its representation matches totally or partially to the user information's needs. The process of information retrieval can be seen as a linguistic issue in which the document information content and the user information need are represented by a set of terms. Its efficiency depends on the quality of the representations of the documents and the terms used to represent the user's information need. The more compatible these representations were, the more efficient the retrieval process. Based on an exploratory and descriptive research substantiated in a specific bibliography, this paper offers to use computational ontologies in information retrieval systems based on the Vector Space Model. The ontologies are applied as external terminological structures used in the indexing terms expansion as well as in the expansion of the terms which compound the query expression. The indexing terms expansion is made as soon as the extraction of the more representative terms of the document in analysis during the indexing process, consisting on the adding of new conceptually related terms in order to improve the document representation. Query expansion is obtained from adding new related terms to the existent ones in the query expression to better contextualize them. In this propose, only the terminological and hierarchical structure offered by an OWL computational ontology was used, regardless other possible relations and logical restrictions that could be described, saving these resources to be used in further works in an attempt to improve the retrieval process efficiency. The shown proposition can be implemented and become a fully operational information retrieval system.

Keywords: information retrieval; ontology; automatic indexing; query expansion; OWL; OWL2.

Lista de Figuras

Figura 1 — Processo de Recuperação de Informação	27
Figura 2 — Exemplo gráfico do cálculo de similaridade por cosseno entre vetores	35
Figura 3 — Lei de Zipf.....	50
Figura 4 — Comportamento não linear do índice IDF de um termo	52
Figura 5 — Representação da Necessidade de Informação	56
Figura 6 — Métodos de Expansão de Consulta	57
Figura 7 — OWL exemplo declaração de prefixo	71
Figura 8 — OWL exemplo prefixo default	72
Figura 9 — OWL exemplo sem o uso de prefixo.....	72
Figura 10 — OWL exemplo de estrutura completa de uma ontologia.....	73
Figura 11 — OWL exemplo de Declarações.....	76
Figura 12 — OWL exemplo de Propriedade Funcional de Objetos.....	77
Figura 13 — OWL exemplo de Propriedades de Dados	78
Figura 14 — OWL exemplo de Asserção de Anotação	78
Figura 15 — OWL exemplo de Asserção de Anotação do tipo rdfs:label	79
Figura 16 — OWL exemplo de Axiomas de Classe	80
Figura 17 — OWL exemplo de Axiomas de Classe do tipo União Disjunta.....	81
Figura 18 — OWL exemplo de Definição de indivíduos.....	81
Figura 19 — OWL exemplo estrutura das classes	82
Figura 20 — OWL exemplo completo utilizando a sintaxe funcional.....	82
Figura 21 — Exemplo de termos extraídos com seus pesos atribuídos	88
Figura 22 — Exemplo de ontologia com termos hierárquicos.....	89
Figura 23 — Fórmula do peso final de um termo	90
Figura 24 — Representação vetorial do documento	91
Figura 25 — Representação vetorial da expressão de busca.....	93

Lista de Quadros

Quadro 1 — Exemplo de pesos termo vs documento	33
Quadro 2 — Primeiros passos do Porter Stemming	48
Quadro 3 — Exemplo frequência de termo vs documento	53
Quadro 4 — Exemplo de termo vs documento, ponderado pelo IDF	54
Quadro 5 — Matriz de frequência dos termos da expressão de busca	54
Quadro 6 — Prefixos do vocabulário da OWL	72
Quadro 7 — OWL Tipos de Axiomas	74
Quadro 8 — OWL Axiomas de Declaração	75
Quadro 9 — Cálculo da similaridade entre documentos e expressão de busca	94

Lista de Tabelas

Tabela 1 — Cálculo dos pesos dos termos derivados	90
Tabela 2 — Cálculo dos pesos dos termos da busca	92
Tabela 3 — Matriz termo/documento	93

Sumário

1	Introdução.....	14
1.1	Problema e hipótese de pesquisa	17
1.2	Objetivo	17
1.3	Objetivos específicos	17
1.4	Metodologia.....	18
1.5	Da terminologia utilizada	19
1.6	Trabalhos relacionados	19
1.7	Organização do trabalho	22
2	Recuperação de Informação.....	24
2.1	O processo de Recuperação de Informação.....	26
2.1.1	Documentos (corpus)	27
2.1.2	Representação dos documentos.....	27
2.1.3	Usuário	28
2.1.4	Expressão de busca	28
2.1.5	Representação da expressão de busca	29
2.1.6	Função de busca	29
2.1.7	Resultado da busca.....	30
2.2	Modelo Espaço Vetorial	30
2.2.1	Pressupostos do Modelo Vetorial.....	31
2.2.2	Fórmula da similaridade por cosseno.....	32
2.2.3	Exemplos de uso da fórmula da similaridade	33
2.2.4	Fórmula da similaridade de Jaccard.....	36
3	Indexação Automática	38
3.1	O processo de indexação	38
3.2	Os precursores da Indexação Automática.....	40
3.3	A extração automática dos termos.....	42
3.4	Um processo completo para a extração dos termos.....	45
3.4.1	<i>Stemming</i>	46
3.4.2	Stopword List.....	49
3.4.3	Atribuição automática de pesos aos termos	51

4	Expansão de Consulta.....	55
4.1	O conceito de relevância.....	55
4.2	Métodos de expansão de consulta.....	57
4.2.1	Expansão de consulta baseada nos resultados da busca.....	59
4.2.2	Expansão de consulta baseada em estruturas de conhecimento dependentes do <i>corpus</i>	60
4.2.3	Expansão de consulta baseada em estruturas de conhecimento independentes do <i>corpus</i>	61
4.3	Expansão de consulta baseada em ontologias	61
5	Ontologias	63
5.1	Ontologias computacionais e a Linguagem OWL.....	67
5.2	Sintaxes para OWL.....	69
5.3	Sintaxe Funcional OWL	70
5.3.1	Prefíxos na OWL.....	71
5.3.2	Estrutura da ontologia	72
5.3.3	Declarações implícitas	73
5.3.4	Axiomas	74
5.3.5	Definição dos indivíduos.....	81
5.4	Ontologia OWL de exemplo.....	81
6	Proposta de utilização de ontologias na recuperação de informação	86
6.1	Indexação automática de documentos	87
6.1.1	Associando uma ontologia ao documento.....	87
6.1.2	Extração de termos	87
6.1.3	Atribuição de conceitos	88
6.1.4	Lista de termos potenciais	91
6.2	Especificação da busca	91
6.2.1	Escolha da ontologia	91
6.2.2	Expansão da consulta	92
6.2.3	Lista de termos potenciais	94
7	Considerações finais.....	96

1

Introdução

O termo “recuperação de informação” (*information retrieval*) foi definido por Calvin Mooers em 1951. Neste trabalho, Mooers também define os problemas a serem abordados por esta que se tornaria uma nova disciplina e um campo de pesquisa. Desde seu surgimento, a Recuperação de Informação é uma área voltada aos aspectos intelectuais bem como aos sistemas, técnicas e máquinas envolvidos no acesso à informação. A definição dada por Mooers (MOOERS, 1951) é importante, pois situa o termo de uma maneira precisa, para que não existam ambiguidades com outras atividades de gerenciamento do conhecimento que também objetivam facilitar o acesso à informação, já consagradas e em uso naquela época.

Fundamentalmente a recuperação de informação é feita por meio da comparação entre as representações dos documentos de um acervo e a representação da necessidade de informação do usuário que busca por documentos que venham a atender tal necessidade. Um documento é recuperado somente se a sua representação coincidir total ou parcialmente com a representação da necessidade do usuário.

Na maioria dos sistemas de recuperação de informação e no contexto deste trabalho as representações dos documentos e a representação das necessidades dos usuários são formalizadas textualmente. Cada documento é representado por um conjunto de termos (termos de indexação) que têm como objetivo sintetizar o seu conteúdo informacional. No outro extremo do processo há o usuário que traduz a sua necessidade por meio da especificação de um conjunto de termos (termos de busca).

Em sistema de recuperação de informação as representações dos documentos e das necessidades de informação do usuário precisam ser formalmente especificadas em uma estrutura que permita realizar comparações. Um modelo de recuperação de informação é a

especificação formal de três elementos: a representação dos documentos, a representação da necessidade de informação do usuário por meio de sua expressão de busca e a função de busca. No centro do processo de recuperação de informação está a função de busca, que compara as representações dos documentos com a expressão de busca dos usuários e recupera os itens que supostamente fornecem a informação que o usuário procura. (FERNEDA, 2003, p. 18-20).

Desta forma os Modelos de Recuperação de Informação estabelecem uma estrutura uniforme de representação dos documentos e das buscas permitindo calcular o grau de semelhança entre tais representações. Os documentos resultantes de uma busca podem assim ser ordenados (ranqueados) segundo essa semelhança. Este ranqueamento reflete o quanto um determinado documento poderá, supostamente, ser relevante para satisfazer a necessidade de informação do usuário.

O conceito de relevância parte do pressuposto de que existe dentro do acervo documental um subconjunto ideal de documentos contendo todos aqueles que satisfazem, mesmo que parcialmente, a necessidade de informação do usuário. Esse subconjunto possui uma ordenação ideal segundo o grau de relevância presumido de cada documento. O ranqueamento torna-se indispensável quando o número de documentos recuperados for elevado, não permitindo a avaliação individual da relevância de cada documento pelo usuário. A capacidade de ranquear os resultados está presente em alguns Modelos de Recuperação como é o caso do Modelo Espaço Vetorial adotado neste trabalho.

A eficiência do processo de Recuperação de Informação é mensurada por meio das chamadas *medidas de avaliação* que consistem em uma análise numérica do conjunto de documentos recuperados em relação à totalidade do acervo documental por meio da avaliação de relevância do usuário. Tentam basicamente mensurar qual é a assertividade dos resultados em relação à necessidade do usuário, índice denominado *precision*¹, bem como qual é a abrangência dos resultados em relação ao todo, este último índice é denominado *recall*². É importante destacar que esta análise é feita somente após a recuperação do primeiro conjunto de resultados. Essa eficiência depende da qualidade das representações dos documentos

¹ A **precisão** (*precision*) mede a capacidade de um sistema em recuperar **apenas documentos relevantes** para uma determinada busca. É calculada pela divisão entre o número de documentos relevantes recuperados e o número de documentos recuperados.

² A **revocação** (*recall*) é calculada pela divisão entre o número de documentos relevantes recuperados e o número total de documentos relevantes existentes no *corpus*. Portanto, a **revocação** mede a capacidade de um sistema em recuperar **todos os documentos relevantes** existentes no *corpus* para uma determinada busca.

(resultante do processo de indexação) e dos termos empregados pelo usuário para representar de sua necessidade de informação.

Sobre a qualidade da indexação, Gil-Leiva e Fujita (2012, p.78) apresentam quatro elementos que caracterizam tanto o processo quanto o resultado da indexação: exaustividade, especificidade, correção e consistência. (1) *exaustividade* está relacionada com a quantidade de conceitos que caracterizam o conteúdo do documento. (2) *especificidade* refere-se à precisão com que um termo de indexação representa fielmente um conceito particular que aparece em um determinado documento. (3) *correção* é a utilização de termos de indexação relevantes para evitar omissões ou inclusões de termos sem necessidade. (4) *consistência* na indexação refere-se ao grau de consenso na escolha dos termos de um documento quando considerado cada um dos indexadores de um grupo.

A especificação da busca é dependente do usuário, do domínio que este usuário tem sobre a terminologia da área ou assunto de interesse. Geralmente as buscas dos usuários são expressas por meio de um número reduzido de termos, não permitindo uma interpretação exata e inequívoca de sua necessidade de informação. Esta especificação pode ser melhorada com o auxílio de ferramentas que modifiquem a expressão de busca do usuário, acrescentando novos termos originários de uma estrutura terminológica externa, chegando a uma expressão de busca mais representativa da necessidade de informação do usuário e, conseqüentemente, melhorando a qualidade dos resultados obtidos.

Este trabalho propõe a utilização da estrutura terminológica das ontologias para melhorar a representação dos documentos e a representação das necessidades de informação do usuário a partir da melhoria de sua expressão de busca.

Com sua origem na Filosofia, o conceito de ontologia foi apropriado pela Ciência da Computação para referir-se a “um conjunto de conceitos e termos que podem ser usados para descrever alguma área do conhecimento ou construir sua representação” (MOREIRA, 2010, p. 51). Neste sentido, as ontologias são artefatos computacionais contendo a representação lógica do conhecimento de uma determinada área do saber. Podem ser vistas como estruturas terminológicas que permitem deduções lógicas e navegabilidade conceitual. Isto possibilita a elaboração de listas contendo termos hierarquicamente relacionados (relações verticais) e termos que possuem relações associativas (relações horizontais), permitindo a especificação precisa sobre o tipo de relação existente entre os termos de uma determinada lista.

As ontologias se popularizaram a partir da proposta da Web Semântica, apresentada publicamente na edição de maio de 2001 da revista *Scientific American* (BERNERS-LEE; HENDLER; LASSILA, 2001). Embora o projeto da Web Semântica ainda não foi realizado em sua totalidade, as tecnologias envolvidas se desenvolveram de forma independente, sendo aplicadas em outros domínios. Esse é o caso das pesquisas com ontologias computacionais, que antecedem a proposta da Web Semântica, mas ganham maior atenção da comunidade científica após o surgimento desta e atualmente são aplicadas em diversas áreas de pesquisa, entre elas a Recuperação de Informação.

1.1 Problema e hipótese de pesquisa

Conforme o contexto destacado anteriormente, o problema de pesquisa apresenta o seguinte questionamento: Como utilizar as ontologias em sistemas de recuperação da informação baseados no Modelo Espaço Vetorial?

Como hipótese de pesquisa considera-se que a estrutura terminológica das ontologias pode possibilitar um enriquecimento das representações dos documentos e, portanto, pode ser empregada como uma ferramenta para auxiliar o usuário na tradução de sua necessidade de informação em um conjunto de termos de busca assim como servir para enriquecer a indexação de um documento.

1.2 Objetivo

O objetivo da pesquisa é propor um modelo de utilização das ontologias em sistemas de recuperação de informação baseados no Modelo Espaço Vetorial, buscando maior eficiência no processo de recuperação da informação.

1.3 Objetivos específicos

- Abordar as questões de recuperação da informação;
- Elucidar o Modelo Espaço Vetorial;
- Esclarecer sobre os aspectos de indexação automática;
- Explicar as questões de expansão de consulta;

- Analisar o uso das ontologias como estrutura terminológica processável por computador, que auxiliam na melhoria das representações dos documentos e das buscas.

1.4 Metodologia

A metodologia utilizada no trabalho se caracterizou, no que se refere a sua natureza, como sendo uma pesquisa exploratória e descritiva. O caráter exploratório teve o objetivo de obter uma nova percepção dos fenômenos e descobrir novas ideias que possam contribuir para o esclarecimento dos objetivos propostos. O caráter descritivo da pesquisa busca caracterizar e correlacionar as variáveis do estudo sem manipulá-las, que se constituem nos objetos de estudo desta pesquisa (CERVO; BERVIAN, 2003). Trata-se de uma pesquisa qualitativa, quanto a abordagem do problema (SILVA; MENEZES, 2005), que buscou analisar e elucidar o uso das ontologias no processo de busca e recuperação da informação.

Em relação aos procedimentos técnicos e metodológicos para coleta de dados, a pesquisa caracteriza-se como bibliográfica (SILVA; MENEZES, 2005). Deste modo, o levantamento bibliográfico foi realizado a partir dos seguintes critérios: a) em relação aos temas de pesquisa: buscou-se pelos temas recuperação da informação, indexação automática, expansão de consulta, ontologias, modelo espaço vetorial; b) em relação aos tipos de materiais bibliográficos: livros, periódicos, anais de congressos, teses e dissertações, sites especializados sobre o tema, base de dados nacionais e internacionais (ex.: portal de periódicos Capes, repositórios digitais, etc); c) em relação a seleção de materiais: foi considerado os idiomas inglês e português; nos principais autores que se destacam na área sobre o tema, como por exemplo, Gerard Salton, Hans Peter Luhn, Stefano Mizzaro, Efthimis Efthimiadis, Mike Uschold, Pascal Hitzler, Markus Krötzsch; e, nos documentos dos últimos 10 anos assim como alguns documentos clássicos com mais de 30 anos.

Assim, os procedimentos metodológicos da pesquisa apresentam as seguintes etapas:

- 1) *Levantamento bibliográfico*: busca de materiais bibliográficos relacionados ao tema de pesquisa, de acordo com os critérios apontados anteriormente;
- 2) *Leitura e análise dos textos*: após a seleção inicial dos textos foi realizada a leitura dos materiais e posterior análise das principais características identificadas na literatura, para criar o referencial teórico da pesquisa e esclarecer o problema estabelecido;

- 3) *Sistematização do estudo exploratório*: reunião das características identificadas na literatura para esclarecer o problema de pesquisa e atingir os resultados por meio dos objetivos propostos;
- 4) *Elaboração da redação final da pesquisa*: para divulgação à comunidade científica dos resultados alcançados e considerações da pesquisa.

1.5 Da terminologia utilizada

O tema principal deste trabalho, Recuperação de Informação, envolve dois campos científicos: a Ciência da Informação e a Ciência da Computação. Disso podem surgir problemas terminológicos decorrentes das diferentes nomenclaturas utilizadas para um mesmo conceito.

Considerando que este trabalho envolve interesses de investigação em Ciência da Informação, a terminologia utilizada será preferencialmente dessa área. Porém, alguns termos originários da Ciência da Computação já estão consolidados e são amplamente utilizados em diversos domínios científicos. Nesse caso a preferência será pelo termo mais comumente utilizado.

1.6 Trabalhos relacionados

Dentro desta temática é possível encontrar na literatura diversas pesquisas sobre recuperação de informação baseada em ontologia. A seguir serão apresentadas algumas destas pesquisas.

Sistema *OntoSeek* (GUARINO; MASOLO; VETERE, 1999): recuperação de informação contida em páginas amarelas e catálogos on-line. Assume que o idioma empregado neste tipo de conteúdo é uma linguagem natural genérica que utiliza um vocabulário técnico bem detalhado, por isso o sistema utiliza uma ontologia de fundamentação (*The Penman Upper Model*) associada a uma base lexicográfica extensa (*WordNet*). Tanto os documentos quanto as consultas dos usuários são representados por meio de grafos conceituais originários da ontologia. O problema da recuperação fica restrito a busca, comparação e ranqueamento entre os nós destes grafos.

Sistema *On-Air – Ontology-Aided Information Retrieval* (PAZ-TRILLO; WASSERMANN; BRAGA, 2005): recuperação de informação em curtos fragmentos de vídeo de longa duração utilizando linguagem natural. A partir de uma ontologia a respeito de arte

contemporânea, muitas horas de entrevistas em vídeo com a artista brasileira Ana Teixeira foram indexados por palavras-chave atribuídas por um especialista do domínio em conjunto com aquelas extraídas da transcrição do vídeo. Cada vídeo é dividido em diversos trechos, sendo cada trecho indexado individualmente. O sistema permite consulta em texto livre, realizando a atribuição automática de pesos aos termos consultados, expandindo a consulta com a utilização de termos encontrados por meio das relações estabelecidas na ontologia.

Segura *et al.* (2011) descreve a expansão de consultas, utilizando ontologias, em Repositórios de Objetos de Aprendizagem (*Learning Objects Repositories – LOR*). O procedimento consiste em: primeiramente extrair os conceitos da consulta inicial; tais conceitos são utilizados para gerar novas consultas. Estas são todas as consultas que provavelmente o usuário com o mesmo interesse temático faria; tais consultas têm seus termos expandidos por meio de uma ontologia; em seguida todas são submetidas ao repositório MERLOT³, sendo todos os resultados cruzados entre si e eliminados aqueles que aparecem uma única vez. Os demais ganham um aumento na posição na lista de resultados de forma proporcional ao número de vezes que aparecerem duplicados em cada conjunto. Esta lista de resultados é apresentada ao usuário que pode categorizá-los como relevantes, parcialmente relevantes e irrelevantes; este feedback será utilizado pelo sistema para recalcular os pesos de uma forma individualizada (cada usuário possui um perfil cadastrado) bem como de forma global. Este procedimento foi testado com o uso de dicionários (como o *WordNet*) na tentativa de expandir os termos com sinônimos, mas isso apenas adicionou, segundo os autores, ambiguidade. Concluem que a expansão de consultas baseada em ontologias aumenta a taxa de novidade (fração dos documentos relevantes no conjunto resposta que não são conhecidas pelo usuário) mantendo os níveis de precisão similares.

Fernandez *et al.* (2011) propuseram um sistema de Recuperação de Informação não apenas apoiado em palavra-chave, mas que tenta considerar os conceitos envolvidos na expressão de busca como critérios. O núcleo do sistema é baseado nos modelos clássicos de Recuperação de Informação compreendendo as fases de indexação, elaboração da consulta, busca e ordenação por relevância da lista contendo os resultados; o diferencial do método proposto está na consulta que é expressa em termos de uma linguagem específica de consulta a ontologias (*SPARQL*), utilizando recursos externos no apoio à indexação e no processamento da consulta. O sistema proposto recebe a expressão de busca e a transforma em uma consulta *SPARQL*. Esta consulta é executada sobre a base de conhecimento representada pela ontologia,

³ <http://www.merlot.org>

retornando uma lista de entidades que satisfaçam os critérios da consulta, nesta fase o processo de consulta é booleano. Os documentos que, durante a fase de indexação, foram associados às entidades da ontologia, são recuperados, ranqueados e apresentados ao usuário. Este último processo faz uma busca aproximada (não booleana), que permite encontrar resultados aproximados e estabelecer um nível de casamento entre o que foi buscado e o que foi encontrado, que é utilizado como critério para o ranqueamento. A indexação é vista como um processo de anotação semântica baseada em ontologias, no índice invertido estão associadas as entidades semânticas da ontologia aos termos do documento. Concluem que os modelos clássicos de recuperação também podem se beneficiar das abordagens desenvolvidas para a Web Semântica envolvendo representação do conhecimento de forma externa ao sistema.

HAHM *et al.* (2014) propuseram um *framework* de busca semântica, que inclui uma abordagem personalizada da expansão de consulta para engenheiros. Ele considera os interesses do usuário, visando, ao mesmo tempo, um serviço personalizado e um método de aprendizado para uma ontologia baseado no perfil do usuário. Um perfil de usuário é gerado para cada um dos utilizadores (engenheiros) e, baseado na ontologia de domínio através da atribuição específica, são dadas pontuações (pesos para a expansão dos termos), chamados valores de preferência, para cada indivíduo. Uma ontologia de Engenharia foi construída usando o *software Protegé*, definindo conceitos, relacionamentos e hierarquias para busca semântica. Os documentos foram processados com técnicas de desambiguação e indexados. A eficácia deste *framework* foi comparada com as de outros quatro sistemas e seus resultados baseados no índice *Mean Average Precision* (MAP) comprovaram a melhoria na recuperação da informação contida em documentos de Engenharia.

JIMENO-YEPES; BERLANGA-LLAVORI; REBHOLZ-SCHUHMANN (2010) utilizando as ontologias como base para uma normalização conceitual, propõem um modelo de consulta denominado por eles de *ontology query model* (OQM). O modelo consiste em gerar as consultas a partir dos conceitos presentes na ontologia e selecionados pelo usuário. Na implementação utilizaram o pacote de software Lemur com uma ontologia biomédica, dois conjuntos de dados (um sobre doenças genéticas e outro sobre interação de proteínas), e fizeram a validação utilizando o *TREC 2005 Genomics collection*. Os resultados relatados na melhoria da recuperação foram promissores e a metodologia empregada permitiu identificar conhecimento incompleto ou ausente na base de conhecimento.

No domínio biomédico, Dinh e Tamine (2012) utilizaram quatro fontes termino-ontológicas: o tesauro *Medical Subject Headings* (MeSH), um vocabulário

padronizado desenvolvido pela *National Library of Medicine*; o *Systematized Nomenclature of Medicine* (SNOMED) padronizado pelo *College of American Pathologists*; a *International Statistical Classification of Diseases* (ICD-10) que é uma lista de classificação médica para a codificação de doenças, sinais e sintomas, achados anormais, reclamações sociais, circunstâncias e causas externas de lesões ou doenças, mantida pela Organização Mundial de Saúde (OMS); o *Gene Ontology* (GO) e duas coleções TREC *Genomics* 2004 e TREC *Genomics* 2005 (compõem um subconjunto de quase 4.6 milhões de citações do MEDLINE de 1994 a 2003, sob a plataforma de recuperação de informação *Terrier*. As expansões das consultas foram realizadas com sinônimos, abreviações e termos hierarquicamente relacionados, sendo a lista com os documentos recuperados ordenada por meio do modelo probabilístico BM25 de peso dos termos para *Document Expansion* (DE) e/ou *Query Expansion* (QE). Os resultados (MAP) demonstraram melhora significativa com relação às técnicas clássicas de Recuperação de Informação. Extração automática com base em uma monoterminologia ou várias terminologias poderia ser uma forma eficaz de melhorar o desempenho de RI.

Zenz *et.al.* (2009) descrevem um sistema computacional denominado “QUICK” cujo objetivo é auxiliar usuários a criarem consultas semânticas em um determinado domínio. O sistema trabalha com bases de conhecimento elaboradas no modelo de dados *Resource Description Framework* (RDF) e *RDF Shema* (RDFS), podendo operar também com *Web Ontology Language* (OWL). Quando o usuário faz uma consulta baseada em palavras-chave, esta consulta é expandida para um conjunto contendo todas as combinações de consultas semânticas possíveis de serem feitas para a base de conhecimento atualmente ativa. Com o objetivo de otimizar o processo de geração de consultas todas as prováveis consultas que poderiam ser feitas para uma determinada base de conhecimento são geradas e armazenadas para posteriormente servirem de *template* durante a expansão das palavras-chave. Todo o sistema de expansão de termos é baseado nos conceitos de inferência do RDF/RDFS/OWL e correspondência textual entre termos internos e externos à base de conhecimento.

