UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ARTHUR MORAIS DE ANDRADE

DESCOBERTA DE RELACIONAMENTOS SEMÂNTICOS NÃO TAXONÔMICOS ENTRE TERMOS ONTOLÓGICOS

São Carlos-SP
Junho de 2017

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESCOBERTA DE RELACIONAMENTOS SEMÂNTICOS NÃO TAXONÔMICOS ENTRE TERMOS ONTOLÓGICOS

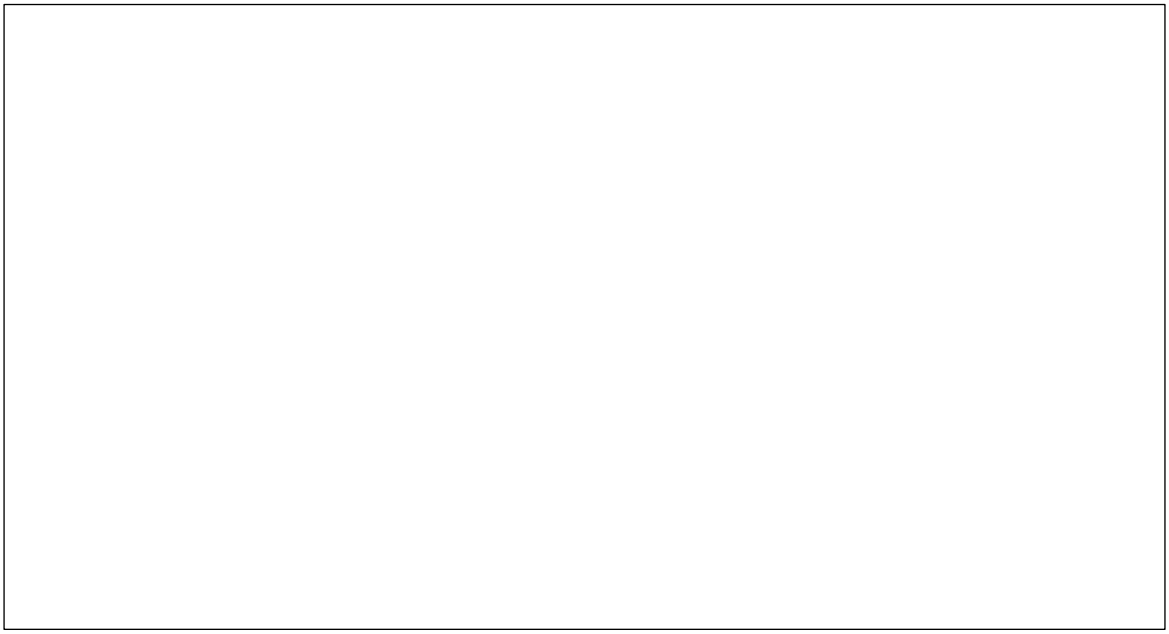
Dissertação de Mestrado elaborada por Arthur Morais de Andrade e apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Banco de Dados.

Orientadora: Profª Dra. Marilde Terezinha Prado Santos

São Carlos-SP
Junho de 2017

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária (BCo)/UFSCar.



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de Dissertação de Mestrado do candidato Arthur Morais de Andrade, realizada em 14/02/2017.

Profª. Drª. Marilde Terezinha Prado Santos
(UFSCar)

Profª. Drª. Helena de Medeiros Caseli
(UFSCar)

****

Profª. Drª. Sílvia Maria Wanderley Moraes
(PUC)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Sílvia Maria Wanderley Moraes, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Arthur Morais de Andrade.

Profª. Drª. Marilde Terezinha Prado Santos
Presidente da Comissão Examinadora
(UFSCar)

Aos meus pais.

AGRADECIMENTOS

Agradeço à minha família pelo apoio constante e por sempre acreditarem em mim, em especial aos meus pais e irmão.

À minha orientadora, Marilde, por sua amizade e carinho, e pela sua dedicação e paciência durante minha jornada na Universidade Federal de São Carlos (UFSCar).

A todos os professores do Departamento de Computação (DC) da UFSCar que, de certa forma, se dedicam para manter o departamento funcional.

Aos amigos que conheci ao ingressar no Mestrado: Claudineia, Pablo, Rafael, João, Luiz, Steve, Bento, Ana, Francielle, Renata, Carol, Fernando, Diego, Kathiani, Alessandro, Guido, entre outros.

Aos integrantes do Laboratório de Currículo Funcional (LCF), que me apoiaram no processo de validação da minha proposta.

À Capes, pelo apoio financeiro, e ao DC da UFSCar, pela estrutura durante todo o período vigente do Mestrado.

"Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça o que a maioria não faz."

(Bill Gates)

RESUMO

Ontologias têm se tornado um importante instrumento para a estruturação do conhecimento. Porém, a construção de uma ontologia envolve um cuidadoso processo de definição de termos representativos do domínio e seus relacionamentos, exigindo muito tempo dos engenheiros de ontologias em conjunto com especialistas de domínio. Esses relacionamentos podem ser taxonômicos (hiponímia e meronímia), representando uma taxonomia de conceitos, e não taxonômicos, referentes aos demais relacionamentos que ocorrem entre os nós dessa taxonomia. As principais dificuldades estão relacionadas ao tempo gasto pelos especialistas de domínio e às garantias necessárias para a qualidade das ontologias criadas, tornando-as confiáveis. Neste sentido, são bem-vindos os esforços para a elaboração de abordagens que visam diminuir o tempo de dedicação do especialista sem redução de qualidade da ontologia criada. Neste trabalho foi desenvolvida uma abordagem para a descoberta de relações semânticas não taxonômicas entre termos ontológicos, a partir de documentos semiestruturados redigidos com vocábulos informais do Português variante brasileira. A abordagem visa auxiliar engenheiros de ontologias e especialistas de domínio na árdua tarefa de descoberta dos relacionamentos entre termos ontológicos. Após a descoberta dos relacionamentos semânticos, estes foram convertidos em uma estrutura conceitual, gerada pelo método *Formal Concept Analysis* (FCA). Essa abordagem foi avaliada em dois experimentos, com auxílio de especialistas de domínio em Educação Especial. O primeiro experimento consistiu em uma comparação entre os relacionamentos extraídos de forma manual e a extração automática, apresentando um bom valor de precisão, cobertura e medida F, obtendo, respectivamente, 92%, 95% e 93%. Já o segundo experimento consistiu em avaliar os relacionamentos extraídos automaticamente na estrutura gerada pelo FCA, obtendo precisão média 86,5%. Esses resultados indicam a eficácia da abordagem de descoberta de relacionamentos semânticos.

Palavras-chave: Ontologia; Extração de Informação; Processamento da Linguagem Natural.

ABSTRACT

Ontologies have become an important tool to structure knowledge. However, the construction of an ontology involves a careful process of defining representative terms of the domain and its relationships, which requires a lot of time from ontology engineers and domain experts. These relationships can be taxonomic (hyponymy and meronymy), representing a taxonomy of concepts, and non-taxonomic, referring to the other relationships that occur between the nodes of this taxonomy. The main difficulties of constructing an ontology are related to the time spent by domain specialists and the necessity of guaranteeing the quality and reliability of the ontologies create. In this way, we are welcome the efforts to elaborate approaches that aim to reduce the amount of time dedicated by specialists without reducing the quality of the ontology created. In this master's project, an approach was developed for the discovery of semantic relationships between non-taxonomic ontological terms from semi-structured documents written with informal vocabularies of the Brazilian Portuguese language. Thus, it aids ontology engineers and domain experts in the arduous task of discovering the relationships between ontological terms. After the discovery of semantic relationships, the relationships were converted into a conceptual structure, generated by the Formal Concept Analysis (FCA) method. This approach was validated in two experiments, with the help of domain experts in special education. The first experiment consisted of a comparison between manually extracted relationships and automatic extraction, presenting a good value of precision, coverage and measurement F, respectively, 92%, 95% and 93%. The second experiment evaluated the relationships extracted, automatically, in the structure generated by the FCA, it gets average accuracy 86,5%.These results prove the effectiveness of the semantic relationship discovery approach.

Keywords: Ontology, Information Extraction, Natural Language Processing.

LISTA DE FIGURAS

Figura 1 – Exemplo de ontologia 26
Figura 2 – Metodologia 101 28
Figura 3 – Metodologia *Methontology* 30
Figura 4 – Exemplo de autômato 42
Figura 5 – Representação de um reticulado de conceitos 46
Figura 6 – FCA alterado 46
Figura 7 – Etapas do modelo proposto por Scheicher (2013) 49
Figura 8 – Extração de relacionamentos semânticos 50
Figura 9 – Processo proposto para extração de tratamentos 51
Figura 10 – Exemplo de regra da Estratégia 1 53
Figura 11 – Exemplo de regra da Estratégia 2 53
Figura 12 – Relações semânticas de interesse 54
Figura 13 – Sentenças anotadas com relações semânticas entre os termos 55
Figura 14 – Padrões definidos para as seis relações semânticas 56
Figura 15 – Padrões definidos a partir de uma iteração do algoritmo de Hearst 56
Figura 16 – Desempenho em medida F para três experimentos 57
Figura 17 – Descrição do *framework* de enriquecimento 58
Figura 18 – Arquitetura do GRAONTO 60
Figura 19 – Exemplo de texto 68
Figura 20 – Um extrato da ontologia base 69
Figura 21 – Arquitetura $Embed_{NT}$RelOnto 70
Figura 22 – Pré-processamento 72
Figura 23 – Processamento 77
Figura 24 – Definição do autômato determinístico 78
Figura 25 – Hierarquia da taxonomia 92
Figura 26 – Ponderação do relacionamento extraído 93
Figura 27 – Diagrama do banco de dados 94
Figura 28 – Pós-processamento 95
Figura 29 – Seleção de relacionamentos no buscador 96

Figura 30 – Busca ponderada 97
Figura 31 – Representação do um 99
Figura 32 – Inserir relacionamento 100
Figura 33 – Relacionamento incorporado na ontologia 101
Figura 34 – Relacionamentos inseridos 108
Figura 35 – Gráfico da Etapa 1, Especialista 1 110
Figura 36 – Gráfico da Etapa 1, Especialista 2 110
Figura 37 – Comparação da primeira etapa 111
Figura 38 – Gráfico da Etapa 2, Especialista 1 112
Figura 39 – Gráfico da Etapa 2, Especialista 2 112
Figura 40 – Gráfico da comparação entre especialistas, Etapa 2 113

LISTA DE ALGORITMOS

Algoritmo 1 – SemanticExtr 79
Algoritmo 2 – extrairRelacionamento 81
Algoritmo 3 – verificarContadores 82
Algoritmo 4 – processaEstadoQ0 83
Algoritmo 5 – isTermoOntologico 84
Algoritmo 6 – processaEstadoQ1 85
Algoritmo 7 – processaEstadoQ2 86
Algoritmo 8 – processaEstadoQ3 87
Algoritmo 9 – processaEstadoQ4 88
Algoritmo 10 – processaEstadoQ5 89
Algoritmo 11 – Produto cartesiano 90
Algoritmo 12 – Marcação de sentença 98
Algoritmo 13 – isTermoOntologico alterado 130

LISTA DE TABELAS

Tabela 1 – δ em forma de tabela.....41
Tabela 2 – Contexto formal definido a partir dos relacionamentos.....43
Tabela 3 – Tabela-resumo dos trabalhos relacionados.....66
Tabela 4 – Descrição do fluxo do autômato.....79
Tabela 5 – Dados dos documentos.....105
Tabela 6 – Tempo gasto e grau de cansaço por especialista.....106
Tabela 7 – Resultado extração automática.....107
Tabela 8 – Precisão, experimento 2.....113
Tabela 9 – Tabela de tempo no experimento 2.....114

LISTA DE ABREVIATURAS E SIGLAS

AM – Aprendizado de Máquina

API – *Application Programming Interface*

ARS – Anotador de Relações Semânticas

CV – *Child Voting*

EI – Extração de Informação

FCA – Formal Concept Analysis

FP – Falsos Positivos

FN – Falsos negativos

JSON – *JavaScript Object Notation*

HAC – *Hierarchical Agglomerative Clustering*

LCF – Laboratório de Currículo Funcional

MD – Mineração de Dados

MT – Mineração de Texto

NEEs – Necessidades Educacionais Especiais

OMCS – *Open Mind Common Sense*

ONEESP – Observatório Nacional da Educação Especial

PCFG – Gramáticas Livres de Contexto Probabilístico

PLN – Processamento de Linguagem Natural

POS – *Part-Of-Speech*

RDF – *Resource Description Framework*

RDFS – *RDF Schema*

SRM – Sala de Recursos Multifuncionais

TFICF – *Term Frequency and Inverse Cluster Frequency*

UFSCar – Universidade Federal de São Carlos

UML – *Unified Modeling Language*

VP – Verdadeiros Positivos

XML – *eXtensible Markup Language*

SUMÁRIO

CAPÍTULO 1 • INTRODUÇÃO 18

1.1 Considerações iniciais 18
1.2 Contexto e motivação 19
1.3 Objetivos 21
1.4 Hipóteses 22
1.5 Organização do trabalho 22

CAPÍTULO 2 • REFERENCIAL TEÓRICO 23

2.1 Introdução 23
2.2 Ontologia 24
2.2.1 Definição formal da Ontologia 25
2.2.2 Linguagens de representação de ontologias 27
2.2.3 Metodologias de construção de ontologias 27
2.2.4 Tecnologias para a construção de ontologias 31
2.3 Mineração de Texto 32
2.3.1 Processo de Mineração de Texto 33
2.3.1.1 Coleta de documentos 33
2.3.1.2 Pré-processamento 33
2.3.1.3 Extração de padrões 35
2.3.1.4 Métricas para a Avaliação dos Resultados 36
2.4 Extração de Informação 37
2.4.1 Abordagem baseada em Aprendizado de Máquina 38
2.4.2 Abordagem baseada em Dicionário 38
2.4.3 Abordagem baseada em Regras 39
2.5 Autômato finito determinístico 40
2.6 *Formal Concept Analysis* 42
2.7 Considerações finais 47

CAPÍTULO 3 • TRABALHOS CORRELATOS 48

3.1 Considerações iniciais 48
3.2 Extração de relacionamento semântico 48

3.2.1 Um método para descoberta de relacionamentos semânticos do tipo "causa e efeito" em sentenças de artigos científicos do domínio biomédico 49
3.2.2 Um processo baseado em parágrafos para a extração de tratamentos de artigos científicos do domínio biomédico 51
3.2.3 Extração automática de relações semânticas a partir de textos escritos em Português variante brasileira 53
3.2.3.1 Recursos 54
3.2.3.2 Experimentos com Padrões Textuais 55
3.2.3.3 Experimentos com Aprendizado de Máquina 56
3.3 Construção de ontologias por meio de documentos textuais 57
3.3.1 Extração de relacionamentos semânticos para o enriquecimento de ontologia de domínio.....58
3.3.2 GRAONTO: uma abordagem baseada em grafos para a construção de ontologia de domínio.....59
3.3.3 Uma abordagem automática para a construção de ontologias expressivas a partir de linguagem natural 62
3.3.4 Construção de estruturas ontológicas a partir de textos: um estudo baseado no método *Formal Concept Analysis* e em papéis semânticos 63
3.4 Considerações finais 65

CAPÍTULO 4 • ABORDAGEM SEMÂNTICA PARA INCORPORAÇÃO DE RELACIONAMENTOS NÃO TAXONÔMICOS EM ONTOLOGIAS 67

4.1 Definições.....67
4.2 Recursos 68
4.3 Abordagem $Embed_{NT}$RelOnto 70
4.4 Pré-processamento 71
4.4.1 Limpeza 72
4.4.2 Sentenciação 73
4.4.3 Tokenização.....73
4.4.4 Etiquetação 73
4.4.5 Lematização.....74
4.4.6 Identificação de termos sinônimos.....74
4.4.7 Identificação de termos ontológicos.....75
4.4.8 Estrutura criada para conter as informações processadas 76

4.5 Método de extração de relacionamentos semânticos entre termos ontológicos .76
4.5.1 Filtro de sentenças que contenham termos ontológicos 77
4.5.2 Método SemanticExtr.....78
4.5.3 Ponderação da relação.....91
4.5.4 Salvar os relacionamentos extraídos no banco de dados.....94
4.6 Pós-processamento 95
4.6.1 Extração de relacionamentos válidos 96
4.6.1.1 Destaque dos relacionamentos semânticos na sentença.....97
4.6.2 Representação gráfica dos relacionamentos.....98
4.6.3 Enriquecimento da ontologia.....100
4.7 Otimização do SemanticExtr 101
4.8 Considerações finais 102
CAPÍTULO 5 • VALIDAÇÃO DA PROPOSTA 104
5.1 Considerações iniciais.....104
5.2 Experimento 1 – Avaliação automática do SemanticExtr 104
5.3 Experimento 2 - Identificação de relacionamentos com o auxílio do FCA.....105
5.4 Experimento de performance 114
5.5 Discussão sobre os resultados.....115
5.6 Validação das hipóteses.....117
5.7 Considerações finais 117
CAPÍTULO 6 • CONCLUSÕES E TRABALHOS FUTUROS 119
6.1 Síntese dos resultados e contribuições.....120
6.2 Dificuldades encontradas 121
6.3 Limitações da abordagem 121
6.4 Trabalhos futuros 122
REFERÊNCIAS.....124
APÊNDICE A – Tratamento de termos rotulados errados.....130
APÊNDICE B – Termos ontológicos selecionados para o experimento 1.....131
APÊNDICE C – FCA 132
APÊNDICE D – Tabela: experimento 2.....133

CAPÍTULO 1

Introdução

Este Capítulo apresenta o contexto em que o presente trabalho está inserido, bem como a motivação que deu origem a esta pesquisa de Mestrado. São discutidos em seguida os objetivos e, por fim, é descrita a organização desta dissertação.

1.1 Considerações iniciais

O aumento expressivo na quantidade de informações produzidas pela sociedade tem sido cada vez mais discutido na academia e nas empresas. Isso se deve ao avanço das Tecnologias de Informação e Comunicação (TIC) que, consequentemente, gera maior volume de informações, facilitando o acesso por parte dos indivíduos.

Neste campo de pesquisa, a web semântica aponta um conjunto de tecnologias capazes de atribuir informações semânticas, as quais possuem potencial de representar o conhecimento. Neste sentido, os modelos conceituais são estruturas para conceber este conhecimento; as estruturas conceituais são artefatos produzidos com o objetivo de representar uma dada porção da realidade, segundo um determinado conceito (GUIZZARDI, 2000). Assim, as ontologias despontam como estruturas consistentes de representação.

Ontologias possuem uma grande a aplicabilidade mas, em contrapartida, apresentam um alto o custo de construção e manutenção. Por esta razão – e devido ao grande volume de informações e de documentos textuais digitais disponíveis atualmente –, muitas pesquisas têm sido realizadas com o objetivo de criar e

enriquecer ontologias a partir de textos. Exemplo disso é o caso do método *Formal Concept Analysis* (FCA),[1] que foi apresentado nos anos 80 para análise de dados (WILLE, 1997) e que vem sendo aplicado na construção ontologias (CIMIANO, 2006; CIMIANO; HOTHO; STAAB, 2005).

Segundo Higuchi (2012), a interpretação automática de textos ainda é um desafio que vem sendo enfrentado para a construção ou o enriquecimento de ontologias. Na área de Processamento de Linguagem Natural (PLN) o desafio está em fornecer aos computadores a capacidade de "entender" textos escritos em linguagem humana com o intuito não só de identificar termos[2] relevantes para a ontologia, mas também de identificar relacionamentos semânticos.

No presente trabalho de pesquisa, o foco está no processamento automático de texto para a extração de relacionamentos semânticos não taxonômicos entre termos ontológicos, para o enriquecimento de ontologias. Nesse contexto, termos ontológicos são representados por um termo pertencente às classes de uma ontologia preliminar. O relacionamento não taxonômico[3] refere-se aos relacionamentos que não contenham as relações de hiponímia (relação *is-a* ou é-um) ou meronímia (relação *part-of* ou parte-de). Por exemplo, a partir da sentença "Os alunos fazem prova", considerando que "aluno" e "prova" são termos ontológicos, é possível extrair o relacionamento não taxonômico (aluno-fazer-prova). Diante dos relacionamentos extraídos, uma estrutura conceitual foi gerada por meio do FCA, com o intuito de apoiar especialistas na identificação dos relacionamentos válidos.