1.7 Organização do trabalho

O presente trabalho está organizado em sete capítulos, que abordam alguns tópicos principais e seguem a seguinte estrutura:

Capítulo 1: Explicita os objetivos e a metodologia empregados e apresenta alguns trabalhos relacionados;

Capítulo 2: Recuperação de Informação. Apresenta a área e o processo de recuperação de informação, em especial trata do Modelo Espaço Vetorial. Como mensurar a similaridade entre documentos e consultas; formas de conseguir ordenar por relevância uma lista contendo referências a documentos que supostamente atendam a necessidade de informação de um usuário manifesta na expressão de busca;

Capítulo 3: Indexação Automática. Discute o processo de indexação automática e a extração de termos. Explica como são criados os vetores associados aos documentos e a expressão de busca, em particular, sobre como é feito o cálculo dos pesos dos termos;

Capítulo 4: Expansão de consulta. Neste capítulo é apresentada a técnica chamada “expansão de consulta”, que em síntese, buscar diminuir a ambiguidade da expressão de busca na tentativa de melhorar a precisão do sistema;

Capítulo 5: Ontologias. Discute o que são as ontologias, em especial, as chamadas ontologias computacionais;

Capítulo 6: Proposta de utilização de ontologias na Recuperação de Informação. É discutido em detalhes a proposta sobre o uso de ontologias no processo de indexação automática e expansão de consulta utilizando o Modelo Vetorial;

Capítulo 7: Considerações finais. São discutidos os resultados alcançados e apresentadas as oportunidades de pesquisa que não puderam ser exploradas neste trabalho.

Recuperação de Informação

Recuperação de Informação é o processo de buscar informações que satisfaçam a necessidade informacional do usuário. Para que este processo seja viável é necessário adotar alguns pressupostos conceituais e compreender que, mesmo empregando técnicas cada vez mais elaboradas, o processo possui limitações próprias, que não são apenas limitadas pela técnica ou pela tecnologia.

O conceito de “informação”, no contexto da Ciência da Informação, possui algumas definições. Neste trabalho, é adotada a definição proposta por BUCKLAND (1991) de “informação como coisa”, como algo tangível. Esta é a forma como os sistemas de recuperação de informação lidam com a informação.

Em consonância com a definição de Buckland, existe também Le Coadic (1996):

A informação é um conhecimento inscrito (gravado) sob a forma escrita (impressa ou numérica), oral ou audiovisual. A Informação comporta um elemento de sentido. É um significado transmitido a um ser consciente por meio de uma mensagem inscrita em um suporte espacial-temporal: impresso, sinal elétrico, onda sonora, etc. Essa inscrição é feita graças a um sistema de signos (a linguagem) [...]. (LE COADIC, 1996, p. 5).

Portanto, na definição adotada neste trabalho, a informação é algo físico, codificada e inscrita em algum suporte, que poderá no futuro funcionar como uma mensagem a algum indivíduo e este por meio de processos cognitivos próprios produzir conhecimento a partir dela. Isso será possível desde que aquele indivíduo seja capaz de decodificar o código, ou seja, ele deve compreender a linguagem utilizada.

O termo “recuperação de informação”, segundo SARACEVIC (1999, p. 1057), teve sua origem no artigo “*Zatocoding applied to mechanical organization of knowledge*”, elaborado

por Calvin Mooers, publicado em 1951. Neste artigo, Mooers propôs um novo sistema para organização mecânica do conhecimento visando recuperação de informação⁴, e para isso define o que ele denomina de “recuperação de informação”:

Recuperação de informação é o nome dado ao processo ou método pelo qual um potencial usuário de informação é capaz de converter a sua necessidade de informação em uma lista real de citações a documentos em um acervo contendo informações úteis para ele. [...] Recuperação de informação abrange os aspectos intelectuais da descrição da informação e sua especificação para a busca, e também quaisquer sistemas, técnicas ou máquinas que são utilizadas para realizar a operação. A recuperação de informação é crucial para a documentação e organização do conhecimento

[...]

O assunto de cada documento ou outra unidade de informação é caracterizado ou descrito por meio de um conjunto de “descritores” tirados de um vocabulário formal de tais termos. Uma “lista de cabeçalho de assuntos” remeterá a uma aproximação grosseira do seu significado (MOOERS, 1951, p.25, tradução nossa).⁵

É importante destacar a importância dada à descrição da informação, notadamente à utilização de elementos externos aos documentos, tais como “vocabulário formal”, “lista de cabeçalhos de assuntos”. A partir desta definição inicial fica delimitado a existência dos três elementos fundamentais envolvidos no processo de recuperação de informação: (1) um conjunto de documentos representados de alguma forma; (2) a necessidade de informação do usuário também representada de alguma forma e; (3) o sistema de recuperação de informação, neste caso a função de busca, atuando como um mediador entre essas representações, retornando um novo conjunto de documentos (ou referências a eles) que supostamente satisfaçam a necessidade de informação manifesta pelo usuário.

Um outro conceito importante é que a atividade de “recuperação de informação” é uma atividade que não informa (altera o conhecimento do usuário sobre algo), ela simplesmente informa sobre a existência (ou não) de documentos que possam interessar ao usuário (LANCASTER, 1968, p. 1). Portanto, um sistema de recuperação de informação recupera documentos ou referências a documentos, nunca a informação em si.

⁴ mechanical organization of knowledge for retrieval of stored information (p. 20).

⁵ Information retrieval is the name for the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him. [...] Information retrieval embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, techniques, or machines that are employed to carry out the operation. Information retrieval is crucial to documentation and organization of knowledge.

[...]

The subject matter of each document or other unit of information is characterized or described by means of a set of "descriptors" taken from a formal vocabulary of such terms. A "subject heading list" will call to mind a rough approximation of what is meant here.

Para que um sistema possa recuperar documentos, o conteúdo destes devem ser representados de alguma forma. Lancaster ao tratar sobre indexação de assunto e redação de resumos posiciona a figura do resumidor como alguém que redige uma descrição narrativa ou síntese do documento, e o indexador como alguém que descreve o conteúdo de um documento empregando um ou vários termos de indexação “comumente selecionados de algum tipo de vocabulário controlado” (LANCASTER, 2004, p. 6).

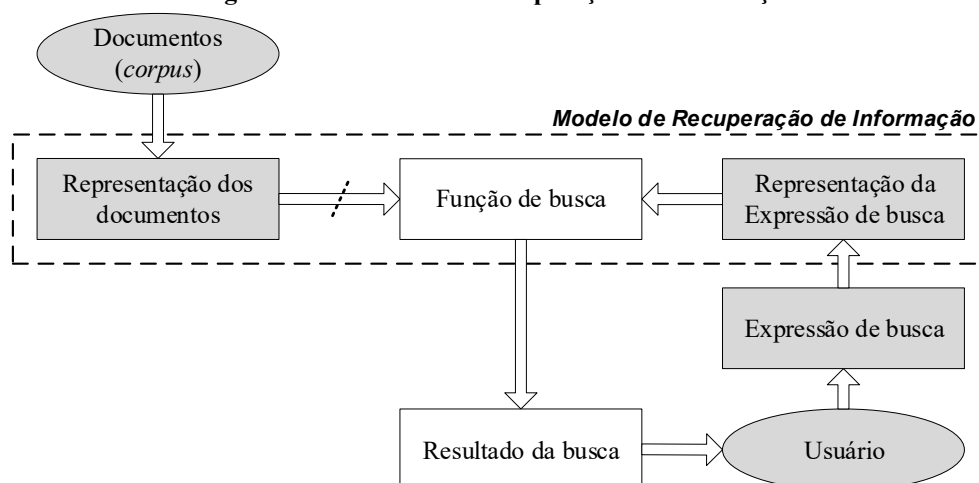
Considerando apenas os três componentes essenciais, a eficiência do processo dependerá principalmente de dois fatores: (1) a função de busca adotada e; (2) a precisão das representações elaboradas. Mesmo que as representações sejam precisas, elas devem possuir compatibilidade entre si, ou seja, as representações dos documentos e a representação da necessidade de informação do usuário devem representar os mesmos conceitos da mesma forma, para que a função de busca consiga realizar comparações entre as elas.

Na Ciência da Informação as linguagens documentárias são empregadas no caso específico das representações que demandam precisão conceitual. Segundo Fujita (2004), estas linguagens são compostas por um conjunto controlado de termos que visam representar os conceitos mais significativos dos assuntos tratados nos documentos, e são empregadas tanto na fase de indexação quanto durante a formulação da busca. Elas oferecem uma convergência entre a linguagem do indexador e a linguagem do usuário de um sistema de recuperação; exercem uma função mediadora entre a linguagem do usuário e os interesses de busca deste usuário, e entre os conceitos intrínsecos ao documento e suas representações.

2.1 O processo de Recuperação de Informação

A Figura 1 apresenta o processo de Recuperação de Informação, conforme concebido por FERNEDA (2012) evidenciando a forma como os documentos e os usuários interagem com o modelo de recuperação de informação.

Figura 1 — Processo de Recuperação de Informação



Fonte: baseada em FERNEDA, 2012, p. 14

Na Figura 1, os três blocos em destaque no retângulo tracejado, denominado “Modelo de Recuperação de Informação” são exatamente os três componentes essenciais mencionados anteriormente: (1) representação dos documentos; (2) representação da expressão de busca e; (3) função de busca (que faz a medição entre as representações). A “representação dos documentos” é feita em momento anterior à busca, e por isso o diagrama possui um seccionamento na seta que aponta em direção ao bloco “função de busca”. Nas próximas subseções serão descritos os componentes constituintes deste modelo:

2.1.1 Documentos (corpus)

Corpus é o nome dado a um conjunto de documentos. Documentos são os suportes materiais onde a informação está contida. É adotada a concepção mais abrangente possível, incluindo aqui não só documentos textuais, como em Suzanne Briet, que define documento como:

Qualquer sinal simbólico ou concreto, preservado ou gravado para os fins de representação, de reconstituição ou de prova de fenômeno físico ou intelectual⁶ (BRIET, 1951, p. 7, tradução nossa).

2.1.2 Representação dos documentos

Na etapa de representação dos documentos, seus conceitos serão extraídos por meio do processo de indexação. É criada uma representação interna ao sistema sobre o conteúdo de

⁶ A tout indice concret ou symbolique, conservé ou enregistré, aux fins de représenter, de reconstituer ou de prouver un phénomène ou physique ou intellectuel

cada documento do *corpus* e esta representação deve ser compatível com a que será elaborada para a expressão de busca, viabilizando a comparação entre elas.

A indexação é feita em tempo anterior à busca (representado pela seta seccionada na Figura 1). Este processo pode ser feito de forma manual, semiautomática, automática ou mesmo formas híbridas entre as citadas. Existem vantagens e desvantagens inerentes a cada método sendo perfeitamente válido, dentro de uma mesma implementação de um determinado modelo de recuperação, empregar mais de uma técnica de indexação.

Uma característica a ser mencionada é que a representação interna não precisa, necessariamente, empregar termos textuais, sendo a única exigência a de que esta representação seja compatível com a representação elaborada para a expressão de busca.

2.1.3 Usuário

Um outro elemento envolvido é o usuário. Ele é fundamental, pois a razão de existir a recuperação de informação é, em fim último, exclusivamente para satisfazer a necessidade de informação do usuário. Sob uma perspectiva social, o motivo principal de organizarmos a informação é para maximizar o seu caráter utilitário, tornando-a útil para a sociedade.

2.1.4 Expressão de busca

A expressão de busca é o meio que o usuário utiliza para comunicar ao sistema a sua necessidade de informação. Isso não significa, necessariamente, que o usuário deve expressar essa necessidade empregando termos textuais semelhantes aos empregados na indexação, geralmente o usuário não conhece a forma que o sistema utilizou para indexar nem o vocabulário empregado. É nesta etapa que estão situadas as interfaces de busca, que servem para mediar a interação do usuário que tenta expressar sua necessidade de informação e o sistema de recuperação que necessita de uma expressão de busca para iniciar o processamento.

O comportamento humano em relação à busca por informação relevante pode ser estudado pela ótica do “comportamento de consulta”, sendo o modelo proposto por Taylor (1962) um dos primeiros. Neste modelo, o autor afirma que a necessidade de informação do usuário para fins de análise, pode ser classificada em quatro categorias: necessidade visceral (*visceral need*), necessidade consciente (*conscious need*), necessidade formalizada (*formalized need*), e necessidade comprometida (*compromised need*). É nesta última fase (comprometida)

que o usuário precisa traduzir a sua necessidade de informação na linguagem do sistema de recuperação de informação utilizado.

2.1.5 Representação da expressão de busca

A representação da expressão de busca é necessária para que a expressão de busca que o usuário elaborou, direta ou indiretamente, e submeteu ao sistema, seja compatibilizada com a representação adotada na indexação dos documentos. O crucial desta etapa é:

[...] independentemente dos recursos oferecidos pelo sistema é necessário que a expressão de busca seja representada de forma similar a utilizada na representação dos documentos. Essa homogeneidade permitirá a comparação entre a busca e todos os documentos do corpus do sistema por meio da função de busca (FERNEDA, 2012, p. 19)

2.1.6 Função de busca

A função de busca é o bloco que efetivamente fará a comparação entre as duas representações e produzirá um resultado. Em sua forma mais comum, como resultado do cruzamento e comparação entre as representações, ela retornará uma lista ordenada por relevância contendo referências aos documentos.

Existe a possibilidade de ambos os conjuntos de representações (documentos e expressão de busca) sofrerem filtragem a partir um ou mais metadados⁷ específicos utilizados como critério para que aquela representação seja ou não incluída nos testes realizados pela função de busca. Por exemplo, poderíamos limitar a busca aos documentos que tenham uma data de publicação específica ou a um tipo específico de suporte; ou limitarmos a quantidade de termos da expressão de busca. Essa filtragem pode servir para o sistema otimizar ou adequar as buscas dentro das limitações técnicas do sistema, reduzindo o conjunto de documentos a ser considerado nas comparações. Há também a possibilidade de filtragem na saída da função de busca, principalmente para limitarmos a quantidade de resultados retornados para o usuário.

⁷ “Metadata, the information we create, store, and share to describe things, allows us to interact with these things to obtain the knowledge we need. The classic definition is literal, based on the etymology of the word itself - metadata is *data about data*.” National Information Standards Organization (NISO), 2017, p. 1.

2.1.7 Resultado da busca

O resultado da busca é o final do processo de recuperação de informação, ou pelo menos, deste ciclo nos casos em que é utilizado algum mecanismo de *feedback*. Aquela lista originária da função de busca pode ou não sofrer reordenação antes de ser apresentada ao usuário; esta reordenação pode se basear em algum metadado dinâmico ou estático, como por exemplo, popularidade do documento segundo algum indicador cientométrico⁸.

Neste bloco “resultados de busca” encontramos uma das duas facetas da interface de busca: a interface de apresentação dos resultados. A suposta relevância que o sistema inferiu está implícita na ordem dos resultados ou em sua apresentação gráfica, porém o julgamento final sobre o que é ou não relevante será feito pelo próprio usuário. A interface mais comum, simplesmente apresenta o conjunto de resultados como uma lista ordenada contendo relativamente poucos resultados, com os itens de maior relevância (inferida pelo sistema) ocupando as primeiras posições.

2.2 Modelo Espaço Vetorial

Salton (1989, p. 313) destaca que dentre os modelos de recuperação propostos até 1989 poderíamos classificá-los em três grandes categorias: (1) modelos booleanos (baseado na comparação entre termos da *query*⁹ com os termos dos documentos indexados), (2) probabilísticos (baseado na estimativa da probabilidade de relevância dos documentos em relação à *query*) e (3) espaço-vetoriais (que representam tanto as *queries* quanto os documentos por conjuntos de termos e calculam a similaridade global entre eles). Ele, um dos pioneiros em empregar este modelo de recuperação, considera o modelo espaço vetorial como o mais simples de ser utilizado e, em algumas situações, como sendo o mais produtivo.

O modelo de recuperação Espaço Vetorial (*Vector Space Model* – VSM) está situado na categoria de modelos de recuperação por similaridade. Nesta categoria há a pressuposição de que quanto maior for a semelhança entre a expressão de busca (*query expression*) e um documento do *corpus*, maior será a relevância deste documento para o usuário que elaborou a expressão (ZHAI, 2009, p. 11).

⁸ Cientometria é a disciplina que estuda aspectos quantitativos da ciência e da produção científica.

⁹ *Query* é a expressão de busca representada em linguagem específica do sistema de recuperação de informação utilizado, geralmente é composta por termos e operadores lógicos.

Este modelo de recuperação representa tanto os documentos quanto as consultas por meio de vetores multidimensionais onde cada termo receberá um peso (ou relevância). Ele deve ser visto como modelo conceitual para a recuperação, pois não há definição detalhada de uma implementação, por exemplo, o modelo não define exatamente como é feita a escolha dos termos, nem como posicionar os documentos e as consultas neste espaço vetorial, bem como também não define com precisão a função de similaridade.

No processo de recuperação de informação, a linguagem é empregada de duas maneiras: (1) para descrever a necessidade percebida de informação e (2) para discriminar o que é desejável dentro do conjunto de informações disponíveis (BLAIR, 2003, p. 4). Este é um princípio que sustenta este modelo: o uso da mesma linguagem tanto na representação temática quanto na recuperação.

2.2.1 Pressupostos do Modelo Vetorial

Para que o Modelo Espaço Vetorial (VSM) possa ser implementado, alguns pressupostos são assumidos (SALTON, 1989): (1) Os termos empregados na indexação são os mesmos utilizados na elaboração da expressão de busca (*query expression*); (2) Expressão de busca e documentos podem ser representados como vetores de termos (*term vectors*) na forma $D = (a_{i1}, a_{i2}, \dots, a_{it})$ e $Q = (q_{i1}, q_{i2}, \dots, q_{it})$, nos quais os coeficientes a_{ik}, q_{ik} possuem valor próximo de 1 quando o termo k aparecer no documento d_i ou query q_i . Este valor será mais próximo de 1 quanto mais importante este termo for considerado na caracterização do documento ou da expressão de busca.

Atendido os pressupostos acima, a similaridade entre os vetores x e y pode ser medida pelo produto matricial $x \cdot y = |x||y| \cos \alpha$, onde $|x|$ é a magnitude do vetor e α é o ângulo formado entre os dois vetores em questão (SALTON, 1989, p. 314).

Para o problema da correlação dos termos é assumido que os termos sempre serão não-correlacionados, o que permite assumir que os vetores serão ortogonais, ou seja, $T_i \cdot T_j = 0$ (caso fossem paralelos: $T_i \cdot T_j = 1$). Por causa deste pressuposto é possível simplificar o cálculo da similaridade, reduzindo-o a uma soma de produtos (SALTON, 1989, p. 315):

$$sim(D_r, Q_s) = \sum_{i,j=1}^t a_{ri} \cdot q_{sj}$$

Os autores SALTON, WONG, YANG (1975) discutem sobre um modelo de recuperação de informação. Neste modelo o espaço onde residem os documentos indexados é definido pelo conjunto de vetores de indexação $C = (D_1, D_2, \dots, D_n)$, onde n é o número de documentos do corpus, sendo cada elemento deste conjunto definido por $D_i = (d_{i1}, d_{i2}, \dots, d_{it})$ onde o i é o número do documento e t é o número de termos reconhecidos, aqui, d_{ij} é o peso atribuído ao j -ésimo termo do documento i . A similaridade entre dois documentos qualquer pode ser aferida a partir do cálculo do coeficiente de similaridade entre dois vetores de indexação definido pela função $S(D_i, D_j)$, onde D_i, D_j são os respectivos vetores.

$$D_i = (d_{i1}, d_{i2}, \dots, d_{it})$$

De uma outra forma, mais simplificada: O *corpus* é definido por $C = (D_1, D_2, \dots, D_n)$ onde n é o número de documentos do *corpus*, existe, portanto, um conjunto denominado C composto pelos documentos $D_1, D_2, D_3 \dots D_n$. Todos os termos (compostos ou não) que o sistema manipula formam um outro conjunto, o vocabulário, definido por $V = (T_1, T_2, \dots, T_t)$, onde t é o número de termos distintos que o sistema reconhece. E por fim, cada documento é definido como um conjunto de termos, composto por valores (pesos) associados a cada termo de V , para um determinado documento i definimos $D_i = (d_{i,1}, d_{i,2}, \dots, d_{i,t})$, aqui os números compreendidos entre 1 e t correspondem aos termos.

Um exemplo prático: dois documentos (D_1, D_2) e três termos (T_1, T_2, T_3) :

- o *corpus* será definido como $C = (D_1, D_2)$;
- o vocabulário como $V = (T_1, T_2, T_3)$;
- cada documento será definido por:
 - $D_1 = (d_{1,1}, d_{1,2}, d_{1,3})$;
 - $D_2 = (d_{2,1}, d_{2,2}, d_{2,3})$;

2.2.2 Fórmula da similaridade por cosseno

Salton e McGill (1983, p. 124) fornecem uma fórmula para o cálculo da similaridade por cosseno entre dois documentos Doc_i e Doc_j :

$$sim_{cosine}(DOC_i, DOC_j) = \frac{\sum_{k=1}^t (TERM_{ik} \cdot TERM_{jk})}{\sqrt{\sum_{k=1}^t (TERM_{ik})^2 \cdot \sum_{k=1}^t (TERM_{jk})^2}}$$

A fórmula acima pode ser vista de uma outra maneira:

$$sim_{cosine}(DOC_i, DOC_j) = \frac{TERMS_i \cdot TERMS_j}{\| TERMS_i \| \cdot \| TERMS_j \|}$$

Na fórmula acima, $TERMS_i$ é um vetor contendo os pesos de todos os termos de Doc_i e $TERMS_j$ é um vetor contendo os pesos de todos os termos de Doc_j , respectivamente. A fórmula também utiliza o conceito de módulo do vetor, onde em um determinado espaço euclidiano n-dimensional, representado por R^n , o comprimento de um vetor $x = (x_1, x_2, \dots, x_n)$ será representado por $\| x \|$, sendo calculado pela fórmula da raiz quadrada da soma dos quadrados de todos os elementos do vetor, definida como:

$$\| x \| = \sqrt{\sum_{i=1}^n x_i^2}$$

Ao se realizar a divisão pelo módulo dos dois vetores envolvidos na comparação normaliza-se o resultado da função $sim_{cosine}(DOC_i, DOC_j)$ para a faixa de valores entre 0 (zero) e 1 (um), ou seja, para vetores idênticos (geometricamente paralelos) o resultado será 1 (um), para vetores totalmente diferentes (geometricamente ortogonais) o resultado será 0 (zero), e nas demais situações, valores compreendidos entre esses dois extremos.

2.2.3 Exemplos de uso da fórmula da similaridade

O Quadro 1 exemplifica o cálculo de similaridade por cosseno entre dois documentos hipotéticos Doc_A e Doc_B . Os valores correspondem aos pesos de cada termo em cada documento; neste exemplo, utiliza-se um vocabulário de indexação composto por 8 termos:

Quadro 1 — Exemplo de pesos termo vs documento

	Termo1	Termo2	Termo3	Termo4	Termo5	Termo6	Termo7	Termo8
Doc_A	2	1	0	2	0	1	1	1
Doc_B	2	1	1	1	1	0	1	1

Fonte: Elaborado pelo autor.

OBS: Os pesos aqui mencionados foram previamente determinados durante a fase de indexação dos documentos.

O Quadro 1 poderá ser representado em sua forma vetorial:

$$Doc_A = [2, 1, 0, 2, 0, 1, 1, 1]$$

$$Doc_B = [2, 1, 1, 1, 1, 0, 1, 1]$$

Uma vez obtidos os vetores são efetuados os cálculos para estimar a similaridade entre os dois documentos:

$$sim_{cosine}(Doc_A, Doc_B) = \frac{Doc_A \cdot Doc_B}{\|Doc_A\| \cdot \|Doc_B\|}$$

$$Doc_A \cdot Doc_B = (2 \times 2) + (1 \times 1) + (0 \times 1) + (2 \times 1) + (0 \times 1) + (1 \times 0) + (1 \times 1) + (1 \times 1) = 9$$

$$\|Doc_A\| = \sqrt{2^2 + 1^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 1^2} = \sqrt{12}$$

$$\|Doc_B\| = \sqrt{2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2} = \sqrt{10}$$

$$= \frac{9}{\sqrt{12} \cdot \sqrt{10}} = \frac{9}{\sqrt{120}} \cong \frac{9}{10,954} \cong 0,822$$

$$sim_{cosine}(Doc_A, Doc_B) \cong 0,822$$

O valor obtido representa o grau de similaridade entre os dois documentos envolvidos. Quanto mais próximo de 1 (um) for o valor obtido, mais semelhantes serão os dois vetores, e conseqüentemente, os dois documentos.

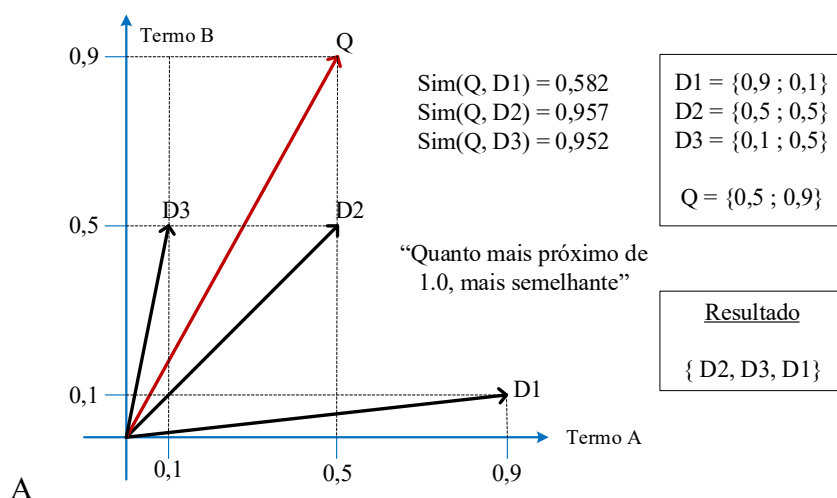
A seguir um outro exemplo de cálculo de similaridade por cosseno envolvendo um vocabulário composto por três termos: gato, cão e rato.

$$V = (gato, cão, rato)$$

$$Doc_A = (gato, cão, cão) = (1, 2, 0)$$

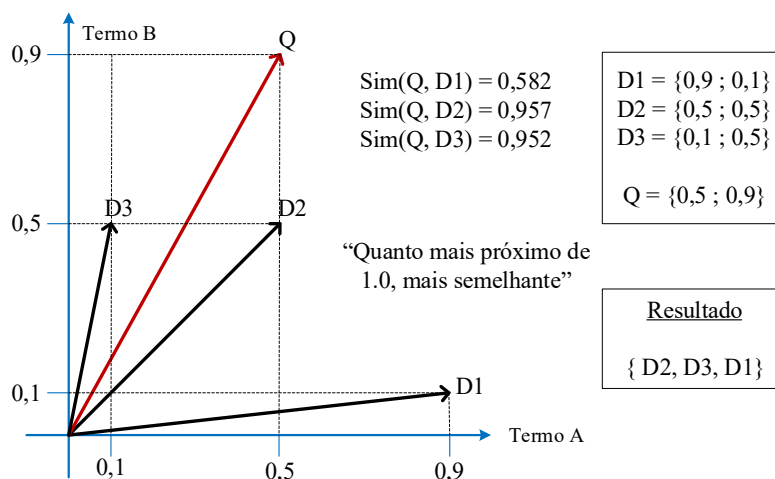
$$Doc_B = (gato, cão, rato, rato) = (1, 1, 2)$$

$$sim_{cosine}(Doc_A, Doc_B) = \frac{(1 \times 1) + (2 \times 1) + (0 \times 2)}{\sqrt{1^2 + 2^2 + 0^2} \cdot \sqrt{1^2 + 1^2 + 2^2}} = \frac{1 + 2 + 0}{\sqrt{5} \cdot \sqrt{6}} = \frac{3}{\sqrt{30}} \cong 0,548$$



traz um outro exemplo de cálculo de similaridade entre expressão de busca e documentos. O exemplo é composto por três documentos: “D1”, “D2” e “D3” e uma expressão de busca “Q”. Para simplificar a representação gráfica, o vocabulário é composto por apenas dois termos: “Termo A” e “Termo B”.