1.2 Contexto e motivação

Nos cursos de graduação e de pós-graduação em Educação Especial da Universidade Federal de São Carlos (UFSCar), existem pesquisadores e

[1] O método FCA será apresentado e discutido em profundidade na seção 2.6 do Capítulo 2.
[2] Entende-se por *termo* uma unidade atômica, a qual é representada por uma entidade (palavra) ou que tem algum significado em uma sentença.
[3] Relacionamento taxonômico é uma relação de hierarquia na qual um elemento (por exemplo, o hipônimo "cão") pode ser a subclasse de um elemento mais geral (por exemplo, o hiperônimo "animal"), obtendo-se assim uma relação hiponímia "*is-a*/é-um (cão, animal)". Ou, por ser um elemento homônimo "cão", ele pode ser formado por outros elementos merônimos, como "cauda", formando assim uma relação meronímia "*part-of*/parte de (cauda, cão)".

pesquisadoras pertencentes ao Laboratório de Currículo Funcional, que fazem parte do projeto Observatório Nacional da Educação Especial (ONEESP). O foco do ONEESP é a produção de estudos integrados sobre políticas e práticas voltadas para a questão da inclusão escolar na educação brasileira. Este projeto envolve profissionais de diversas regiões do Brasil, que coletaram dados de suas regiões referentes às políticas públicas de inclusão de pessoas com deficiências físicas e cognitivas.

O projeto ONEESP se concentrou em um estudo nacional sobre políticas governamentais para estimular a criação de serviços de apoio aos alunos com Necessidades Educacionais Especiais (NEEs) nas escolas regulares, considerados no âmbito das denominadas Salas de Recursos Multifuncionais (SRM) (MENDES; CIA, 2015).

Os dados do ONEESP foram organizados em um banco de dados central, mantido na UFSCar, composto de diversos documentos relacionados à avaliação de políticas públicas para a Educação Especial. A análise preliminar destes dados indicou que as políticas públicas dos municípios, apesar de seguirem algumas diretrizes do Ministério da Educação, foram traduzidas para a realidade e assumiram uma postura diferente, dependendo em parte da história local do desenvolvimento de serviços de Educação Especial em determinada realidade (MENDES; CIA, 2015).

Diante da grande quantidade de documentos gerados pelas entrevistas focais e com o apoio dos especialistas alocados no LCF, foi possível realizar a extração de conhecimento dos documentos. A extração de conhecimento foi iniciada por Fernandes (2016), que realizou a construção de uma ontologia preliminar refletindo o domínio da política pública inclusiva brasileira. No entanto, observou-se que apesar do tempo dedicado às discussões entre os especialistas, foram identificadas poucas relações entre os termos ontológicos.

A necessidade de aumentar o poder de inferência na ontologia preliminar do domínio da política pública inclusiva brasileira e o fato de o LCF ser detentor de um grande conjunto de documentos coletados no âmbito do ONEESP constituíram a motivação que impulsionou, por sua vez, uma pesquisa por abordagens, a qual se baseia na detecção dos relacionamentos semânticos não taxonômicos entre os termos ontológicos a partir de documentos caracterizados como semiestruturados, formados por vocábulos informais em Português variante brasileira.

A pesquisa realizada apontou para algumas iniciativas na área de Engenharia de Ontologias, porém voltadas para o inglês.[4] Trabalhos realizados para a busca de relacionamentos semânticos em textos redigidos em Português estão normalmente associados à área de PLN – que não se relaciona com a Engenharia de Ontologias – e/ou estão restritos a documentos bem estruturados, cuja estrutura auxilia mais na percepção semântica de seu conteúdo.

Portanto, a motivação para a realização da presente dissertação de Mestrado se baseia na percepção de que o processo de construção de ontologias pode ser beneficiado tanto no quesito *tempo* quanto no quesito *confiabilidade*, contando com uma abordagem para detecção de relacionamentos semânticos entre os termos ontológicos a partir da análise em documentos semiestruturados e em Português variante brasileira, considerando vocábulos formais e informais.

1.3 Objetivos

O objetivo principal do presente trabalho é propor um método computacional para solucionar as seguintes necessidades:

- Identificar termos ontológicos nos textos em análise;
- Realizar a extração de relacionamentos não taxonômicos entre termos ontológicos a partir de textos informais escritos em Português variante brasileira, fornecidos pelo projeto ONEESP;
- Gerar uma estrutura conceitual por meio do método FCA que auxilie os especialistas de domínio na validação dos relacionamentos semânticos extraídos.

[4] Estas iniciativas serão apresentadas no Capítulo 3.

1.4 Hipóteses

A partir do objetivo identificado anteriormente, foram levantadas hipóteses a respeito da extração de relacionamentos semânticos não taxonômicos entre termos ontológicos e do uso do FCA para representação dos relacionamentos extraídos.

As hipóteses propostas, portanto, são as seguintes:

- É possível realizar a extração de relacionamentos semânticos não taxonômicos entre termos ontológicos a partir de textos informais no Português variante brasileira;
- O método FCA auxilia os especialistas de domínio na identificação dos relacionamentos semânticos.

1.5 Organização do trabalho

O presente trabalho está organizado da seguinte forma: no Capítulo 2 é apresentado o referencial teórico que constitui base para o entendimento da proposta. No Capítulo 3, são discutidos os trabalhos correlatos à abordagem proposta e no Capítulo 4 é apresentada a proposta de fato. O Capítulo 5 apresenta a validação da proposta e, por fim, o Capítulo 6 encerra as conclusões e trabalhos futuros a partir da abordagem sugerida.

CAPÍTULO 2

Referencial teórico

Neste Capítulo serão contextualizados os principais conceitos encontrados na literatura da área, utilizados para compreender o processo de extração de relacionamentos semânticos entre termos ontológicos não taxonômicos em textos informais escritos em Português variante brasileira. Uma definição de ontologias, conceitos sobre Mineração de Textos e fundamentos usados em Mineração de Textos relacionados ao Processamento de Linguagem Natural (PNL) e Extração de Informação também serão apresentados.

2.1 Introdução

O mundo está repleto de dados que são gerados toda vez que alguém utiliza seu celular ou navega na internet, por exemplo. Estes dados são passíveis de serem analisados, processados e transformados em informações de valor. Apesar do grande volume de informações, uma pequena parte deles é utilizada para solucionar problemas do cotidiano (BREITMAN, 2005).

Como forma de organizar a informação surgiu, em 2001, o conceito de *Web Semântica*, em que se busca utilizar recursos provenientes, dentre outras áreas, da Inteligência Artificial, da Engenharia de Software e da Computação Distribuída para executar atividades na Web que antes só eram possíveis graças a agentes humanos.

A concepção e uso das ontologias fazem parte da proposta da Web Semântica. Elas estão presentes na arquitetura proposta por Tim Berners-Lee[5] e têm configurado uma das tecnologias-chave na criação de aplicativos mais adequados para lidar com grandes quantidades de informações de maneira inteligente (HORROCKS et al., 2007).

Ontologia é uma abordagem para organização de informações que vem recebendo atenção especial nos últimos anos, principalmente no que diz respeito à representação formal de conhecimento (GUARINO, 1995). A ontologias geralmente são desenvolvidas por especialistas, e sua estrutura é baseada na descrição de conceitos e dos relacionamentos semânticos estabelecidos entre eles. A popularização das ontologias é devido a sua promessa de compartilhamento e entendimento comum de algum domínio de conhecimento que possa ser comunicado entre pessoas e computadores.

Neste sentido, elas têm sido desenvolvidas para facilitar o compartilhamento e reutilização de informações, possibilitando não só o reuso do domínio de conhecimento, mas também a sua análise, tornando explícitas hipóteses sobre este domínio (MORAIS; AMBRÓSIO, 2007, p. 21).

2.2 Ontologia

Gruber (2009) afirma que, para a computação, uma ontologia define primitivas de representação, com as quais se modelam um domínio de conhecimento. Essas primitivas são as classes, os atributos e as relações entre as classes. As classes podem ser subdivididas em superclasses e subclasses, sendo organizadas em uma hierarquia taxonômica. Cada classe possui propriedades que descrevem suas características e restrições (NOY; MCGUINNESS, 2001). Guarino (1998), por sua vez, considera que uma ontologia é um conjunto de axiomas lógicos designados para explicar o significado de um vocábulo.

Cimiano et al. (2005), Maedche et al. (2002) e Sumida et al. (2006) citam métodos que apoiam um engenheiro de ontologias na criação e manutenção de

[5] Tim Berners-Lee é engenheiro e cientista da computação britânico, criador da *World Wide Web* e precursor da Web Semântica.

ontologias. Os métodos se concentram em estruturas taxonômicas, pois a representação do conhecimento se concentra na classificação da hierarquia. Na hierarquia, os termos superiores são mais gerais, enquanto que os termos inferiores são mais particulares. Os relacionamentos taxonômicos (é-um) capturam parcialmente o conhecimento relevante, enquanto que relacionamentos semânticos entre termos não taxonômicos fornecem mais associações de domínio específico (SHEN et al., 2012).

2.2.1 Definição formal da ontologia

Maedche e Staab (2000) definem formalmente a ontologia em uma tupla: O = (*C,H,I,R,P,A*), onde:

C = $C_C \cup C_I$ é um conjunto de entidades de uma ontologia. O Conjunto C_C consiste nas classes (como por exemplo, “Pai” $\in C_C$) enquanto o conjunto C_i é constituído de instâncias (como por exemplo, “John” $\in C_I$).

H = {é_um (c_1, c_2) | $c_1 \in C_C$, $c_2 \in C_C$} é um conjunto de relações taxonômicas entre conceitos, o qual define o conceito de hierarquia e é denotado por “é_um (c_1, c_2)”, significando que c1 é uma subclasse c2 (como por exemplo “é_um (Pai, Progenitor)”).

I = {é_um (c_1, c_2) $c_1 \in C_I \wedge c_2 \in C_C$} $\cup$ {$prop_K$ (c_i, valor) | $c_i \in C_I$} $\cup$ {rel_K (c_1, c_2, ..., c_n) | $\forall i$, $c_i \in C_I$} é um conjunto de relacionamento entre elementos da ontologia e suas instâncias (como por exemplo “é_um(John, Pai)” ou “pai_de(John, Jane)”).

R = {rel_k (c_1, c_2, ..., c_n) | $\forall i$, $c_i \in C_C$} é um conjunto de relações da ontologia que não são nem “tipo_de” nem “é_um”. Por exemplo, pai_de(Progenitor, Filho).

P = {$prop_K$ (c_i, tipo_de_dado) | $c_i \in C_C$} é um conjunto de propriedades das classes da ontologia. Por exemplo, CPF(Pessoa, xxx.xxx.xxx-xx).

A = {condição$_x$ $\Rightarrow$ *conclusão*$_y$ (c_1, c_2, ..., c_n) | $\forall j$, $c_j \in C_C$} é um conjunto de axiomas, regras que permitem verificar a consistência de uma ontologia e inferir novos conhecimentos através de algum mecanismo de inferência. O termo condição$_x$ é dado por condição$_x$ = {($cond_1$, $cond_2$, ..., $cond_n$) | $\forall z$, $cond_z \in H \cup I \cup R$}. Por exemplo, ($\forall$ Pai, Filho1, Filho2, pai_de(Pai, Filho1), pai_de(Pai, Filho2) $\Rightarrow$ irmão_de(Filho1, Filho2) é uma regra que indica que dois filhos possuem o mesmo Pai; logo, os filhos são irmãos.

Diante da definição formal da ontologia, é possível fazer a associação de seus conceitos com algumas apreciações consideradas nesse texto. Por exemplo, os termos ontológicos e relacionamentos não taxonômicos são representados, respectivamente, por C e R.

Os conceitos apresentados por Maedche e Staab (2000) são abordados na Figura 1. Por meio dela, é possível entender o conceito de *axioma*. Axiomas são regras pertinentes ao domínio em questão, as quais expressam sempre verdade. Por exemplo, se definirmos que “pessoa” é subclasse de “homem”, pode-se concluir que uma determinada instância de homem é também uma instância de pessoa. Ainda de acordo com a Figura 1, é possível definir uma classe por meio de um axioma. Por exemplo: um homem é definido como pessoa não mulher.

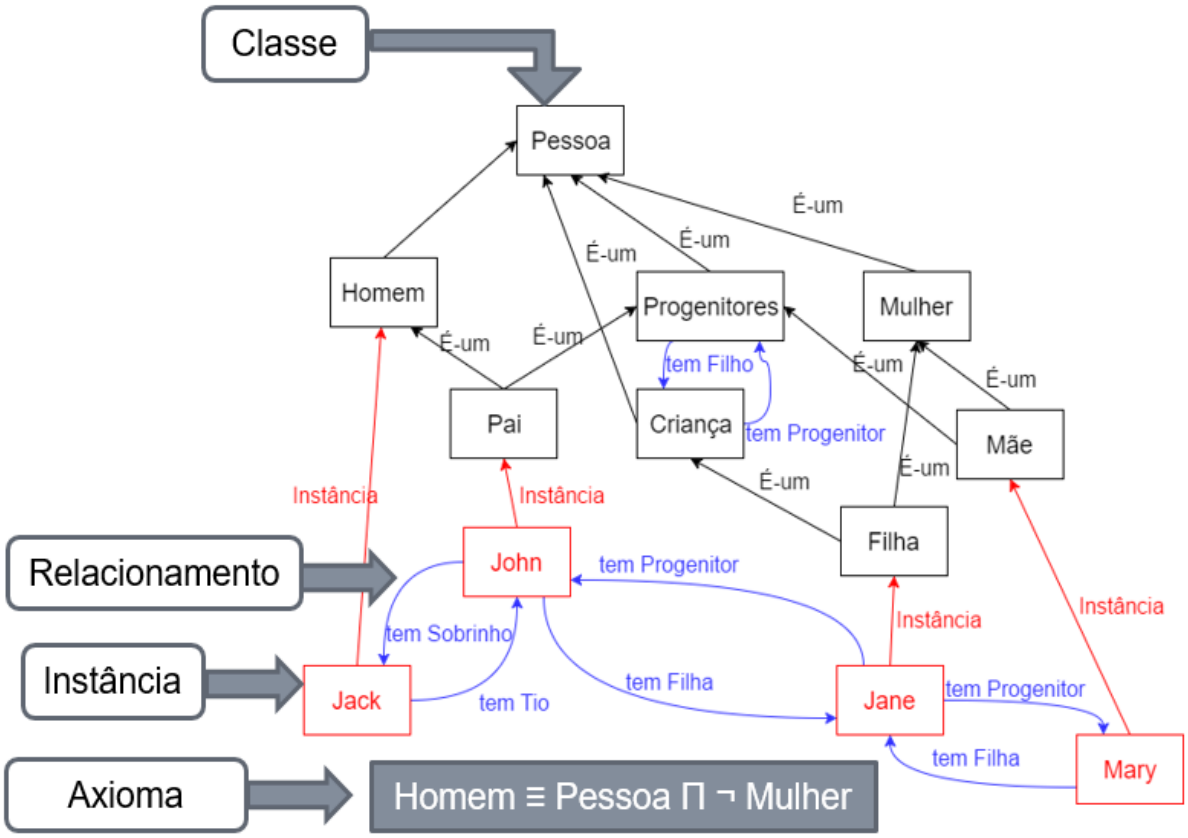


Figura 1 Exemplo de ontologia.
Fonte: elaboração própria.

2.2.2 Linguagens de representação de ontologias

As ontologias são representadas por algum tipo de linguagem. A mais utilizada é a *Ontology Web Language* (linguagem OWL). Ela é a linguagem padrão para a representação de ontologias, além de ser recomendada para o uso pela W3C (LIMA; CARVALHO, 2005, p. 22). A OWL foi criada para aplicações que precisam processar o conteúdo da informação ao invés de apenas apresentá-la aos humanos.

A OWL é subdividida em três linguagens: *OWL Lite*, *OWL DL* e *OWL Full*. De fato, a *OWL Lite* é um subconjunto da *OWL DL*, que é subconjunto da *OWL Full*. As linguagens OWL são classificadas conforme sua expressividade (SMITH et al., 2004).

- *OWL Lite*: é a menos expressiva. É uma sublinguagem sintaticamente simples e é específica para necessidades básicas dos usuários. Destina-se a situações em que apenas são necessárias restrições e uma hierarquia de classificação;
- *OWL DL*: é mais expressiva que a *OWL Lite* A sigla DL possui correspondência com a lógica descritiva (*Description Logics*), passível de raciocínio automático;
- *OWL Full*: a*OWL Full* é a sublinguagem mais expressiva. Destina-se a situações nas quais a alta expressividade é mais importante para garantir a decidibilidade ou completude da linguagem.

No contexto do presente trabalho, a ontologia criada por Fernandes (2016) é representada pela linguagem *OWL Lite*.

2.2.3 Metodologias de construção de ontologias

A construção de ontologias é um processo demorado e trabalhoso. Logo, são necessárias metodologias que auxiliem o processo de construção. Nesta direção, as existem metodologias de desenvolvimento de ontologias cujo intuito é sistematizar sua construção e manipulação.

São apresentadas a seguir três metodologias que apoiam a construção de ontologias.

Metodologia 101

Noy e McGuinness (2001) sugerem um processo para a construção de ontologias, denominado *Ontology Development 101*. O processo consiste em um guia de passos interativos, os quais são executados livremente no desenvolvimento de ontologias. A parte (a) da Figura 2 ilustra os sete passos sugeridos pelos pesquisadores, e a parte (b) apresenta um exemplo de como os passos podem ser empregados durante o desenvolvimento de uma ontologia.

Na primeira etapa, "Determinar o escopo da ontologia", deve-se identificar o propósito e os cenários de utilização da ontologia a ser desenvolvida. Após a delimitação do escopo, aconselha-se verificar a existência de ontologias que podem ser reutilizadas em um novo projeto de ontologia. Há bibliotecas de ontologias reutilizáveis na internet e na literatura. Por exemplo, podemos usar a *Ontolingua*[6] ou a DAML.[7]

Na etapa de enumeração de termos importantes do domínio da ontologia, devem ser listados todos os termos. É importante obter uma lista abrangente de termos, sem se preocupar com a sobreposição dos conceitos que representam.

Na etapa de definição das propriedades das classes, a partir da lista de termos, devem ser observados se os termos correspondem a propriedades de dados ou de relações de classe para uma determinada classe.

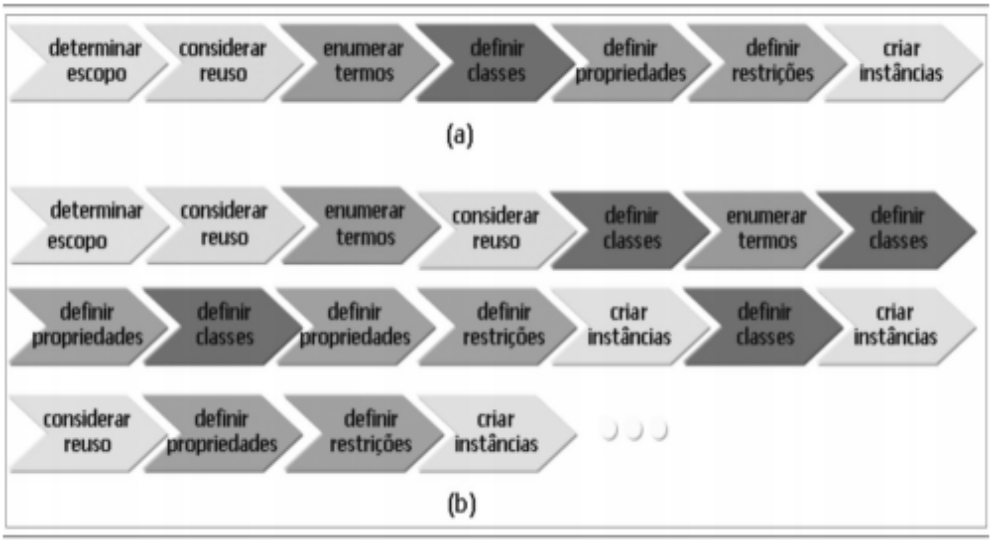


Figura 2 Metodologia 101.
Fonte: (RAUTENBE*RG et al*, 2008).

[6] Disponível em <http://www.ksl.stanford.edu/software/ontolingua/>.
[7] Disponível em <http://www.daml.org/ontologies/>.

Para a definição das restrições das propriedades, deve-se levar em consideração a propriedade de classe como dados, observando-se o tipo de dado que a propriedade comporta (*string* ou número, por exemplo). Caso a propriedade seja uma relação, deve-se definir a quais classes a relação se refere. Restrições sobre cardinalidade e valores válidos para as propriedades são consideradas neste passo.

A última etapa refere-se à criação das instâncias do domínio: são criadas instâncias da ontologia a partir da definição das classes, atribuindo valores as suas propriedades de dados e relações.

Metodologia *Methontology*

Essa metodologia foi desenvolvida no Laboratório de Inteligência Artificial da *Universidad Politécnica de Madrid* (UPM) e possibilita a criação de ontologias apoiada em processos de reengenharia de ontologia, auxiliando também na criação (com reutilização) de outras ontologias (GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004).

A Figura 3 ilustra a metodologia *Methontology*, na qual é possível observar as atividades previstas pelo método para desenvolvimento de ontologias. O planejamento, que se encontra dentro das atividades de gestão, é o primeiro passo para a construção de quaisquer ontologias.

A ontologia criada durante a atividade de desenvolvimento possui o seguinte ciclo de vida: especificação, conceitualização, formalização, desenvolvimento e manutenção. A ontologia em construção poderá sofrer alterações durante as fases do seu ciclo de vida.