Figura 2 — Exemplo gráfico do cálculo de similaridade por cosseno entre vetores



Fonte: Elaborada pelo autor.

É justamente por utilizar o mesmo tipo de representação (vetores), tanto dos documentos quanto da expressão de busca que é possível empregar uma única função para mensurar a similaridade entre a expressão de busca e um documento qualquer do *corpus*, bem como mensurar a similaridade entre os documentos do *corpus*

2.2.4 Fórmula da similaridade de Jaccard

Além da fórmula para estimar a similaridade entre vetores baseada no cosseno, Salton e McGill (1983, p. 200-204) mencionam a existência de outras, e concluem que:

[...] a medida de *Jaccard* e a medida por cosseno têm características similares, variando a resposta entre os valores no mínimo 0 e no máximo 1 para elementos não negativos nos vetores. Estas medidas são fáceis de calcular e aparentam ser tão eficientes na recuperação quanto outras funções mais complicadas[...]¹⁰ (SALTON; MCGILL, 1983, p. 204, tradução nossa)

A seguir, a fórmula para o cálculo da similaridade denominado Coeficiente de *Jaccard* (SALTON; MCGILL, 1983, p. 203):

$$sim_{jaccard}(DOC_i, DOC_j) = \frac{\sum_{k=1}^t (TERM_{ik} \cdot TERM_{jk})}{\sum_{k=1}^t TERM_{ik} + \sum_{k=1}^t TERM_{jk} - \sum_{k=1}^t (TERM_{ik} \cdot TERM_{jk})}$$

Comparada com a função de similaridade por cosseno, a principal diferença entre elas está no termo normalizador (denominador da divisão).

Exemplo, extraído de Salton e McGill (1983, p. 202). A partir dos documentos a seguir, a similaridade de *Jaccard* pode ser calculada:

$$Doc_1 = (3, 2, 1, 0, 0, 0, 1, 1)$$

$$Doc_2 = (1, 1, 1, 0, 0, 1, 0, 0)$$

$$\begin{aligned} & \sum_{k=1}^t (TERM_{1k} \cdot TERM_{2k}) \\ &= (3 \times 1) + (2 \times 1) + (1 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 0) \\ &+ (1 \times 0) = 3 + 2 + 1 + 0 + 0 + 0 + 0 + 0 = 6 \end{aligned}$$

$$\sum_{k=1}^t TERM_{1k} = 3 + 2 + 1 + 0 + 0 + 0 + 1 + 1 = 8$$

¹⁰ [...] The Jaccard and the cosine measures have similar characteristics, ranging from a minimum 0 to a maximum of 1 for nonnegative vector elements. These measures are easy to compute and they appear to be as effective in retrieval as other more complicated functions. [...]

$$\sum_{k=1}^t TERM_{2k} = 1 + 1 + 1 + 0 + 0 + 1 + 0 + 0 = 4$$

$$\begin{aligned} sim_{Jaccard}(DOC_1, DOC_2) &= \frac{\sum_{k=1}^t (TERM_{1k} \cdot TERM_{2k})}{\sum_{k=1}^t TERM_{1k} + \sum_{k=1}^t TERM_{2k} - \sum_{k=1}^t (TERM_{1k} \cdot TERM_{2k})} \\ &= \frac{6}{8 + 4 - 6} = 1 \end{aligned}$$

Se os mesmos valores fossem utilizados com o cálculo da similaridade por cosseno:

$$\begin{aligned} sim_{cosine}(DOC_1, DOC_2) &= \frac{\sum_{k=1}^t (TERM_{1k} \cdot TERM_{2k})}{\sqrt{\sum_{k=1}^t (TERM_{1k})^2 \cdot \sum_{k=1}^t (TERM_{2k})^2}} \\ &= \frac{6}{\sqrt{(3^2 + 2^2 + 1^2 + 0^2 + 0^2 + 0^2 + 1^2 + 1^2) \cdot (1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 1^2 + 0^2 + 0^2)}} \\ &= \frac{6}{\sqrt{16 \cdot 4}} = \frac{6}{\sqrt{64}} = 0,75 \end{aligned}$$

Como resultado do exemplo, há os seguintes valores para as duas fórmulas:

$$sim_{Jaccard}(DOC_1, DOC_2) = 1,00$$

$$sim_{cosine}(Doc_1, Doc_2) = 0,75$$

Ambos os índices são adequados para aferição da similaridade sendo uma das vantagens da fórmula de *Jaccard* dispensar o cálculo da raiz quadrada no denominador, o que em determinadas circunstâncias, principalmente questões técnicas referentes ao *hardware*, pode ser uma opção

Indexação Automática

A indexação tem como objetivo representar o conteúdo dos documentos na esperança de que isso possa servir para, durante uma busca conduzida por um usuário, priorizar alguns documentos dentro de um *corpus* potencialmente extenso que tenham a possibilidade de servir para satisfazer a necessidade de informação daquele usuário. O processo de representação de um documento pode ser descritivo ou temático, neste trabalho a ênfase será na representação temática.

Quando a indexação é feita automaticamente com o uso de dispositivos computacionais, estes dispositivos não conseguem realmente compreender o conteúdo dos documentos para realizar uma representação temática mais precisa como um indexador humano profissional faz. De uma forma bem genérica, os assuntos dos documentos são inferidos, basicamente, por métodos estatísticos envolvendo frequência e coocorrência entre palavras, sendo então, algumas destas palavras promovidas ao status de “termos” e empregadas como termos de indexação.

3.1 O processo de indexação

O processo de indexação pode ser definido como: “ato de identificar e descrever o conteúdo de um documento com termos representativos dos seus assuntos e que constituem uma linguagem de indexação” (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, NBR12676, 1992, p. 2). Neste trecho nota-se o caráter representativo e sintético do processo, pois toda representação implica inevitavelmente em algum tipo de redução (síntese).

Segundo Gil Leiva a Indexação tem sua origem nas tarefas dos antigos escribas da Mesopotâmia, elaborando etiquetas para identificar cópias das tabuletas de argila com textos

armazenados em “prateleiras de madeira, colocados em nichos nas paredes ou eram dispostos em caixas de madeira”; estas etiquetas serviam para descrever o conteúdo daqueles documentos. “Nessas tarefas rudimentares, vemos os primeiros passos do que hoje conhecemos como a indexação” (GIL LEIVA; RODRÍGUEZ MUÑOZ, 1996, p. 53 *apud* GIL LEIVA; FUJITA, 2012, p. 65). A Indexação é um processo abrangente, pois “qualquer objeto pode ser indexado, ou seja, reduzido a representações conceituais que facilitem seu armazenamento e recuperação em bases de dados” (GIL LEIVA; FUJITA, 2012, p. 65).

No caso específico da indexação automática existe mais um obstáculo: o processo de indexação será realizado por um dispositivo eletrônico, que não consegue efetivamente compreender os conceitos expressos no item o qual ele está representando. Apesar de ser possível indexar automaticamente diferentes tipos de conteúdo disponíveis nos mais variados meios, o enfoque deste trabalho está em material textual já disponível em algum formato digital legível por máquina.

Os processos automáticos para manipular a informação documental a tratam como coisa, conforme o conceito estabelecido por Buckland (1991) onde os itens que compõem os sistemas de informações documentais são basicamente registros relativos a objetos (coisas), sendo estes os reais portadores de informação; neste contexto, o termo “documento” é utilizado para designar coisa informativa.

Além do conceito de informação como coisa, ainda existe a questão sobre os níveis de representação. Assumindo que todas as representações podem ser pensadas em dois níveis: primário e secundário. Em um nível primário a representação é “feita pelos autores no momento da expressão dos resultados de seus pensamentos” (ALVARENGA, 2003, p. 20); já a representação secundária é aquela feita sobre uma representação primária com o objetivo de incluí-la em sistemas documentais referenciais. Em ambos os níveis, a representação é produzida mediante um processo cognitivo.

A representação secundária é um substituto do documento que é armazenado no sistema (catálogo, banco de dados de fontes de informação, repositórios digitais, etc.) e será

utilizada para a recuperação de uma referência que leva ao documento. Lídia Alvarenga sintetiza os dois níveis de uma representação como:

[...] no processo de tratamento ou processamento dos registros de conhecimento para fins de armazenagem nos sistemas de informação, é requerido um novo estágio de representação, desta vez partindo-se não do ser ontológico em si, mas do conhecimento sobre o ser, expresso em documentos. Esta seria uma representação secundária. Nesse sentido, a representação secundária teria por objeto prioritário não o acervo da ontologia, das coisas e seres existentes, mas o acervo de conhecimentos sobre essas coisas e seres, objetos da epistemologia (ALVARENGA, 2003, p. 22).

Portanto, segundo a própria autora, a representação primária tem preocupação com a ontologia (em seu sentido filosófico e não computacional), já a representação secundária lida com os objetos da epistemologia. Esta representação secundária tem o objetivo de tornar um artefato extremamente complexo e extenso, portanto, inviável de ser manipulado, em um outro menos complexo e extenso, possibilitando sua manipulação por meio de uma representação. É aceitável e até inevitável, que neste processo de simplificação ocorrerão perdas que ocasionarão deformações neste novo artefato.

3.2 Os precursores da Indexação Automática

Reconhecidos como pioneiros pela literatura consultada, alguns autores realizaram contribuições expressivas para o estabelecimento das bases teóricas da área de recuperação de informação. Entre eles destacamos Calvin Mooers (1919-1994†), George Kingsley Zipf (1902-1950†), Hans Peter Luhn (1896-1964†), Harold P. Edmundson (1921-2009†), Karen Spärck Jones (1935-2007†) e Gerard Salton (1927-1995†).

Na década de 50 o principal enfoque na área de recuperação de informação (RI) eram o hardware e a recuperação eletromecânica dos documentos. Ainda nos anos 50 ocorre a introdução de vocabulários controlados na RI automatizada por computadores para representação de assuntos. Um segundo passo, foi o gradual abandono destes vocabulários controlados para a indexação e análise de conteúdo, havendo um maior enfoque no uso de descritores baseados em um único termo: as palavras-chave; nesse momento o índice coordenado de palavras se tornou uma espécie de consenso entre os sistemas automáticos de RI (SALTON, 1987, p. 375-376).

A partir de 1950, com o desenvolvimento da computação, surgem algumas pesquisas com o objetivo de empregar computadores tanto para auxiliar quanto para realizar de forma

totalmente automática a indexação de documentos com conteúdo textual. Um dos problemas ainda não solucionado completamente até a atualidade é a extração automática dos termos, pois “a única coisa que os computadores podem fazer para nós é manipular símbolos e produzir resultados com estas manipulações”¹¹ (DIJKSTRA, 1988, tradução nossa) e a tarefa do indexador é realizar a representação temática e, portanto, este deve compreender o conteúdo do documento sendo descrito. Este é o grande dificultador da tarefa de indexação automática: a compreensão pela máquina das ideias e conceitos presentes no material a ser indexado.

Logo após o término da Segunda Guerra Mundial (1939-1945) havia uma grande quantidade de informação técnica e científica, resultado das inúmeras pesquisas realizadas durante o período do conflito. Esse volume de informação, fundamentalmente impressa, foi disponibilizado por meio de inúmeras publicações científicas (SOLLA PRICE, 1963), gerando uma nova demanda: lidar com um grande volume de informação disponível em suportes distintos e abrangendo diversos assuntos.

Vannevar Bush que ocupava o cargo de Diretor do Escritório de Pesquisa Científica e Desenvolvimento norte-americano durante este período, sintetiza essa preocupação em seu artigo propondo a criação de uma máquina, teórica, chamada MEMEX, cujo objetivo era coletar, armazenar e recuperar informações (BUSH, 1945, p. 101-108).

Empregar técnicas e sistemas de indexação tradicionais, já utilizados e consagrados, para atender a estas novas demandas que surgiram não foram suficientes. Tais demandas incluem a necessidade de lidar com um perfil cada vez mais diversificado de material (artigos, relatórios, jornais) em suportes variados (papel, fotográfico, digital).

Quando nós formos além de um único livro para um sistema de informação envolvendo centenas de livros, relatórios, artigos, etc., a chance de sucesso [na elaboração de] um sumário sistemático adequado ou na classificação de toda a informação contida no sistema se torna remota. Além disso, se o sistema de classificação deve funcionar em qualquer situação, não só para todas as informações contidas no sistema [de informação], mas também para toda a informação que poderá ser adicionada no futuro, este sistema de classificação perderá rapidamente toda a sua importância como um dispositivo de organização para a informação dentro do sistema [de informação]¹² (TAUBE, 1955, tradução nossa)

¹¹ the only thing computers can do for us is to manipulate symbols and produce results of such manipulations

¹² When we go beyond a single book to a system of information involving thousands of books, reports, articles, etc., the possibility of a successful and adequate systematic table of contents or a classification of all the information in the system becomes remote. Further, if the classification system must provide not only for all the

O Sistema *Uniterm* proposto por Taube (1955) buscava atender essa demanda “empregando palavras e termos na forma como ocorressem, sendo extraídos dos documentos”¹³ (MOOERS, 2003, p. 818, tradução nossa).

O custo operacional de um Sistema de Recuperação de Informação (sistema de RI) está relacionado aos custos da indexação, dos equipamentos utilizados e da recuperação em si. O primeiro é influenciado pelo tempo empregado, salários e outras despesas com os indexadores; o segundo inclui além do investimento inicial, todas as despesas operacionais dos equipamentos necessários ao processo; já o terceiro envolve a busca no índice incluindo aqui o custo do tempo empregado nesta busca bem como as despesas decorrentes da disponibilização física dos documentos (CLEVERDON, 1958, p. 687-688).

3.3 A extração automática dos termos

Entre os anos de 1957 e 1959 destacam-se as publicações de Hans Peter Luhn. Salton afirma que Luhn foi pioneiro em afirmar que os computadores poderiam realizar a análise do conteúdo de textos disponíveis em formato digital, propondo a elaboração automática de *abstracts* e métodos de atribuir pesos aos termos baseados em frequência e localização das palavras nos textos considerados (SALTON, 1987, p. 376).

Hans Peter Luhn adota a ideia de utilizar termos extraídos do texto para descrever o seu conteúdo informacional “caracterizar um tópico por meio de uma série de elementos de identificação”¹⁴, “quanto mais termos forem associados, mais específico o tópico estará delineado”¹⁵ (LUHN, 1953, p. 14, tradução nossa).

O método de Luhn (1958) para a elaboração automática de *abstracts* consiste em: inicialmente os textos, já em formato digital, são processados palavra por palavra. Todas as palavras com grande ocorrência como pronomes, preposições e artigos são ignoradas. As palavras restantes são consolidadas da seguinte forma: todas são comparadas letra a letra entre si aos pares e, a partir do ponto de dissimilaridade, são contadas as letras restantes da maior palavra. Se houver menos de seis letras dissimilares, as palavras são consideradas semelhantes pois, provavelmente, referem-se a um mesmo conceito (*similar notion*); havendo mais do que

information in the system at any time but for all information which may be added at any future date, it will rapidly lose all significance as an organizing apparatus for the information in the system.

¹³ employed words and terms as they occurred in, and were selected from, the documents.

¹⁴ The new method uses the principle of characterizing a topic by a set of identifying elements or criteria

¹⁵ When identifying a topic by a set of criteria or identifying terms, the more terms are stated the more specifically the topic is delineated

seis letras as palavras são consideradas distintas. Todas as palavras semelhantes são consideradas similares em significado, idênticas em valor, e compõem um grupo. Estes grupos formam um conjunto ordenado pelo número de elementos de cada grupo. Grupos com uma contagem abaixo de um valor, determinado experimentalmente de acordo com o conjunto de documentos, são descartadas; os grupos restantes têm todas as suas palavras consideradas como “palavras significativas”.

Este método considera o número de ocorrências das palavras significativas dentro de uma sentença bem como a distância linear entre elas, pois podem haver palavras não significativas entre algumas palavras significativas da sentença. Para viabilizar o processamento disso criou-se o conceito de que toda sentença é composta por um ou mais conjuntos de palavras e que cada conjunto é composto por palavras significativas e não significativas, sendo cada conjunto delimitado por duas palavras significativas. Determinou empiricamente que o número máximo de palavras não significativas admitido em cada conjunto deve ser quatro ou cinco palavras. Uma vez estabelecido estes conjuntos, seus valores de *significance factor* serão dados pela contagem do número de palavras significativas em cada um dos conjuntos elevado ao quadrado, dividido pelo número total de palavras de cada conjunto. O *significance factor* da sentença é a somatória do valor de todos os conjuntos que compõem a sentença, e o *significance factor* do parágrafo, é a somatória do valor de suas sentenças.

Edmundson (1969, p. 270-272) também elabora um método para extração automática de termos. Este método é composto por quatro técnicas originadas das características observadas nos documentos: duas características estruturais que são o corpo do texto e o formato do texto (título, cabeçalhos, formato); e duas características linguísticas, que são as características do corpus e características do documento. O cruzamento destas duas categorias de características (estruturais e linguísticas) dá origem as quatro técnicas: *Cue method* (corpo do texto / características do corpus), *Location method* (formato do texto / características do corpus), *Key method* (corpo do texto / características do documento) e *Title method* (formato do texto / características do documento).

Na implementação destas técnicas o autor define duas estruturas de dados, uma chamada *dictionary* (lista de palavras com um coeficiente numérico associado, independente das palavras do documento atualmente em análise) e outra chamada *glossary* (lista de palavras com um coeficiente numérico de associado, é composta por palavras do texto atualmente em processamento) (EDMUNDSON, 1969, p. 270-272).

O *Cue method* (corpo do texto / características do corpus), baseia-se na hipótese de que a provável relevância de uma sentença é determinada pela presença de palavras pragmáticas no corpo do texto em análise como "significante", "impossível", "dificilmente". Estas palavras estão armazenadas no *cue dictionary* que distingue três categorias de palavras: *bonus words* (positivamente relevantes), *stigma words* (negativamente relevantes) e *null words* (que são irrelevantes). O *cue weight* de cada sentença é definido como a somatória desses valores obtidos na consulta ao *cue dictionary* (EDMUNDSON, 1969, p. 271).

O *Key method* (corpo do texto / características do documento) compila uma *key glossary* para cada documento, assemelhando-se segundo o próprio Edmundson, ao método de Luhn (1958). O resultado é uma lista contendo uma parte das palavras que compõe o documento juntamente com sua frequência (EDMUNDSON, 1969, p. 271-272).

Title method (formato do texto / características do documento) apoia-se na hipótese de que o autor sintetiza no título (e títulos das seções) o principal assunto de cada trecho do documento, por isso este método considera estas palavras como “positivamente relevantes”. Todas estas palavras que não forem *null words* receberão um peso positivo. O valor final para cada sentença será a somatória do valor das palavras que a compõe (EDMUNDSON, 1969, p. 272).

Location method (formato do texto / características do corpus) pressupõe que os trechos mais importantes de um documento estão localizados em posições específicas, como logo após um título de uma seção. Sentenças tópicas tendem a ocorrer no primeiro e último parágrafo de cada seção de um documento e por isso nesta técnica recebem uma pontuação adicional. Para cada localização da palavra no documento, esta pode receber uma pontuação diferenciada (EDMUNDSON, 1969, p. 272).

Nos métodos de Luhn e Edmundson não há uma preocupação com a compreensão, em sentido linguístico, dos termos extraídos. Luhn define que o fator de significância de cada sentença (*significance factor*) é um método probabilístico que não tenta fazer qualquer tentativa de interpretação ou compreensão do significado da palavra em si ou dos “argumentos expressos pela combinação das palavras” (LUHN, 1958, p. 160); “A importância do grau de proximidade é baseada nas características da linguagem falada e escrita, na qual as ideias intelectualmente

semelhantes são as expressas pelas palavras fisicamente mais próximas”¹⁶ (LUHN, 1958, p. 161, tradução nossa).

Ambos utilizam como base a contagem de frequência das palavras significativas:

Já faz algum tempo que foi aceito que a contagem de frequência das palavras significativas de um documento (principalmente substantivos, adjetivos e verbos) pode servir para isolar o vocabulário especial utilizado para transmitir informações em qualquer área específica do universo do discurso¹⁷ (OSWALD; LAWSON, 1953, p. 1-11 *apud.* EDMUNDSON; WYLLYS, 1961, p. 226, tradução nossa).

Eles também empregavam o conceito de eliminar palavras muito comuns, o que atualmente é denominado “*stop list*”. É difícil de estabelecer quando o termo “*stop list*” foi cunhado, mas Luhn preferia utilizar “*non-significant or common words*” (LUHN, 1960, p. 289), e Edmundson prefere empregar apenas a denominação “*common words*” (EDMUNDSON, 1969).

A ideia básica de extrair palavras-chave a partir de uma massa de texto (sendo títulos ou textos completos) e associa-las de algum modo ao documento como um todo ou partes dele é simplesmente muito boa e muito barata para desaparecer¹⁸ (WILLIAMS, 2010, p. 848, tradução nossa).

Há também o procedimento de “consolidação das palavras escritas da mesma forma em seu início, exemplo ‘similar’ e ‘similaridade’”¹⁹ que serão consideradas idênticas para fins estatísticos (LUHN, 1958, p. 162, tradução nossa), procedimento este que seria aperfeiçoado e conhecido mais tarde, provavelmente após Lovins (1968), como *stemming*.

3.4 Um processo completo para a extração dos termos

Os autores Salton e McGill (1983, p. 71-75) descrevem um processo genérico completo para a extração dos termos. A proposta deles foca especificamente em documentos textuais que já estejam em formato processável por máquina.

¹⁶ The significance of degree of proximity is based on the characteristics of spoken and written language in that ideas most closely associated intellectually are found to be implemented by words most closely associated physically

¹⁷ For some time it has been understood that a frequency count of the significant words of a document (mostly nouns, adjectives, and verbs) can serve to isolate the special vocabulary used to convey information in any particular realm of discourse

¹⁸ The basic idea of extracting keywords from a mass of text (whether in titles or in full text) and linking them in some way to the entire document, or parts of the document, is simply too good and too inexpensive to go away

¹⁹ consolidation of words which are spelled in the same way at their beginning, such as similar and similarity

O processo inicia a partir de um documento. Nele, todas as palavras de uso muito comum, com grande frequência de ocorrência entre os documentos, são eliminadas com o uso de um “dicionário negativo” (*stop list*). Esta lista composta pelas palavras que possuem um baixo poder discriminatório é específica de cada idioma, os autores exemplificam que no idioma Inglês existem aproximadamente 250 palavras que compõem entre 40 e 50% de qualquer documento, ou seja, ao eliminá-las o documento será reduzido a praticamente metade das palavras originais.

Após, as palavras restantes da fase anterior sofrem um tratamento linguístico denominado *stemming*, no qual as palavras flexionadas sofrerão a remoção de prefixos e sufixos, sendo reduzidas a sua forma raiz. Como exemplo, as palavras: *operate*, *operating*, *operates*, *operation*, *operative*, *operatives* e *operational*, se submetidas ao algoritmo denominado *Porter Stemmer* (PORTER, 1980) seriam reduzidas para a raiz “*oper*”. São essas raízes que serão candidatas a termo.

Uma vez obtida uma lista com os candidatos a termos, deve ser empregada alguma técnica baseada em frequência (DOCFREQ, DISCVALUE ou SIGNAL) para, a partir de um valor de corte (*threshold*), eleger as raízes que se tornarão termo. Os termos assim obtidos sofrerão atribuição de pesos, geralmente a partir de alguma técnica estatística, para que possam compor o vetor de índice ou de busca.

3.4.1 *Stemming*

A escrita, tomada como uma representação gráfica do idioma humano, possui uma flexibilidade quase tão grande quanto a da comunicação oral. Por essa razão um conceito pode ser expresso por diferentes termos, que podem ser grafados de inúmeras formas consideradas válidas. Os motivos para esta variabilidade vão desde exigências gramaticais, como conjugações e flexões, até chegarmos a questões estilísticas e figuras de linguagem. Um problema que surge devido a essa flexibilidade da ortografia é como agrupar essas variações, que remetem ao mesmo conceito, para que sejam contabilizadas como idênticas durante a análise estatística do texto. Uma proposta para isso é o processo denominado de “*stemming*”.

Um dos trabalhos pioneiros na área de *stemming* foi o de Julie Beth Lovins, publicado em 1968. A autora definiu o processo como:

Um algoritmo de *stemming* é um procedimento computacional que reduz todas as palavras com a mesma raiz para uma forma comum, normalmente eliminando de cada palavra os sufixos derivacionais e inflexionais²⁰ (LOVINS, 1968, p. 22, tradução nossa)

A ideia de transformar várias palavras que possuem uma origem comum em um termo, que não precisa, necessariamente, ser gramaticalmente válido antecede o trabalho de Lovins. Em 1958, Peter Luhn já propunha o que ele denominou de “consolidação”, onde palavras grafadas da mesma forma em seu início e que divergissem em até seis letras em seu final seriam consideradas idênticas, Hans Peter Luhn explica o processo:

[...] consolidação das palavras que são escritas da mesma forma em seu início, como “similar” e “similaridade”. Este procedimento era uma simples rotina de análise estatística consistindo em comparar letra a letra os pares de palavras sucessivas em uma lista alfabética. A partir do ponto onde acaba a coincidência entre as letras, um contador combinado era feito com as letras diferentes subsequentes em ambas as palavras. Quando esta contagem dava seis ou menos, as palavras eram presumidas como similares; acima de seis, como diferentes. Mesmo esse método de consolidação não sendo infalível, erros de até 5% não pareciam afetar os resultados finais do processo de elaboração de resumos. A máquina então contava as ocorrências das palavras similares derivadas desta maneira²¹ (LUHN, 1958, p. 162, tradução nossa).

No caso do idioma Inglês, um algoritmo popular é o chamado *Porter Stemming*, desenvolvido por Martin F. Porter (PORTER, 1980). Neste, há uma série de passos a serem seguidos que resultam em uma gradativa transformação de uma palavra em sua raiz. Como exemplo, no Quadro 2, os três primeiros passos do algoritmo de Porter aplicados a algumas palavras de exemplo.

²⁰ A stemming algorithm is a computational procedure which reduces all words with the same root [...] to a common form, usually by stripping each word of its derivational and inflectional suffixes.

²¹ [...] consolidation of words which are spelled in the same way at their beginning, such as similar and similarity. This procedure was a simple statistical analysis routine consisting of a letter-by-letter comparison of pairs of succeeding words in the alphabetized list. From the point where letters failed to coincide, a combined count was taken of the non-similar subsequent letters of both words. When this count was six or below, the words were assumed to be similar notions; above six, different notions. Although this method of word consolidation is not infallible, errors up to 5% did not seem to affect the final results of the abstracting process. The machine then counted the occurrence of similar words derived in this way.

Quadro 2 — Primeiros passos do Porter Stemming

passo	terminação		exemplo	
	de	para	de	para
1a	<i>sses</i>	<i>ss</i>	caresses	caress
1a	<i>ies</i>	<i>i</i>	ponies	poni
1a	<i>ss</i>	<i>ss</i>	caress	caress
1a	<i>s</i>	[eliminar]	cats	cat
1b	[vogal] + “ <i>ing</i> ”	[eliminar]	walking sing	walk sing
1b	[vogal] + “ <i>ed</i> ”	[eliminar]	plastered	plaster
2	<i>ational</i>	<i>ate</i>	relational	relate
2	<i>izer</i>	<i>ize</i>	digitalizer	digitalize
2	<i>ator</i>	<i>ate</i>	operator	operate

Fonte: Adaptado de Porter (1980).

Para o idioma Português, há o *RSLP Stemmer*, criado por Viviane Pereira Moreira (na época chamada Viviane Moreira Orenge) e Christian Huyck (ORENGO; HUYCK, 2001).

Donna Harman (1991) faz algumas ponderações sobre vantagens e desvantagens em empregar técnicas de stemming na indexação automática e sugere uma solução híbrida:

uma abordagem realística para recuperação online seria o uso automático de um stemmer, empregando um algoritmo como o de Porter (1980) ou Lovins (1968), mas oferecer a possibilidade de manter o termo que foi submetido ao processo de stemming (o inverso de truncar), se o usuário perceber que a consulta com o termo radicalizado produziu muitos documentos não relevantes, a consulta poderia ser reenviada com aquele termo sinalizado para não sofrer radicalização. Desta maneira, os usuários aproveitariam das vantagens do stemming e seriam capazes de melhorar os resultados das consultas prejudicadas pelo procedimento de stemming²² (HARMAN, 1991, p. 14, tradução nossa).

A transformação de palavras que remetam a um mesmo conceito em grafias idênticas é necessária devido às limitações das técnicas de análise estatística empregadas para estimativa dos pesos que será tratada mais adiante. Quando o documento analisado é puramente textual, a extração de termos será baseada nas palavras, porém, o que realmente nos interessa são os conceitos nos quais estas palavras estão conectadas. O que denominamos por “termos” são as

²² a realistic approach for online retrieval would be the automatic use of a stemmer, using an algorithm like Porter (1980) or Lovins (1968), but providing the ability to keep a term from being stemmed (the inverse of truncation). If a user found that a term in the stemmed query produced too many nonrelevant documents, the query could be resubmitted with that term marked for nostemming. In this manner, users would have full advantage of stemming, but would be able to improve the results of those queries hurt by stemming

representações, geralmente textuais, e não necessariamente ortograficamente válidas, que formam grupos compostos por uma ou mais palavras para que durante o processo de contagem de termos, possamos efetivamente “contar conceitos”.

3.4.2 Stopword List

Nesta etapa, o objetivo é eliminar as palavras que possuam uma frequência muito elevada que é observada tanto entre os documentos que compõem o *corpus* quanto dentro de cada documento. Estas palavras são conhecidas como “palavras vazias” (*stop words*), pois justamente pelo fato de ocorrerem com muita frequência, perdem completamente o seu caráter discriminatório e passam a não oferecer qualquer representatividade conceitual.

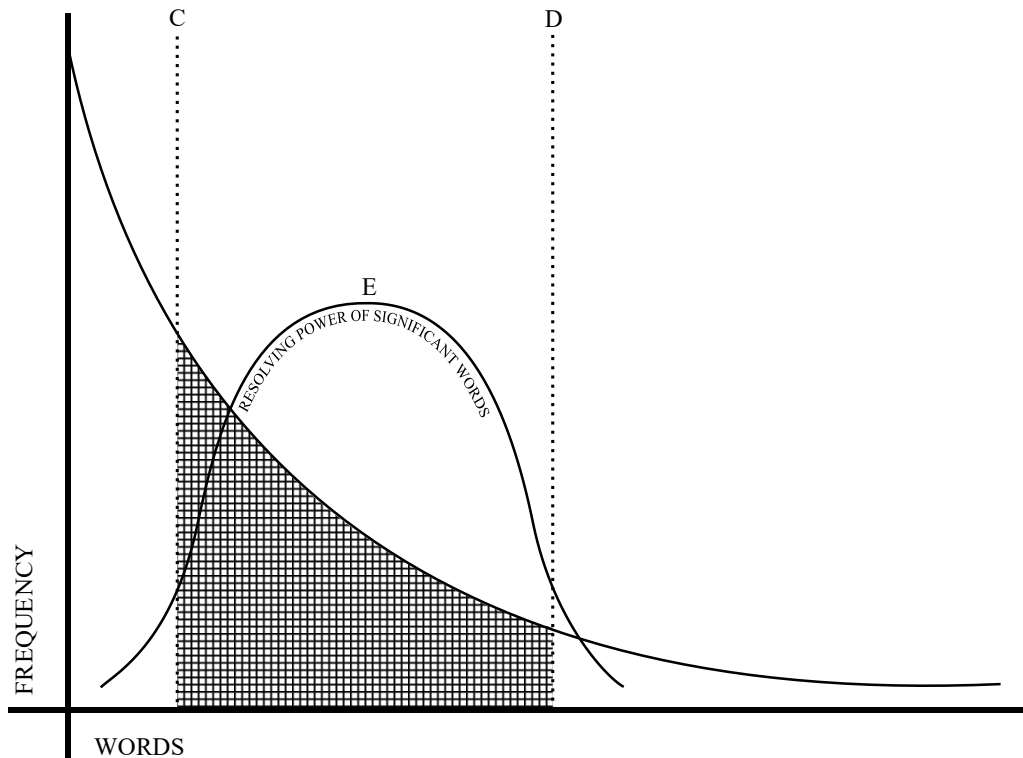
O princípio que embasa essa eliminação de palavras antes de realizar a contagem é que os conceitos centrais de que trata um documento trata estão expressos nas palavras que ocorrem no texto com uma frequência mediana. George Kingsley Zipf, em 1949, já havia discutido a relação entre a frequência de palavras de um texto e sua contribuição com os conceitos nele representados, sugerindo evidência empírica da existência de uma relação entre elas. Powers fez uma síntese deste trabalho de Zipf:

O principal trabalho de Zipf sobre este assunto explora uma teoria baseada em um processo competitivo de equilíbrio entre minimizar o esforço tanto daquele que fala quanto daquele que escuta. Ele utiliza uma analogia em que as palavras são consideradas ferramentas, que são construídas e organizadas para serem capazes de cumprir a tarefa de comunicação da forma mais eficiente possível²³. (POWERS, 1998, p. 152, tradução nossa).

Na Figura 3, as palavras que compõem um determinado documento são dispostas, por ordem de frequência, no eixo horizontal (eixo X) de um gráfico. Ao considerar a frequência em que estas palavras ocorrem dentro deste documento, valor no eixo vertical (eixo Y), obtém-se uma curva. Portanto, as palavras situadas na faixa delimitada pelos marcadores “C” e “D” (área hachurada), seriam as portadoras do maior poder de discriminação. O poder discriminatório das palavras que compõem este gráfico está representado pelo sino “E”.

²³ Zipf's major work on this subject explores a theory based on a competitive process balancing the minimization of the effort of both speaker and hearer. He uses an analogy in which words are regarded as tools, which are so constructed and arranged as to be able to achieve the communication task as efficiently as possible.

Figura 3 — Lei de Zipf



Fonte: LUHN, 1958, p. 161

Historicamente, Hans Peter Luhn (1958, p. 160, tradução nossa) já empregava o conceito de remoção das palavras muito frequentes e, portanto, vazias de significado. Ele usava o termo “*common-word list*” e não o termo “*stopword*”, mas conceitualmente são idênticos: “Este ruído pode ser materialmente reduzido pela técnica de eliminação na qual as palavras do texto são comparadas com uma lista armazenada de palavras comuns”²⁴.

Autores como Ellen Riloff (1995) tecem algumas críticas sobre a remoção de palavras utilizando listas predefinidas e algoritmos de *stemming*, bem como o uso de palavras isoladas como termos de indexação:

[...] palavras isoladas normalmente não fornecem contextualização suficiente para serem indicadores confiáveis para um domínio [porém,] frases um pouco maiores podem ser confiáveis.²⁵ (RILOFF, 1995, p. 131, tradução nossa)

²⁴ This noise can be materially reduced by an elimination technique in which text words are compared with a stored common-word list

²⁵ single words do not usually provide enough context to be reliable indicators for a domain, slightly larger phrases can be reliable.

3.4.3 Atribuição automática de pesos aos termos

O Modelo de Recuperação Espaço Vetorial oferece uma maneira formal que permite ordenar objetos por meio da similaridade e diferença entre suas propriedades. Para isso, cada característica do objeto representada em uma ou mais propriedades recebe um valor numérico que atuará como peso, beneficiando ou penalizado aquela característica daquele objeto em relação a mesma característica em outros objetos do conjunto durante a ordenação. A atribuição destes pesos às propriedades ocorre em dois momentos: na elaboração do vetor de busca e no momento da indexação, podendo esta atribuição ser feita de maneira manual, semiautomática ou automática. No caso específico da atribuição automática, o cálculo destes pesos geralmente é realizado por métodos estatísticos.

O processo de discriminação trata sobre a relação entre os documentos desejados e os outros documentos disponíveis. É um problema enfrentado quando se faz necessário indexar um conjunto de documentos com conteúdo semelhante, pois neste caso existirão diversos termos que mesmo sendo representativos sob o ponto de vista temático, não teriam muito utilidade se empregados isoladamente porque seriam incapazes de segmentar adequadamente o conjunto, portanto, estes termos possuem um poder discriminatório baixo.

Uma das mais conhecidas abordagens sobre o processo de discriminação é encontrada na obra de Karen Spärck Jones, que a partir dos conceitos de exaustividade e especificidade, delinea o que culminaria no conceito de TF-IDF. Segundo a autora, a especificidade é uma propriedade semântica do termo de indexação²⁶ (SPÄRK JONES, 1972, p. 14, tradução nossa), é esta propriedade que define a capacidade de discriminação de um termo, em outra dimensão teórica, a exaustividade é uma propriedade das descrições produzidas pelo processo de indexação, quanto mais termos descritores relacionados associados ao objeto, mais exaustiva será a indexação. A tese central da autora é que a especificidade de um termo deve ser interpretada em função do uso deste termo: “[a especificidade] deve ser interpretada como uma propriedade estatística em vez de semântica dos termos de indexação”²⁷ (SPÄRK JONES, 1972, p. 13, tradução nossa). Esta é a base teórica que fundamenta a abordagem estatística da extração de termos.

Quando utilizado o Modelo Espaço Vetorial se faz necessário definir quais serão as propriedades de cada objeto com seus respectivos pesos. No caso específico da indexação

²⁶ is a semantic property of index terms.

²⁷ It should be interpreted as a statistical rather than semantic property of index terms.

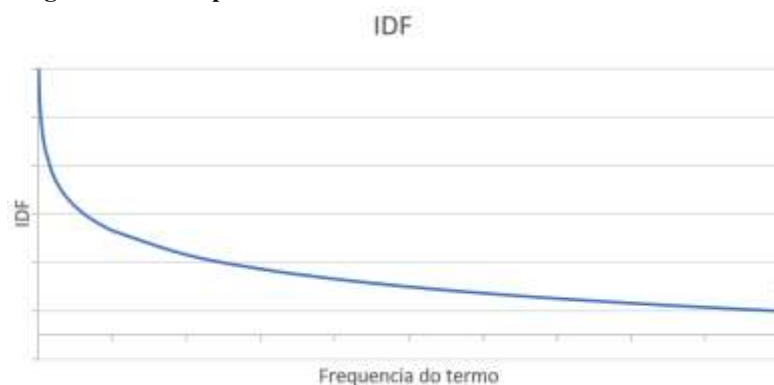
realizada de forma automática de conteúdo textual, tradicionalmente, os objetos serão os textos a serem recuperados e as propriedades serão alguns termos que possuam um grau moderado de discriminação com seus respectivos pesos. A escolha das palavras, simples e compostas, que serão transformadas em termos tem como critério beneficiar as palavras que aparecem com frequência mediana dentro de poucos textos, pois isso daria origem a termos com maior poder discriminatório. Da mesma forma os pesos seriam calculados objetivando beneficiar os termos que aparecem em poucos documentos do conjunto de documentos a ser indexado.

Uma fórmula para o cálculo do IDF de um termo k (SPARCK JONES, 1972 *apud* SALTON; MCGILL, 1983, p. 63):

$$IDF_k = \log_2 \frac{n}{Docfreq_k} + 1 = \log_2(n) - \log_2(Docfreq_k) + 1$$

Nesta fórmula n é o número de documentos que compõe o *corpus* e $Docfreq_k$ é a contagem do número de documentos em que o termo k aparece no mínimo uma vez. O uso da função *logaritmo de base 2* é necessário para que os termos raros não sejam muito beneficiados, em outras palavras, não obtenham um IDF_k baixo muito próximo de 1. Isso pode ser visto na Figura 4, onde fica evidenciado que se um termo for muito frequente seu IDF será baixo, porém essa proporção entre frequência de um termo no *corpus* e redução de seu IDF não é linear, conforme comprovado pelo padrão da curva do gráfico.

Figura 4 — Comportamento não linear do índice IDF de um termo



Fonte: Elaborado pelo autor.

Exemplo de aplicação desta fórmula: Dado um corpus contendo 1000 documentos e analisando apenas três termos (“A”, “B” e “C”): O termo “A” ocorrendo em 100 documentos, o termo “B” ocorrendo em 500 e o termo “C” ocorrendo em 900 documentos. O cálculo do IDF será:

$$IDF_A = \log_2 \frac{1000}{100} + 1 = \log_2 10 + 1 = 4,322$$

$$IDF_B = \log_2 \frac{1000}{500} + 1 = \log_2 2 + 1 = 2,000$$

$$IDF_C = \log_2 \frac{1000}{900} + 1 = \log_2 1,11 + 1 = 1,152$$

Neste exemplo, o termo C será o menos beneficiado, seguido do termo B e termo A.

Utilizando a fórmula de IDF_k , Salton e McGill (1983, p. 63) propõem uma função para cálculo dos pesos de cada termo extraído durante a indexação automática. O peso (WEIGHT) de um termo k em um documento i é dado por:

$$WEIGHT_{ik} = FREQ_{ik} * [\log_2(n) - \log_2(Docfreq_k) + 1]$$

Esta fórmula nada mais é do que $WEIGHT_{ik} = FREQ_{ik} * IDF_k$, onde $FREQ_{ik}$ é o *Term Frequency* (TF) do termo k especificamente no documento i , por isso o nome mais popular para esta fórmula de estimar os pesos dos termos é *TF-IDF*.

Continuando o exemplo anterior, se naquele mesmo corpus contendo 1000 documentos, nos três primeiros documentos (D1, D2, D3) a ocorrência dos três termos (A, B, C) considerados obedecer a seguinte frequência de ocorrência dentro de cada documento (Quadro 3).

Quadro 3 — Exemplo frequência de termo vs documento

	Termo A IDF = 4,322	Termo B IDF = 2,000	Termo C IDF = 1,152
Documento D1	3	10	3
Documento D2	10	16	6
Documento D3	2	7	5

Fonte: Elaborado pelo autor.

Ao efetuar o cálculo do $WEIGHT_{ik}$, que consiste em multiplicar o IDF de cada termo por sua frequência de ocorrência dentro do texto, resulta nos seguintes pesos para cada termo em cada documento (Quadro 4).

Quadro 4 — Exemplo de termo vs documento, ponderado pelo IDF

	Termo A	Termo B	Termo C
Documento D1	12,966	20,000	3,456
Documento D2	43,220	32,000	6,912
Documento D3	8,644	14,000	5,760

Fonte: Elaborado pelo autor.

Obtendo os seguintes vetores para cada documento:

$$D_1 = [12,966 \quad 20,000 \quad 3,456 \quad \dots]$$

$$D_2 = [43,220 \quad 32,000 \quad 6,912 \quad \dots]$$

$$D_3 = [8,644 \quad 14,000 \quad 5,760 \quad \dots]$$

São estes os vetores que serão armazenados pelo Sistema de Recuperação de Informação, em estruturas de dados próprias para este fim.

O mesmo procedimento é válido durante a elaboração do vetor de busca. Por exemplo, supondo que a expressão de busca seja composta com os termos “A” e “C”, obtém-se o Quadro 5.

Quadro 5 — Matriz de frequência dos termos da expressão de busca

	Termo A IDF = 4,322	Termo B IDF = 2,000	Termo C IDF = 1,152
Expressão de busca	1	0	1

Fonte: Elaborado pelo autor.

O vetor resultante para esta query será:

$$Query = [4,322 \quad 0 \quad 1,152 \quad \dots]$$

Uma vez obtidos ambos os vetores dos documentos do *corpus* e da *query*, é possível a comparar a query com os documentos, ou mesmo a comparar um documento com outro, por meio de uma função de similaridade. Esta função matemática fornece um único valor numérico que representa a similaridade ou a relevância, utilizado para ordenar a resposta.

Expansão de Consulta

O processo de recuperação de informação pode ser pensado como um ciclo interativo e incremental: (1) formulação da expressão de busca inicial; (2) julgamento de relevância dos resultados obtidos, e (3) reformulação da expressão de busca na tentativa de aproximar os resultados (documentos) obtidos da necessidade de informação. Este processo é repetido até que o usuário consiga satisfazer, ainda que parcialmente, a sua necessidade de informação. A principal dificuldade do usuário está em prever, por meio de sua expressão de busca, os termos que foram usados para representar os documentos que satisfarão a sua necessidade.

O processo de expansão de consulta visa melhorar a eficiência da recuperação de informação baseados no pressuposto de que as consultas (expressões de busca) definidas pelos usuários muitas vezes não refletem suas reais necessidades de informação. O objetivo principal é tentar aproximar a expressão de busca à necessidade do usuário por meio da adição de novos termos à expressão inicialmente formulada por ele e presumivelmente obter um conjunto de documentos mais relevante.

A relevância é um dos principais conceitos que perpassa todo o processo de recuperação de informação, mas é particularmente importante na implementação de ferramentas que auxiliem o usuário a refinar sua expressão de busca, obtendo resultados mais úteis e pertinentes.

4.1 O conceito de relevância

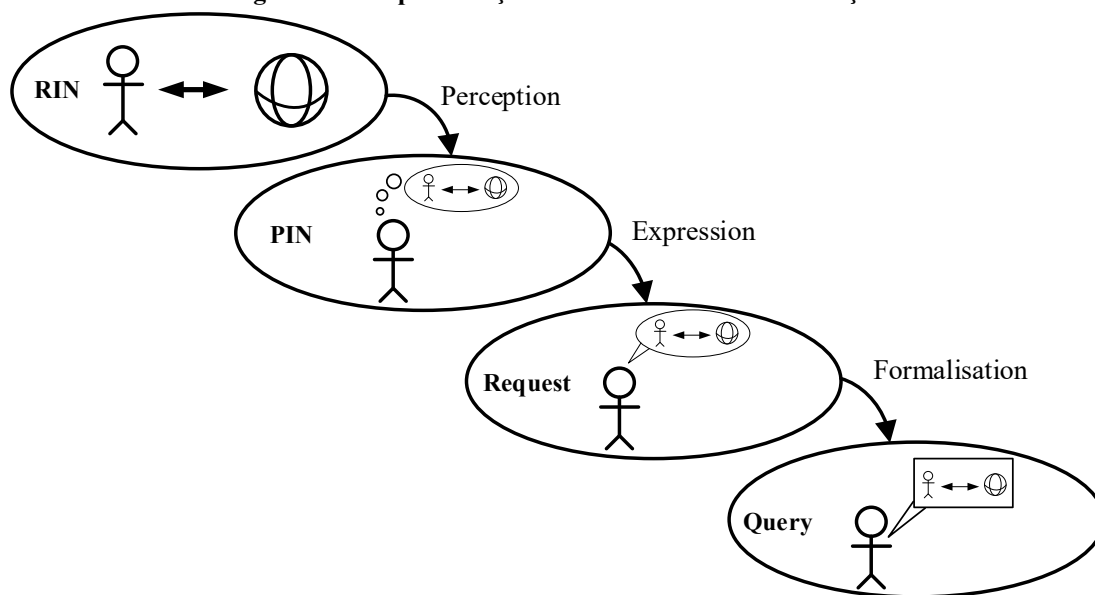
Fortemente relacionado à ideia de expansão de consulta está o conceito de relevância. A tentativa de aprimorar os resultados obtidos em uma busca pode ser analisada como uma

tentativa de aproximação entre o que o sistema de recuperação considera relevante com o que o usuário percebe como tal.

Stefano Mizzaro (1998) ilustra o processo da representação da necessidade informação do usuário em uma expressão de busca (*query*) a partir do diagrama apresentado na Figura 5.

O usuário possui uma necessidade de informação real (*Real Information Need* - RIN). Ao perceber sua RIN, o usuário constrói sua necessidade de informação percebida (*Perceived Information Need* - PIN). Portanto, PIN é a representação mental da RIN e não é, necessariamente, completa ou mesmo correta. A seguir, o usuário expressa a PIN por meio de uma *request*, sendo esta uma representação da PIN feita em uma linguagem humana, normalmente linguagem natural. A *request* será formalizada em uma *query* (expressão de busca) utilizando a linguagem e os recursos disponibilizados pelo sistema de recuperação de informação.

Figura 5 — Representação da Necessidade de Informação



Fonte: MIZZARO, 1998, p. 306

Ocorre redução no processo de representação da necessidade de informação, sendo representada pela seguinte expressão:

$$Query < Request < PIN < RIN$$

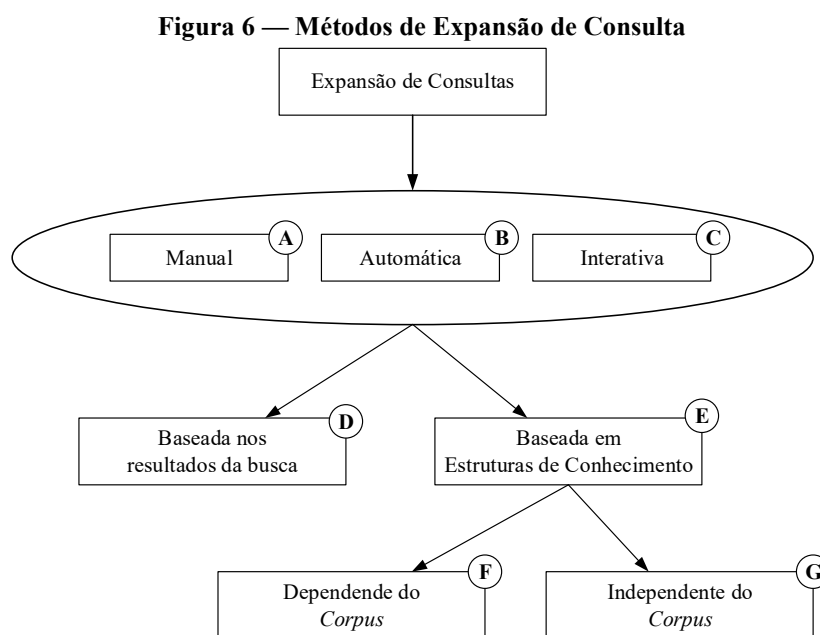
A questão apresentada por Mizzaro (1998, p. 307, tradução nossa) é que “há normalmente apenas uma tradução parcial da RIN na PIN, então na *request* e finalmente na

query”²⁸, ou seja, a cada nível de representação ocorre uma perda e uma distorção na representação da necessidade de informação em relação ao nível anterior.

A partir deste panorama, percebe-se que a tradução da necessidade de informação do usuário em uma expressão de busca é um processo que naturalmente envolve uma degradação progressiva da precisão conceitual desde a necessidade de informação real do usuário. Neste sentido, a Expansão da Consulta tenta minimizar essa degradação, aproximando conceitualmente, por meio da terminologia, a necessidade de informação percebida pelo usuário (PIN) com aquela formalizada por meio da expressão de busca (*Query*).