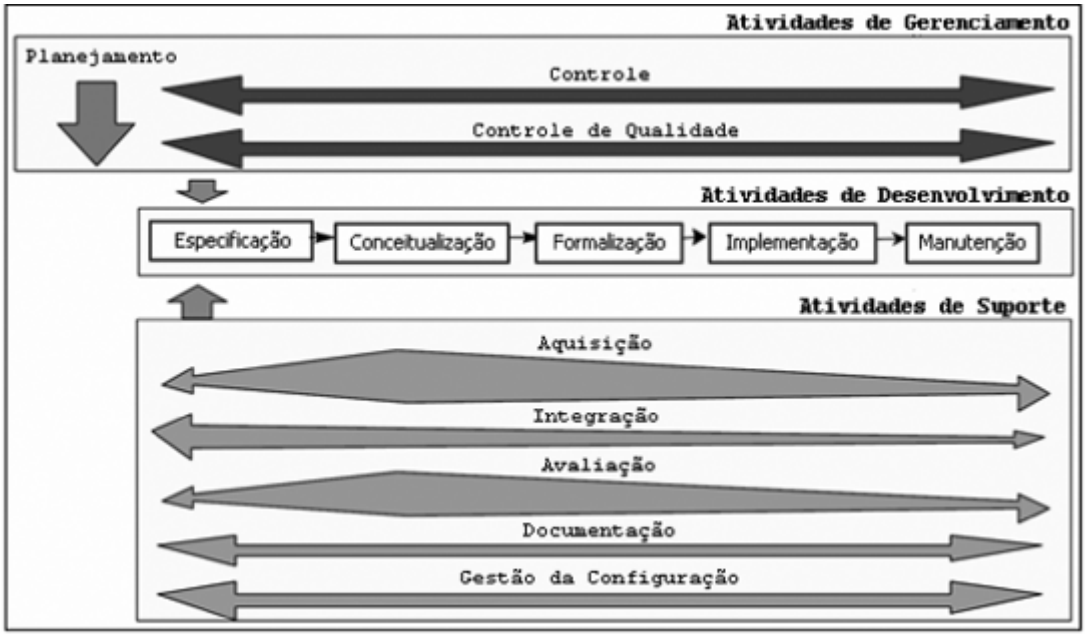


Figura 3 Metodologia *Methontology*.
Fonte: extraída de gómez-Pérez, Fernández-López e Corcho (2004).

O objetivo da fase de especificação da ontologia é a criação de um documento em linguagem natural, que contemple o principal objetivo da ontologia e os níveis de granularidade e de alcance. Essa especificação da ontologia deve ser a mais completa e concisa, o quanto possível.

A fase de conceitualização visa à organização dos conhecimentos não estruturados, adquiridos nas fases anteriores. Essa fase converte as informações do domínio em estudo em uma especificação semiformal, utilizando um conjunto de representações intermediárias, que possam ser compreendidas pelos especialistas do domínio e pelos desenvolvedores da ontologia.

A fase de formalização consiste em identificar e formalizar a criação dos componentes (classes, relações, axiomas e instâncias) da ontologia em desenvolvimento.

Já a fase de implementação de ontologias requer a utilização de ambientes capazes de dar suporte às características das meta-ontologias selecionadas na atividade de integração, descrita a seguir. O resultado da fase de codificação é a ontologia construída em alguma linguagem, como a OWL.[8]

[8] Cf. a seção "2.2.2 Linguagens de representação de ontologias" do presente Capítulo.

Na fase de manutenção, a ontologia recebe melhorias ou reparos em possíveis defeitos, quando possível. Na Figura 3, nota-se que a fase de manutenção possui um fluxo para fase de especificação. Esse ciclo de vida das atividades de desenvolvimento é amparado pelas atividades de apoio, representadas na Figura 3, quais sejam: aquisição do conhecimento, integração, avaliação, documentação e gestão de configuração. Pela Figura 3 observam-se as tarefas que são enfatizadas durante algumas fases, enquanto que a execução de outras tarefas permanece constante.

A atividade de aquisição do conhecimento é uma atividade independente no processo de desenvolvimento da ontologia. Ela normalmente diminui conforme evolui o desenvolvimento, pois é natural que os pesquisadores se tornem mais familiarizados com o domínio.

Já a atividade de avaliação diz respeito ao julgamento técnico da ontologia, verificando se a implementação e a documentação estão de acordo com o que foi planejado na fase de especificação.

2.2.4 Tecnologias para a construção de ontologias

As ferramentas que apoiam a construção de ontologias utilizadas pela comunidade da área são o Protégé[9] e o WebOnto[10]. Alguns raciocinadores de ontologia são o Jena,[11] Jess[12] e Pellet.[13]

O Protégé é um ambiente gráfico que permite construir aplicações de base ontológica simples e complexas. Os desenvolvedores podem integrar a saída do Protégé com os sistemas de regras ou outros solucionadores de problemas, para a construção de uma ampla gama de sistemas inteligentes. Ela possui mecanismos de inferência, baseados em lógica de descrição, para a verificação de consistência da ontologia.

WebOnto é um ambiente gráfico que possibilita a navegação, criação e edição de ontologias. Ele permite o gerenciamento de ontologias por interface

[9] Disponível em <http://protege.stanford.edu/>.
[10] Disponível em <http://projects.kmi.open.ac.uk/webonto/>.
[11] Disponível em <https://jena.apache.org/>.
[12] Disponível em <http://www.jessrules.com/jess/index.shtml>.
[13] Disponível em <https://github.com/complexible/pellet>.

gráfica, inspeção de componentes, verificação da consistência da herança e trabalho cooperativo (DOMINGUE, 1998).

O Jena é um raciocinador formado por um conjunto de APIs (*Application Programming Interface*) escritas em Java e de código aberto, que auxiliam a construção de aplicações da web semântica. Foi desenvolvido por pesquisadores da *Hewlett Packard* (HP).[14]

O Jess é uma ferramenta desenvolvida em Java para a construção de regras, desenvolvida nos laboratórios da *Sandia National Laboratories*[15] em *Livermore*, Califórnia.

O Pellet, por sua vez, é um raciocinador OWL DL *open source* baseado em Java. O Pellet atua como um raciocinador lógico, e seus objetivos são: verificar a consistência de ontologias; classificar a taxonomia; verificar vínculos; e responder a um subconjunto de consultas.

2.3 Mineração de Texto

Com o passar dos anos, em decorrência das altas taxas de geração de conhecimento, as tecnologias para aquisição e armazenamento de dados tiveram um avanço significativo. Grande parte desse conhecimento é formada por dados não estruturados, ou seja, dados em formato de texto, tais como e-mails, livros, entre outros.

A organização desses documentos textuais é de grande interesse para as instituições. O grande volume de dados textuais armazenados dificulta a análise e compreensão dos documentos perante a visão humana (REZENDE et al*.,* 2011). Nessa lógica, a Mineração de Texto nos fornece um meio de extrair conhecimento útil a partir de um grande volume de dados.

A Mineração de Texto (MT) tenta resolver a crise de excesso de informação por meio da combinação de técnicas de Mineração de Dados (MD), de Aprendizado de Máquina (AM), de Processamento de Linguagem Natural (PLN), e de recuperação de informação e gestão do conhecimento (FELDMAN; SANGER, 2006).

[14] Disponível em <http://www.hp.com/>.
[15] Disponível em <www.sandia.gov>.

2.3.1 Processo de Mineração de Texto

Existem, na literatura, variações do processo de Mineração de Textos. Esse processo pode ser resumido em quatro etapas principais: (i) coleta de documentos; (ii) pré-processamento; (iii) extração de padrões; e (iv) análise e avaliação dos resultados (DUQUE et al*.,* 2011; EBECKEN et al*.,* 2003; FELDMAN; SANGER, 2006).

2.3.1.1 Coleta de documentos

O objetivo da etapa de coleta de documentos é a criação de uma base textual, ou seja, é realizada a captação de toda a base de documentos e textos a ser trabalhada nas próximas etapas (ARANHA; PASSOS, 2006; MATOS, 2010).

A coleta de dados é feita em repositórios, nos quais estão disponíveis máquinas de busca que indexam o conteúdo existente na Web a partir de palavras-chave, como por exemplo o Google. Na área científica, é possível mencionar máquinas de busca como a *Scopus*, a *Web of Science*, os periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), entre outras (MATOS, 2010).

2.3.1.2 Pré-processamento

O pré-processamento é responsável por obter a representação estruturada dos documentos. (FELDMAN; SANGER, 2006; ARANHA, 2007).

O Processamento de Linguagem Natural (PLN) é uma subárea da Inteligência Artificial, tendo como propósito de desenvolvimento os métodos computacionais que permitam compreender a linguagem natural (falada ou escrita) da mesma forma que o ser humano (ARANHA, 2007; JACKSON; MOULINIER, 2007).

Os métodos de PLN são usados na Mineração de Textos, especificamente na etapa de pré-processamento, com o intuito de melhorar a representação dos documentos. Ou seja, a PLN promove o reconhecimento e a classificação de termos a partir de documentos textuais (ARANHA, 2007).

A PLN realista a análise de documentos em diversos níveis linguísticos. Jurafsky e Martin (2000) classificam a análise linguística em seis categorias diferentes:

- **Fonética** e **Fonológica**: estudo dos sons linguísticos;
- **Morfológica**: formação e construção dos termos;
- **Sintática**: análise da relação entre termos em uma sentença;
- **Semântica**: análise dos significados dos termos e das sentenças;
- **Pragmática**: compreensão do uso da língua utilizando conhecimentos do mundo;
- **Discursiva**: interpreta a estrutura e o significado de um texto por completo.

Várias técnicas podem ser aplicadas durante a etapa de pré-processamento automático de texto (SPASIC et al., 2005). Dente elas, é possível citar as seguintes:

- **Tokenização**: divide o texto em unidades básicas, conhecidas como *tokens*; a divisão é feita por meio de delimitadores, os quais podem ser espaço em branco, pontuação ou até mesmo sentenças.
- **Etiquetador gramatical (*Part-Of-Speech* – *POS tagger* ou *POS tagging*)**: identifica as classes gramaticais de cada termo do texto, podendo ser aplicada em nível morfológico (substantivo, adjetivo, artigo).
- **Lematização**: substitui um termo flexionado pela forma base eliminando número e gênero (por exemplo: brincamos ⇒ brincar);
- ***Stemming***: elimina as variações morfológicas de um termo. Os prefixos, sufixos, características de gênero, número e grau dos termos são retirados (por exemplo: brincamos ⇒ brinc);
- **Remoção de *Stopwords***: são eliminados termos que não devem ser considerados no documento. *Stopwords* são termos não relevantes na análise de textos. Geralmente são preposições, pronomes, artigos, entre outros;
- **Identificação de termos**: identificação dos termos presentes no texto, em que os termos podem ser simples ou compostos (por exemplo: aluno, sala de recurso, professor);

Atualmente, o PLN disponibiliza um leque razoavelmente grande de ferramentas que implementam essas técnicas. Contudo, essas ferramentas possuem maior destaque na língua inglesa. Uma das ferramentas mais difundidas é a *Natural Language Toolkit* (NLTK)[16] ou as ferramentas desenvolvidas em Stanford.

Com relação às ferramentas que suportam o idioma Português, tem-se a Apache OpenNLP[17] e a Aelius.[18] No entanto, nem todas as técnicas do PLN estão implementadas nessas ferramentas – como é o caso da OpenNLP, em que não há suporte para a lematização, ou a Aelius, que realiza somente a anotação das sentenças. A OpenNLP é utilizada como base para o desenvolvimento de outras ferramentas, tais como o lematizador LemPORT (RODRIGUES; GONÇALO OLIVEIRA; HUGO GOMES, 2014).

Atualmente, as ferramentas voltadas para o PLN no Português estão sendo aprimoradas nos núcleos de pesquisas das universidades. Como exemplo podemos citar o *Interinstitutional Center for Computational Linguistics* (NILC)[19] e o *NLX-Group*.[20]

O desenvolvimento da presente pesquisa de Mestrado utiliza a ferramenta OpenNLP pelo fato de esta suportar o idioma Português e possuir as técnicas de tokenização e etiquetador gramatical. Uma das facilidades do OpenNLP é a possibilidade de integração a projetos desenvolvidos em Java. Também é utilizadO o lematizador LemPORT, uma vez que ele é facilmente integrado com a OpenNLP.

2.3.1.3 Extração de padrões

O objetivo da etapa de extração de padrões é aplicar técnicas úteis para a extração de conhecimentos, fazendo o uso de combinações de algoritmos e técnicas de Mineração de Dados (MD) provenientes de diversas áreas do conhecimento , tais como: Aprendizado de Máquina (AM), estatística e bancos de dados (ARANHA, 2007). As principais tarefas da Mineração de Dados são: classificação, regressão, agrupamento, associação e extração da informação (NEVES; CORRÊA; CAVALCANTI, 2013).

[16] Disponível em <http://nlp.stanford.edu/software/>.
[17] Disponível em <https://opennlp.apache.org/documentation/manual/opennlp.html>.
[18] Disponível em <http://aelius.sourceforge.net/>.
[19] Disponível em <http://www.nilc.icmc.usp.br/nilc/index.php>.
[20] Disponível em <http://nlxgroup.di.fc.ul.pt/>.

No contexto dessa dissertação foi utilizada a tarefa de Extração de Informação, para realizar a extração de relacionamentos semânticos em textos informais.[21]

2.3.1.4 Métricas para a Avaliação dos Resultados

Nesta fase é necessário contar com o fator humano – preferencialmente um especialista no domínio – para avaliar se os resultados obtidos estão de acordo com algumas métricas, as quais se baseiam na noção de relevância. Por fim, o usuário é o responsável por julgar a aplicabilidade destes resultados.

As métricas provêm informações importantes, que permitem uma avaliação dos dados gerados, bem como da qualidade de cada uma das etapas, individualmente. As principais métricas existentes para análise dos dados extraídos são: precisão, cobertura (ou revocação) e medida F.

A precisão é uma medida de fidelidade. Conforme expressado na Equação I, a precisão fornece a taxa do número de elementos relevantes recuperados (Verdadeiros Positivos – VP) frente ao número total de elementos recuperados. O número total de elementos recuperados corresponde à soma do número de elementos relevantes recuperados com o número de elementos recuperados que não são relevantes (Falsos Positivos – FP).

Equação I Precisão.

$$\text{Precisão} = \frac{Número\ de\ elementos\ relevantes\ recuperados}{Número\ total\ de\ elementos\ recuperados}$$

A cobertura (ou revocação) informa a taxa de acertos. Conforme expressado na Equação II, a cobertura é definida como o número de elementos relevantes recuperados (VP) frente ao número total de elementos relevantes. O número total de elementos relevantes corresponde à soma entre o número de elementos relevantes recuperados (VP) e o número de elementos relevantes que não foram recuperados (Falsos negativos – FN).

[21] Esta tarefa será detalhada logo mais, na seção 2.4 do presente Capítulo.

Equação II Cobertura.

$$\text{Cobertura} = \frac{Número\ de\ elementos\ relevantes\ recuperados}{Número\ total\ de\ elementos\ relevantes}$$

A medida F é a média harmônica ponderada da precisão e cobertura, como demonstrado na Equação III, onde P e R são respectivamente precisão e revocação. Quando a precisão e revocação têm o mesmo peso, $\beta = 1$, a medida F é expressada na Equação III.

Equação III Medida F_β.

$$Medida\ F_\beta = \frac{2\ x\ P\ x\ R}{\text{P} + \text{R}}$$

2.4 Extração de Informação

As técnicas de Extração de Informação (EI) permitem a localização e, posteriormente, realizam a extração de sentenças de um texto ou de elementos específicos de textos não estruturados, armazenando o conteúdo extraído em formato estruturado. Normalmente o resultado da EI é armazenado em um banco de dados, para que possa ser utilizado por algoritmos de Mineração de Dados (MD) para identificar padrões de interesse (FELDMAN; SANGER, 2006; GUPTA; LEHAL, 2009).

Existem três principais abordagens para a EI. São elas: Aprendizado de Máquina (COHEN; HUNTER, 2008; KOU; COHEN; MURPHY, 2005), regras (HEARST, 1992; SPASIC et al., 2005) e dicionário (KRAUTHAMMER; NENADIC, 2004). A abordagem baseada em Aprendizado de Máquina faz o uso de classificadores com o intuito de separar ou identificar sentenças de interesse. A abordagem baseada em regras identifica padrões por meio de expressões regulares. A abordagem baseada em dicionários faz o uso de um ou mais dicionários para a identificação de termos (palavras).

2.4.1 Abordagem baseada em Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial, concentrada em desenvolver modelos que possam “aprender” por meio da experiência (SANCHES, 2003). Segundo Taba (2013), o aprendizado acontece quando os dados são comparados com determinadas características úteis e relevantes; em inglês, são chamadas de *features*, as quais são atributos que descrevem uma instância. Os métodos de Aprendizado de Máquina fazem uso das *features* como forma de generalizar e discriminar instâncias.

Krauthammer e Nenadic (2004) afirmam que técnicas de Aprendizado de Máquina usam dados de treinamento para aprender as características que são úteis e relevantes para o reconhecimento e a classificação de termos (palavras). Porém, o grande volume de dados para o treinamento torna-se um problema para o Aprendizado de Máquina (ANANIADOU, MCNAUGHT, 2005; MONARD et al., 2000).

Vários algoritmos de Aprendizado de Máquina têm sido utilizados para identificação e classificação de termos, incluindo *Naive Bayes*, *Support Vector Machines* (SVMs) e o algoritmo C4.5 para a obtenção de árvores de decisão (EBECKEN; LOPES; COSTA, 2003).

2.4.2 Abordagem baseada em Dicionário

O Dicionário consiste em uma lista em um arquivo de texto puro, em linguagem estruturada de marcação de dados – por exemplo, XML e JSON (*JavaScript Object Notation*) – ou ainda em uma tabela de um banco de dados. O dicionário contém um conhecimento prévio sobre o domínio do problema.

Ananiadou e McNaught (2005) afirmam que a abordagem baseada em dicionário provê uma lista de termos para encontrar as ocorrências no texto. Rebholz-Schuhmann et al.(2005) complementam que o uso de dicionário em conjunto com ontologias permite relacionar as informações citadas em textos de domínio com as informações armazenadas em um banco de dados.

Duque et al.(2011) mencionam uma desvantagem da abordagem de dicionário, a qual está relacionada à restrição de nomes que estão presentes no dicionário, sendo, assim, indispensável o armazenamento de palavras com variações (como por exemplo as de gênero e número). Visando aumentar a precisão

dos resultados das extrações, é necessário que os sinônimos, homônimos e as variações sejam incluídos nos dicionários.

2.4.3 Abordagem baseada em Regras

A abordagem baseada em Regras é uma das abordagens mais simples da Extração de Informação, sendo conhecida, também, como abordagem baseada em Padrões Textuais (COHEN; HUNTER, 2008).

Esta abordagem utiliza termos padrões de formação e fundamenta-se no desenvolvimento e na aplicação de regras que descrevem estruturas de nomes comuns para certas classes de termos, usando ortografia léxica descrita por expressão regular ou recursos morfossintáticos complexos (ANANIADOU; MCNAUGHT, 2005).

A definição de expressões regulares pode ser expressa através da teoria de linguagens formais. Elas consistem em constantes e operadores que denotam conjuntos de cadeias de caracteres e operações sobre esses conjuntos, respectivamente. Dado um alfabeto finito Σ, as seguintes constantes são definidas:

- (conjunto vazio) ∅ denotando o conjunto ∅;
- (cadeia de caracteres vazia) ε denotando o conjunto {ε};
- (literal) a em Σ denotando o conjunto {a}.

Taba (2013) afirma que a extração de relações semânticas com base em padrões textuais é a proposta mais antiga e mais simples. Trabalhos que seguem esta abordagem fazem uso de padrões textuais como indicativos (pistas) de que determinada construção textual denota uma relação entre duas entidades.

Hearst (1992) foi uma das primeiras estudiosas a explorar a abordagem de padrões textuais. Um dos padrões definidos em seu trabalho é apresentado na forma de expressão regular.

O padrão a seguir indica uma relação hiponímia, conhecida como *is-a* (é-um): NP_0 *such as* {NP_1, NP_2, ..., (*and* | *or*)} NP_n, onde “NP” denota um sintagma nominal (*noun phrase*), “{” e “}” representam a repetição de 0 (zero) ou mais vezes do padrão entre as chaves e “|” indica uma opção de escolha entre valores.

Um exemplo que utiliza a relação hiponímia é o seguinte: dada a sentença “O Brasil é um país em desenvolvimento”, temos a relação “*is-a* (Brasil, país)”.

Existem outros trabalhos relacionados à extração por regras, mudando o tipo de relação e aplicando o algoritmo de Hearst (1992). Podemos mencionar Berland e Charniak (1999) para a extração de relação de meronímia (*part-of* ou parte-de) e o trabalho de Taba (2013), interessado nas relações *propert-of*, *is-a*, *part-of*, *location-of*, *effect-of*, *made-of* e *used-for*.