4.2 Métodos de expansão de consulta

A expansão de consulta tenta diminuir a ambiguidade inerente da linguagem natural por meio do agrupamento de um conjunto de termos que remeta a um conceito único que se aproxime da necessidade de informação do usuário. Os métodos de expansão de consulta podem ser analisados a partir do esquema proposto por Efthimiadis (1996), exposto na Figura 6:



Fonte: Adaptado de Efthimiadis, 1996.

Os elementos-chave a serem considerados na aplicação de qualquer forma de expansão de consulta são: (1) a fonte que fornecerá os termos para a expansão e (2) método que será usado para selecionar os termos utilizados na expansão.

²⁸ there is usually only a partial translation of the RIN into the PIN, then into the request and finally into the query

Efthimiadis (1996) identifica três métodos para realizar a expansão de consulta: (A) *manual*: quando o próprio usuário, com base nos resultados obtidos, altera a expressão de busca adicionando novos termos; (B) *automática*, quando o sistema, baseando-se nos resultados, adiciona automaticamente novos termos na expressão de busca original, sem qualquer interferência ou mesmo conhecimento do usuário; (C) *interativa*: o usuário interfere sobre a seleção de termos de expansão a partir de um conjunto de termos apresentados pelo sistema.

O modo manual de expandir a consulta (Figura 6 – A) é a prática mais convencional. Nela o usuário ao interagir com o sistema parte de alguns pressupostos, não necessariamente fundamentados, sobre quais termos o sistema possivelmente empregou para representar os documentos que lhe sejam interessantes e elabora sua expressão inicial de busca usando tais termos; somente após a primeira lista de resultados é que o usuário fará seu julgamento de relevância e tentará adequar os termos empregados, enviando uma nova expressão de busca já reformulada manualmente.

Em outro extremo temos a expansão automática de consulta (Figura 6, item B). Nesta situação, a reformulação é feita exclusivamente pelo sistema de forma automática, sem a interação com o usuário. Após a submissão da primeira expressão de busca elaborada pelo usuário, o sistema empregando algum critério cria um subconjunto destes resultados obtidos nesta consulta inicial. Alguns destes termos usados na indexação deste subconjunto escolhido serão adicionados à nova expressão de busca que será então submetida novamente. Este processo é repetido sucessivas vezes, a critério do sistema, na esperança de que consiga aumentar a quantidade de documentos relevantes para o usuário apresentados nas primeiras posições do resultado.

Também existe uma outra possibilidade: expansão interativa da consulta (Figura 6, item C). Este modo assemelha-se muito com a expansão manual na fase da primeira submissão da expressão de busca, porém ao obter a primeira lista de resultados o usuário de alguma forma informará ao sistema quais resultados foram (ou não) relevantes; neste momento o próprio sistema fará a reformulação da expressão de busca inicial, seu reenvio e a subsequente apresentação dos resultados novamente para o usuário fazer seu julgamento. Este processo pode ser repetido indefinidamente até que o usuário esteja satisfeito com os resultados obtidos.

Além dos modos, válidos para qualquer forma de expansão de consulta, precisamos considerar também a origem dos termos que serão utilizados no processo de expansão. Estes podem ser originados (1) a partir dos resultados da busca (Figura 6, item D) ou (2) baseados em estruturas de conhecimento externas (Figura 6, item E).

No primeiro caso (1) é a situação que ocorre com o procedimento de *relevance-feedback*. Neste procedimento, os documentos identificados como relevantes pelo usuário em iterações anteriores de busca, têm seus termos utilizados nesta iteração. Aqui, a seleção dos termos está implícita na seleção dos documentos relevantes pelo usuário, ou seja, quando o usuário indica os documentos relevantes, os termos de indexação empregados nestes documentos serão utilizados para a expansão da busca na iteração seguinte.

No outro caso (2), quando os termos são originários de estruturas externas de conhecimento (Figura 6, item E), estas estruturas podem ainda ser classificadas em duas subcategorias: (1) dependente do *corpus* (Figura 6, item F) como nos casos de procedimentos para expansão de siglas, remoção de sufixos, etc. ou (2) independente do *corpus* (Figura 6, item G), utilizando instrumentos de controle terminológico, como as ontologias.

Os termos utilizados na expansão de uma consulta podem ser gerados, como mostrado na Figura 6, de duas formas: originados dos *resultados de busca* (D) ou de *estruturas de conhecimento* (E). No primeiro caso (D), estes termos têm sua origem nos documentos recuperados a partir da consulta inicial, sendo a eficácia desta técnica de expansão de consulta fortemente dependente da qualidade dos resultados proporcionados pela consulta original. Por outro lado, no segundo caso (E) não há esta dependência qualitativa da consulta inicial pois a expansão ocorrerá baseada exclusivamente em estruturas de conhecimento. Estas estruturas de conhecimento podem ser *dependentes do corpus* (F) ou *independentes do corpus* (G).

Os mecanismos de expansão dependentes do *corpus* (F) consideram o acervo documental a fim de selecionar de forma automática os termos que serão utilizados para a expansão da consulta. Existem também mecanismos de expansão independentes do *corpus* (G) que se apoiam em estruturas de conhecimento que não possuem uma correspondência direta com os documentos, como os léxicos, glossários, dicionários, tesouros.

Nas seções seguintes será apresentado um detalhamento de cada uma das possíveis fontes de origem de termos para expansão de consulta.

4.2.1 Expansão de consulta baseada nos resultados da busca

A expansão de consulta baseada nos resultados da busca está fortemente relacionada à técnica de *Relevance Feedback*, que parte da ideia de que é mais intuitivo para o usuário avaliar a relevância de um conjunto de documentos recuperados do que formular uma primeira consulta que resulte um conjunto de documentos com relevância satisfatória. Como exemplo existem os

sistemas de recuperação de imagens, nos quais o usuário consegue rapidamente julgar a relevância das imagens recuperadas auxiliando no refinamento do processo de busca.

Os autores Manning, Raghavan e Schütze (2008, p. 163) resumem *Relevance Feedback* nos seguintes passos:

- O usuário elabora uma consulta inicial e a submete ao sistema;
- O sistema retorna um conjunto inicial de documentos;
- O usuário seleciona como relevante ou não-relevante alguns dos documentos recuperados e submete este subconjunto novamente ao sistema;
- O sistema adequa a representação da necessidade de informação baseada no *feedback* fornecido pelo usuário e refaz a consulta.
- O sistema apresenta o novo conjunto de documentos obtidos, supostamente com uma melhor precisão dos resultados.

Estes passos podem ser repetidos até que o usuário esteja satisfeito com o resultado obtido.

Com relação ao tipos de *Relevance Feedback*, Ruthven e Lalmas (2003) afirmam que existem basicamente dois tipos: (1) *User Relevance Feedback* e (2) *Pseudo Relevance Feedback*. Na primeira situação - *User Relevance Feedback*, ocorre a indicação pelo usuário dos documentos relevantes ou não resultantes de uma consulta, sendo essa informação reenviada ao sistema que a utilizará na reelaboração da consulta. No segundo caso - *Pseudo Relevance Feedback*, o sistema utilizará os documentos considerados mais relevantes para aperfeiçoar a consulta; note que neste caso não existe intervenção por parte do usuário, o sistema obterá o primeiro conjunto de resultados e a partir dele escolherá alguns dos documentos considerados mais relevantes e fará o *feedback* os utilizando, por isso que “Esta técnica depende fortemente da qualidade da consulta inicial e de sua aptidão em recuperar documentos relevantes” (FERNEDA, 2013, p. 72).

4.2.2 Expansão de consulta baseada em estruturas de conhecimento dependentes do *corpus*

Os métodos dependentes do *corpus* (F) consideram todos os documentos que compõem o acervo. Normalmente empregam técnicas estatísticas, como o cálculo de coocorrência para extrair termos que serão empregados na expansão da consulta.

Uma vantagem em empregar tais métodos é a possibilidade de formular a consulta inicial, sendo que tais estruturas de conhecimento podem ser criadas automaticamente, facilmente adaptáveis a um *corpus* com características dinâmicas (KRISTENSEN, 1993 *apud* FERNEDA, 2013, p. 73).

4.2.3 Expansão de consulta baseada em estruturas de conhecimento independentes do *corpus*

Com o uso de estruturas de conhecimento é possível, mesmo na consulta inicial, beneficiar-se do processo de expansão e também nos casos em que o número de documentos que compõem o *corpus* é muito reduzido ou quando esses documentos não possuem textos cuja extensão permita construir tais estruturas. Uma outra característica desejável é a possibilidade de usa-la a qualquer momento da elaboração da busca, não necessitando de um primeiro conjunto de resultados (Bhokal *et al.*, 2007).

A expansão de consulta é um elemento importante no processo de recuperação de informação favorecendo um melhor casamento entre a terminologia empregada pelo usuário durante a externalização de sua necessidade de informação por meio da elaboração da expressão de busca como com a terminologia empregada pelo sistema durante o processo de indexação do acervo documental.

4.3 Expansão de consulta baseada em ontologias

Expansão de consulta baseada em ontologias tem como objetivo, assim como é feito na indexação, resolver a ambiguidade dos termos por meio de sua contextualização. A ontologia age como um vocabulário controlado, permitindo que os termos sejam expandidos, ou seja, obtendo novos termos com significação semelhante para que possam ser adicionados à expressão de busca inicial.

No caso da desambiguação utilizando ontologias, o termo ambíguo não é substituído por outro termo não ambíguo, há a adição de novos termos relacionados ao contexto do termo original. Por exemplo: o termo “manga” poderia ter como complemento, adicionado automaticamente pelo processo de expansão, os termos: “fruta”, “suco”, “bebida”, “alimento”; com isso a ambiguidade entre “manga fruta” e “manga de roupa” seria resolvida, sem a necessidade do termo “manga” ser suprimido da expressão de busca.

Na expansão das consultas as ontologias são utilizadas como instrumento para compatibilização da terminologia empregada pelo usuário na elaboração de sua expressão de busca, trazendo novos termos hierarquicamente mais específicos e termos relacionados por semelhança conceitual.

5

Ontologias

A origem do termo “ontologia” pode ser datada em 1613, sendo atribuída a dois filósofos que possivelmente o criaram de maneira independente: Rudolf Göckel (Goclenius) que trouxe o termo em sua obra *Lexicon philosophicum* (1613) e Jacob Lorhard (Lorhardus) que também menciona o termo na segunda edição de *Theatrum philosophicum* (1613). A primeira ocorrência registrada em inglês segundo o Dicionário Oxford de Inglês apareceu no dicionário de Bailey em 1721 (SMITH, 2004, p. 155).

A popularização do termo ocorreu alguns anos mais tarde:

Foi apenas no ano de 1730, com a publicação da obra *Philosophia prima sive Ontologia* (Figura 2.4), de Christian Wolff (1679-1754), que o termo ontologia tomou visibilidade nos círculos filosóficos, sendo considerado sinônimo de *metaphysica generalis* – parte da metafísica que analisa as características do ser em geral. O livro de Wolff propõe investigar os predicados mais gerais de todos os entes por meio de um “método demonstrativo”, racional e dedutivo. (FERNEDA, 2013, p. 25)

Uma explicação para a etimologia deste termo é dada por Marilena Chaui:

A palavra ontologia é composta de duas outras: *onto* e *logia*. *Onto* deriva-se de dois substantivos gregos, *ta onta* (os bens e as coisas realmente possuídas por alguém) e *ta eonta* (as coisas realmente existentes). Essas duas palavras, por sua vez, derivam-se do verbo **ser**, que, em grego, se diz *einai*. O particípio presente desse verbo se diz *on* (sendo, ente) e *ontos* (sendo, entes). Dessa maneira, as palavras *onta* e *eonta* (as coisas) e *on* (ente) levaram a um substantivo: *to on*, que significa o **Ser**. O Ser é o que é realmente e se opõe ao que parece ser, à aparência. Assim, ontologia significa: estudo ou conhecimento do Ser, dos entes ou das coisas tais como são em si mesmas, real e verdadeiramente (CHAUÍ, 2012, p. 229 *apud* FERNEDA, 2013, p. 26).

Os autores Guarino, Oberle e Staab (2009, p. 1) distinguem dois significados, cada qual com grafias distintas: substantivo incontável “Ontologia” e substantivo contável “uma

ontologia”. A primeira forma (“Ontologia”) é uma ramificação da Filosofia, que lida com a natureza e a estrutura da realidade; enquanto que a segunda forma (“uma ontologia”) é definida como um objeto informacional ou um artefato computacional. Nesta segunda forma, a existência é condicionada à possibilidade de representação: “Para sistema de Inteligência Artificial o que *existe* é aquilo que pode ser representado” (GUARINO; OBERLE; STAAB, 2009, p. 2).

Ontologia segundo a abordagem da Filosofia:

A palavra ontologia tem origem no grego *ontos* (ser) e *logos* (palavra), e apesar do estudo do *ontos* originar-se com Aristóteles e Platão, a utilização do termo Ontologia para designar um ramo da Filosofia é muito mais recente, tendo sido introduzido na transição da Idade Média para a Idade Moderna, na escolástica, por volta dos séculos XVII e XVIII. Segundo Welty e Guarino (2001), o termo foi cunhado na área de Filosofia em 1613 por Rudolf Goclenius e aparentemente de forma independente por Jacob Lorhard (BOCATTO; RAMALHO; FUJITA, 2008, p. 202).

Outra definição, a partir da Lógica:

Na lógica, o quantificador existencial \exists é a notação para afirmar que alguma coisa existe. Mas a lógica não tem um vocabulário para descrever as coisas que existem. A ontologia preenche esta lacuna: é o estudo da existência de todos os tipos de entidades – abstratas e concretas – que compõe o mundo²⁹ (SOWA, 2000, p. 51, tradução nossa).

Ontologia, na visão da computação:

Uma ontologia é uma especificação explícita de uma conceitualização. O termo é emprestado da filosofia, onde a ontologia é um relato da existência. Nos sistemas baseados em conhecimento, o que “existe” é exatamente o que pode ser representado. Quando o conhecimento de um domínio é representado em um formalismo declarativo, o conjunto de objetos que podem ser representados é chamado universo do discurso. Este conjunto de objetos e as relações descritas entre eles são refletidas no vocabulário de representação que o programa baseado em conhecimento utiliza para representar o conhecimento³⁰. (GRUBER, 1993, p. 199, tradução nossa).

²⁹ In logic, the existential quantifier \exists is a notation for asserting that something exists. But logic itself has no vocabulary for describing the things that exist. Ontology fills that gap: it is the study of existence, of all the kinds of entity – abstract and concrete – that make up the world.

³⁰ An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what “exists” is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge.

A definição de ontologia feita por Thomas Gruber, citada anteriormente: **“Uma ontologia é uma especificação explícita de uma conceitualização”**, pode ser considerada uma definição clássica. Ela pode ser complementada por esta definição de outro autor:

“Ontologia” é o termo usado para referenciar uma compreensão compartilhada de um domínio de interesse [...]. Uma ontologia necessariamente implica ou inclui alguma espécie de visão do mundo em relação ao domínio dado. A visão de mundo é frequentemente concebida como um conjunto de conceitos (exemplo: entidades, atributos, processos), suas definições e suas interrelações; isto é referenciado com uma conceitualização³¹.

Tal conceitualização pode estar implícita, e existir apenas na mente de alguém, ou incorporada em um software. Por exemplo, uma suíte financeira pressupõe uma visão de mundo envolvendo conceitos como fatura e departamentos em uma corporação. A palavra “ontologia” é às vezes utilizada para fazer referência a esta conceitualização implícita. Entretanto, um uso mais padronizado e que adotaremos é que a ontologia é uma explícita representação de (ou parte de) uma conceitualização³². (USCHOLD; GRUNINGER, 1996, p. 96-97, tradução nossa).

As ontologias possuem uma característica muito peculiar em relação às ciências: seu objeto de estudo nunca será exclusivo, sempre será compartilhado com outras ciências. Não há como existir um objeto de estudo exclusivo das ontologias (HENNING, 2008, p. 39). Todos os objetos estudados em uma ontologia também são estudados por outra disciplina; a preocupação da ontologia está centrada na própria definição de existência e naquilo que possa vir a existir, ou seja, “A tarefa da ontologia é representar a realidade, ou melhor, apoiar as Ciências em sua representação da realidade”³³ (JANSEN, 2008, p. 173, tradução nossa).

Autores como Arp, Smith e Spear (2015, p. 2, tradução nossa) definem Ontologias como “artefatos representacionais” que buscam representar a realidade para que esta possa ser utilizada no desenvolvimento, teste e formalização de teorias científicas. Eles adotam as seguintes definições para os termos artefato e representação: Artefato é “alguma coisa que é projetada deliberadamente (ou, em certos casos específicos, escolhida) pelo humano para

³¹ “Ontology” is the term used to refer to the shared understanding of some domain of interest [...]. An ontology necessarily entails or embodies some sort of world view with respect to a given domain. The world view is often conceived as a set of concepts (e.g. entities, attributes, processes), their definitions and their inter-relationships; this is referred to as a conceptualization.

³² Such a conceptualisation may be implicit, e.g. existing only in someone's head, or embodied in a piece of software. For example, an accounting package presumes some world view encompassing such concepts as invoice, and a department in an organisation. The word "ontology" is sometimes used to refer to this implicit conceptualisation. However, the more standard usage and that which we will adopt is that the ontology is an explicit account or representation of (some part of) a conceptualisation.

³³ The task of ontology is to represent reality or, rather, to support the sciences in their representation of reality

atender a um determinado propósito”³⁴; Representação é “uma entidade (por exemplo, um termo, uma ideia, uma imagem, uma etiqueta, uma descrição, um ensaio) que faz referência a alguma outra entidade ou entidades”³⁵.

Uma ontologia pode ser entendida com um vocabulário de representação especializado em algum domínio. Uschold deixa claro este papel de vocabulário que ela pode assumir para uma determinada área do conhecimento e as consequências que isso traz:

Uma ONTOLOGIA pode possuir uma variedade de formas, mas necessariamente incluirá um vocabulário de termos, e alguma especificação de seus significados. Isto inclui definições e uma indicação de como conceitos estão inter-relacionados, o que impõe uma estrutura geral no DOMÍNIO e restringe as possíveis interpretações dos termos.³⁶

Uma ONTOLOGIA é virtualmente sempre a manifestação de um entendimento compartilhado de um DOMÍNIO que é consensual entre um número de agentes. Esta consensualidade facilita a precisão e efetividade da comunicação do significado, o que por sua vez acarreta outros benefícios como interoperabilidade, reuso e compartilhamento.³⁷(USCHOLD, 1998, p. 12, tradução nossa).

O conhecimento pode ser classificado em dois tipos: conhecimento factual (ou declarativo) e conhecimento procedural. O conhecimento declarativo explica o que as coisas são, por exemplo: “um cachorro é um mamífero, carnívoro, que possui quatro patas e uma cauda”. O conhecimento procedural tenta explicar como as coisas funcionam, por exemplo: “se o cachorro estiver faminto ele irá procurar alimento; para localizar o alimento ele tentará farejá-lo; se após comer este alimento ele ainda continuar faminto irá procurar mais” (RAMIREZ; VALDES, 2012, p. 47). Em particular para este trabalho, é interessante o conhecimento declarativo, que atualmente é empregado na representação do conhecimento pelas ontologias computacionais.

Na representação do conhecimento (*knowledge representation*), Sowa (2000, p. xi-xii) preceitua que é necessário o envolvimento de três áreas: Lógica, Ontologia e Computação, cada

³⁴ Artifact = def. something that is deliberately designed (or, in certain borderline cases, selected) by human beings to address a particular purpose

³⁵ Representation = def. an entity (for example, a term, an idea, an image, a label, a description, an essay) that refers to some other entity or entities

³⁶ An ONTOLOGY may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the DOMAIN and constrain the possible interpretations of terms

³⁷ An ONTOLOGY is virtually always the manifestation of a shared understanding of a DOMAIN that is agreed between a number of agents. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing

uma exercendo um papel complementar com as outras duas. Neste contexto, a Lógica tem o papel de fornecer a estrutura formal e postular as regras de inferência eliminando a redundância e a contradição; a Ontologia serve para delimitar os tipos de “coisas” que existem no domínio em questão e, por fim, a Computação viabilizando a aplicabilidade prática das duas áreas anteriores. É desta forma que a Ciência da Computação precisa representar o conhecimento em um artefato denominado ontologia computacional, viabilizando seu processamento por computadores.

5.1 Ontologias computacionais e a Linguagem OWL

As ontologias computacionais são formais. O formalismo de uma disciplina não está necessariamente associado ao simbolismo formal adotado na representação. Pode-se dizer “todos os homens são mortais” ou simplesmente grafar esta afirmação de forma simbólica $\forall x: Homem(x) \rightarrow Mortal(x)$. Não é o fato de haver esta simbologia formal mais densa que caracteriza o formalismo da afirmação. O formalismo está na forma e nas características das estruturas adotadas (HENNING, 2008, p. 44). Portanto, a definição de uma ontologia formal independe da simbologia utilizada, podendo até ser feita no formato de narrativa.

Dentre as propostas de linguagem para criação de ontologias computacionais, uma que vem recebendo bastante atenção no meio acadêmico desde sua publicação é a chamada *Web Ontology Language* (OWL). Esta surgiu como uma tecnologia de base para a proposta da Web Semântica (BERNERS-LEE; HENDLER; LASSILA, 2001), o que não impede que ela também possa ser utilizada em sistemas independentes da Web (WORLD WIDE WEB CONSORTIUM, 2012).

A primeira publicação do padrão de linguagem OWL foi em 2004, resultado do grupo de trabalho *WebOnt* instituído pela *World Wide Web Consortium* (W3C).³⁸ A linguagem OWL sofreu uma revisão com a publicação da recomendação W3C em 27 de outubro de 2009 (MOTIK; PATEL-SCHNEIDER; PARSIA, 2009) dando origem a linguagem denominada OWL2, (atualmente estamos na segunda edição desta especificação: MOTIK; PATEL-SCHNEIDER; PARSIA, 2012a).

A OWL foi definida como uma linguagem para representar o conhecimento, projetada para intercâmbio e inferência sobre um domínio de interesse. Tem como fundamento os conceitos de: Axiomas (*Axioms*), Entidades (*Entities*) e Expressões (*Expressions*); também

³⁸ <https://www.w3.org/2004/01/sws-pressrelease>

adota os pressupostos de mundo aberto e não unicidade dos nomes (HITZLER *et. al.*, 2012). A especificação OWL2 oferece extensões ao padrão OWL inicial, permitindo que haja compatibilidade entre OWL e OWL2, o que significa que todas as ontologias desenvolvidas em OWL são completamente compatíveis com a nova especificação OWL2, e a menos que tais extensões sejam empregadas ambas as linguagens são idênticas, por esse motivo, todas as referências à OWL também são aplicáveis à OWL2 e neste trabalho será adotado apenas a designação OWL para ambas.

A linguagem OWL está apoiada em outras tecnologias propostas e padronizadas pelo W3C, sendo de especial interesse para este trabalho o RDF e o RDFS.

O *Resource Description Framework* (RDF) é uma “linguagem formal para descrever informação estruturada” (HITZLER; KRÖTZSCH; RUDOLPH, 2009, p. 19). Estabelece uma forma de fazer descrições baseado no conceito de triplas. Ela é a base de outras tecnologias como a OWL, empregada na elaboração de ontologias. Seu desenvolvimento iniciou-se em 1990, porém, a primeira especificação oficial foi publicada pela W3C somente em 1999. Neste período, entre 1990 e 1999, a ênfase foi a representação dos metadados sobre os recursos Web tais como: autoria e licenciamento; após a proposta da Web Semântica, em 2001, o RDF foi estendido para representar os relacionamentos em geral entre recursos sob a forma de um grafo direcionado, sendo publicada em 2004 uma nova especificação. O RDF oferece a base na qual outras tecnologias o utilizam como alicerce, por exemplo o RDFS e a OWL.

Intimamente ligado ao *Framework* RDF está o conceito de Vocabulário RDF ou simplesmente “vocabulário”. Ele “define conjuntos de elementos e a relação destes conjuntos com as propriedades que descrevem estes elementos”³⁹ (ALLENMANG, HENDLER; 2008, p. 94), ou seja, é um conjunto de termos com uma semântica formal já pré-estabelecida entre eles. O próprio framework RDF além de ser um modelo de dados, também é um Vocabulário RDF para a representação destes dados, pois define um conjunto mínimo de termo e uma semântica formal entre eles.

Todo vocabulário é um documento RDF bem formado, isso permite o seu processamento por qualquer ferramenta com suporte à RDF. Permitir o processamento não significa que qualquer ferramenta consiga realizar processos de inferência para qualquer vocabulário, para que isto ocorra a ferramenta deve suportar a semântica formal introduzida pelo vocabulário que se queira realizar inferências.

³⁹ It defines set of elements and the relationship of those sets to the properties that describe the elements.

Apoiado no modelo de dados e vocabulário estabelecidos pelo RDF, existe o vocabulário RDFSchema (RDFS), que traz uma expressividade maior em relação ao vocabulário RDF. Autores como Hitzler, Krötzsch, Rudolph (2009, p. 47) concebem o RDFS como uma linguagem para ontologias, portanto, nesta ótica **uma descrição RDFS é uma ontologia:**

A capacidade de especificar este tipo de conhecimento torna o RDFS uma linguagem para representação do conhecimento ou linguagem de ontologia, pois fornece meios para descrever uma parte considerável das interdependências semânticas de um particular domínio de interesse⁴⁰. [... entendendo que ...] uma ontologia é a descrição do conhecimento sobre um domínio de interesse, cuja essência é uma especificação processável por computadores com significado formalmente definido⁴¹(HITZLER; KRÖTZSCH; RUDOLPH, 2009, p. 47, tradução nossa).

O vocabulário RDFSchema (RDFS) possui um nível de expressividade que permite a representação de conhecimento terminológico, denominado *schema knowledge* e permite alguns tipos de inferência. Utilizando apenas o RDF e o RDFS é possível descrever taxonomias com todas as suas relações de dependência entre subclasses e superclasses.

A linguagem OWL traz um vocabulário cuja definição utiliza-se dos vocabulários RDF e RDFS. Esta relação entre os três vocabulários fica evidente durante o uso da linguagem OWL para a criação de ontologias, na qual a todo momento é utilizado termos de um ou de outro vocabulário.