Este trabalho de Mestrado utiliza a extração de relacionamentos baseado em Regras. O processo de extração é apoiado pelo uso de autômato finito determinístico. A seção a seguir apresenta a definição e exemplo para o uso deste autômato. A representação dos relacionamentos extraídos é feita por meio do *Formal Concept Analysis*.[22]

2.5 Autômato finito determinístico

Autômato finito determinístico é um modelo matemático de sistema com entradas e saídas discretas. Pode assumir um número finito e pré-definido de estados. Cada estado possui informações necessárias para determinar as ações para a próxima entrada (HOPCROFT; ULLMAN, 1969).

No presente trabalho, aborda-se o uso de autômato finito determinístico para apoiar a extração de relacionamentos semânticos, por meio da abordagem baseada em padrões textuais. Como descrito por Menezes (1997), autômatos finitos e expressões regulares fazem parte do grupo de linguagens regulares definida na hierarquia do linguista Noam Chomsky. Hopcroft e Ullman (1969) provam em seu livro, por meio de conceitos matemáticos, que autômatos finitos e expressões regulares são equivalentes. Logo, o autômato finito determinístico reconhece um conjunto de linguagens regulares que são, dentre outros aspectos, úteis para a realização de análise léxica e reconhecimento de padrões.

Um autômato finito determinístico é definido formalmente por uma quíntupla: $M = (\Sigma, Q, \delta, q0, F)$, onde:

[22] O método FCA será descrito mais adiante, na seção 2.6.

- Σ – Alfabeto de símbolos de entrada;
- Q – Conjunto finito de estados possíveis do autômato;
- δ – Função de Transição ou Função Programa δ: Q x Σ → Q. Se M está no estado Q e vê a entrada a, o autômato vai para o estado δ (q,a);
- q0 – Estado inicial tal que q0 ∈ Q;
- qF – Conjunto de estados finais, tais que F ⊆ Q, ou seja, F está contido em Q.

O processamento de um autômato, para uma palavra de entrada w, consiste na sucessiva aplicação de função programada para cada símbolo de w (da esquerda para direita), até ocorrer uma condição de parada.

Para melhor compreensão de um autômato finito, considere o seguinte exemplo, no qual é definida a seguinte linguagem:

$$L = \{ w \mid w \text{ possui aa ou bb como subpalavra} \}$$

O Autômato que reconhece essa linguagem é definido como:

$$M = (\{a,b\}, \{q0,q1,q2,qf\}, \delta, q0, \{qf\})$$

Onde δ é representado pela Tabela 1.

Tabela 1 δ em forma de tabela.

δ	a	b
q0	q1	q2
q1	qf	q2
q2	q1	qf
qf	qf	qf

Fonte: elaboração própria.

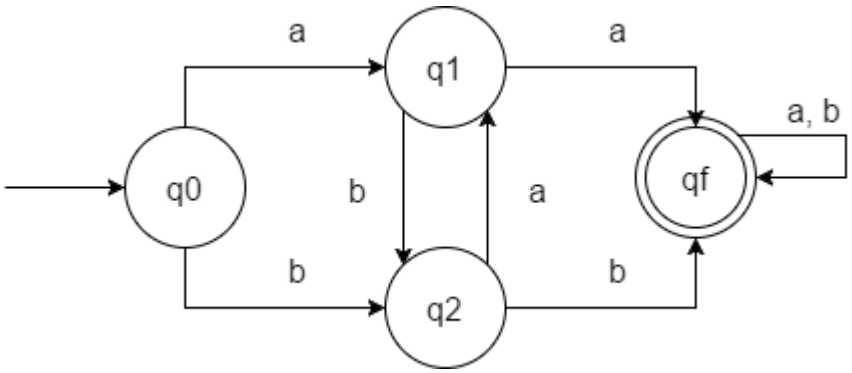


Figura 4 Exemplo de autômato.
Fonte: adaptada de Menezes (1997).

O autômato pode ser representado pelo grafo na Figura 4. O algoritmo apresentado usa os estados q1 e q2 para armazenar o símbolo anterior. Assim, q1 contém “a” como símbolo anterior e q2 contém “b” como símbolo anterior. Após identificar dois a’s ou dois b’s consecutivos, o autômato assume o estado final (qf) e varre o sufixo do termo de entrada, sem qualquer controle lógico.

O autômato sempre para ao processar qualquer entrada. O processo de parada do autômato pode ser de duas maneiras: aceitando ou rejeitando uma entrada w. Existem ainda três concisões de parada: as duas primeiras acontecem após o processamento do último símbolo, quando o autômato assume um estado final ou não final – sendo que, no estado final, o autômato aceita o dado de entrada, e no estado não final o autômato rejeita o termo de entrada.

O último caso de parada refere-se a quando um símbolo lido não é definido pelo autômato, obrigando-o a parar e rejeitar o dado de entrada.

No contexto desta dissertação, o autômato foi utilizado como forma de identificar os relacionamentos presentes. No caso, o autômato é responsável por entrar os termos associados a um verbo.[23]

2.6 Formal Concept Analysis

Como a construção de estruturas ontológicas a partir de textos é considerada um problema de difícil solução, outras abordagens têm surgido. É o caso do método

[23] O autômato utilizado no presente trabalho será detalhando mais adiante, na seção 4.5.2 do Capítulo 4.

Formal Concept Analysis (FCA), que vem sendo aplicado na construção de tais estruturas (CIMIANO; HOTHO; STAAB, 2005).

Moraes (2012) realizou um estudo comprovando que a junção do FCA e papéis semânticos geram estruturas ontologias as quais podem ser úteis para a criação de ontologias. Um exemplo dado pela autora, é que a partir de dependências sintáticas entre os verbos e seus argumentos, podemos definir contextos formais.

Formal Concept Analysis (FCA) é um método usado principalmente para a análise dos dados. Os dados são estruturados em unidades que são abstrações formais de conceitos do pensamento humano, permitindo interpretação significativa. Assim, FCA pode ser vista como uma técnica de agrupamento conceitual, como também fornece descrições intencionais para os conceitos abstratos ou unidades de dados que produz (CIMIANO; HOTHO; STAAB, 2005).

Tabela 2 Contexto formal definido a partir dos relacionamentos.

	aluno-ter	professor-trabalhar	aluno-possuir	aluno-chegar
sala	x	x	x	x
professor	x		x	
laudo	x	x		

Fonte: elaboração própria.

O modelo matemático que permite descrever os conceitos formais introduz, inicialmente, a noção de contexto formal. Contextos formais são caracterizados pela tripla (G; M; I) (GANTER; WILLE, 1999; PRISS, 2007), onde:

- G é o conjunto formado pelas entidades do domínio, ditas objetos formais;
- M é constituído pelas características dessas entidades, seus atributos formais;
- I é uma relação binária sobre G × M, denominada relação de incidência, que associa um objeto formal ao seu atributo. Desta forma, a relação gIm pode ser lida como “o objeto g tem o atributo m”.

Como forma de melhor representar o FCA, o exemplo a seguir retoma o contexto da Tabela 2, que apresenta um subconjunto de relacionamentos semânticos:

- Aluno-ter-sala;

- Aluno-ter-professor;
- Aluno-ter-laudo;
- Professor-trabalhar-sala;
- Professor-trabalhar-laudo;
- Aluno-possuir-sala;
- Aluno-possuir-professor;
- Aluno-chegar-sala.

Diante da Tabela 2, é possível definir o conjunto G de objetos formais por {sala, professor, laudo, amigo} e o conjunto M de atributos formais por {aluno-ter, professor-trabalhar, aluno-possuir, aluno-chegar}. Já o conjunto I corresponde à relação binária entre as relações G-M.

Os conceitos formais são definidos a partir de um contexto formal, e são necessários operadores de derivação. Assim, para dois conjuntos arbitrários de objetos e atributos, denotados, respectivamente, por O e A, os operadores O' e A' podem ser definidos por Wille (2005) como:

$$O' = \{m \in M / gIm \text{ para todo } g \in O\}$$
$$A' = \{g \in G / gIm \text{ para todo } m \in A\}$$

Considerando que O' determina todos os atributos em M compartilhados pelos objetos em O e A' determina todos os objetos em G que compartilham os atributos em A, um conceito formal em (G, M, I) é definido pelo par (O, A) se e somente se $O \supseteq G$, $A \supseteq M$, tal que O' = A e A' = O (PRISS, 2007; WILLE, 2005). Da Tabela 3.1 pode-se, então, extrair os conceitos formais:

({sala}, {aluno-ter, professor-trabalhar, aluno-possuir, aluno-chegar})
({sala, professor}, {aluno-ter, aluno-possuir})
({sala, laudo}, {aluno-ter, professor-trabalhar})

A Figura 5 ilustra[24] uma representação para o reticulado de conceitos formado a partir da Tabela 2. A Figura é resultante da técnica de "etiquetagem reduzida" (*reduced labeling*) (WILLE, 1997). Essa técnica é utilizada para reticulados com grande número de conceitos, facilitando, assim, a visualização da estrutura. Por meio da *reduced labeling*, se um rótulo de objeto pertencer a todos esses nós, apenas o modo mais inferior desse caminho exibirá tal rótulo, ficando implícita a sua presença nos seus ascendentes. No caso de atributos, é o inverso: o rótulo aparecerá apenas em um nó superior, tornando-se sua presença implícita em seus nós descendentes.

Pela Figura 5, em conjunto com a técnica *reduced labeling,* pode-se notar que o atributo formal "aluno-ter" está relacionado com todos os objetos descendentes (amigo, laudo, professor e sala). Já o atributo "professor-trabalhar" está relacionado com os objetos laudo e sala.

Como se pode notar, na representação gráfica do FCA, as relações ajudam na melhor representação do conhecimento.

A partir da observação da Figura 5 é possível notar que a ferramenta *Concept Explorer* gera esferas representadas por cores diferentes. A ferramenta atribui à esfera "azul-branca" atributos que não possuem um objeto ligado de forma direta, como é o caso do atributo "aluno-ter". Contudo, é possível interpretar, pela imagem, que o atributo possui uma relação com os objetos "laudo", "professor" e "sala".

[24] O diagrama da Figura 5 foi construído com o auxílio da ferramenta *Concept Explorer 1.3*.

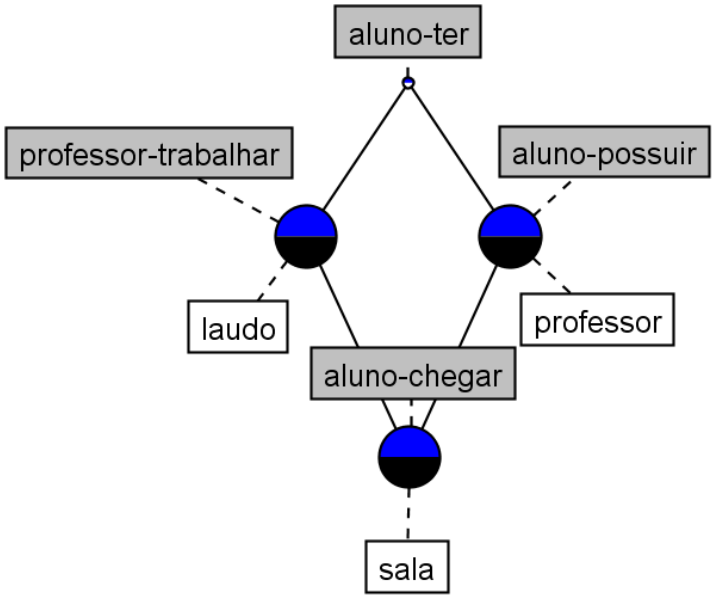


Figura 5 Representação de um reticulado de conceitos.
Fonte: elaboração própria.

Já a esfera “azul-preto” contém os atributos e objetos ligados a ela. Ao analisar o FCA, o atributo “professor-trabalhar” está relacionado com os objetos “laudo” e “sala”.

Ao omitir o relacionamento “aluno-ter-sala”, o FCA é alterado, mostrando novas esferas, como é observado na Figura 6. A esfera branca não conterá atributos e relacionamentos, sendo usada para iniciar/finalizar o reticulado. Por fim, a esfera “branco-preta” aparecerá quando não há atributos ligados diretamente ao objeto.

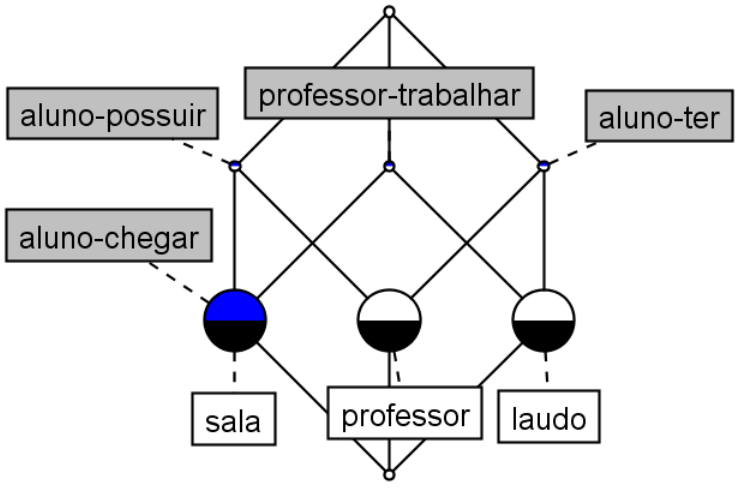


Figura 6 FCA alterado.
Fonte: elaboração própria.

No contexto desta dissertação, o FCA é utilizado como ferramenta de apoio ao especialista de domínio na identificação de relacionamentos relevantes extraídos.

2.7 Considerações finais

Neste Capítulo, apresentou-se o conceito de ontologias e sua definição formal. Em seguida, foram abordados os tipos de linguagem em que as ontologias podem ser representadas.

Após a definição conceitual de ontologia, foi abordado o processo de criação de ontologias e as metodologias que apoiam esta construção, por meio das quais é possível notar a evolução do processo de criação no decorrer dos tempos. Dentre elas, mencionamos as metodologias 101 e *methontology*.

Em seguida, foram apresentados os conceitos sobre Mineração de Textos (MT), que podem ser divididos em quatro etapas principais: (i) coleta de documentos, que vão constituir a base textual; (ii) pré-processamento, etapa responsável por obter uma representação estruturada dos documentos; (iii) extração de padrões, fase em que é possível aplicar técnicas de extração; e (iv) análise e avaliação dos resultados, etapa de avaliação do resultado gerado a partir dos passos anteriores.

Foram discutidos fundamentos da Extração de Informação (EI) para obtenção de informações relevantes em dados não estruturados. Para a extração, são utilizadas três abordagens: (i) abordagem baseada em Aprendizado de Máquina (AM), que utiliza classificadores para separar ou identificar sentenças de interesse; (ii) abordagem baseada em Regras, utilizada para identificar padrões de extração com expressões regulares; e (iii) abordagem baseada em Dicionário, que utiliza informações de um dicionário para auxiliar na identificação dos termos ou das entidades no texto.

Foi introduzido, também, o conceito de autômato finito determinístico, uma abordagem de extração de relacionamentos baseada em regras que auxilia o processo de construção de ontologias. Por fim, foi apresentado o método FCA: uma estrutura que, no contexto deste trabalho, oferece suporte à visualização dos relacionamentos semânticos.

Os trabalhos relacionados que abordam os conceitos apontados neste Capítulo serão apresentados no Capítulo seguinte.

CAPÍTULO 3

Trabalhos correlatos

3.1 Considerações iniciais

O presente Capítulo está dividido em duas seções principais: a primeira, referente aos trabalhos correlatos voltados apenas para a descoberta de relacionamento semântico; e a segunda relacionada à construção de ontologias a partir de documentos textuais. Na última seção, são apresentadas as considerações finais.

3.2 Extração de relacionamento semântico

Nesta seção são abordados trabalhos referentes à descoberta de relacionamentos semânticos em sentenças. Os relacionamentos semânticos são identificados por meio dos processos de Mineração de Texto (MT), Padrões Textuais e Aprendizado de Máquina (AM).

3.2.1 Um método para descoberta de relacionamentos semânticos do tipo "causa e efeito" em sentenças de artigos científicos do domínio biomédico

Na dissertação de Scheicher (2013), foi proposto um método para a extração de relacionamentos semânticos, especificamente relações de "causa e efeito" em artigos científicos do domínio biomédico (Anemia Falciforme), identificadas em textos formais no idioma inglês. O trabalho compreendeu quatro etapas: (1) preparação de dados; (2) extração de termos do domínio biomédico de documentos científicos; (3) identificação de relacionamentos existentes nos textos, com base nos termos extraídos; e (4) sugestão de uma rede de conhecimento baseada nos relacionamentos extraídos. As etapas mencionadas podem ser visualizadas na Figura 7.

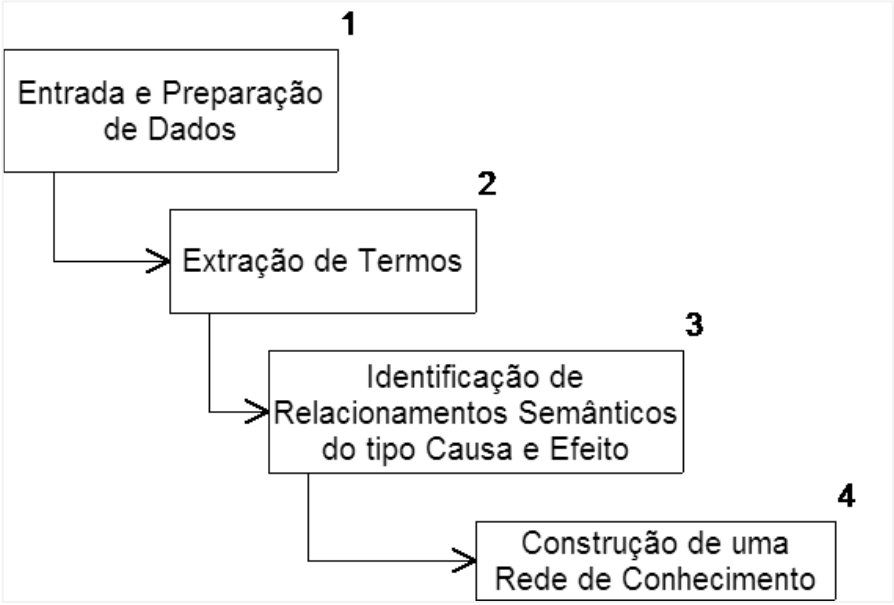


Figura 7 Etapas do modelo proposto por Scheicher (2013)**.**
Fonte: extraída de Scheicher (2013).

Na etapa 1 é realizado o processo de anotação morfossintática do corpus, por meio do etiquetador de *Part-of-Speech* (POS). A saída é um arquivo no formato JSON que contém as sentenças anotadas.

Na etapa de extração, são extraídos dois conjuntos: termos e *tip words*. Os termos são palavras consideradas relevantes do domínio biomédico, tais como

"genes", "doenças", "proteínas" etc. Já as *tip words* são termos que podem indicar que determinada sentença possui um relacionamento de causa e efeito.

A extração automática de termos e *tip words* é feita por meio de dois dicionários, um para termos e outro para *tip words*. Para a construção do dicionário de termos foi realizado o mapeamento de um conjunto de ontologias do domínio biomédico. Logo, foram extraídos das ontologias os nomes dos termos, descrições, categorias e sinônimos para serem colocados no dicionário. Na fase de extração, o arquivo de entrada é o JSON gerado pela etapa de preparação. A saída será o arquivo JSON com os termos e *tip words* anotados. Diante da sentença "*With endothelial dysfunction and vascular injury, the levels of endothelial bound and soluble adhesion molecules increase*" obtém-se, como termos, as "*endothelial dysfunction*", "*vascular injury*" e "*soluble adhesion molecules*"; já as *tip words* aparecem representadas por "*increase*".

Em seguida, o algoritmo faz a extração de relações de causa e efeito, com auxílio de um padrão textual. Existem dois padrões para a extração: a MetaRegra de Associação e a MetaRegra *Increase*/*Decrease*. A MetaRegra de associação possui a função de identificar uma sentença como associação e a MetaRegra *Increase*/*Decrease* de identificar uma sentença como *Increase*/*Decrease*. Os padrões utilizados juntamente com os exemplos são demostrados na Figura 8.

MetaRegra de Associação:

```
((?:and|or| , | .)?(?:<.*>)(?:<protein>|<disease>|<sca complication>).*)
((?:<associated with>)<tip word>) ((?:and|or| , |.*| .)?(?:<.*>)
(?:<protein>|<disease>|<sca complication>).*)
```

MetaRegra *Increase/Decrease*:

```
((?:and|or| , | .)?(?:<increase><tip word> )?(?:<.*>) (?:<sca complication>|<protein>)
(?: <increase><tip word>)?).*?((?:and|or| , | .)? (?:<increase><tip word> )?(?:<.*>)
(?:<sca complication>|<protein>)(?: <increase><tip word>)?)
```

Figura 8 Extração de relacionamentos semânticos.
Fonte: extraída de Scheicher (2013).