5.2 Sintaxes para OWL

A W3C define a linguagem OWL a partir de duas perspectivas: sintática e semântica. Na perspectiva semântica, que define os significados de cada estrutura, estabelece que existem duas semânticas possíveis: a direta (*Direct Semantic*) e a baseada em RDF (*RDF-Based Semantic*). Na perspectiva sintática faz uma separação entre definição abstrata (estrutural) e concreta (serializada) (WORLD WIDE WEB CONSORTIUM OWL Working Group, 2012).

A noção abstrata de uma ontologia OWL é discutida pela W3C no documento *OWL 2 Structural Specification*. Este aborda todos os elementos estruturais de uma ontologia

⁴⁰ The capability of specifying this kind of schema knowledge renders RDFS a knowledge representation language or ontology language as it provides means for describing a considerable part of the semantic interdependencies which hold in a domain of interest.

⁴¹ An ontology is a description of knowledge about a domain of interest, the core of which is a machine-processable specification with a formally defined meaning.

computacional deste tipo, especificando os papéis e funcionalidades de cada um dos elementos envolvidos. Para uso prático, além da definição abstrata é necessária uma sintaxe concreta que permita o armazenamento, intercâmbio e manipulação concreta destas ontologias.

A estrutura de uma ontologia OWL (conceitos abstratos) pode ser representada como um grafo RDF (modelo abstrato) para processamento. Este mapeamento entre os elementos estruturais OWL e o grafo RDF é tratado no documento da W3C chamado *Mapping to RDF Graphs* (PATEL-SCHNEIDER; MOTIK, 2012).

O grafo RDF é um modelo de dados (nível conceitual) e não um formato lógico ou físico, ou seja, não existe um arquivo em formato de grafo RDF ou mesmo em formato OWL. Para que estes dados sejam processáveis por computadores é necessário que este modelo de dados seja representado em algum esquema lógico, existindo para isso os “formatos de serialização” (*serialization format*). Alguns dos formatos de serialização propostos pela W3C são:

- RDF 1.1 Turtle (BECKETT; BERNERS-LEE; PRUD’HOMMEAUX, 2014)
- RDF 1.1 N-Quads (CAROTHERS, 2014)
- RDF 1.1 N-Triples (BECKETT, 2014)
- RDF 1.1 TriG (BIZER; CYGANIAK, 2014)
- RDF 1.1 XML Syntax (GANDON; SCHREIBER, 2014)
- JSON-LD 1.0 (SPORNY *et al.*, 2014)
- OWL/XML (MOTIK; PATEL-SCHNEIDER; PARSIA, 2012b)
- Manchester Syntax (HORRIDGE; PATEL-SCHNEIDER, 2012)
- Functional Syntax (MOTIK; PATEL-SCHNEIDER; PARSIA, 2012a)

5.3 Sintaxe Funcional OWL

Para este trabalho usa-se a sintaxe funcional. A opção pelo uso da sintaxe funcional (*functional syntax*) foi motivada pelo fato dela ser muito próxima da especificação estrutural da OWL, fato esse destacado pela W3C e usado como justificativa para a adoção desta sintaxe nos exemplos oficiais. O documento base empregado ao longo desta seção foi *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)* dos editores Boris Motik, Peter F. Patel-Schneider e Bijan Parsia, publicado como recomendação pela W3C em 11 de dezembro de 2012 (MOTIK; PATEL-SCHNEIDER, 2009).

Na sintaxe funcional existem três categorias sintáticas empregadas na especificação da estrutura conceitual: (1) Entidades, (2) Expressões e (3) Axiomas. Todos serão discutidos nas seções seguintes.

Além destas categorias, qualquer entidade, axioma ou mesmo ontologia podem receber anotações. O uso para estas anotações será definido pela aplicação que as utiliza pois segundo os aspectos lógicos da ontologia estas anotações não são consideradas, funcionando apenas como dados associados aos itens para uso por terceiros.

5.3.1 Prefixos na OWL

O uso de nomes únicos para as entidades e relacionamentos é um pressuposto fundamental do modelo estrutural OWL. Cada nome deve remeter exclusivamente a um único elemento, entidade ou relacionamento, não precisando que cada elemento possua apenas um único nome, isso é obtido mediante o emprego de identificadores no formato IRI (*International Resource Identifier*), com isso cada identificador possuirá uma entidade responsável pelo controle daquele espaço de nomes, e cada entidade deve garantir a unicidade mencionada. Um dos inconvenientes da adoção de identificadores do tipo IRI é o comprimento, em caracteres, que o nome completo (canônico) pode atingir, sendo interessante o uso de artifícios para a abreviação destes nomes.

A sintaxe concreta do tipo funcional, assim como algumas outras sintaxes concretas, pode utilizar abreviações (prefixos) para os nomes das entidades e relacionamentos. Com o uso de prefixos é possível abreviar um nome muito extenso, tornando a notação mais enxuta e por isso mais fácil de ser visualizada e compreendida. A definição de uma ontologia nesta sintaxe, portanto, pode iniciar-se com a declaração dos prefixos, conforme Figura 7.

Figura 7 — OWL exemplo declaração de prefixo

```
Prefix(rdfs:<http://www.w3.org/2000/01/rdf-schema#>)
```

Fonte: Elaborada pelo autor.

Posteriormente, ao utilizar o nome "rdfs:label" este será idêntico, em sentido de identificação, ao uso do nome completo "http://www.w3.org/2000/01/rdf-schema#label".

Existe um prefixo especial denominado "empty" ou "default", que será utilizado durante a expansão dos prefixos quando nenhum prefixo for especificado explicitamente, conforme exemplificado na Figura 8.

Figura 8 — OWL exemplo prefixo default

```
Prefix( := <http://www.janaite.net/ontologies/2018/3/informatica#> )  
Declaration( Class( :AllInOne ) )
```

Fonte: Elaborada pelo autor.

Será funcionalmente idêntico à declaração feita na Figura 9:

Figura 9 — OWL exemplo sem o uso de prefixo

```
Declaration( Class( http://www.janaite.net/ontologies/2018/3/informatica#:AllInOne ) )
```

Fonte: Elaborada pelo autor.

Os prefixos `rdf:`, `rdfs:`, `xsd:`, e `owl:`, conforme Quadro 6, formam o vocabulário reservado da linguagem OWL2, sendo opcional sua declaração no início de uma ontologia. Eles sempre estarão definidos, mesmo que não apareçam explicitamente.

Quadro 6 — Prefixos do vocabulário da OWL

prefixo	IRI completa
<code>rdf:</code>	<code><http://www.w3.org/1999/02/22-rdf-syntax-ns#></code>
<code>rdfs:</code>	<code><http://www.w3.org/2000/01/rdf-schema#></code>
<code>xsd:</code>	<code><http://www.w3.org/2001/XMLSchema#></code>
<code>owl:</code>	<code><http://www.w3.org/2002/07/owl#></code>

Fonte: Elaborado pelo autor

5.3.2 Estrutura da ontologia

Toda definição de ontologia (na sintaxe funcional), vista como uma estrutura hierárquica possui apenas dois elementos no topo da hierarquia: prefixos (que são opcionais) e a definição propriamente dita, contida no elemento sintático `Ontology(...)`.

Figura 10 — OWL exemplo de estrutura completa de uma ontologia

```
Prefix( : =<http://www.janaite.net/ontologies/2018/3/informatica#>

Ontology( <http://www.janaite.net/ontologies/2018/3/informatica>
  Import( <http://www.example.com/ontology2> )
  Annotation( rdfs:comment "Ontologia de exemplo"@pt-br )
  Annotation( owl:versionInfo "1.0"@pt-br

  [AXIOMAS...]
)
```

Fonte: Elaborada pelo autor.

Dentro do elemento sintático `ontology(...)` podem existir até quatro outras categorias de elementos, todos opcionais, sendo elas: (1) a definição do nome base da ontologia, (2) as importações de outras ontologias que forem necessárias, (3) as anotações referentes a esta ontologia e (4) todos os axiomas.

A Figura 10 ilustra todos esses elementos: (1) é definido o nome base desta ontologia como "`http://www.janaite.net/ontologies/2018/3/informatica`", (2) é feita a importação de uma outra ontologia hipotética disponível no recurso nomeado "`http://www.example.com/ontology2`", (3) são associadas duas anotações (metadados) a esta ontologia, sendo a primeira um comentário do tipo `rdfs:comment` e a segunda um metadado de versionamento do tipo `owl:versionInfo`.

5.3.3 Declarações implícitas

Um detalhe importante são as declarações implícitas que existem em todas as ontologias OWL. Todas estas declarações não são visíveis no formato serializado da ontologia. Elas existem por causa da maneira como foi definida a semântica formal com suas regras de inferência. Podem ser vistas como pares de conjuntos de entidades, existindo o conjunto que contém todas as entidades de um determinado tipo e o conjunto negado que não contém.

São elas:

- `Declaration(Class(owl:Thing))`
- `Declaration(Class(owl:Nothing))`
- `Declaration(ObjectProperty(owl:topObjectProperty))`
- `Declaration(ObjectProperty(owl:bottomObjectProperty))`

- `Declaration(DataProperty(owl:topDataProperty))`
- `Declaration(DataProperty(owl:bottomDataProperty))`
- `Declaration(Datatype(rdfs:Literal))`
- `Declaration(Datatype(I))`
 - Para cada tipo de dado “I” existe implicitamente esta declaração
- `Declaration(AnnotationProperty(I))`
 - Para cada propriedade de anotação “I” existe implicitamente esta declaração

Todas estas declarações trazem consequências globais: Todos os indivíduos do universo pertencem à classe `owl:Thing`. Todos eles estão relacionados entre si pela propriedade `owl:topObjectProperty`. Assim como todos eles também estão relacionados a todos os valores literais existentes por meio de `owl:topDataProperty`.

Por sua vez, não há qualquer indivíduo pertencente à classe `owl:Nothing`, assim como não há qualquer indivíduo relacionado a outro por meio de `owl:bottomObjectProperty` e relacionado a valores por meio de `owl:bottomDataProperty`.

5.3.4 Axiomas

Os axiomas são o principal componente de uma ontologia. Eles são afirmações sobre tudo aquilo que é verdade no domínio descrito. A especificação estrutural de um axioma pode tomar as formas descrita no Quadro 7.

Quadro 7 — OWL Tipos de Axiomas

Tipo de Axioma	Palavra-chave da OWL	Exemplificado em
Axiomas de Declarações	<code>Declaration</code>	Figura 11
Axiomas de Classe	<code>SubClassOf</code> <code>EquivalentClasses</code> <code>DisjointClasses</code> <code>DisjointUnion</code>	Figura 16, Figura 17
Axiomas de Propriedades de objetos	<code>SubObjectPropertyOf</code> <code>EquivalentObjectProperties</code> <code>DisjointObjectProperties</code> <code>InverseObjectProperties</code> <code>ObjectPropertyDomain</code> <code>ObjectPropertyRange</code> <code>FunctionalObjectProperty</code> <code>InverseFunctionalObjectProperty</code> <code>ReflexiveObjectProperty</code> <code>IrreflexiveObjectProperty</code> <code>SymmetricObjectProperty</code> <code>AsymmetricObjectProperty</code> <code>TransitiveObjectProperty</code>	Figura 12 (apenas <code>FunctionalObjectProperty</code> , <code>ObjectPropertyDomain</code> e <code>ObjectPropertyRange</code>)

Axiomas de Propriedades de Dados	SubDataPropertyOf EquivalentDataProperties DisjointDataProperties DataPropertyDomain DataPropertyRange FunctionalDataProperty	Figura 13 (apenas DataPropertyDomain e DataPropertyRange)
Axiomas de definição de tipo de dado	DatatypeDefinition	(Não exemplificado)
Chaves (semelhante ao conceito usado em bancos de dados)	HasKey	(Não exemplificado)
Axioma de Asserção	SameIndividual DifferentIndividuals ClassAssertion ObjectPropertyAssertion NegativeObjectPropertyAssertion DataPropertyAssertion NegativeDataPropertyAssertion	Figura 18 (apenas ClassAssertion, ObjectPropertyAssertion, DataPropertyAssertion)
Axioma de Anotação	AnnotationAssertion	Figura 14, Figura 15

Fonte: Elaborado pelo autor, com base em Motik, Patel-Schneider, Parsia, 2012a

Desta lista serão detalhados apenas os axiomas empregados na definição da ontologia utilizada como exemplo neste trabalho (Figura 20).

5.3.4.1 Axiomas de Declaração

Estes axiomas são utilizados para declarar entidades em uma ontologia. Não são considerados axiomas lógicos, pois não influenciam nas implicações lógicas. Eles servem para declarar as entidades⁴², definindo sua categoria e nome que serão manipuladas na ontologia em elaboração. Toda declaração é feita mediante o uso de `Declaration`, suportando as entidades demonstradas no Quadro 8.

Quadro 8 — OWL Axiomas de Declaração

Palavra-chave OWL	Forma de uso
Class	Declaration (Class (p:NOME))
Datatype	Declaration (Datatype (p:NOME))
ObjectProperty	Declaration (ObjectProperty (p:NOME))
DataProperty	Declaration (DataProperty (p:NOME))
AnnotationProperty	Declaration (AnnotationProperty (p:NOME))
NamedIndividual	Declaration (NamedIndividual (p:NOME))

Fonte: Elaborado pelo autor.

⁴² Entidades são os blocos fundamentais das ontologias OWL. Todas as entidades são definidas por uma URI.

No exemplo a seguir, Figura 11, existe a declaração de que os nomes `:Computador`, `:Desktop` e `:Fabricante` serão tratados como classes; `:marca` será uma propriedade de objeto; `:modelo` será uma propriedade de dados; e que os nomes `:desktop1`, `:fabricante-dell` e `:laptop1` serão indivíduos.

Figura 11 — OWL exemplo de Declarações

```
Declaration(Class(:Computador))
Declaration(Class(:Desktop))
Declaration(Class(:Fabricante))
Declaration(ObjectProperty(:marca))
Declaration(DataProperty(:modelo))
Declaration(NamedIndividual(:desktop1))
Declaration(NamedIndividual(:fabricante-dell))
Declaration(NamedIndividual(:laptop1))
```

Fonte: Elaborada pelo autor.

O conceito de Classe vem da taxonomia que objetiva agrupar os indivíduos por semelhança. No caso das ontologias, este agrupamento em classe pode ser realizado por asserção axiomática declarativa (como demonstrado no exemplo da Figura 11) ou por meio de regras de restrição existencial.

Propriedades de objeto são empregadas para criar relações entre indivíduos. Toda propriedade de objeto está hierarquicamente subordinada ao `owl:topObjectProperty`, sendo esta uma declaração implícita. No exemplo da Figura 11, a propriedade de objeto denominada `:marca` é, implicitamente, descendente da propriedade `owl:topObjectProperty`.

Propriedade de dados são utilizadas para relacionar um indivíduo a um literal (valor fixo). Assim como ocorre em outros tipos de propriedade, esta propriedade está hierarquicamente subordinada ao `owl:topDataProperty`, sendo esta, também, uma declaração implícita.

Por fim, os indivíduos são os elementos referenciáveis mediante o nome declarado, que poderão pertencer às classes, possuir relacionamentos entre si e valores literais associados

5.3.4.2 Axiomas de Propriedades de Objetos

Ao relacionar, de forma rotulada e direcional, dois objetos é preciso estabelecer o sentido desta relação. O domínio (*domain*) define, no caso das ontologias, os valores possíveis do sujeito da relação. Ao passo em que a imagem (*range*) define os valores possíveis do objeto

da relação. Relembrando que, conceitualmente, toda relação é feita no formato de tripla: sujeito → predicado → objeto, significando que o “sujeito” mantém uma relação caracterizada pelo “predicado” com o “objeto”.

No caso específico das propriedades de objetos, `ObjectPropertyDomain` define o rótulo da relação e o nome de uma classe à qual deve pertencer o indivíduo cujo nome esteja listado como sujeito desta relação. Já o "`ObjectPropertyRange`" faz algo semelhante, porém relativo ao objeto da relação.

Figura 12 — OWL exemplo de Propriedade Funcional de Objetos

```
FunctionalObjectProperty (:marca)
ObjectPropertyDomain (:marca :Computador)
ObjectPropertyRange (:marca :Fabricante)
ObjectPropertyAssertion (:marca :laptop1 :fabricante-desconhecido)
ObjectPropertyAssertion (:marca :laptop1 :fabricante-hp)

# será inferido: SameIndividual (:fabricante-hp :fabricante-desconhecido)
```

Fonte: Elaborada pelo autor.

A Figura 12 contém um exemplo de propriedade funcional, `:marca` é uma propriedade que permite relacionar alguma entidade que pertença à classe `:Computador` com outra entidade que pertença à classe `:Fabricante`, ao passo que a `FunctionalObjectProperty (:marca)` explicita um que cada entidade pertencente a classe `:Computador` pode possuir um única entidade pertencente a classe `:Fabricante` associado por meio da propriedade `:marca`. Isto traz implicações para o sistema de inferência, ao afirmar que o indivíduo `:laptop1` possui como `:marca` o indivíduo `:fabricante-hp` e o indivíduo `:fabricante-desconhecido`; como a propriedade `:marca` é funcional, assume-se que `:fabricante-hp` e `:fabricante-desconhecido` são o mesmo indivíduo, resultando na inferência de `SameIndividual (:fabricante-hp :fabricante-desconhecido)`, trazendo com consequência o pertencimento de `fabricante-desconhecido` a classe `:Fabricante`.

Além do mencionado, quando estabelecido este relacionamento o sistema de inferência atribui a todas as entidades que participem deste relacionamento como pertencentes a uma das duas classes, de acordo com o papel de cada entidade na relação.

5.3.4.3 Axiomas de Propriedades de Dados

As propriedades de dados são semelhantes às propriedades de objeto, a diferença está na classe da imagem (range) do relacionamento. Estas propriedades servem para relacionar entidades pertencentes a determinada classe com dados literais (valores literais fixos).

Figura 13 — OWL exemplo de Propriedades de Dados

```
DataPropertyDomain(:modelo :Computador)
DataPropertyRange(:modelo xsd:string)
```

Fonte: Elaborada pelo autor.

No exemplo na Figura 13, `:modelo` é uma propriedade que relaciona alguma entidade pertencente à classe `:Computador` com um valor literal do tipo `xsd:string`. Em caso de inferência, todas as entidades que constarem com sujeito de um predicado "modelo" serão também associadas à classe `:Computador`.

5.3.4.4 Axiomas de Anotação

Os axiomas de anotação servem para agregar dados às entidades. São multivalorados, ou seja, é possível atribuir diversas relações com mesmo rótulo, porém com valores distintos a uma mesma entidade.

Figura 14 — OWL exemplo de Asserção de Anotação

```
AnnotationAssertion(rdfs:label :Computador "computador"@pt)
AnnotationAssertion(rdfs:label :Computador "computer"@en)
AnnotationAssertion(rdfs:label :Computador "ordenador"@es)
```

Fonte: Elaborada pelo autor.

No exemplo da Figura 14, a entidade `:Computador` tem associado a ela três `rdfs:label`, cada um em idioma distinto. Este axioma oferece uma oportunidade de adição terminológica às entidades, mesmo que a nomenclatura das entidades seja semelhante, as vezes até idêntica, ao termo que a descreve, isso é por conveniência e não há qualquer garantia dessa correspondência. A identificação das entidades é usada internamente na ontologia, podendo inclusive ser puramente simbólica (exemplo: "abc123"). Uma forma de garantir a associação terminológica à uma entidade qualquer é mediante o uso de axiomas de anotação.

Conforme mencionado, qualquer entidade, axioma ou mesmo a ontologia como um todo podem receber anotações. Estas anotações não causam qualquer efeito sobre as definições

ou aspectos lógicos das ontologias, elas servem exclusivamente para associar dados e relacionamentos aos elementos definidos. Não há uma semântica formal definida para o tratamento destas anotações, elas são simplesmente ignoradas, devendo, se for o caso, serem tratadas por sistemas externos. Sua utilidade é grande, pois é a partir destas anotações que os elementos definidos podem ser relacionados com o ambiente no qual a ontologia irá operar. Geralmente as anotações utilizam outros vocabulários ao estabelecer os relacionamentos.

Na relação entre termos e conceitos, um predicado importante é o `rdfs:label`, definido no vocabulário RDFS (RDFS). Para a expansão de termos utilizando ontologias, a representação textual de um conceito é fundamental, por isso é necessário utilizar algum recurso para associar termos a conceitos definidos em uma ontologia, e uma das formas, possivelmente a mais direta, seria utilizar o `label` sendo ele específico para este fim: “`rdfs:label` é uma instância de `rdf:Property` que pode ser utilizada para fornecer uma versão do nome do recurso que seja legível por humanos”⁴³ (BRICKLEY; GUHA, 2014, tradução nossa). Existe uma diferença entre `label` e `comment` sendo “`rdfs:comment` é uma instância de `rdf:Property` que pode ser utilizada para fornecer uma descrição de um recurso”⁴⁴ (BRICKLEY; GUHA, 2014, tradução nossa). A associação entre o termo e o conceito poderá ser multilíngue. Os literais ao serem definidos, podem carregar consigo uma indicação de idioma, sendo permitido, inclusive, a definição de mais de um termo no mesmo idioma.

Figura 15 — OWL exemplo de Asserção de Anotação do tipo `rdfs:label`

```
Ontology(<http://www.janaite.net/ontologies/2018/3/informatica>
Annotation(rdfs:comment "Ontologia de exemplo"@pt-br)
Annotation(owl:versionInfo "1.0"@pt-br)

AnnotationAssertion(rdfs:comment :Netbook "small laptop without CD drive"@en)
AnnotationAssertion(rdfs:label :Netbook "netbook"@pt)
SubClassOf(:Netbook :Notebook)

)
```

Fonte: Elaborada pelo autor.

⁴³ *rdfs:label* is an instance of *rdf:Property* that may be used to provide a human-readable version of a resource's name

⁴⁴ *rdfs:comment* is an instance of *rdf:Property* that may be used to provide a human-readable description of a resource

Como exemplo prático, é demonstrado na Figura 15 como é possível associar indivíduos à nomes legíveis, termos. Seria possível, usando vocabulários externos, associar classes à elementos externos da ontologia.

5.3.4.5 Axiomas de Classe

Considerando que o conceito de classe remete diretamente a ideia de classificação taxonômica, é necessária uma forma de definir hierarquias entre as classes, isso é feito mediante o axioma de classe `SubClassOf`. Na inferência, uma entidade pertencente a classe filha é considerada como também pertencente à classe pai e assim sucessivamente. Implicitamente existe a classe `owl:Thing`, que é o topo da hierarquia, sendo que todas as classes definidas em qualquer ontologia têm como ancestral comum esta classe.

Figura 16 — OWL exemplo de Axiomas de Classe

```
SubClassOf(:Desktop :Computador)
SubClassOf(:Notebook :Computador)
SubClassOf(:Netbook :Notebook)
EquivalentClasses(:Laptop :Notebook)
DisjointClasses(:Computador :Fabricante)
DisjointUnion(:Computador :Desktop :Notebook :Tablet)
```

Fonte: Elaborada pelo autor.

No exemplo, temos uma hierarquia bem simples onde "Computador" tem como descendentes diretos `:Desktop` e `:Notebook`; por sua vez `:Notebook` tem como descendente `:Netbook`.

O axioma `EquivalentClasses` traz a ideia de classes equivalentes, que são manifestações com outros nomes para a mesma classe. Em uma situação mais simples ocorrerá a junção entre todos os atributos e indivíduos de uma com outra classe, tornando-as efetivamente sinônimas.

O axioma `DisjointClasses` serve para definir que o pertencimento de uma entidade à uma classe nega o seu pertencimento a uma outra classe. Na inferência esta negação pode ser confirmada.

No exemplo `:Computador` e `:Fabricante` são classes disjuntas, pois nenhuma entidade pode pertencer simultaneamente a ambas as classes. Na inferência ao associar uma entidade à classe "Computador" todos os caminhos inferenciais que concluem o pertencimento desta mesma entidade à classe "Fabricante" serão invalidados.

O axioma `DisjointUnion` é conhecido como "*covering axiom*". Ele define que ao pertencer a primeira classe, a entidade somente poderá pertencer a uma das outras classes. Veja o exemplo na Figura 17.

Figura 17 — OWL exemplo de Axiomas de Classe do tipo União Disjunta

```
DisjointUnion(:Computador :Desktop :Notebook :Tablet)
```

Fonte: Elaborada pelo autor.

Na Figura 17 o axioma estabelece que uma entidade qualquer ao pertencer a classe "`Computador`", poderá pertencer também a apenas uma outra das classes enumeradas `:Desktop`, `:Notebook` OU `:Tablet`.

5.3.5 Definição dos indivíduos

Após o estabelecimento das relações lógicas existentes entre as classes que compõem o domínio, segue a definição, por asserção, dos indivíduos.

Figura 18 — OWL exemplo de Definição de indivíduos

```
ClassAssertion(:Tablet :tablet1)
ObjectPropertyAssertion(:marca :tablet1 :fabricante-apple)
DataPropertyAssertion(:modelo :tablet1 "Ipad")
```

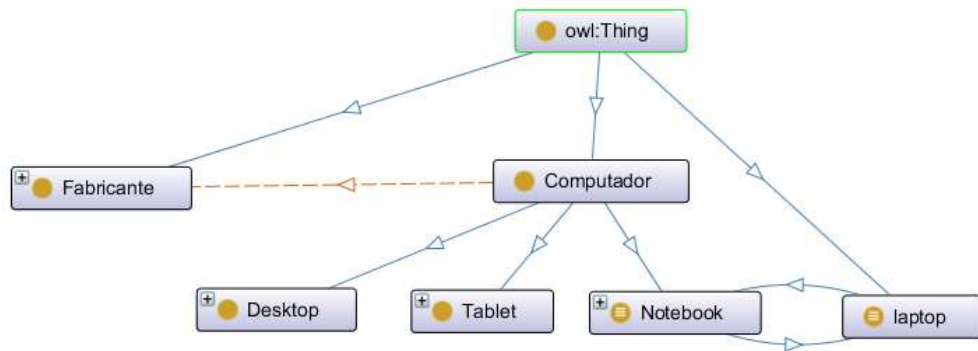
Fonte: Elaborada pelo autor.