A última etapa consiste na construção de uma rede semântica. A rede consiste em uma estrutura de grafo. Nos nós, são armazenadas informações tais como termo, se há (e qual é) uma *tip word* associada, número e nome do artigo do qual foi extraída.

A avaliação da metodologia de extração de sentenças foi comparada à extração manual. Por se tratar de uma extração em um contexto bem específico, os resultados para a metodologia foram excelentes: a precisão, revocação e medida F foram, respectivamente, 94,83%, 98,10% e 96,43%.

3.2.2 Um processo baseado em parágrafos para a extração de tratamentos de artigos científicos do domínio biomédico

Na dissertação de mestrado de Duque et al. (2011), a proposta foi a de realizar a extração de informações sobre tratamentos da anemia falciforme em textos formais no idioma inglês. Para a extração desse relacionamento, foram empregadas técnicas de Extração de Informação (EI) e de abordagem baseada em Regras, Aprendizado de Máquina e em Dicionário.

O processo apresentado é composto de seis passos, e pode ser visualizado na Figura 9.

Em resumo, são utilizados dois classificadores, chamados de C1 (Classificador 1) e de C2 (Classificador 2), ambos com o objetivo de separar as sentenças de interesse que terão, respectivamente, termos de complicação e termos de tratamento, das sentenças que possivelmente não terão nenhum destes termos. As sentenças são etiquetadas, e por fim, são utilizados dicionários e regras para identificar e extrair as partes de interesse dentro das sentenças pré-selecionadas.

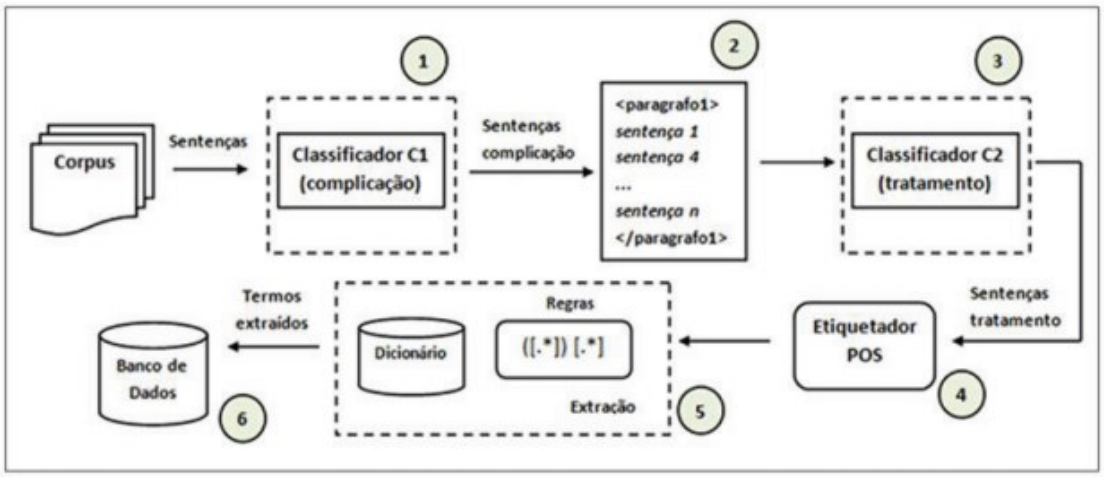


Figura 9 Processo proposto para extração de tratamentos.
Fonte: extraída de Duque et al. (2011).

O primeiro estágio da Extração de Informação é a fase de classificação de sentenças, cujo propósito é a construção de um modelo de classificação adequado, que melhor represente as particularidades das sentenças de treinamento e, com isso, possa predizer qual é a classe de uma nova sentença. A classificação de sentenças é supervisionada, ou seja, os rótulos das classes são previamente conhecidos.

O processo supervisionado de classificação de sentenças é avaliado por dois algoritmos estatísticos clássicos de Aprendizado de Máquina: *Support Vector Machine* (SVM) e *Naive Bayes* (NB).

No passo quatro, as sentenças com termos de tratamento são etiquetadas pelo etiquetador POS, cujo objetivo é auxiliar as regras a encontrar padrões de escrita de textos. A partir das sentenças classificadas, é possível realizar a identificação de termos relevantes em cada uma das sentenças de interesse. Duas abordagens para a Extração de Informação são utilizadas: Dicionário e Regras. O dicionário possui a função de identificar os termos validados e previamente conhecidos nas sentenças de interesse. Já as regras extraem novos termos das sentenças de interesse e os armazena no dicionário.

Para a abordagem baseada em Regras são estabelecidas duas estratégias: (1) Verbo ou palavra superficial em conjunto com etiqueta proveniente do etiquetador *POS tagger*; e (2) somente etiquetador *POS tagger*. Na primeira, o foco do processamento é realizado em parte da sentença, ou seja, a expressão regular somente casará com a sentença se, e somente se, existir o verbo ou palavra representativa na sentença, como mostra a Figura 10.

Na segunda regra, há somente o uso de POS, e o processamento é realizado na sentença por completo, fazendo com que o padrão POS case com algum padrão POS descoberto e criado por meio da análise realizada previamente no subconjunto de sentenças, como demonstrado na Figura 11. Nessas abordagens, o caractere \w significa uma sequência de letras, números ou sublinhado; a etiqueta IN indica preposição; NN indica substantivo no singular; NNP indica nome próprio; NNS indica substantivo comum no plural; e as etiquetas VBD e VBN indicam verbos.

```
(?:[\w]*_IN) (?:[\w\-/\\]* )?([a-z]{7,12}_NN|[a-zA-Z]{2,3}_NNP|[a-z]{7,12}_NNS) (?:treatment_NN|therapy_NN)
(?:were_VBD|was_VBD|while_IN|went_VBD|while_NN)(?:[\w\-/\\]* on_IN) ([\w\-]*)
```

Figura 10 Exemplo de regra da Estratégia 1.
Fonte: extraída de Duque et al. (2011)

```
(?:[\w-/\\]*_NNS) (?:[\w-/\\]*_IN|[\w-/\\]*_VBD|[\w-/\\]*_VBN)* ([\w-/\\]*_NN|[\w-/\\]*_NNP|[\w-/\\]*_NNS) (?:[\w-/\\]*_NN)?
```

Figura 11 Exemplo de regra da Estratégia 2.
Fonte: Duque et al. (2011).

A abordagem de Extração de Informação baseada em Dicionário reconhece quais sentenças apresentam termos válidos. Os termos validados foram consolidados manualmente por um especialista. Segundo Duque et al. (2011), o dicionário possui duas características: a primeira possui a capacidade de armazenar termos extraídos dos artigos provenientes do processo de Extração de Informação, e a segunda a de identificar sentenças que possuem termos existentes no dicionário.

Para validar a metodologia proposta foram realizados vários experimentos. Em geral, os melhores resultados atingiram taxas de 96% de precisão, 19% de cobertura e 31% de medida F. Em outros experimentos, a cobertura foi de 100% na identificação de termo distintos, representando uma extração completa dos termos relevantes existentes no corpus testado.

3.2.3 Extração automática de relações semânticas a partir de textos escritos em Português variante brasileira

A dissertação de Taba (2013) teve por objetivo verificar como e quais métodos automáticos poderiam ser aplicados para a extração de relações semânticas em textos escritos em Português.

O trabalho apresentou enfoque em duas principais abordagens: 1) abordagem de Padrões Textuais; e 2) abordagem de Aprendizado de Máquina com diferentes classificadores (árvores de decisão C4.5 e *Support Vector Machines* – SVMs).

A primeira abordagem baseia-se nos trabalhos de Freitas e Quental (2007) e Hearst (1992), sendo utilizada para encontrar instâncias da relação de hiponímia (*is-a*). A segunda abordagem utiliza *corpora* anotados e algoritmos de Aprendizado de Máquina para encontrar as sete relações de interesse, descritas na Figura 12.

	Relação semântica	**Sentença exemplo**	**Relação extraída**
1	location-of(algo/alguém, local)	Uma secretária pode ser encontrada em um escritório	location-of(secretária, escritório)
2	is-a(subclasse, superclasse)	Maçã é uma fruta	is-a(maçã, fruta)
3	property-of(algo/alguém, característica)	O prédio é alto	property-of(prédio, alto)
4	part-of(todo, parte)	Parafuso é uma parte de uma máquina	part-of(máquina, parafuso)
5	made-of(produto, substância)	Cacau é utilizado para fazer chocolate	made-of(chocolate, cacau)
6	effect-of(ação/estado, consequência)	Gripe causa febre	effect-of(gripe, febre)
7	used-for(entidade, função)	Pás são usadas para cavar	used-for(pás, cavar)

Figura 12 Relações semânticas de interesse.
Fonte: extraída de Taba (2013).

3.2.3.1 Recursos

Para a tarefa de extração automática das relações semânticas com o uso de métodos de Aprendizado de Máquina, faz-se necessário um corpus de treinamento, anotado manualmente com as relações semânticas existentes entre os termos de cada sentença.

São utilizados dois *corpora* para o estudo. O primeiro se refere a artigos científicos da Revista *Pesquisa FAPESP*.[25] Esse corpus foi enriquecido com informações linguísticas inseridas com base nos dados do ReTraTos (CASELI, 2007, p. 186) e do PALAVRAS (BICK, 2000, p. 505).

O segundo corpus é composto de artigos provenientes do Jornal *Folha de São Paulo*. O corpus foi enriquecido com informações morfossintáticas pelo *parser* PALAVRAS. A Figura 13 mostra exemplos de sentenças do corpus da Revista, anotadas com algumas relações semânticas de interesse.

[25] Disponível em <http://revistapesquisa.fapesp.br/>.

Para auxiliar a tarefa de anotação manual de relações semânticas, foi desenvolvida uma ferramenta específica: o ARS (Anotador de Relações Semânticas). Essa ferramenta, desenvolvida em Java, realiza a manipulação de sentenças codificadas no formato JSON, permitindo a marcação visual de termos e de relações entre eles (TABA; CASELI, 2013). O ARS possui, como uma de suas funcionalidades, a conversão de documentos processados pelo *parser* PALAVRAS em um arquivo JSON, facilitando a manipulação do arquivo para a anotação de relacionamentos semânticos.

1. uma equipe da [universidade estadual de campinas]1 ([unicamp]2) conseguiu isolar e caracterizar pela primeira vez o [vírus respiratório sincicial bovino]3 ([brsv]4) no brasil , que causa graves problemas respiratórios sobretudo em bezerros . [is-a(1,2), is-a(3,4)]
2. uma nova [espécie de hominídeo]1 encontrado na [tailândia]2 , com estimados 12 milhões de anos , tornou - se o parente mais remoto dos atuais [orangotangos]3 ([pongo pygmaeus]4) . [location-of(1,2), is-a(3,4)]
3. mas só agora começam a reunir condições de responder à pergunta : quando é que virá o próximo el niño ? []

Figura 13 Sentenças anotadas com relações semânticas entre os termos.
Fonte: extraída de Taba (2013).

3.2.3.2 Experimentos com Padrões Textuais

Hearst (1992) foi uma das primeiras a utilizar padrões para encontrar relações semânticas, no caso a hiponímia. Além disso, a pesquisadora definiu um algoritmo iterativo para a descoberta de novos padrões e relações. Com base no trabalho de da autora, Freitas e Quental (2007) traduziram seus Padrões Textuais para o Português variante brasileira.

Inicialmente, por meio do corpus, foram definidos manualmente os 13 padrões para seis relações, apresentados na Figura 14. As notações “_N”, “_ADJ” e “_V” indicam que um termo deve ser um substantivo, um adjetivo ou um verbo, respectivamente. Apenas a primeira iteração do algoritmo de Hearst (1992) foi aplicada usando a base de dados do projeto OMCS-Br (advindo do projeto OMCS)

como semente. O resultado do algoritmo foram quarto novos Padrões Textuais, definidos na Figura 15.

Relação	#	Padrão
property-of	1	T1_N T2_ADJ
	2	T2_ADJ T1_N
	3	T1_N " T2_ADJ "
part-of	1	T1 com T2
	2	T1 {verbo fazer} parte de T2
	3	T1 {verbo ser} parte de T2
made-of	1	T1_N de T2_N
	2	T1 (é|são)? feit(o|a|os|as) de T2
location-of	1	T1 entrou em T2
	2	T1 ,? localizad(a|o) em T2
effect-of	1	T2_V .* devido=a T1
	2	T2_V por=causa=de (a|o|as|os)? T1
used-for	1	T1 (que podem ser)? usadas? para T2_V

Figura 14 Padrões definidos para as seis relações semânticas.
Fonte: extraída de Taba (2013).

Relação	#	Padrão
property-of	4	de T1_ADJ T2_N
part-of		–
made-of		–
location-of	3	T1 chega a o T2
	4	T1 em (o|a|os|as) T2
effect-of		–
used-for	2	T1 para (o|a|os|as) T2_V

Figura 15 Padrões definidos a partir de uma iteração do algoritmo de Hearst.
Fonte: extraída de Taba (2013).

A precisão obtida na extração de instâncias da relação *is-a* no corpus do Jornal *Folha de São Paulo* foi de 61,1%; a média da precisão utilizando os 24 padrões foi de 36,5% e a cobertura de 18,2%. Diante dos resultados, é afirmado que os padrões textuais tiveram boa precisão, mas baixa cobertura.

3.2.3.3 Experimentos com Aprendizado de Máquina

Os experimentos desenvolvidos fazem o uso de modelos de Aprendizado de Máquina (AM) supervisionado. O experimento emprega algoritmos de árvore de decisão e consistiu no treinamento do algoritmo de árvore de decisão C4.5

(QUINLAN, 1993, p. 302) sobre as cerca de 100 mil instâncias de treinamento do corpus da *Folha de São Paulo*. A precisão média foi de 54,4% e, quanto à cobertura média, obteve-se a taxa de 30,4%. Diante desses resultados, pode-se afirmar que árvores de decisão possuíram melhores valores em comparação com Padrões Textuais.

O segundo algoritmo de AM escolhido foram as *Support Vector Machines* (SVMs) (VAPNIK, 2000). A precisão, cobertura e medida F médias obtidas com SVMs foram 61,6%, 39,2% e 47,9%, respectivamente. Consequentemente, os valores médios obtidos com as SVMs também foram maiores do que os obtidos com Padrões Textuais. Por fim, Taba (2013) construiu o gráfico apresentado na Figura 16, no qual são comparados os resultados da medida F para os experimentos de Padrões Textuais, árvores de decisão e SVMs. O autor conclui que, na maioria dos casos, o método por SVMs obtém melhores resultados.

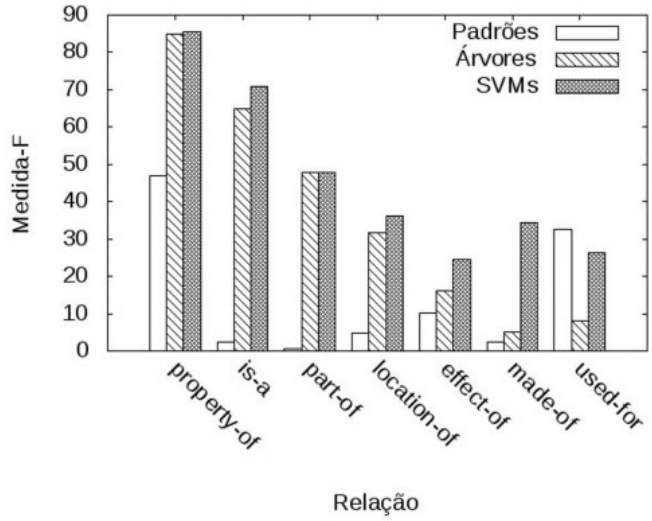


Figura 16 Desempenho em medida F para três experimentos.
Fonte: extraída de Taba (2013).

3.3 Construção de ontologias por meio de documentos textuais

Essa seção apresenta trabalhos que realizam a descoberta de novas relações semânticas em documentos textuais com intuito de promover a construção/extensão de ontologias.

3.3.1 Extração de relacionamentos semânticos para o enriquecimento de ontologia de domínio

Shen et al. (2012) propõem uma abordagem semisupervisionada para a extração de relações semânticas de um domínio por meio de documentos textuais no idioma inglês. O objetivo do estudo foi o de encontrar novas relações semânticas para ontologias. Inicialmente, a abordagem faz o uso de uma ontologia base. A abordagem adiciona manualmente as novas relações semânticas, extraídas dos documentos textuais. Os autores descreveram um *framework*, o qual é dividido em quatro partes e pode ser visualizado na Figura 17.

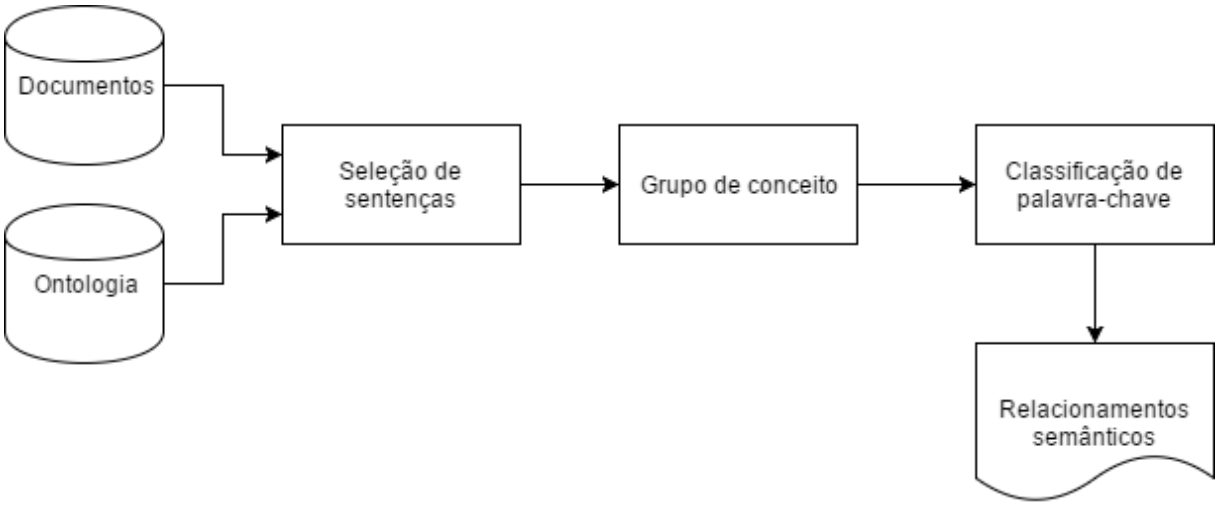


Figura 17 Descrição do ***framework*** de enriquecimento.
Fonte: adaptada de Shen et al. (2012).

Na etapa “Seleção das Sentenças”, realiza-se a segmentação de sentenças em conceitos. Cada sentença é dividida e em seguida etiquetada, associando entidades que correspondem a algum conceito ontológico. Os conceitos que não se encaixam na ontologia são etiquetados pela técnica *Part-Of-Speech* (POS). Ou seja, palavras são anotadas com *tags part-of-speech* (verbos ou substantivos) ou conceitos ontológicos. Por fim, as sentenças devem conter pelo menos dois conceitos ontológicos e um verbo. As sentenças que não possuírem essas caraterísticas são descartadas.

Nos grupos de contexto, as sentenças que determinam o contexto de cada par de conceitos descobertos são transformadas em vetores para o agrupamento. Para cada vetor, as palavras passam por um dicionário de sinônimos, em busca de

palavras de um mesmo significado. Por fim, os vetores são aplicados no algoritmo *Hierarchical Agglomerative Clustering* (HAC), para a criação de grupos que representam a relação semântica de um par de conceitos (MANNING; RAGHAVAN; SCHÜTZE, 2009).

A etapa de "Classificação das palavras-chaves" refere-se à recomendação de palavras-chave adequadas para a rotulagem das relações semânticas dos pares de conceitos. As palavras-chave candidatas foram obtidas na etapa anterior, e a classificação é feita por meio da atribuição de pesos para a palavra-chave. Dois esquemas de pesos são aplicados: *Term Frequency and Inverse Cluster Frequency* (TFICF) e *Child Voting* (CV).

As palavras-chave candidatas são classificadas usando combinação linear baseadas em pesos para rotular as relações semânticas. O esquema TFICF estima a importância de uma palavra em um grupo e sua discriminação ao longo do grupo. "TF" é a ocorrência de uma palavra em um grupo, e "ICF" é o número de grupos em que a palavra aparece.

O experimento descobriu relações semânticas desconhecidas, as quais foram validadas por um especialista de domínio. O desempenho da extração da relação semântica é validado em termos de acurácia, que representa o número de pares de conceito obtidos em função do número de ocorrência dos rótulos previstos. Por fim, Manning, Raghavan e Schütze (2009) fazem uma comparação utilizando os métodos TF, TFICF e TFICF+CV, onde TFICF+CV obteve melhor acurácia (em torno de 74%).

3.3.2 GRAONTO: uma abordagem baseada em grafos para a construção de ontologia de domínio

Hou et al. (2011) propuseram uma abordagem baseada em grafos para a construção automática de ontologias de domínio a partir de documentos formais no idioma inglês. O diferencial dessa metodologia é a construção a partir do zero, ou seja, sem o auxílio de uma ontologia base, tendo como entrada documentos textuais. Nela se destaca, ainda, a representação do conteúdo textual por grafos.

Os autores dividiram o *framework* em cinco etapas, que podem ser observadas na Figura 18.

Na etapa de pré-processamento dos documentos são retirados os termos que são considerados irrelevantes (processo "*stop words removel*"); os termos são reduzidos à sua forma normal (processo "*text stemming*"), aplica-se o "*Part-of speech (POS tagging)*" para identificação de palavras como substantivos, verbos, adjetivos, advérbios etc. e, por fim, é realizada uma estatística sobre o documento.

Na etapa de representação dos documentos em grafo, para cada texto é gerado um grafo, cuja estrutura é definida por uma quádrupla: G = (V, E, α, β), onde V é o conjunto de vértices (ou nó), E é o conjunto de arestas conectando os vértices, α é a função de rotulagem dos vértices e β é a função de rotulagem das arestas.

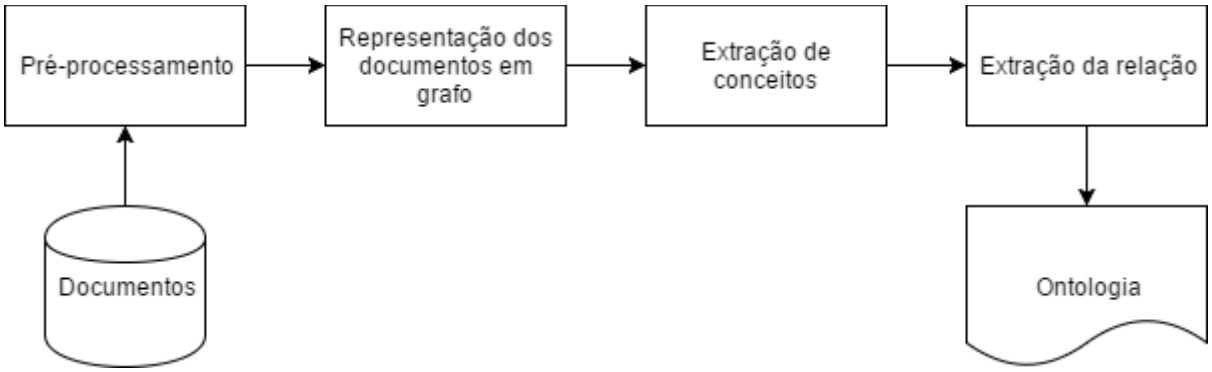


Figura 18 Arquitetura do GRAONTO.
Fonte: adaptada de Hou et al. (2011).

Cada vértice é rotulado com o termo que representa. A aresta conecta dois termos, indicando que ambos são adjacentes no documento. Cada vértice e cada aresta são rotulados com a frequência dos termos, o que representa o número total de ocorrências de vértices e arestas. Finalizando essa etapa, os valores dos grafos são normalizados em um intervalo [0,1].

A etapa de extração de conceito consiste em duas partes: a primeira, nomeada por pesagem dos termos, é calculada pelo caminho randômico em um grafo. A segunda parte utiliza o algoritmo *Markov Cluster* (MCL) (DONGEN, 2000) para agrupar os termos conforme seu peso para o documento.

O caminho randômico para o cálculo do termo é um procedimento iterativo, e a pontuação de cada vértice é atualizada a cada iteração, levando em consideração os novos pesos de seus vértices adjacentes. O procedimento é executado até que todos os vértices convirjam para um limiar pré-definido.

Por fim, o algoritmo MCL é aplicado, para criar uma matriz de *Markov* a partir do grafo. O processo de expansão e inflação é aplicado na matriz até que ela seja

uma matriz idempotente.[26] A partir da obtenção de uma matriz duplamente idempotente, o grafo é finalmente segmentado em diferentes grupos.

A extração da relação para construção de ontologias pode ser convertida no problema de descoberta de subgrafos frequentes dentro de grafos, levando em consideração as informações do subgrafo. Os autores fazem o uso do algoritmo *gSpan* (YAN; HAN, 2002) para descobrir os subgrafos frequentes. Nesse algoritmo foi adicionada uma função para estimar a capacidade informativa de um subgrafo. Após a mineração do subgrafo frequente, os subgrafos obtidos são interpretados como as relações entre os conceitos. As relações são representadas por triplas {conceito1, relação, conceito2}.

Para a validação da abordagem foram usados documentos de teste de domínios, os quais não foram utilizados para construção da ontologia. Caso ocorram contradições, ou seja, caso haja erros ou conflitos na ontologia construída, ela precisa ser melhorada. Assim, os conceitos ou relações que contribuem para as contradições são alimentados de volta para o sistema, e os documentos originais e os documentos de teste de domínio são usados em conjunto para refinar a ontologia. Os processos descritos são iterados até que nenhum erro ou conflito ocorra.

Os autores propuseram dois experimentos para o teste do GRAONTO. O primeiro examina a eficiência da extração dos conceitos e a extração da relação comparada com os métodos tradicionais. O segundo experimento se refere à comparação entre a ontologia criada pela ferramenta e a ontologia criada manualmente por um especialista. O resultado do primeiro experimento demonstra que a abordagem proposta obteve melhores resultados ao ser comparada com os seguintes métodos: TF-IDF, SIGNUM e TF-IDF-ART. Isto porque GRAONTO realiza a pesagem dos termos em perspectivas locais e globais; um algoritmo de agrupamento em grafo é usado para agrupar e retirar a ambiguidade de termos semelhantes, o que contribui para a melhoria da precisão.

Com relação à extração da relação, o GRAONTO obteve os melhores resultados ao ser comparado com os outros três métodos. Isto porque ele considera o problema de extração de relação como um problema de mineração de subgrafo informativo frequente. Uma função de informação é introduzida para avaliar o quão informativo é um subgrafo e, com base nisto, o desempenho de GRAONTO em

[26] *Matriz idempotente* é uma matriz que, ao ser multiplicada por si mesma, resulta em si mesma.

comparação com a ontologia criada manualmente é o seguinte: a precisão é da ordem de 71% e revocação é da ordem de 77%.

3.3.3 Uma abordagem automática para a construção de ontologias expressivas a partir de linguagem natural

Azevedo et al. (2014) propuseram uma abordagem para construção de ontologias expressivas por meio da linguagem natural. Esta abordagem consiste na utilização de um método híbrido, que combina análise sintática e semântica. Os autores demostraram um tradutor para a criação de ontologias que formaliza e codifica conhecimento em OWL DL $\mathcal{ALC}$ (HORROCKS et al., 2007).

A arquitetura proposta por Azevedo et al. (2014) é composta de três módulos: análise sintática, análise semântica e axiomas OWL DL.

Para o módulo de análise sintática é usada uma *Probabilistic Context-Free Grammar* (PCFG, Gramática Livre de Contexto Probabilística), para a realização da análise sintática nas sentenças fornecidas para o sistema. Este módulo realiza duas atividades: rotulagem léxica e análise de dependência entre os termos (PCFG). O módulo da análise semântica possui quatro etapas: extração do termo, concatenação, quebra de frases e extração da relação.

Na etapa de extração de termos, as palavras que são classificadas como preposições, conjunções, números, artigos e verbos são descartadas, e as classificadas como substantivos e adjetivos são indicados como possíveis conceitos da ontologia.

A etapa de concatenação realiza a junção dos termos com base nos resultados de dependência entre os termos analisados na etapa anterior.

Já na etapa de quebra de frases, para todo termo ou sinal de pontuação encontrados – como vírgula (,), ponto (.), “e”, “ou”, “que”, “quem” ou “o que” (disjuntores de frases) –, as sentenças são divididas em subsentenças e analisadas separadamente.

Na etapa de extração da relação, as relações entre os termos são verificadas e validadas por meio de verbos encontrados nas frases e padrões observados. Os verbos são separados, e os termos dependentes dos verbos extraídos.

O módulo de axiomas OWL DL possui o princípio de encontrar/aprender axiomas que impedem interpretações ambíguas e de limitar as possíveis

interpretações do discurso. O módulo reconhece conjunções (“e” e “ou”), indicando união (disjunção) e interseção (conjunção), respectivamente, para os conceitos e/ou propriedades, bem como reconhece palavras ligadas a verbos seguidos de negações, para axiomas de negação (¬). Por fim, ele também reconhece “é” e “são” como relações taxonômicas (⊑ – hierárquicos). As transformações ocorrem em quatro passos e fazem uso dos resultados obtidos pelos módulos anteriores.

Para a avaliação da abordagem, os autores do trabalho utilizaram 120 frases. Foram obtidos os seguintes resultados: em 75% das sentenças analisadas (90 sentenças), o tradutor detectou e criou de forma coerente os axiomas, enquanto que em 30 sentenças (25%, incluindo também as que o tradutor não poderia resolver de qualquer maneira). O tradutor não detectou os axiomas de forma coerente e cometeu erros. No entanto, em 24 das 30 sentenças, o tradutor criou axiomas e tornou possível a recriação das ontologias através do processo de inserção de novas definições.

3.3.4 Construção de estruturas ontológicas a partir de textos: um estudo baseado no método *Formal Concept Analysis* e em papéis semânticos

Moraes (2012) propôs em seu trabalho uma nova metodologia, denominada *Semantic FCA* (SFCA), para a construção de estruturas ontológicas a partir de textos utilizando estruturas *Formal Concept Analysis* (FCA) baseadas em papéis semânticos. Por meio do método de agrupamento do FCA, a autora tem por objetivo combinar o método FCA com papéis semânticos para construir, de forma automática e a partir de informações textuais, estruturas ontológicas baseadas em relações não taxonômicas. O método FCA, quando comparado a outros métodos de agrupamento, permite delinear, do ponto de vista semântico, os grupos e subgrupos de uma hierarquia.

Como premissa da metodologia, o corpus deve ser anotado com informações léxico-semânticas, por meio de um etiquetador de papéis semânticos e um anotador de *POS tagger*.

Antes de apresentar os passos para metodologia, é necessário o entendimento de papel semântico. A autora afirma que papéis semânticos expressam a relação semântica entre um verbo e seus argumentos. Essa relação de

dependência nas estruturas predicado-argumento pode ser facilmente percebida analisando-se as sentenças enumeradas a seguir:

1. [Lucas $_{Agente}$] quebrou [a janela $_{Paciente}$].
2. [A janela $_{Paciente}$] quebrou.
3. [Lucas $_{Agente}$] abriu [a porta $_{Paciente}$].
4. [A chave $_{Instrumento}$] abriu [a porta $_{Paciente}$].

Onde o “Agente” é associado ao sujeito da sentença e corresponde a uma entidade, tipicamente humana ou pelo menos animada, que provoca uma ação de forma voluntária. Já “Paciente” é uma entidade diretamente afetada por uma ação, mudando o seu estado (animado ou inanimado). Por fim, “Instrumento” é objeto inanimado que participa de forma secundária da ação, sendo também uma causa do evento descrito pelo verbo.

A autora descreve sua metodologia (SFCA) em nove passos:

1. A metodologia se inicia normalizando morfologicamente o corpus.
2. Em seguida, devem ser analisadas as sentenças, identificando os verbos e extraindo os argumentos desses verbos e os papéis semânticos associados a esses argumentos. As sentenças cujos argumentos de verbos não tenham informações úteis perante o processo de anotação devem ser descartadas;
3. Identificar os sintagmas nominais existentes, por meio dos substantivos comuns anotados como papéis semânticos;
4. Formar tuplas usando as informações extraídas das sentenças nos passos dois e três, seguindo o formato: (sn_1, ps_1, sn_2, ps_2), onde sn_i e ps_i correspondem, respectivamente, ao sintagma nominal e ao seu respectivo papel semântico. Nas tuplas, os sintagmas nominais devem ser formados pelos lemas de seus substantivos. Cada tupla deve é formada por dois (sintagmas nominais) cujos papéis semânticos foram atribuídos por um verbo, em uma mesma sentença. Em uma mesma sentença, se o verbo possuir mais de dois argumentos anotados semanticamente, as duplas devem ser formadas perante as combinações possíveis. Caso apenas o verbo possua anotação, a sentença deve ser descartada.

5. Construir as tuplas de (objeto, valor) seguindo o formato (sn_1; ps_1_sn_2) e (sn_2; ps_2_sn_1). Os pares devem ser inseridos em uma lista, e suas frequências devem ser contabilizada.
6. Devem ser selecionados os pares mais significativos. Os pares mais significativos se dão pela frequência dos pares.
7. Aplicar técnicas de agrupamentos para os atributos quando esses forem muito específicos, a fim de evitar um contexto formal muito esparso (como, por exemplo, o coeficiente de Dice).
8. Construir o contexto formal usando os pares (objeto; atributo) resultantes dos passos 5, 6 e 7.
9. Gerar a estrutura SFCA a partir do contexto formal. O reticulado de pode ser formado a partir de algoritmos de agrupamento ou de uma ferramenta.

Como forma de avaliação de sua abordagem, a autora utilizou dois *corpora* presentes em Wikicorpus 1.0. O primeiro refere-se ao corpus WikiFinance, que possui 482 textos do domínio “Finanças”, e o segundo corpus, WikiTourism, que contém 442 textos do domínio “Turismo”. Ambos os corpus passaram pelos processos anotados com papéis semânticos ao estilo PropBank pelo processador F-EXT-WS, pelo *POS tagger* e também foram normalizados pelo lematizador e etiquetados pelo *TreeTagger*.

Por fim, a autora comparou sua abordagem com relação ao algoritmo k-NN. Em sua abordagem (baseada em conceitos formais), ela obteve como resultado as seguintes taxas: as regras baseadas em conceitos formais produziram 0,92 na média F1, e o algoritmo k-NN, 0,95.

3.4 Considerações finais

Neste Capítulo foram apresentados os trabalhos correlatos à atual pesquisa, discutindo estudos que possuem maior grau de similaridade com a proposta. Os três primeiros trabalhos são relacionados à extração de relacionamentos semânticos a partir de textos. No que se referem aos últimos três trabalhos, os autores e autora estão preocupados com a extração dos relacionamentos e, posteriormente, com a

construção da ontologia. A **Tabela 3** possui um quadro comparativo dos trabalhos correlatos contidos neste Capítulo.

Diante dos trabalhos que realizam a extração de relacionamentos semânticos por meio de padrões textuais, ressalta-se que contribuíram diretamente para o desenvolvimento do presente trabalho.

Tabela 3 Tabela-resumo dos trabalhos relacionados.

(Autores(as))	Pré-processamento	Extração de relacionamentos	Consideração da ontologia
(SHEN et al., 2012)	Segmentação de sentenças, *Part-of-Speech* (*POS tagging*)	TFICF, *Child Voting*	Manual
(HOU et al., 2011)	Remoção *stop words*, *text stremming*,	Mineração de subgrafos frequentes (gSpan)	Automática
(AZEVEDO et al., 2014)	*Lexical tagging*	Regras e Construção de axiomas	Automática
(TABA, 2013)	PALAVRAS; ReTraTos; ARS	Regras e Aprendizado de Máquina (AM)	Não realiza
(DUQUE et al., 2011)	Classificação de sentenças (Aprendizado de Máquina) *Part-of-Speech* (*POS tagging*)	Regras e Dicionários	Não realiza
(SCHEICHER, 2013)	*Part-of-Speech* (*POS tagging*)	Dicionários e Regras	Não realiza
(MORAES, 2012)	*Part-of-Speech* (*POS tagging*), lematizador, anotador papel semântico	frequência e FCA	Automática

Fonte: elaboração própria.

CAPÍTULO 4

Abordagem semântica para incorporação de relacionamentos não taxonômicos em ontologias

As ontologias de domínio facilitam a organização, o compartilhamento e o reuso do conhecimento de domínio. Os métodos automatizados para a construção de ontologias são focados na identificação de relações taxonômicas.

A extração de conhecimento a partir de bases de dados textuais é uma das técnicas utilizadas para a construção de ontologias, extraindo conhecimentos úteis de documentos de texto para auxiliar o processo de descoberta de novas relações semânticas.

Este Capítulo apresenta a proposta do presente trabalho, mostrando os passos a serem realizados para atingir o objetivo final, que é a extração de novas relações não taxonômicas entre termos ontológicos. Propõe-se aqui, portanto, uma ferramenta de apoio ao especialista neste processo – considerado, nos dias atuais, um processo trabalhoso.

4.1 Definições

Antes de iniciar a descrição da pesquisa, ressaltam-se algumas definições utilizadas no presente texto.

- ***Token***: uma sequência de quaisquer caracteres, com exceção do espaço;

- **Termo**: uma sequência de *tokens* que representa uma entidade ou tem algum significado específico em uma sentença;
- **Termo ontológico**: corresponde a um termo de uma sentença, o qual pode ser encontrado na ontologia;
- **Relação semântica**: nesta pesquisa em específico, refere-se a uma tripla < to_1, *v*, to_2>, onde *v é* representa o verbo da relação, e to_1 e to_2 são termos ontológicos distintos em uma sentença.

4.2 Recursos

Para o desenvolvimento desta pesquisa de mestrado, foi necessária a utilização de alguns recursos textuais. Com a ajuda de especialistas de domínio da área da Educação Especial, obteve-se o corpus, que contêm um total de 290 documentos frutos do projeto ONEESP.

TEMA: ORGANIZAÇÃO DO ENSINO NAS SRMS E CLASSES COMUNS

1. **De modo geral qual é a função da escolarização para alunos com NEEs? O que a escola tem condições e oferecer a eles?**

Enicéia: Então hoje é o terceiro tema, terceiro eixo né dessa etapa primeiro de descrição e o tema hoje é sobre a organização do trabalho pedagógico né, do funcionamento dentro da sala de recursos. Também tem um, vocês já conhecem como que é já vou entrar nas questões aqui, ai tem uma questão. De um modo geral, qual é a função da escolarização para alunos com NEE o que a escola tem condições de oferecer a eles? Pra vocês né. Quem começa? A função da escolarização, pensando no conjunto dos alunos né desde os mais prejudicados aos mais leves.

Vanessa: Eu avalio que é importante pela questão da aprendizagem mesmo e mesmo que eles não acomp alguns deles né, não acompanham o conteúdo curricular correspondente a série, mas tem acesso a algum conteúdo a alguma aprendizagem correspondente a condição dele e também pela questão dele tá em contado com os pares, isso é importante pro, acaba sendo importante pro desenvolvimento dele também a interação social.

Figura 19 Exemplo de texto.
Fonte: acervo do Observatório Nacional da Educação Especial (ONEESP).

O corpus é constituído de documentos que servem de base para a identificação dos relacionamentos semânticos para a ontologia. Esses documentos são caracterizados como semiestruturados, em formato de texto, salvos em .doc e .docx. No estudo de caso adotado no presente trabalho, os documentos abordam o conteúdo relacionado à políticas públicas de Educação Especial de várias regiões do país. A estruturação de tais documentos corresponde a questionários, em que as respostas são apresentadas sob a forma discursiva, em linguagem coloquial, ou seja, não formal. Apresenta-se, na Figura 19, um exemplo de um dos textos desse corpus.

Outro recurso utilizado foi a taxonomia de uma ontologia criada a partir do mesmo corpus utilizado neste trabalho, ou seja, a partir de dados provenientes do projeto ONEESP. A taxonomia utilizada é a base de uma ontologia de domínio já desenvolvida por Fernandes (2016), em sua tese de Doutorado em Educação Especial na UFSCar. Um extrato dessa ontologia pode ser visualizado na Figura 20 visualizada a seguir.

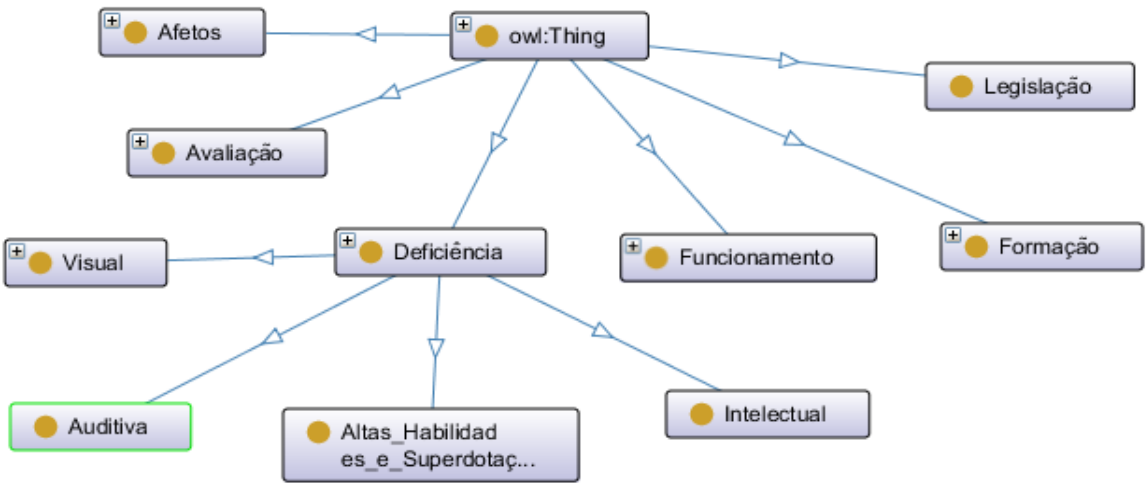


Figura 20 Um extrato da ontologia base.
Fonte: elaboração própria.