No Figura 18 é retratada uma associação não inferencial da entidade "`tablet1`". Esta entidade é associada como pertencente à classe "`Tablet`", e são estabelecidas mais duas relações `:marca` e `:modelo`. O relacionamento `:marca`, associa a entidade `fabricante-apple` como objeto da relação; já o relacionamento `:modelo` associa o literal "`Ipad`" como objeto desta relação. Em síntese, é uma afirmação de que a entidade `:tablet1` é da classe `:Tablet`, da `:marca :fabricante-apple` e do `:modelo "Ipad"`.

5.4 Ontologia OWL de exemplo

Com o emprego de todos os axiomas discutidos e demonstrados mediante exemplos obtém-se a ontologia representada graficamente na Figura 19. Nesta ilustração estão em evidência as classes definidas bem como a hierarquia formada entre elas e também é observável a classe implícita `owl:Thing` ancestral comum de qualquer classe definida.

Figura 19 — OWL exemplo estrutura das classes



Fonte: Elaborada pelo autor.

Na Figura 20 é feita a definição completa de uma ontologia de exemplo (retomando muitos dos exemplos demonstrados em figuras anteriores) utilizando a sintaxe funcional.

Figura 20 — OWL exemplo completo utilizando a sintaxe funcional

```

Prefix(:=<http://www.janaite.net/ontologies/2018/3/informatica#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.janaite.net/ontologies/2018/3/informatica>
Annotation(rdfs:comment "Ontologia de exemplo"@pt-br)
Annotation(owl:versionInfo "1.0"@pt-br)

Declaration(Class(:AllInOne))
Declaration(Class(:Computador))
Declaration(Class(:Desktop))
Declaration(Class(:Fabricante))
Declaration(Class(:Laptop))
Declaration(Class(:Netbook))
Declaration(Class(:Notebook))
Declaration(Class(:Tablet))
Declaration(ObjectProperty(:marca))
Declaration(DataProperty(:modelo))
Declaration(NamedIndividual(:allinone1))
Declaration(NamedIndividual(:desktop1))
Declaration(NamedIndividual(:fabricante-apple))
Declaration(NamedIndividual(:fabricante-dell))
Declaration(NamedIndividual(:fabricante-desconhecido))
Declaration(NamedIndividual(:fabricante-hp))
Declaration(NamedIndividual(:laptop1))
    
```

```

Declaration (NamedIndividual (:netbook1))
Declaration (NamedIndividual (:notebook1))
Declaration (NamedIndividual (:tablet1))

ObjectPropertyDomain (:marca :Computador)
ObjectPropertyRange (:marca :Fabricante)

DataPropertyDomain (:modelo :Computador)
DataPropertyRange (:modelo xsd:string)

#####
#   Classes
#####

AnnotationAssertion (rdfs:label :AllInOne "All in one"@pt-br)
SubClassOf (:AllInOne :Desktop)

AnnotationAssertion (rdfs:label :Computador "computador"@pt)
AnnotationAssertion (rdfs:label :Computador "computer"@en)
AnnotationAssertion (rdfs:label :Computador "ordenador"@es)

AnnotationAssertion (rdfs:label :Desktop "computador de mesa"@pt-br)
AnnotationAssertion (rdfs:label :Desktop "desktop")
SubClassOf (:Desktop :Computador)

AnnotationAssertion (rdfs:label :Laptop "laptop")
EquivalentClasses (:Laptop :Notebook)

AnnotationAssertion (rdfs:comment :Netbook "small laptop without CD drive"@en)
AnnotationAssertion (rdfs:label :Netbook "netbook"@pt)
SubClassOf (:Netbook :Notebook)

AnnotationAssertion (rdfs:label :Notebook "notebook"@pt)
SubClassOf (:Notebook :Computador)

AnnotationAssertion (rdfs:label :Tablet "tablet"@en)
AnnotationAssertion (rdfs:label :Tablet "tablet"@pt-br)
AnnotationAssertion (rdfs:label :Tablet "táblete"@pt-pt)
SubClassOf (:Tablet :Computador)

#####
#   Named Individuals
#####

ClassAssertion (:AllInOne :allinone1)
ObjectPropertyAssertion (:marca :allinone1 :fabricante-dell)

```

```

DataPropertyAssertion(:modelo :allinone1 "Inspiron 24 5000")

ClassAssertion(:Desktop :desktop1)
ObjectPropertyAssertion(:marca :desktop1 :fabricante-hp)
DataPropertyAssertion(:modelo :desktop1 "EliteDesk 705 G3 Mini")

AnnotationAssertion(rdfs:label :fabricante-apple "Apple")
ClassAssertion(:Fabricante :fabricante-apple)

AnnotationAssertion(rdfs:label :fabricante-dell "Dell")
ClassAssertion(:Fabricante :fabricante-dell)

AnnotationAssertion(rdfs:label :fabricante-hp "HP"@en)
AnnotationAssertion(rdfs:label :fabricante-hp "Hewlett-Packard"@en)
AnnotationAssertion(rdfs:label :fabricante-hp "Hewlett-Packard Company"@en)
ClassAssertion(:Fabricante :fabricante-hp)

ClassAssertion(:Laptop :laptop1)
ObjectPropertyAssertion(:marca :laptop1 :fabricante-desconhecido)
ObjectPropertyAssertion(:marca :laptop1 :fabricante-hp)
DataPropertyAssertion(:modelo :laptop1 "ProBook 440")

ClassAssertion(:Netbook :netbook1)
ObjectPropertyAssertion(:marca :netbook1 :fabricante-hp)
DataPropertyAssertion(:modelo :netbook1 "STREAM 11")

ClassAssertion(:Notebook :notebook1)
ObjectPropertyAssertion(:marca :notebook1 :fabricante-dell)
DataPropertyAssertion(:modelo :notebook1 "Inspiron 15")

ClassAssertion(:Tablet :tablet1)
ObjectPropertyAssertion(:marca :tablet1 :fabricante-apple)
DataPropertyAssertion(:modelo :tablet1 "Ipad")

DisjointClasses(:Desktop :Notebook :Tablet)

)

```

Fonte: Elaborada pelo autor.

De uma maneira bem sucinta, uma ontologia OWL é composta por classes e subclasses, propriedades descritivas e relacionais, expressões lógicas e indivíduos. Além destes elementos, existem as anotações que podem aparecer em qualquer um dos elementos anteriormente mencionados. Essas anotações permitem a associação de metadados genéricos,

que serão completamente ignorados pelo sistema de inferência, a qualquer elemento de uma ontologia, possibilitando, por exemplo, a associação de bases terminológicas aos conceitos formalizados logicamente dentro dela.

Proposta de utilização de ontologias na recuperação de informação

A recuperação de informação é um processo de comparação entre as representações dos documentos de um acervo e a representação da necessidade de informação do usuário. A proposta a seguir utiliza ontologias como estrutura terminológica para enriquecer as representações dos documentos durante o processo de indexação automática. Propõem-se também a utilização de ontologias para auxiliar o usuário na tradução de sua necessidade de informação em uma expressão de busca que produza resultados mais adequados.

Os conceitos estabelecidos em uma ontologia OWL não possuem, necessariamente, uma definição terminológica explícita. Para a proposta apresentada neste trabalho, a representação textual de um conceito é fundamental, por isso é necessário utilizar algum recurso para associar termos a conceitos definidos em uma ontologia. Uma das formas, possivelmente a mais direta, seria utilizar o axioma de anotação do tipo *label* para este fim. Portanto, é pressuposto que todas as ontologias envolvidas possuam uma associação entre termos e conceitos, tornando possível uma consulta terminológica em tais ontologias.

Um questionamento muito comum é o porque da escolha de ontologias computacionais OWL em detrimento a outros instrumentos terminológicos como tesouros, taxonomias, etc. A opção por elas é fundamentada por: (1) possibilidade nativa de processamento completamente automático por computadores, (2) disponibilidade imediata e com relativa abundância⁴⁵ de ontologias prontas, (3) possibilidades de usos mais ambiciosos se

⁴⁵ Exemplo o site <<http://bioportal.bioontology.org>> que possui atualmente 716 ontologias da área biomédica (Acesso em 13 jun. 18)

utilizada restrições, inferências e demais recursos existentes exclusivamente neste tipo de instrumento.

6.1 Indexação automática de documentos

Nesta proposta, o processo de indexação automática segue as seguintes etapas: (1) associação de uma ontologia ao documento sendo indexado; (2) extração dos termos deste documento; (3) atribuição dos conceitos e cálculo dos pesos de cada termos representativo do conceito; (4) geração de uma lista de termos potenciais, extraídos, relevantes, mas não encontrados na ontologia. A seguir uma descrição detalhada de cada um destes passos.

6.1.1 Associando uma ontologia ao documento

Inicialmente é necessário que os documentos a serem gerenciados pelo sistema estejam agrupados em conjuntos temáticos, sendo atribuída uma ontologia de domínio específico para cada conjunto antes da indexação. Esta atribuição poderá ser feita manualmente, mediante a intervenção de um profissional da área ou mesmo de maneira automática. Se considerado o ambiente da Web, pode inclusive ser utilizado um *Crawler Focado*, que faz o *crawling* a partir de um critério temático, obtendo ao final um corpus temático que será submetido ao processo de indexação automática.

6.1.2 Extração de termos

Existem técnicas específicas para a extração de termos a partir de um documento textual. Algumas técnicas mais elementares já foram discutidas na seção 3.3 (A extração automática dos termos, p. 42) e exemplificadas na seção 3.4 (Um processo completo para a extração dos termos, p. 45).

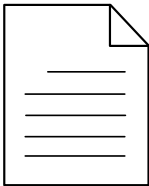
O processo utilizado para obter termos pertinentes ao assunto abordado em algum documento em formato textual é tratado pelo campo de pesquisa, na Ciência da Computação, denominado *Information Extraction*:

Isolar fragmentos de texto relevante, extrair informação relevante a partir dos fragmentos e, em seguida, agrupar as informações em uma estrutura coerente [...] O objetivo da pesquisa em *extração de informação* é construir sistemas que encontrem informações relevantes e ignorem informação estranha e irrelevante⁴⁶ (COWIE; LEHNERT, 1996, p. 81, tradução nossa).

Este trabalho independe da técnica utilizada para a extração de termos. O único requisito é que a extração forneça termos relevantes e que ser contabilizados para posteriormente receberem um coeficiente numérico (peso).

A indexação automática inicia com a extração dos termos e atribuição automática de pesos (valor numérico associado a cada termo que representa a importância conceitual que o termo possui dentro do texto analisado). Para isto, no exemplo, são utilizados os métodos de indexação propostos por Salton, Wong e Yang (1975), que foram discutidos no capítulo 3 (p. 38) sobre indexação automática. Ao final desse processo, obtém-se termos de indexação e seus respectivos pesos, conforme apresentado na Figura 21.

Figura 21 — Exemplo de termos extraídos com seus pesos atribuídos



T1	0,90
T2	0,82
T3	0,80
T4	0,60
T5	0,45
T6	0,30

Fonte: elaborada pelo autor

Neste exemplo foram extraídos seis termos e atribuídos pesos empregando as fórmulas mencionadas anteriormente. Foi predefinido um limite numérico mínimo do peso para que o termo seja considerado termo de indexação (0,80 neste caso), por isso, apenas três termos (T1, T2 e T3) seriam utilizados como termos de indexação.

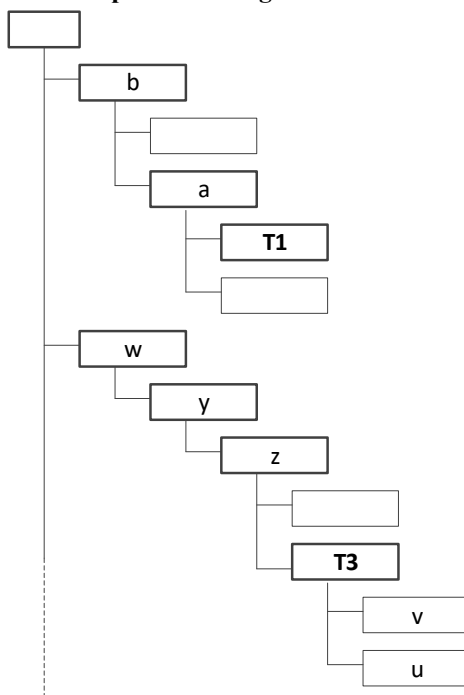
6.1.3 Atribuição de conceitos

Um termo extraído do texto deve coincidir com um termo encontrado na ontologia. Conforme já estipulado, é imprescindível que a ontologia empregada possua uma representação que de alguma forma permita consulta e navegação terminológica.

⁴⁶ It isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework. [...] The goal of IE research is to build systems that find and link relevant information while ignoring extraneous and irrelevant information.

Os termos de indexação, uma vez obtidos, serão considerados como conceitos centrais da ontologia associada ao *corpus*. Esta ontologia exerce duas funções: (1) expandir o conjunto de termos de indexação de cada documento; e (2) atribuir pesos a cada um dos termos. Na Figura 22 é apresentada uma possível visualização de uma ontologia como uma estrutura hierárquica de termos, não sendo considerado o tipo de relação existente entre os termos. Apenas os termos que possuem representação na ontologia serão utilizados para a indexação, neste caso, T1 e T3. Todos os demais termos que representarão o documento em questão serão derivados a partir daqueles termos centrais encontrados na ontologia.

Figura 22 — Exemplo de ontologia com termos hierárquicos



Fonte: elaborada pelo autor

Conforme o exemplo da Figura 22, o termo T1 é representado com 100% do valor estimado originalmente. Todos os termos hierarquicamente mais genéricos a T1 recebem pesos decrescentes, proporcionais ao distanciamento hierárquico, neste caso o termo “a” recebe o peso de 80% do valor estimado para T1 e o termo “b” recebe 60%. Para o termo T3, ocorre o mesmo procedimento: ele recebe 100% do valor e os conceitos derivados dele recebem, respectivamente, 80%, 60% e 40% do valor do peso estimado para ele. A Tabela 1 sintetiza estes valores.

Tabela 1 — Cálculo dos pesos dos termos derivados

Termo	Valores envolvidos no cálculo			Peso relativo	Peso final
	Distância hierárquica	Termo base considerado	Valor do termo base		
T1	0	T1	0,9	100%	0,90
a	1	T1	0,9	80%	0,72
b	2	T1	0,9	60%	0,54
T3	0	T3	0,8	100%	0,80
z	1	T3	0,8	80%	0,64
y	2	T3	0,8	60%	0,48
w	3	T3	0,8	40%	0,32

Fonte: elaborada pelo autor

O algoritmo proposto nesta pesquisa utiliza três parâmetros determinados experimentalmente de acordo com o perfil do *corpus* envolvido e de acordo com o conteúdo da ontologia empregada. São parâmetros: (1) peso mínimo do termo; (2) distância hierárquica máxima; (3) penalização hierárquica. A fórmula utilizada para a determinação do peso final de um termo é apresentada na Figura 23, onde.

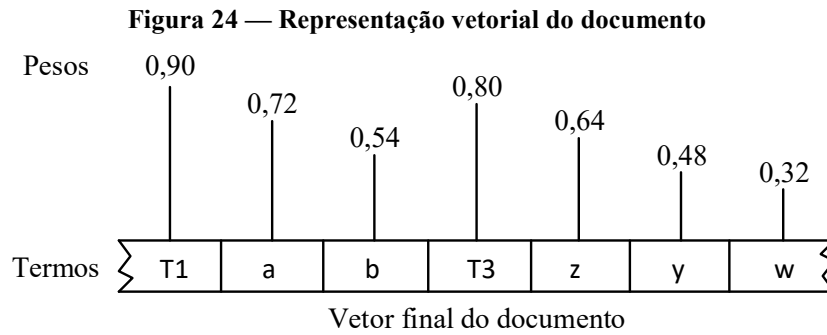
Figura 23 — Fórmula do peso final de um termo

$p(d, t, k) = (1 - d \cdot k) \cdot t$
<p>Onde:</p> <p>$d \{d \in \mathbb{N} \mid d \geq 0\}$: distância hierárquica do termo (zero quando for um termo base)</p> <p>$k \{k \in \mathbb{R} \mid 0 \leq k \leq 1\}$: fator de penalização hierárquica (no exemplo foi utilizado $k=0,2$)</p> <p>$t \{t \in \mathbb{R} \mid t \geq 0\}$: peso estimado para o termo base considerado</p>

Fonte: elaborada pelo autor

O parâmetro limite inferior (peso mínimo do termo) define um limite numérico mínimo do peso para que o termo seja considerado termo de indexação. O limite utilizado no exemplo foi 0.8, o que determinou que apenas os termos T1 e T3 fossem considerados como termos de indexação e conseqüentemente, passassem pelo processo de expansão.

Há um parâmetro que determina qual será a distância hierárquica máxima que será utilizada durante a expansão dos termos (Figura 23, parâmetro d). No caso da expansão de termos para indexação, representa o quanto generalizar na hierarquia. No exemplo foi utilizado o valor 3.



Fonte: elaborada pelo autor

Outro parâmetro é o fator de penalização hierárquica (Figura 23, parâmetro k) que especifica o quanto cada nível hierárquico de distanciamento penalizará o valor estimado de peso para o termo base; no exemplo empregamos um valor de 0,2, ou seja, a cada nível hierárquico de distanciamento reduzimos em 20% valor do peso base. O vetor obtido ao final do processo descrito é apresentado na Figura 24.

6.1.4 Lista de termos potenciais

O termo T2 embora tenha um peso significativo no documento, não está representado na ontologia e por isso foi descartado. Este termo poderia ser armazenado em uma lista separada, formando um conjunto de possíveis conceitos candidatos a serem inseridos manualmente na ontologia.

6.2 Especificação da busca

No Modelo Vetorial uma expressão de busca é representada por um vetor numérico. Neste, cada elemento corresponde a um termo de indexação e o seu valor numérico à importância do respectivo termo na descrição da necessidade de informação do usuário.

6.2.1 Escolha da ontologia

Ao elaborar a expressão de busca, o usuário deve, inicialmente, selecionar uma ontologia de domínio correspondente à sua necessidade de informação. Os termos definidos pelo usuário em sua expressão de busca (consulta) serão utilizados como conceitos centrais da ontologia associada a essa consulta. A ontologia terá duas funções: (1) expandir o conjunto de termos da consulta, acrescentando novos termos conceitualmente próximos; e (2) atribuir pesos a cada um dos termos adicionados à consulta.

6.2.2 Expansão da consulta

Para exemplificar a expansão de consulta, considera-se um usuário hipotético que escolhe uma ontologia correspondente ao seu domínio de interesse, neste caso, a ontologia representada graficamente na Figura 22; elabora sua expressão de busca contendo os termos “T3” e “T4”; submete esta expressão ao sistema e aguarda pelas respostas.

O sistema procederá o cruzamento de todos os termos contidos na expressão de busca com os termos disponíveis na ontologia selecionada, neste caso, o termo “T4” será descartado, pois não foi localizado na ontologia; o termo “T3”, por sua vez, será escolhido como um dos termos centrais, a partir dos quais será realizada a expansão hierárquica descendente. Neste exemplo, consultando a ontologia (Figura 22), o termo “T3” será expandido mediante a adição dos termos “v” e “u”. Obtendo, no final, o vetor de consulta composto pelos termos “T3”, “v” e “u”.

O cálculo dos pesos dos termos que compõe o vetor de consulta é feito da seguinte maneira: (1) todos os termos encontrados na ontologia e, portanto, considerados termos centrais, receberão peso no valor de 1,0; (2) todos os termos descendentes receberão pesos proporcionais à distância hierárquica em relação ao seu termo central.

A diferença entre o cálculo dos pesos dos termos expandidos de indexação em relação ao cálculo dos pesos dos termos expandidos de consulta é que estes últimos quando considerados termos centrais sempre receberão o valor constante de 1,0. A fórmula apresentada na Figura 23 continua válida, sendo o parâmetro “t” uma constante com o valor de 1,0.

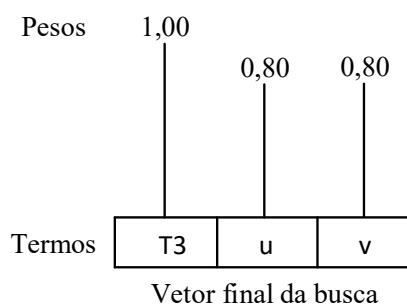
Tabela 2 — Cálculo dos pesos dos termos da busca

Termo	Valores envolvidos no cálculo				Peso final
	Distância hierárquica	Termo base considerado	Valor do termo base	Peso relativo	
T3	0	T3	1,0	100%	1,00
v	1	T3	1,0	80%	0,80
u	1	T3	1,0	80%	0,80

Fonte: Elaborada pelo autor

O vetor de busca resultante que será utilizado efetivamente no sistema pode ser visto na Figura 25.

Figura 25 — Representação vetorial da expressão de busca



Fonte: Elaborada pelo autor

Após a determinação dos pesos aplica-se o modelo de recuperação Espaço Vetorial conforme proposto por Salton e McGill (1983). A Tabela 3 exemplifica os valores associados aos termos de cada documento durante a indexação automática, bem como os valores estimados para o vetor de busca. Estes valores serão utilizados para o cálculo de similaridade conforme as fórmulas do Quadro 9.

Tabela 3 — Matriz termo/documento

	Termo T1	Termo a	Termo b	Termo T3	Termo z	Termo y	Termo w	Termo u	Termo v
Doc_A	0,90	0,72	0,54	0,80	0,64	0,48	0,32	0	0
Doc_B	0	0,85	0,68	0,95	0,76	0,57	0,38	0	0
Doc_C	0	0	0	0	0	0	0	0	0,9
Busca	0	0	0	1,0	0	0	0	0,80	0,80

Fonte: Elaborada pelo autor

Quadro 9 — Cálculo da similaridade entre documentos e expressão de busca

$$\|Busca\| = \sqrt{1^2 + 0,8^2 + 0,8^2} = \sqrt{2,28} \cong 1,510$$

$$\|Doc_A\| = \sqrt{0,9^2 + 0,72^2 + 0,54^2 + 0,80^2 + 0,64^2 + 0,48^2 + 0,32^2} = \sqrt{3,0024} \cong 1,733$$

$$sim_{cosine}(Busca, Doc_A) = \frac{(1 \times 0,8) + (0,8 \times 0) + (0,8 \times 0)}{\sqrt{2,28} \times \sqrt{3,0024}} = \frac{0,8}{1,510 \times 1,733} \cong 0,306$$

$$\|Doc_B\| = \sqrt{0,85^2 + 0,68^2 + 0,95^2 + 0,76^2 + 0,57^2 + 0,38^2} = \sqrt{3,134} \cong 1,770$$

$$sim_{cosine}(Busca, Doc_B) = \frac{(1 \times 0,95) + (0,8 \times 0) + (0,8 \times 0)}{\sqrt{2,28} \times \sqrt{3,134}} = \frac{0,95}{1,510 \times 1,770} \cong 0,355$$

$$\|Doc_C\| = \sqrt{0,9^2} = 0,9$$

$$sim_{cosine}(Busca, Doc_C) = \frac{(1 \times 0) + (0,8 \times 0) + (0,8 \times 0,9)}{\sqrt{2,28} \times 0,9} = \frac{0,72}{1,510 \times 0,9} \cong 0,530$$

Fonte: Elaborado pelo autor

Considerando que **a medida de similaridade por cosseno resulta em valores próximos de 1.0 para itens semelhantes**, a partir deste cálculo concluímos que o documento C (valor 0,530) é mais relevante para a expressão de busca, seguido pelos documentos B (valor 0,355) e A (valor 0,306). O documento que foi considerado mais relevante foi incluído nos resultados graças ao procedimento de expansão de consulta, pois originalmente o documento C não possuía o termo T3 (único termo empregado na busca); com a expansão, os termos “u” e “v” foram adicionados à busca, permitindo que documentos indexados sem os termos originais da busca fossem recuperados.

6.2.3 Lista de termos potenciais

No exemplo o termo “T4” foi descartado por não estar representado por um conceito da ontologia. Como este termo foi utilizado pelo usuário possivelmente ele possui certa importância dentro do domínio tratado. Se um termo específico for repetidamente utilizado por

diversos usuários, pode-se considerar que este termo deveria constar na ontologia. Esses termos serão armazenados em um tipo de repositório, formando um conjunto de potenciais conceitos a serem inseridos na ontologia durante uma atualização manual posterior.

Considerações finais

A recuperação de informação ocorre por meio de coincidências terminológicas entre duas representações. Existem as representações dos documentos de um acervo e a representação da necessidade de informação do usuário sintetizada na expressão de busca. Um documento será recuperado se sua representação coincidir total ou parcialmente com os termos utilizados na expressão de busca.

Partindo da premissa de que estamos diante de um problema linguístico, este trabalho propõe uma aproximação terminológica entre os termos utilizados para representar os documentos com os termos de busca utilizados pelo usuário. Para obter essa compatibilização propõe-se a utilização das ontologias computacionais como um vocabulário específico de uma determinada área do conhecimento (vocabulário de domínio), que permitem reduzir consideravelmente as ambiguidades inerentes à linguagem natural.

Com essas adequações terminológicas em ambas as representações, tanto dos documentos quanto da busca, será possível aumentar a possibilidade de coincidências entre elas, oferecendo melhorias na performance do sistema de recuperação de informação. O Modelo Vetorial oferece uma estrutura formal para essas representações (documentos e buscas) e permite que os resultados sejam ordenados de acordo com o grau de relevância estimado pelo sistema.

A proposta feita neste trabalho tem como característica a delimitação explícita do domínio no qual o processo de recuperação será realizado. Por um lado, os documentos são indexados utilizando termos pertencentes ao domínio específico de uma determinada ontologia; por outro lado, o usuário escolhe uma ontologia que será utilizada na expansão de sua busca, delimitando com isso seu assunto de interesse e restringindo o conjunto de termos a serem

utilizados na expansão de sua busca inicial. O resultado deste estudo demonstra a viabilidade no uso de ontologias como base terminológica na recuperação de informação podendo ser utilizado na implementação um protótipo para a realização de testes comparativos de eficiência entre sistemas convencionais e sistema baseados em ontologia.

Nesta proposta utiliza-se apenas a estrutura terminológica e hierárquica de uma ontologia. Não são considerados os demais tipos de relação possíveis nem as restrições lógicas que podem ser descritas em uma ontologia. Tais recursos podem ser utilizados em trabalhos futuros na tentativa de melhorar ainda mais a eficiência de um sistema, porém a complexidade de implementação e processamento trazidas pela utilização desses recursos será certamente maior e não fica evidente se os resultados obtidos serão significativamente melhores.