É possível notar que a ontologia apresenta, como raiz, o termo “Thing”, e suas subclasses são “Afetos”, “Avaliação”, “Deficiência”, “Formação”, entre outras. As subclasses da classe “Deficiência” são as classes “Visual”, “Auditiva”, “Altas Habilidades e Superdotação” e “Intelectual”.

4.3 Abordagem Embed$_{NT}$RelOnto

A abordagem para incorporação de relacionamentos não taxonômicos em ontologias (*approach to embedding non taxonomic relationships in ontologies* – EmbedRel$_{NT}$Onto) visa a extração de relacionamentos entre termos provenientes de uma ontologia preliminar, a partir de textos informais sobre Educação Especial e, posteriormente, a realização do enriquecimento da ontologia. Nesta dissertação, o conceito de abordagem refere-se a um conjunto de passos que são utilizados para se alcançar um objetivo comum.

A arquitetura do processo de extração de relacionamento em documentos pode ser visualizada Figura 21. A ideia principal é processar os documentos a fim de identificar termos ontológicos e, posteriormente, extrair o relacionamento existente entre eles, incorporando estes relacionamentos na ontologia.

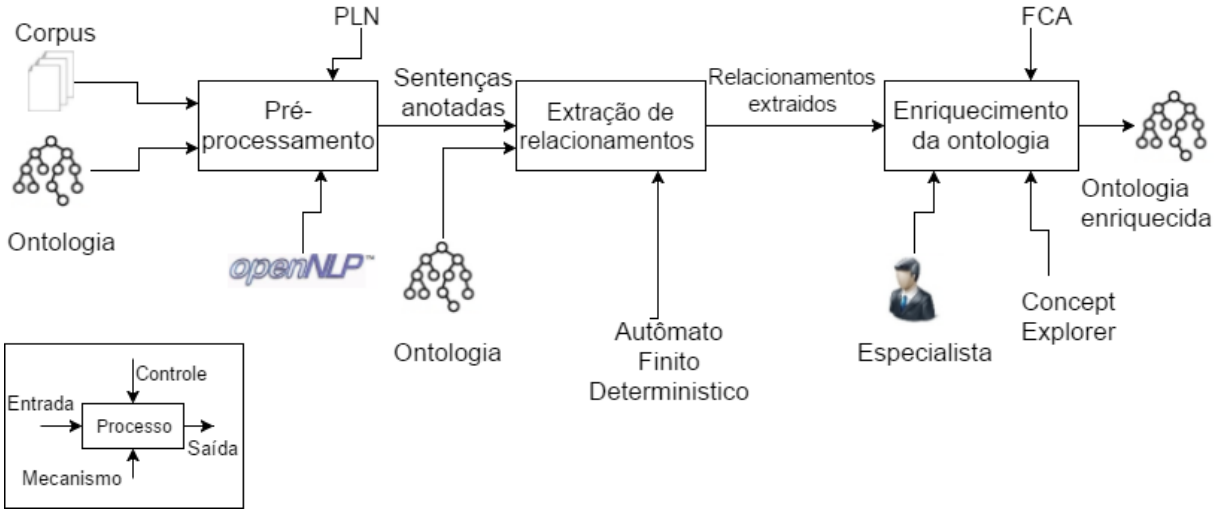


Figura 21 Arquitetura Embed$_{NT}$RelOnto.
Fonte: elaboração própria.

A representação das imagens referentes à arquitetura do processo descrito neste trabalho foi baseada na técnica de Ross (1977), intitulada *Structured Analysis and Design Technique* (SADT). No diagrama, de acordo com a caixa-legenda (Figura 21), os retângulos representam os processos da abordagem. As setas que entram pelo lado esquerdo dos retângulos representam as entradas de dados, e as setas que saem pelo lado direito representam as saídas geradas em cada etapa. As setas superiores representam os controles que orientam a execução de cada etapa.

Já as setas inferiores representam os participantes e as ferramentas que auxiliam a execução das etapas.

Como pode ser visualizado na Figura 21 em questão, a entrada de dados é composta de um corpus que, por sua vez, é composto de diversos documentos nos formatos .doc e .docx cujo domínio se refere à Educação Especial. Os documentos passam pela etapa de pré-processamento, a qual faz o uso da biblioteca OpenNLP para a execução das tarefas relacionadas ao Processamento da Linguagem Natural (PLN). A ontologia está presente no pré-processamento, a qual é utilizada na identificação dos termos ontológicos.

Após o pré-processamento, a etapa de extração de relacionamentos é executada. Nessa etapa, as sentenças anotadas são processadas com o objetivo principal de encontrar e extrair os relacionamentos semânticos entre termos ontológicos não taxonômicos das sentenças. Esse processo de extração é baseado em um autômato finito. A saída dessa etapa contém os relacionamentos extraídos, os quais são salvos em um banco de dados, juntamente relacionados as suas respectivas sentenças.

A etapa de pós-processamento é responsável por auxiliar o especialista ou engenheiro de ontologias no processo de construção. Essa etapa é apoiada pelo método FCA, que fornece um agrupamento gráfico dos relacionamentos extraídos. E, por fim, os relacionamentos são inseridos na ontologia por meio do “gerador de relacionamentos ontológicos”. O resultado final de todo o processo é a ontologia enriquecida, ou seja, a ontologia constituída com os relacionamentos extraídos.

A construção do protótipo que implementa a abordagem EmbedRel$_{NT}$Onto foi inteiramente montada em Java.

4.4 Pré-processamento

A etapa de pré-processamento tem por objetivo preparar os documentos para o processo de extração de relacionamentos. Esta etapa pode ser visualizada na Figura 22. As tarefas realizadas no pré-processamento são as seguintes: limpeza, sentenciação, tokenização, etiquetação, lematização e identificação de termos ontológicos. Todos os passos da etapa de pré-processamento são executados em

ordem sequencial, e tanto a sentenciação quanto a tokenização e a etiquetação são executadas com o auxílio de ferramentas OpenNLP. Para etapa de identificação de termos ontológicos é necessária uma taxonomia preliminar para a extração das classes nela definidas. Como exemplo, para todo o processo descrito neste texto, utiliza-se a seguinte sentença: “Os alunos e professores (inaudível 56:57 – 56:59) frequentam a sala.”

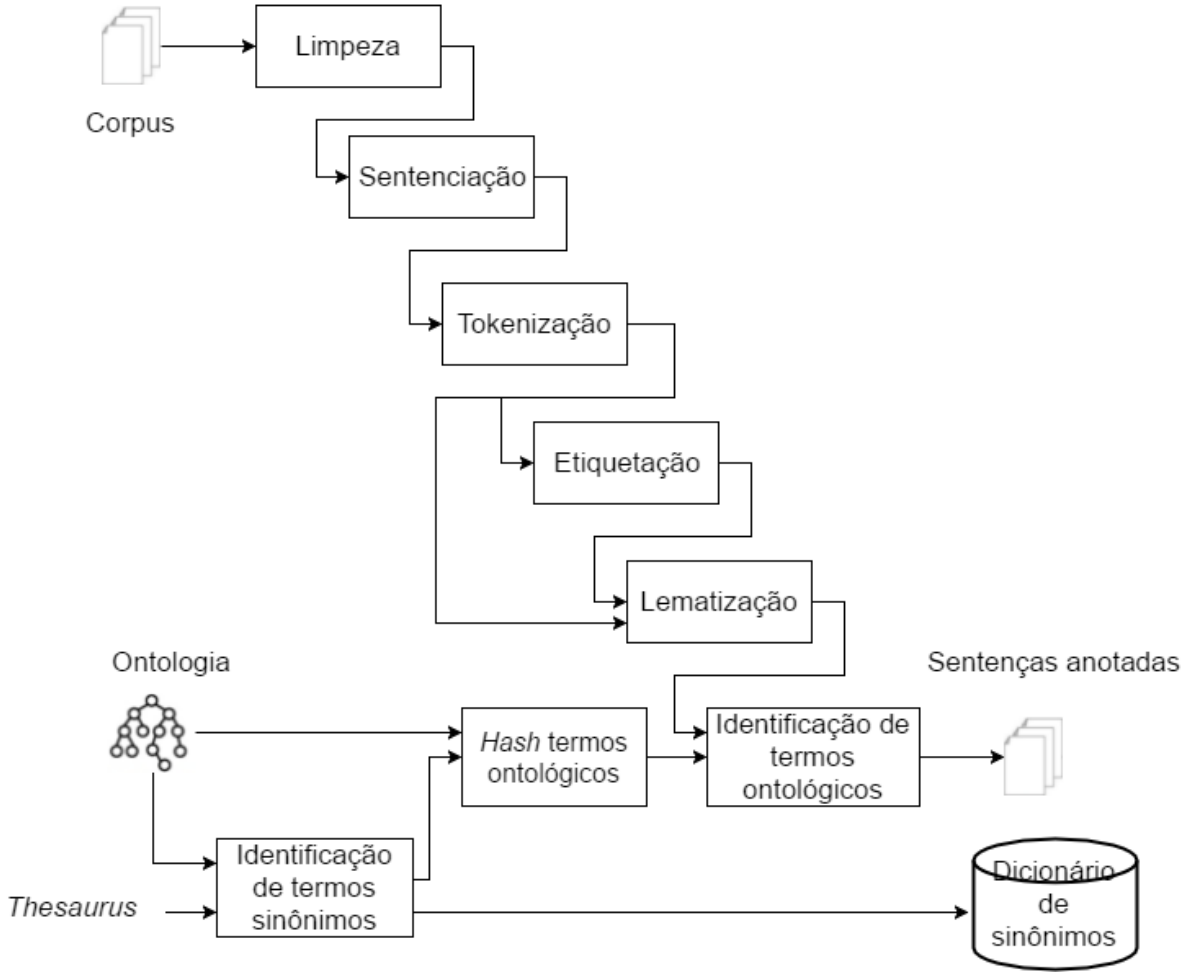


Figura 22 Pré-processamento.
Fonte: elaboração própria.

4.4.1 Limpeza

A etapa de limpeza consiste em aplicar um filtro com o intuito de remover as marcações feitas nas transcrições de áudio, as quais não são importantes para a extração. As marcações são informações referentes, dentre outros aspectos, ao não entendimento de um determinado trecho do áudio, à indicação de uma entonação no

discurso ou a uma sobreposição de discurso. Essas marcações são feitas por meio de variações de uso de parênteses e colchetes. Ou seja, o filtro removerá qualquer conteúdo que esteja dentro de parênteses ou colchetes. Diante da sentença-exemplo, mencionada anteriormente, a subsentença "(inaudível 56:57 – 56:59)" foi removida, obtendo-se como resultado "Os alunos e professores frequentam a sala". Ou seja, foi removida a marcação feita pelos especialistas.

4.4.2 Sentenciação

O próximo passo da etapa de pré-processamento refere-se à sentenciação. A sentenciação tem como objetivo fragmentar o texto em sentenças. A partir de um documento, a sentenciação consegue fragmentar o texto em parágrafos e, conseguintemente, em sentenças. Em geral a sentenciação busca encontrar segmentos de texto que contêm caracteres de pontuação referentes ao fim de sentenças, tais como: "." (ponto-final), "?" (ponto de interrogação) e "!" (ponto de exclamação). A sentenciação tem como entrada o documento e como saída uma lista de sentenças. Esta etapa é executada por meio da ferramenta OpenNLP. Diante do exemplo, o resultado da sentenciação será igual ao dado de entrada, uma vez que existe apenas uma sentença.

4.4.3 Tokenização

Já a tokenização tem como finalidade seccionar as sentenças em unidades mínimas, chamadas de *tokens*. A tokenização tem como entrada as sentenças obtidas pela sentenciação, gerando como saída uma lista de *tokens*. Esta tarefa também é provida pela ferramenta OpenNLP.

Diante da sentença-exemplo, é obtido o seguinte resultado: <Os> <alunos> <e> <professores> <frequentam> <a> <sala> <.>. Ou seja, a sentença é fragmentada em *tokens*, formando uma lista de *tokens*;

4.4.4 Etiquetação

Em sequência é executado o processo de etiquetação. A etiquetação (*POS tagger*) tem como objetivo atribuir uma categoria morfossintática a cada *token* da

sentença processada. O processo de etiquetação também é executado pela ferramenta OpenNLP, e possui como entrada a sentença tokenizada, obtendo como saída uma lista de marcações.

As marcações atribuídas pela etiquetação são: n (nome), adj (adjetivo), v (verbo), pron-pers (pronome pessoal), ',' (vírgula), conj-c (conjunção coordenativa), conj-s (conjunção subordinativa), punc (pontuação), adv (advérbio), art (artigo), entre outras.

Diante da sentença-exemplo, obtém-se como resultado uma lista de etiquetas: <art> <n> <conj-c> <n> <v-fin> <art> <n> <punc>. A lista de *tokens* é processada e cada *token* recebe uma etiqueta. Uma melhor forma de visualizar o resultado é por meio da seguinte notação: <token(etiqueta)>. Exemplo: "<Os(art)> <alunos(n)> <e(conj-c)> <professores(n)> <frequentam(v-fin)> <a(art)> <sala(n)> <.(punc)>".

4.4.5 Lematização

O processo seguinte é o de lematização, que tem como objetivo a transformação de cada *token* em sua forma simples. A lematização possui como entrada a lista de *tokens*, juntamente com suas etiquetas. O processo foi realizado com o apoio da ferramenta LemPORT (RODRIGUES; GONÇALO OLIVEIRA, HUGO GOMES, 2014).

Diante da sentença-exemplo, obtém-se como resultado uma lista de termos lematizados: <o> <aluno> <e> <professor> <frequentar> <o> <sala> <.>

4.4.6 Identificação de termos sinônimos

A etapa de identificação tem como objetivo identificar termos sinônimos que complementam os termos ontológicos, criando assim um dicionário de sinônimos.

Para a construção do dicionário, foi usado o *thesaurus* eletrônico para o Português variante brasileira, o Tep 2.0,[27] desenvolvido pelo NILC. O processo de criação foi feito manualmente, e os termos da ontologia foram usados como base no Tep, obtendo como resultado os sinônimos dos termos pesquisados.

[27] Disponível em <http://www.nilc.icmc.usp.br/tep2/busca.php>.

O dicionário de sinônimos é formado por meio de termos contidos na ontologia e seus sinônimos, em que o dicionário é construído diante de uma estrutura chave-valor. Todos os termos ontológicos e sinônimos são transformados em chaves, e o valor do termo ontológico é o próprio termo ontológico; já o termo sinônimo possui, como valor, o termo ontológico. Por fim, os termos sinônimos são também inseridos na lista de termos extraídos.

Como por exemplo, o termo Ontológico “criança” possui como sinônimos “menino”, “menina”. Logo, o dicionário é representado por:

criança -> criança
menino -> criança
menina -> criança

4.4.7 Identificação de termos ontológicos

A etapa de identificação de termos ontológicos objetiva identificar os *tokens* que são considerados termos ontológicos. Termos ontológicos são palavras ou *tokens* que constituem a ontologia. Ou seja, essa etapa é responsável por identificar quais *tokens* das sentenças correspondem aos termos da ontologia.

A identificação de termos ontológicos das sentenças utiliza um *set* de termos ontológicos, o qual é formado pelos termos da ontologia e de dicionários de sinônimos. A criação do *set* de termos ontológicos corresponde a extrair os termos da ontologia por meio da API Jena, que possibilita a manipulação de ontologias no Java. Além dos termos da ontologia, são incorporados os termos sinônimos do dicionário de sinônimos, o qual é descrito na seção 4.4.6, em que os termos sinônimos também são considerados termos ontológicos.

O processo de identificação de termos ontológicos necessita, como entrada, o *set* de termos ontológicos e a sentença lematizada. A sentença lematizada é processada *token* a *token*, consultado o *token* em análise no *set* de termos ontológicos. Caso o *token* esteja contido no set, ele é etiquetado como termo ontológico.

Diante da sentença-exemplo, são identificados como termos ontológicos os seguintes termos: <aluno>, <professor> e <sala>. Internamente, a aplicação marca

os *tokens* (alunos, professores e sala) como uma *flag*, com “verdadeiro” para termo ontológico e “falso” para termo não ontológico.

4.4.8 Estrutura criada para conter as informações processadas

A estrutura criada para representar a etapa de pré-processamento de uma sentença baseia-se na transformação da sentença em uma lista de *tokens*, em que cada *token* contém as seguintes informações:

- *Token* = termo;
- Etiqueta = etiqueta atribuída ao *token* pelo etiquetador (*POS tagger*);
- *Lemma* = lema atribuído ao *token;*
- Termo ontológico = *flag* com verdadeiro ou falso.

Como forma de demostrar a estrutura de um *token*, assume-se o *token* “aluno”. Logo, “aluno” contém as seguintes informações:

- Token: alunos;
- Etiqueta: n;
- *Lemma*: aluno;
- Termo ontológico: verdadeiro.

4.5 Método de extração de relacionamentos semânticos entre termos ontológicos

Após o pré-processamento são obtidas, como resultado, as sentenças com *tokens* etiquetados, lematizados e a identificação de termo ontológico. A etapa de processamento corresponde à extração dos relacionamentos semânticos não taxonômicos entre termos ontológicos. Esse processamento é automático, decorrente de um autômato finito determinístico. Na Figura 23 pode ser observada a arquitetura que representa o processo de extração de relacionamentos semânticos.

O resultado final obtido nessa etapa são os relacionamentos extraídos e ponderados, os quais são armazenados no banco de dados.

Com o intuito de proporcionar uma melhor extração dos relacionamentos semânticos, foi necessário definir algumas heurísticas:

- Sentenças compostas somente do verbo “Ser” são descartadas;
- Para sentenças que apresentam combinação verbal entre os termos ontológicos, deve-se extrair o último termo da combinação verbal. Por exemplo: “Os alunos estão comparecendo na sala”. O relacionamento extraído é: (Aluno, comparecer, Sala).
- Identificar termos compostos, como é o caso de “sala” e “sala de recursos”.

4.5.1 Filtro de sentenças que contenham termos ontológicos

Primeiramente, como entrada no processo, tem-se as sentenças processadas pela etapa de pré-processamento, estabelecido anteriormente. O filtro de sentenças tem por objetivo selecionar sentenças que apresentam termos ontológicos evitando, assim, um processamento desnecessário.

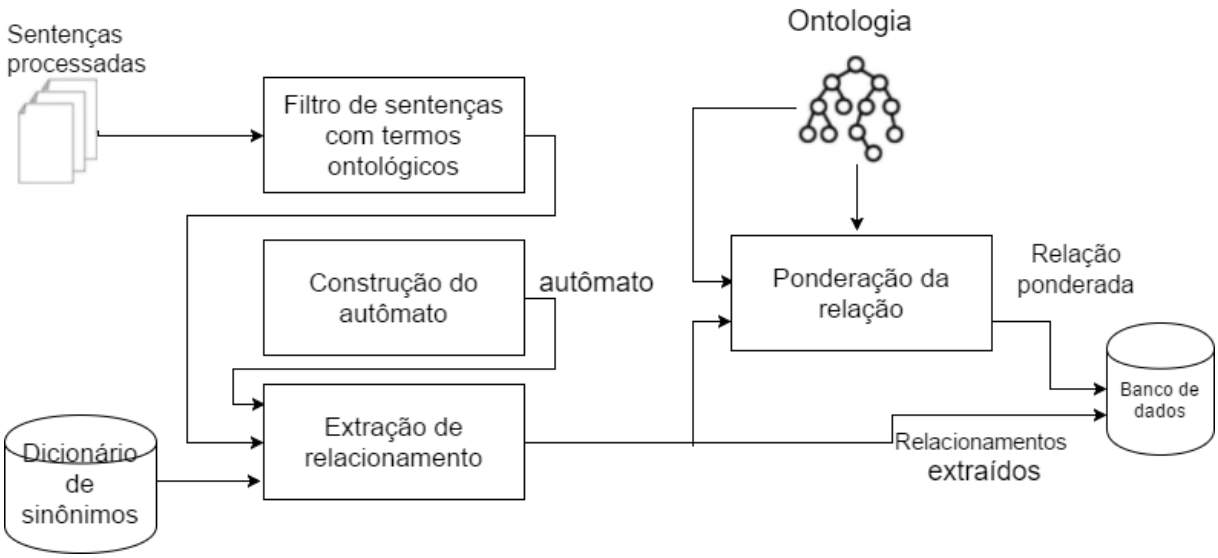


Figura 23 Processamento.
Fonte: elaboração própria.