Referências

ALLENMANG, Dean; HENDLER, James A. **Semantic Web for the working ontologist: effective modeling in RDFS and OWL**. Amsterdam: Morgan Kaufmann. 2008. 330p. ISBN: 978-01-237-3556-0.

ALVARENGA, Lídia. Representação do Conhecimento na Perspectiva da Ciência da Informação em tempo e espaço digitais. **Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação**, Florianópolis, v. 8, n. 15, p. 18-40, 1º sem. 2003. Disponível em: <doi:10.5007/1518-2924.2003v8n15p18>. Acesso em: 08 ago. 2017.

ARP, Robert; SMITH, Barry; SPEAR, Andrew D. **Building Ontologies with Basic Formal Ontology**. Massachusetts: The MIT Press, 2015, 220 p., ISBN 978-0-262-52781-1.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 12.676: Métodos para análise de documentos: Determinação de seus assuntos e seleção de termos de indexação**. Rio de Janeiro, 1992, 4 p.

BECKETT, David. **RDF 1.1 N-Triples: A line-based syntax for an RDF graph**. W3C Recommendation 25 February 2014. 25 fev. 2014. Disponível em: <<http://www.w3.org/TR/2014/REC-n-triples-20140225/>>. Acesso em: 04 out. 2017.

BECKETT, David; BERNERS-LEE, Tim; PRUD'HOMMEAUX, Eric; CAROTHERS, Gavin. **RDF 1.1 Turtle: Terse RDF Triple Language**. W3C Recommendation 25 February 2014, 25 fev. 2014. Disponível em: <<http://www.w3.org/TR/2014/REC-turtle-20140225/>>. Acesso em: 04 out. 2017.

BERNERS-LEE, Tim ;HENDER, James;LASSILA, Ora. The semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. **Scientific American**, New York, May 2001.

BHOGAL, J.; MACFARLANE, A.; SMITH, P. A review of ontology based query expansion. **Information Processing and Management**, v. 43, n. 4, p. 866–886, 2007. Disponível em: <doi:10.1016/j.ipm.2006.09.003>. Acesso em: 15 ago. 2016.

BIZER, Chris; CYGANIAK, Richard. **RDF 1.1 TriG: RDF Dataset Language**. W3C Recommendation 25 February 2014. 25 fev. 2014. Disponível em: <<http://www.w3.org/TR/2014/REC-trig-20140225/>>. Acesso em: 04 out. 2017.

BLAIR, David C. Chapter 1: Information Retrieval and the Philosophy of Language. In: **Annual Review of Information Science and Technology**, v. 37, n. 1, 2003, p. 3-50. ISSN:1550-8382. Disponível em: <doi:10.1002/aris.1440370102>. Acesso em: 13 mai. 2017.

BOCCATO, Vera Regina Casari; RAMALHO, Rogério Aparecido Sá; FUJITA, Mariângela Spotti Lopes. A contribuição dos tesouros na construção de ontologias como instrumento de organização e recuperação da informação em ambientes digitais. **Ibersid: revista de sistemas de información y documentación**, 2008, p. 199-209. Disponível em: <<http://www.iversid.eu/ojs/index.php/iversid/article/view/2235>>. Acesso em: 13. fev. 2017. ISSN 1888-0967.

BRICKLEY, Dan (Ed.); GUHA, R. V. (Ed.) **RDF Schema 1.1**. W3C Recommendation 25 february 2014. Disponível em: <<https://www.w3.org/TR/2014/REC-rdf-schema-20140225/>>. Acesso em: 04 out.17.

BRIET, Suzanne. **Qu'est-ce que la documentation?** Paris: EDIT, 1951, 48 p.

BUCKLAND, Michael K. Information as thing. **Journal of the American Society for Information Science (JASIS)**, v. 42, n. 5, jun. 1991, p. 351-360. Disponível em: <[doi:10.1002/\(SICI\)1097-4571\(199106\)42:5%3C351::AID-ASI5%3E3.0.CO;2-3](https://doi.org/10.1002/(SICI)1097-4571(199106)42:5%3C351::AID-ASI5%3E3.0.CO;2-3)>. Acesso em 21 jul. 2015.

BUCKLAND, Michael K. What is a document? **Journal of the American Society for Information Science (JASIS)**. v. 48, n. 9, 1997, p. 804-809. Disponível em: <[doi:10.1002/\(SICI\)1097-4571\(199709\)48:9%3C804::AID-ASI5%3E3.0.CO;2-V](https://doi.org/10.1002/(SICI)1097-4571(199709)48:9%3C804::AID-ASI5%3E3.0.CO;2-V)>. Acesso em: 21 jul. 2015.

BUSH, Vannevar. As We May Think. **The Atlantic Monthly**, Boston, v. 176, n. 1, p. 101–108, 1945. Disponível em: <<http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>>. Acesso em: 23 jul. 2015.

CAROTHERS, Gavin (Ed.). **RDF 1.1 N-Quads: A line-based syntax for RDF datasets**. W3C Recommendation 25 February 2014, 25 fev. 2014. Disponível em: <<http://www.w3.org/TR/2014/REC-n-quads-20140225/>>. Acesso em: 04 out. 2017.

CERVO, Amado Luiz; BERVIAN, Pedro Alcino. **Metodologia científica**. 5. ed. São Paulo: Prentice Hall, 2003.

CHAUI, Marilena de Souza. **Convite à Filosofia**. 14. ed. São Paulo: Ed. Ática, 2012. 520p.

CLEVERDON, Cyril W. The Evaluation of Systems Used in Information Retrieval. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SCIENTIFIC INFORMATION, 1958. **Proceedings...**, 1958. p. 687-698. Disponível em: <[doi:10.17226/10866](https://doi.org/10.17226/10866)>. Acesso em: 26 ago. 2016.

COWIE, Jim; LEHNERT, Wendy. Information extraction. **Communications of the ACM**, v. 39, n. 1, p. 80–91, 1996. Disponível em: <[doi: 10.1145/234173.234209](https://doi.org/10.1145/234173.234209)>. Acesso em 01 mai. 2018.

DIJKSTRA, Edsger W. On the cruelty of really teaching computing science dez. 1988. In: **E. W. Dijkstra Archive: the manuscripts of Edsger W. Dijkstra (1930-2002)**, The Center for American History of The University of Texas, Austin. Documento EWD1036:14. Manuscrito. Não publicado. Disponível em: <<http://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF>>. Acesso em: 22 ago. 2016.

DINH, Duy; TAMINE, Lynda. Towards a context sensitive approach to searching information based on domain specific knowledge sources, **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 12–13, p.41-52, 2012. Disponível em: <doi:10.1016/j.websem.2011.11.009>. Acesso em: 08 ago. 2016.

EDMUNDSON, Harold P. New Methods in Automatic Extracting. **Journal of the ACM**, v. 16, n. 2, p. 264-285, 1969. Disponível em: <doi:10.1145/321510.321519>. Acesso em: 26 ago. 2016.

EDMUNDSON, Harold P.; WYLLIS, R. E. Automatic abstracting and indexing: survey and recommendations. **Communications of the ACM**, v. 4, n. 5, p. 226-234, 1961. Disponível em: <doi:10.1145/366532.366545>. Acesso em: 26 ago. 2016.

EFTHIMIADIS, Efthimis N. Query Expansion. In: WILLIAMS, M. E. (ed.). **Annual Review of Information Science and Technology (ARIST)**, v. 31, p. 121-187, 1996. Disponível em: <<http://web.archive.org/web/20110604010433/http://faculty.washington.edu/efthimis/pubs/Pubs/qe-arist/QE-arist.html>>. Acesso em: 18 jun. 2016.

FERNANDEZ, Miriam; CANTADOR, Ivan; LOPEZ, Vanesa; VALLET, David, CASTELLS, Pablo; MOTTA, Enrico. “Semantically enhanced Information Retrieval: An ontology-based approach”. **Journal of Web Semantics**. v. 9, n. 4, p. 434–452, 2011. Disponível em: <doi:10.1016/j.websem.2010.11.003>. Acesso em: 8 ago. 2016.

FERNEDA, Edberto. **Introdução aos Modelos Computacionais de Recuperação de Informação**. Rio de Janeiro: Ed. Ciência Moderna, 2012. 166p. ISBN: 978-85-399-0188-3.

FERNEDA, Edberto. **Ontologia como recurso de padronização terminológica em um Sistema de Recuperação de Informação**. 2013. Relatório (Pós-Doutorado em Ciência da Informação) - Universidade Federal da Paraíba, João Pessoa, 2013.

FUJITA, Mariângela Spotti Lopes. A leitura Documentária na Perspectiva de suas Variáveis: leitor-texto-contexto. **DataGramZero: Revista de Ciência da Informação**, Rio de Janeiro, v. 5, n. 4, ago. 2004. Disponível em: <<http://www.brapci.ufpr.br/brapci/index.php/v/a/7547>>. Acesso em: jun. 2015.

GANDON, Fabien; SCHREIBER, Guus (Ed.). **RDF 1.1 XML Syntax**. W3C Recommendation 25 February 2014. 25 fev. 2014. Disponível em: <<http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/>>. Acesso em: 04 out. 2017.

GIL LEIVA, Isidoro.; RODRÍGUEZ MUÑOZ, José Vicente. Los orígenes del almacenamiento y recuperación de información. **Boletín de la Asociación Andaluza de Bibliotecarios**, Málaga, n. 42, p. 51-66, 1996.

GIL LEIVA, Isidoro; FUJITA, Mariângela Spotti Lopes (Ed.). **Política de indexação**. São Paulo: Cultura Acadêmica; Marília: Oficina Universitária, 2012, 260 p. Disponível em: <https://www.marilia.unesp.br/Home/Publicacoes/politica-de-indexacao_ebook.pdf>. Acesso em: 24 ago. 2016.

GRUBER, Thomas R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v. 5, n. 2, p. 199-220, 1993. Disponível em: <doi:10.1006/knac.1993.1008 >. Acesso em: 08 mar. 2016.

GUARINO, Nicola; MASOLO, Cláudio; VETERE, Guido. OntoSeek: Content-Based Access to the Web. **IEEE Intelligent Systems and their Applications**, v. 14, n. 3, p. 70-80, 1999. Disponível em: <doi:10.1109/5254.769887>. Acesso em: 06 mar. 2018.

GUARINO, Nicola; OBERLE, Daniel; STAAB, Steffen. What is an Ontology? In: STAAB, Steffen; STUDER, Rudi (eds.). **Handbook on Ontologies**. 2ªed. Springer, 2009. ISBN 978-3-540-70999-2.

HAHMA, Gyeong June; YI, Mun Yong; LEEC, Jae Hyun; SUH, Hyo Won. A personalized query expansion approach for engineering document retrieval, **Advanced Engineering Informatics**, v.28, n.4, 2014. Disponível em: <doi:10.1016/j.aei.2014.04.002>. Acesso em: 08 ago. 2016.

HARMAN, Donna. How effective is suffixing? **Journal of the American Society for Information Science**, v. 42, n. 1, p. 7–15, 1991. Disponível em: <doi:10.1002/(SICI)1097-4571(199101)42:1%3C7::AID-ASI2%3E3.0.CO;2-P>. Acesso em: 31 ago. 2017.

HENNING, Boris. Chapter 2: What is Formal Ontology? In: MUNN, Katherine; SMITH, Barry (Org.) **Applied Ontology: An Introduction**. Frankfurt: Ontos Verlag, 2008. p. 39-56. ISBN 978-3-938793-98-5.

HITZLER, Pascal; KRÖTZSCH, Markus; PARSIA, Bijan; PATEL-SCHNEIDER, Peter F.; RUDOLPH, Sebastian. **OWL 2 Web Ontology Language Primer** (Second Edition). 2012. Disponível em: <https://www.w3.org/TR/2012/REC-owl2-primer-20121211/>. Acesso em: 01 jan 2017.

HITZLER, Pascal; KRÖTZSCH, Markus; RUDOLPH, Sebastian. **Foundations of Semantic Web Technologies**. Boca Raton: CRC Press, 2009. 427 p., ISBN 978-1-4200-9050-5.

HORRIDGE, Matthew; PATEL-SCHNEIDER, Peter F. **OWL 2 WEB Ontology Language Manchester Syntax** (Second Edition), 2012. Disponível em: <https://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/>. Acesso em: 04 out. 2017.

JANSEN, Ludger. Chapter 8: Categories The Top-Level Ontology. In: MUNN, Katherine; SMITH, Barry (Org.) **Applied Ontology: An Introduction**. Frankfurt: Ontos Verlag, 2008. p. 173-196. ISBN 978-3-938793-98-5.

JIMENO-YEPES, A.; BERLANGA-LLAVORI, R.; REBHOLZ-SCHUHMANN, D. Ontology refinement for improved information retrieval. **Information Processing and Management**, v. 46, n. 4, p. 426–435, 2010. Disponível em: <doi:10.1016/j.ipm.2009.05.008>. Acesso em: 10 ago. 2016.

KRISTENSEN, Jaana. Expanding end-users' query statements for free text searching with a search-aid thesaurus. **Information Processing and Management**, v.29, n.6, 1993.

LANCASTER, Frederick Wilfrid. **Indexação e Resumos: Teoria e Prática**. Brasília: Briquet de Lemos, 2004. 2ªEd. 452p. Tradução de: Antonio Agenor Briquet de Lemos. Original: *Indexing and Abstracting in theory and practice*. 2003. 3ªEd.

LANCASTER, Frederick Wilfrid. **Information Retrieval Systems: Characteristics, Testing and Evaluation**. New York: John Willey & Sons, 1968. 222p.

LE COADIC, Yves-François. **A Ciência da Informação**. Brasília: Briquet de Lemos, 1996, 115p. Tradução de: Maria Yêda F. S. de Filgueiras Gomes. Original: *La Science de l'informacion*. Paris: Presses Universitaires de France, 1994.

LEITE, Maria Angelica de Andrade. **Modelo Fuzzy para Recuperação de Informação Utilizando Múltiplas Ontologias Relacionadas**. 2009. Tese (Doutorado) - Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação, 2009. 183 p.

LOVINS, Julie Beth. Development of a Stemming Algorithm. **Mechanical Translation and Computational Linguistics**, v. 11, n. 1-2, p. 22-31, mar./jun. 1968. Disponível em: <<http://www.mt-archive.info/MT-1968-Lovins.pdf>>. Acesso em: 30 ago. 2016.

LUHN, Hans Peter. A new method of recording and searching information. **American Documentation**, v. 4, n. 1, p. 14-16, 1953. Disponível em: <[doi:10.1002/asi.5090040104](https://doi.org/10.1002/asi.5090040104)>. Acesso em: 23 ago. 2016.

LUHN, Hans Peter. Key word-in-context index for technical literature (kwic index). **American Documentation**, v. 11, n. 4, p. 288–295, 1960. Disponível em: <[doi:10.1002/asi.5090110403](https://doi.org/10.1002/asi.5090110403)>. Acesso em: 30 ago. 2016.

LUHN, Hans Peter. The Automatic Creation of Literature Abstracts. **IBM Journal of Research and Development**, v. 2, n. 2, p. 159-165, 1958. Disponível em: <[doi:10.1147/rd.22.0159](https://doi.org/10.1147/rd.22.0159)>. Acesso em: 23 ago. 2016.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **Introduction to information retrieval**. New York: Cambridge University Press, 2008. ISBN: 978-0-511-41405-3.

MIZZARO, Stefano. How many relevances in information retrieval? **Interacting with Computers**, v. 10, n. 3, p. 303-320, 1998. Disponível em: <[doi:10.1016/S0953-5438\(98\)00012-5](https://doi.org/10.1016/S0953-5438(98)00012-5)>. Acesso em: 03 nov. 2016.

MOOERS, Calvin N. Descriptors. In: DRAKE, Miriam A. (ed.) **Encyclopedia of Library and Information Science**. 2. ed. New York: Marcel Dekker, 2003. p. 813-821.

MOOERS, Calvin. N. Zatocoding applied to mechanical organization of knowledge. **American Documentation**, v. 2, n. 1, p. 20–32, 1951. Disponível em: <[doi:10.1002/asi.5090020107](https://doi.org/10.1002/asi.5090020107)>. Acesso em: 28 set. 2017.

MOREIRA, Walter. A construção de informações documentárias: aportes da linguística documentária, da terminologia e das ontologias. 2010. 156 f. Tese (Doutorado em Ciência da Informação) – Universidade de São Paulo, Escola de Comunicações e Artes, São Paulo, 2010.

MOTIK, Boris (Ed.); PATEL-SCHNEIDER, Peter F. (Ed.); PARSIA, Bijan. (Ed.). **OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (second edition)**: W3C Recommendation 11 December 2012. Disponível em: <<https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>>. Acesso em: 09 mar. 2017. (A)

MOTIK, Boris (Ed.); PATEL-SCHNEIDER, Peter F. (Ed.); PARSIA, Bijan. (Ed.). **OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax**: W3C Recommendation 27 October 2009. Disponível em: <<https://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>>. Acesso em: 09 mar. 2017.

MOTIK, Boris (Ed.); PATEL-SCHNEIDER, Peter F. (Ed.); PARSIA, Bijan. (Ed.). **OWL2 Web Ontology Language XML Serialization** (second edition): W3C Recommendation 11 December 2012. Disponível em: <<https://www.w3.org/TR/2012/REC-owl2-xml-serialization-20121211/>>. Acesso em: 09 mar. 2017. (B)

NATIONAL INFORMATION STANDARDS ORGANIZATION (NISO). **Understanding Metadata: What is Metadata, and What is it For?: A Primer**. Disponível em: <<https://www.niso.org/publications/understanding-metadata-2017>>. Acesso em: 12 jun. 2018.

ORENGO, Viviane Moreira; HUYCK, C. R. A Stemming Algorithm for the Portuguese Language. In: 8TH INTERNATIONAL SYMPOSIUM ON STRING PROCESSING AND INFORMATION RETRIEVAL (SPIRE), 2001, Laguna de San Raphael, Chile. **Proceedings...** p. 183-193. Disponível em: <[doi:10.1109/SPIRE.2001.10024](https://doi.org/10.1109/SPIRE.2001.10024)>. Acesso em: 14 set. 2017.

OSWALD, Victor A. ; LAWSON, Richard. H. An Idioglossary for mechanical translation. **Modern Language Forum**, v. 38, n. 3-4, p. 1-11, set./dez. 1953. Disponível em: <<http://www.mt-archive.info/50/Oswald-1953.pdf>>. Acesso em: 16 out. 2017.

PATEL-SCHNEIDER, Peter F. (Ed.) ; MOTIK, Boris. (Ed.). **OWL 2 Web Ontology Language Mapping to RDF Graphs** (Second Edition): W3C Recommendation 11 December 2012. Disponível em: <<https://www.w3.org/TR/owl2-mapping-to-rdf/>>. Acesso em 09 mar. 2017.

PAZ-TRILLO, Christian; WASSERMANN, Renata; BRAGA, Paula P. An information retrieval application using ontologies. **Journal of the Brazilian Computer Society**, v. 11, n. 2, p. 17-31, 2005. Disponível em: <<http://www.scielo.br/pdf/jbcos/v11n2/02.pdf>>. Acesso em: 05 mar. 2018.

PORTER, Martin F. An algorithm for suffix stripping. **Program: electronic library and information systems**, v. 14, n. 3, p. 130-137, 1980. Disponível em: <[doi:10.1108/00330330610681286](https://doi.org/10.1108/00330330610681286)>. Acesso em: 16 ago. 2016.

POWERS, David M. W. Applications and Explanations of Zipf's Law. In: NEMLAP3/CONLL '98 PROCEEDINGS OF THE JOINT CONFERENCES ON NEW METHODS IN LANGUAGE PROCESSING AND COMPUTATIONAL NATURAL LANGUAGE LEARNING. **Proceedings...** 1998, p. 151-160. Disponível em: <[doi:10.3115/1603899.1603924](https://doi.org/10.3115/1603899.1603924)>. Acesso em: 31 ago. 2016.

RAMIREZ, Carlos; VALDES, Benjamin. A General Knowledge Representation Model of Concepts. In: GUTIÉRREZ, Carlos Ramírez (ed.) **Advances in Knowledge Representation**. Rijeka: InTech, 2012. Cap. 3, p. 43-76. ISBN 978-953-51-0597-8.

RILOFF, Ellen. Little Words Can Make a Big Difference for Text Classification. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 18th, 9-13 jul. 1995, Seattle, WA, USA. **Proceedings...** New York: ACM, 1995. p. 130-136. Disponível em: <[doi:10.1145/215206.215349](https://doi.org/10.1145/215206.215349)>. Acesso em: 28 nov. 2016.

SALTON, Gerard. **Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer**. Boston: Addison-Wesley, 1989. 530p.

SALTON, Gerard. The past thirty years in information retrieval. **Journal of the American Society for Information Science**, v. 38, n. 5, p. 375-380, 1987. Disponível em: <doi:10.1002/(SICI)1097-4571(198709)38:5<375::AID-ASIS>3.0.CO;2-3>. Acesso em: 21 jul. 2015.

SALTON, Gerard.; WONG, A.; YANG, C. S. A Vector Space Model for Automatic Indexing. **Magazine Communications of the ACM**, v. 18, n. 11, p. 613–620, 1975. Disponível em: <doi:10.1145/361219.361220>. Acesso em: 11 nov. 2015.

SALTON, Gerard; MCGILL, Michael J. **Introduction to Modern Information Retrieval**. New York: Mcgraw-Hill Computer Science Series, 1983. 448p.

SARACEVIC, Tefko. Information science. **Journal of the American Society for Information Science**, v. 50, n. 12, p. 1051–1063, 1999. John Wiley & Sons, Inc. Disponível em: <doi:10.1002/(SICI)1097-4571(1999)50:12<1051::AID-ASIS>3.0.CO;2-Z >. Acesso em: 25 jul. 2015.

SEGURA, Alejandra, SALVADOR-SANCHEZ, Alonso; GARCIA-BARRIOCANAL, Elena; PRIETO, Manuel. “An empirical analysis of ontology-based query expansion for learning resource searches using MERLOT and the Gene ontology”. **Knowledge-Based Systems**. v. 24, n. 1, p. 119–33, 2011. Disponível em: <doi:10.1016/j.knosys.2010.07.012>. Acesso em: 9 ago. 2016.

SILVA, Edna Lúcia da.; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. 4ed., Florianópolis: UFSC, p. 19-23, 2005. Disponível em: <http://soniaa.arq.prof.ufsc.br/roteirosmetodologicos/metpesq.pdf>. Acesso em: 27 out. 2015.

SMITH, Barry. Ontology In: FLORIDI, Luciano (ed.). **The Blackwell Guide to the Philosophy and Information**, 2004, Cap. 11, p. 155-166.

SOLLA PRICE, Derek J. de. **Little Science, Big Science**. New York and London: Columbia University Press, 1963, p. 18-19.

SOWA, John F. **Knowledge Representation: Logical, Philosophical and Computational Foundations**. Pacific Groove: Brooks Cole, 2000, 594p., ISBN 0-534-94965-7.

SPÄRCK JONES, Karen. A statistical interpretation of term specificity and its application in retrieval. **Journal of Documentation**, v. 28, n. 1, p. 11–21, 1972. Disponível em: <doi:10.1108/eb026526>. Acesso em 28 jul. 2017.

SPÄRCK JONES, Karen; WALKER, Steve; ROBERTSON, Stephen E. A probabilistic model of information retrieval: development and comparative experiments Part 2. **Information Processing and Management**, v. 36, n. 6, p. 809-840, 2000. Disponível em: <doi:10.1016/S0306-4573(00)00016-9>. Acesso em: 03 dez. 2016.

SPORNY, Manu; LONGLEY, Dave; KELLOGG, Gregg; LANTHALER, Markus; LINDSTRÖM, Niklas (eds.). **JSON-LD 1.0: A JSON-based Serialization for Linked Data**. W3C Recommendation 16 January 2014. 16 jan. 2014. Disponível em: <https://www.w3.org/TR/2014/REC-json-ld-20140116/>. Acesso em: 04 out. 2017.

TAUBE, Mortimer. Storage and retrieval of information by means of the association of ideas. **American Documentation**, v. 6, n. 1, p. 1-18, 1955. Disponível em: <doi:10.1002/asi.5090060103>. Acesso em: 23 ago. 2016.

TAYLOR, Robert S. The process of asking questions. **American Documentation**, v. 13, n. 4, p. 391–396, 1962. Disponível em: <doi:10.1002/asi.5090130405>. Acesso em: 15 out. 2016.

USCHOLD, Mike. Knowledge level modelling: concepts and terminology. **The Knowledge Engineering Review**, v. 13, n. 1, p. 5-29, 1998. Printed United Kingdom, Cambridge University Press, Disponível em: <doi:10.1017/S0269888998001040>. Acesso em 23 mar. 2018.

USCHOLD, Mike; GRUNINGER, Michael. Ontologies: principles, methods and applications. **The Knowledge Engineering Review**, v. 11, n. 2, p. 93-136, 1996. Disponível em: <doi:10.1017/S0269888900007797>. Acesso em 23 mar. 2018.

WILLIAMS, Robert V. Hans Peter Luhn and Herbert M. Ohlman: Their Roles in the Origins of Keyword-in-Context/permutation Automatic Indexing. **Journal of the American Society for Information Science and Technology**, v. 61, n. 4, p. 835-849, 2010. Disponível em: <doi:10.1002/asi.21265>. Acesso em: 19 mai. 2016

WORLD WIDE WEB CONSORTIUM (W3C). **OWL 2 Web Ontology Language, Document Overview (Second Edition)**, OWL Working Group, W3C Recommendation 11 December 2012, disponível em <<https://www.w3.org/TR/2012/REC-owl2-overview-20121211/>>. Acesso em: 2018.

ZENZ, G.; ZHOU, X.; MINACK, E.; SIBERSKI, W.; NEJDL, W. From keywords to semantic queries: Incremental query construction on the semantic web. **Journal of Web Semantics**. v. 7, n. 3, p. 166–176, 2009.

ZHAI, ChengXiang. **Statistical Language Models for Information Retrieval**. Williston: Morgan & Claypool, 2009. 125p. ISBN: 9781598295917.

ZIPF, George Kingsley. **Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology**. Cambridge: Addison-Wesley Press, 1949. 573 p.