4.5.2 Método SemanticExtr

A etapa de geração das regras de extração consiste na modelagem do autômato. Ou seja, na definição dos estados e do valor de aceitação de cada estado. Para definir essas regras não são necessárias as etiquetas morfossintáticas; o fluxo do autômato determinístico pode ser visualizado na Figura 24.

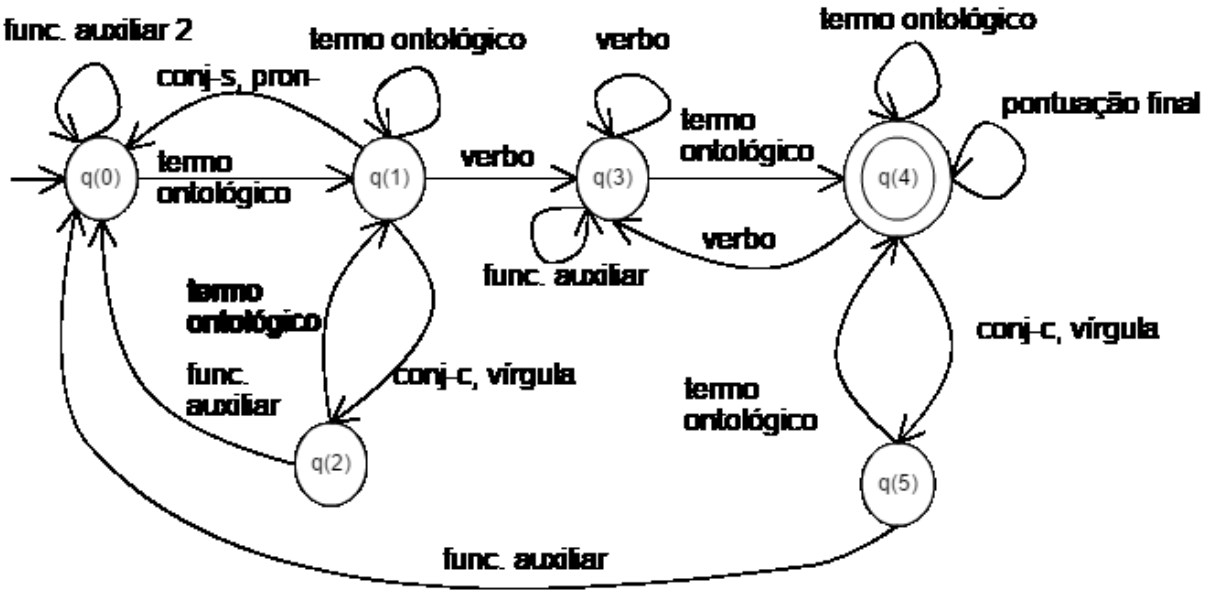


Figura 24 Definição do autômato determinístico.
Fonte: elaboração própria.

O autômato representado pela Figura 24 é definido formalmente como uma quíntupla: M = (Σ, Q, δ, q0, qf), onde:

- Σ – {'n', 'adj', 'v-', 'pron-pers', ',', 'conj-c', 'conj-s', 'punc', 'adv' e 'art'}
- Q – {q0, q1, q2, q3, q4 e q5}
- δ – Pode ser visualizada na Tabela 4
- q0 – q0
- qf – q4

Tabela 4 Descrição do fluxo do autômato.

	termo ontológico	**verbo**	**func. auxiliar**	**func. auxiliar 2**	**conj-c**	**vírgula**	**pont. final**
q(0)	q(1)			q(0)			
q(1)	q(1)	q(3)			q(0)		
q(2)	q(1)		q(0)				
q(3)	q(4)	q(3)	q(3)				
q(4)	q(4)	q(3)			q(5)	q(5)	q(4)
q(5)	q(4)		q(0)				

Fonte: elaboração própria.

O alfabeto presente no autômato é descrito como n (nome), adj (adjetivo), v- (verbo), pron-pers (pronome pessoal), ',' (vírgula), conj-c (conjunção coordenativa), conj-s (conjunção subordinativa), punc (pontuação), adv (advérbio) e art (artigo). A função auxiliar representa um conjunto de marcações do alfabeto do autômato, nas quais podemos citar: conj-s, punc, pron-pers, verb, adv ou conj-c. JÁ a função auxiliar 2 representa o seguinte alfabeto: conj-s, punc, pron-pers, verb, adv, art e conj-c.

A extração dos relacionamentos é feita utilizando um autômato. Essa lógica é refletida no Algoritmo 1. O método SemanticExtr possui, como entrada, a lista de sentenças pré-processadas (S).

A linha 1 corresponde a um laço, o qual percorre a lista de sentenças (S), onde “s” corresponde a estrutura que contém a sentença crua, representada pela variável “sentenca” e “listaToken”, em que é armazenada a lista de *tokens* pré-processados da sentença.

Algoritmo 1 SemanticExtr.

Input: Lista de sentenças pré-processadas (S)
Output: Lista de relacionamentos (R)
1: **for** cada s de S **do**
2: $RE = extrairRelacionamento(s.listaToken, s.sentenca)$
3: $R \leftarrow R \cup RE$
4: **end for**
5: **return** R

Fonte: elaboração própria.

A linha 2 corresponde à chamada do método “extrairRelacionamento”, o qual é representado no

Algoritmo 2. O método “extrairRelacionamento” apresenta como parâmentros “listaToken” e “sentença”, retornando uma lista representada por uma quádrupla (to_1, v, to_2, sentencaRotulada), sendo que “to_1”, “v”, “to_2” são termos que representam o relacionamento extraído, e “sentencaRotulada” corresponde à sentença etiquetada da qual o relacionamento foi extraído. [28]

Na linha 3, há a correspondência à concatenação dos relacionamentos dos resultados obtidos pelo “extrairRelacionamento” com os resultados das sentenças já processadas.

Algoritmo 2, representa-se a implementação do autômato destinado à extração dos relacionamentos semânticos. O estado e o contador de *tokens* são iniciados com zero, e a variável “contador” conta os *tokens* que não fazem parte do relacionamento, iniciando a contagem a partir de um termo pertencente ao relacionamento. Esse contador é criado como medida de distância entre os termos pertencentes ao relacionamento. O valor máximo definido para o contador é 3; passando desse valor o autômato é reiniciado.

A linha 3 contém a estrutura “listasControle”, que armazena os dados processados pelo autômato. Neste sentido, “listasControle” é formada por:

- listaRelacionamentosExtraidos: a qual armazena uma lista de relacionamentos extraídos. O relacionamento é formado pela quádrupla (to_1, V, to_2, sentencaRotulada);
- listaTo1: que contém os termos ontológicos presentes antes do verbo;
- listaTo2: que contém os termos ontológicos presentes depois do verbo;
- V: contém o verbo.

“listasControle” possui, ainda, os seguintes métodos:

- addTo1: adiciona o termo em “listaTo1”;
- addTo2: adiciona o termo em “listaTo2”;
- addV: adiciona o termo em “V”;
- limparListaControles: deleta os dados presentes em “listaTo1”, “listaTo2” e “V”;

[28] A criação da sentencaRotulada é descrita no Algoritmo 12.

- finalizarListaRelacionamentos: responsável por formar relacionamentos presentes em “listaTo1”, “listaTo2” e “V”; e adicioná-los em “listaRelacionamentosExtraidos”;[29]
- retornarlistaRelacionamentosExtraidos: responsável por recuperar a lista de relacionamentos extraídos de “listaRelacionamentosExtraidos”

A estrutura *token* é formada por:

- *token*;
- *lemma*;
- etiqueta;
- termoOntologico.

Algoritmo 2 extrairRelacionamento.

```
Input: lista de tokens (listaToken), sentenca
Output: Lista de relacionamentos formado por (to1, V, to2, sentenca)
1:  estado = 0
2:  contador = 0
3:  listasControle = new listaControle(sentenca)
4:  for cada token (t) de listaTokens do
5:      verificarContadores();
6:      if isQ0(estado) then
7:          processaEstadoQ0(t)
8:      else if isQ1(estado) then
9:          processaEstadoQ1(t)
10:     else if isQ2(estado) then
11:         processaEstadoQ2(t)
12:     else if isQ3(estado) then
13:         processaEstadoQ3(t)
14:     else if isQ4(estado) then
15:         processaEstadoQ4(t)
16:     else if isQ5(estado) then
17:         processaEstadoQ5(t)
18:     end if
19: end for
20: return listasControle.retornarlistaRelacionamentosExtraidos()
```

Fonte: elaboração própria.

[29] Esta lógica é representada no Algoritmo 11.

A linha 4 corresponde ao laço que percorre a lista de *tokens*. Já a linha 5 realiza o controle de contadores (“verificarContadores”), representado pelo Algoritmo 3, a seguir.

Algoritmo 3 verificarContadores.

```
1: if contador > 3 then
2:    if estado > 4 then
3:       listasControle.finalizarListaRelacionamentos(sentenca)
4:    else
5:       listasControle.limparListaControles()
6:    end if
7:    contador = 0
8:    contadorVerbo = 0
9:    estado = 0
10: end if
```

Fonte: elaboração própria.

O objetivo do Algoritmo 3 é o de reiniciar o autômato. É verificado se o contador de *tokens* é maior que 3 e se o estado do autômato é maior que 4. Obedecendo o critério, é possível afirmar que “listasControle” contém pelo menos um relacionamento extraído presente em “listaSn1”, “listaSn2” e “Sv”. O relacionamento é formado no método “finalizarListaRelacionamentos”, e o autômato é reiniciado. Em caso negativo, o autômato é reiniciado e “listasControle” é criada novamente. Ao referir-se em reiniciar o autômato, é afirmado que os contadores de *token* e verbo são zerados, e o estado do autômato passa a ser zero.

Retomando o

Algoritmo 2, na linha 6 é verificado se o estado do autômato é 0; já na linha 7, o método “processaEstadoQ0” é responsável por encontrar o primeiro *token*, no qual conste um termo ontológico. Essa lógica é refletida no Algoritmo 4.

Algoritmo 4 processaEstadoQ0.

```
Input: token (t)
1: if isTermoOntologico(t) then
2:     listasControle.addTo1(t.lemma)
3:     estado = 1
4:     contador = 0
5: end if
```

Fonte: elaboração própria.

O Algoritmo 4, “processaEstadoQ0”, possui como principal objetivo a identificação de um termo ontológico. O método “isTermoOntologico” verifica se o *token* corresponde a um termo ontológico: em caso positivo, o *lemma* do *token* é adicionado em “listaSn1”, o estado é atualizado para 1 e o contador para zero. A correspondência do *token* em termo ontológico é verificado no método “isTermoOntologico”, representado no Algoritmo 5. O método desconsidera os termos presentes na *função auxiliar 2*, uma vez que estes termos não identificam termos ontológicos, mantendo o autômato no estado 0.

O Algoritmo 5 corresponde à função “isTermoOntologico”, a qual possui como entrada o *token* e como saída uma flag (*boolean*). Inicialmente, é verificado se a etiqueta do *token* corresponde a “n” ou “adj” e se o *token* corresponde ao termo ontológico. Obedecendo-se o critério o método retorna o valor *true*, ou seja, o *token* corresponde é um termo ontológico. Devido a problemas no *POS tagger*, foram detectadas inconsistências em algumas marcações, como por exemplo a atribuição de verbo a um *token*, o qual é substantivo. O controle dessas marcações é discutido no Apêndice A.

Algoritmo 5 isTermoOntologico.

```
Input: token
Output: flag
1:  flag = false
2:  if t.etiqueta igual "n" ou "adj" then
3:    if t.termoOntologico não nulo then
4:      flag = true
5:    else
6:      listasControle.limparListaControles()
7:      contador = 0
8:    end if
9:  end if
10: return flag
```

Fonte: elaboração própria.

O Algoritmo 6, “processaEstadoQ1”, é representado pelo estado 1 e possui quatro ações. A primeira verifica se o *token* em análise corresponde ao caractere ‘,’ (vírgula) ou se possui *lemma* igual a ‘conj-c’, atualizando o autômato para o estado 2 e incrementando o contador de *tokens*. A segunda ação consiste em verificar se a etiqueta do *token* é um verbo; em caso afirmativo, o verbo é salvo e o estado é atualizado para 3. Caso o verbo seja “ser”, ele é descartado, uma vez que cria relacionamentos taxonômicos. A terceira ação trata de verificar se o termo é um termo ontológico, adicionando-o na “listaTo1”. A quarta ação corresponde à verificação da etiqueta do *token*, se é “conj-s” ou “pron-prers”, e o autômato é reiniciado. Embora o controle para os demais termos do alfabeto do autômato tenha sido omitido na Figura 24, eles devem manter o autômato no estado atual (estado 1), e o contador é, assim, incrementado.

Tem-se, a seguir, o Algoritmo em questão.

Algoritmo 6 processaEstadoQ1.

```
Input: token (t)
1: if t.token == ',' || t.lemma igual 'conj-c' then
2:    estado = 2
3:    contador = contador + 1
4: else if t.etiqueta contém "V-" then
5:    if t.lema =="ser" then
6:       estado = 1
7:       contador = contador + 1
8:    else
9:       listasControle.addV(t.lemma)
10:      estado = 3
11:   end if
12: else if isTermoOntologico(t) then
13:   listasControle.addTo1(t.lemma)
14:   contador = 0
15: else if t.etiqueta contém ("conj-s", "pron-") then
16:   listasControle.limparListaControles()
17:   contador = 0
18:   estado = 0
19: else
20:   contador = contador + 1
21: end if
```

Fonte: elaboração própria.

O Algoritmo 7, por sua vez, corresponde ao estado 2 do autômato. O método possui duas funções principais: a primeira é a de identificar se o *token* corresponde ao termo ontológico. Obedecendo o critério, o *token* é adicionado em "listaTo1", o contador é marcado como zero e o autômato é atualizado para o estado 1. Caso a etiqueta do *token* contenha conj-s, punc, pron-pers, v-, adv ou conj-c, as listas de controle são limpas, e o autômato é reiniciado. Na Figura 24 foi omitido o controle para os demais termos do alfabeto do autômato, contudo eles devem manter o autômato no estado atual (estado 2), e o contador é incrementado.

Algoritmo 7 processaEstadoQ2.

```
Input: token
1: if isTermoOntologico(t) then
2:     listasControle.addTo1(t.lemma)
3:     contador = 0
4:     estado = 1
5: else if t.etiqueta contém ("conj-", "punc", "adv", "pron-", "v-")
   then
6:     listasControle.limparListaControles()
7:     contador = 0
8:     estado = 0
9: else
10:    contador = contador + 1
11: end if
```

Fonte: elaboração própria.

Já o Algoritmo 8, “processaEstadoQ3”, corresponde ao estado 3, onde são contempladas duas ações possíveis. A primeira verifica se o *token* corresponde a um termo ontológico: em caso afirmativo, o *lemma* do *token* é adicionado em “listaSn2”, o autômato assume o estado 4 e o contador é marcado como zero. A segunda verifica se o *token* possui a etiqueta de verbo, consequentemente, o *lemma* do *token* é adicionado em Sv e o estado permanece em 4.

Sv não é uma lista, logo seu conteúdo pode ser substituído. Essa medida de substituição é adotada perante sentenças que fazem o uso de gerúndio e particípio, o que demostra que o segundo verbo apresenta maior valor significativo para a relação formada. Podemos referenciar, como exemplo, a sentença “Os alunos estão frequentando a sala”. Nela, a relação (aluno-estar-sala) faz menos sentido que (aluno-participar-sala). O verbo “ser” é desconsiderado, pois ele gera relacionamentos taxonômicos. Ao encontrar um dos termos da função auxiliar, o contador é incrementado e o autômato permanece no estado 3.

Algoritmo 8 processaEstadoQ3.

```
Input: token
1: if isTermoOntologico(t) then
2:    listasControle.addTo2(t.lemma)
3:    contador = 0
4:    estado = 4
5: else if t.etiqueta contém ("v-") then
6:    if t.lemma == "ser" then
7:       estado = 3
8:       contador = contador + 1
9:    else
10:      estado = 3
11:      listasControle.addV(t.lemma)
12:   end if
13: else
14:   contador = contador + 1
15: end if
```

Fonte: elaboração própria.

O Algoritmo 9, “processaEstadoQ4”, corresponde ao estado 4 (estado final) e possui quatro ações principais. A primeira verifica se o *token* em análise corresponde ao caractere ‘,’ (vírgula) ou se possui *lemma* igual a conj-c; em caso afirmativo, o autômato é atualizado para o estado 7 e o contador é incrementado. A segunda ação verifica se o *token* é um termo ontológico, então o autômato continua no estado 5 e o *lemma* do *token* é adicionado em “listaSn2”. A terceira ação verifica se o *token* corresponde a alguma pontuação de finalização de sentença – quais sejam, ponto-final (“.”), ponto de exclamação (“!”) ou de interrogação (“?”) –, indicando que a lista de *tokens* foi processada, logo, os termos presentes em “listaTo1”, “v” e “listaTo2” são transformados em relacionamento pelo método “finalizarListaRelacionamentos”.

A quarta ação do Algoritmo 9 verifica se a etiqueta do *token* contém verbo, realizando a cópia do último termo adicionado em “listaTo2”. Os termos contidos em “listaTo1”, “v” e “listaTo2” são transformados em relacionamentos pelo método “finalizarListaRelacionamentos”. Uma nova relação em potencial é iniciada, o termo salvo é adicionado em “listaTo1”, o *token* referente ao verbo é adicionado em “v” e o estado é atualizado para 3.

Na Figura 24 foi omitido o controle para os demais termos do alfabeto do autômato, mas eles devem manter o autômato no estado atual (estado 4), e o contador é incrementado.

Algoritmo 9 processaEstadoQ4.

```
Input: token
1: if t.lemma igual 'conj-c' || t.token igual ',' then
2:    estado = 5
3:    contador = contador + 1
4: else if isTermoOntologico(t) then
5:    listasControle.addTo2(t.lemma)
6:    contador = 0
7:    estado = 4
8: else if t.token contém (".", "!", "?") then
9:    estado = 4
10:   contador = 0
11:   listasControle.finalizarListaRelacionamentos(sentenca)
12: else if t.etiqueta contém "V-" then
13:   if t.lema == "ser" then
14:      estado = 4
15:      contador = contador + 1
16:   else
17:      ultimoTermo = listasControle.obterUltimoTermoTo1()
18:      listasControle.finalizarListaRelacionamentos()
19:      listasControle.addTo1(ultimoTermo)
20:      listasControle.addV(t.lemma)
21:      estado = 3
22:   end if
23: else
24:   contador = contador + 1
25: end if
```

Fonte: elaboração própria.

O Algoritmo 10, "processaEstadoQ5", corresponde ao estado 5 do autômato. O método possui duas funções fundamentais: a primeira identifica se o *token* corresponde ao termo ontológico. Obedecendo este critério, o *token* é adicionado em "listaTo2", o contador é marcado como zero e o autômato é atualizado para o

estado 5. Caso o *token* etiquetado contenha uma das seguintes etiquetas: conj-s, punc, pron-pers, v-, adv ou conj-c, os termos contidos em “listaTo1”, “v” e “listaTo2” são transformados em relacionamentos pelo método “finalizarListaRelacionamentos”, e o autômato é reiniciado.

Outra vez mais, na Figura 24 omitiu-se o controle para os demais termos do alfabeto do autômato, contudo eles devem manter o autômato no estado atual (estado 5), e o contador é incrementado.

Algoritmo 10 processaEstadoQ5.

```
Input: token
1: if isTermoOntologico(t) then
2:     listasControle.addTo2(t.lemma)
3:     contador = 0
4:     estado = 4
5: else if t.etiqueta contém ("conj-", "punc", "adv", "pron-", "v-")
   then
6:     listasControle.finalizarListaRelacionamentos(sentenca)
7:     contador = estado = 0
8:     estado = 0
9: else
10:    contador = contador + 1
11: end if
```

Fonte: elaboração própria.

O método “finalizarListaRelacionamentos” é responsável por realizar o produto cartesiano entre o conjunto formado por “listaTo1”, “listaTo2” e “v”. O resultado do produto cartesiano é uma lista de relacionamentos semânticos, juntamente com a sentença rotulada. O algoritmo do produto cartesiano pode ser visualizado no Algoritmo 11, que será discutido a seguir.

O produto cartesiano de dois conjuntos A e B são todos os pares ordenados (x, y), sendo que x pertence ao conjunto A e y pertence ao conjunto B. Mencionamos, por exemplo, os seguintes conjuntos: A = {1,2,3}; B = {4,5,6}. O produto cartesiano de A por B, representado por A x B, é igual a: {(1,4), (1,5), (1,6), (2,4), (2,5), (3,4), (3,5), (3,6)}.

Algoritmo 11 Produto cartesiano.

Input: listaTo1, listaTo2, V e sentenca
Output: listaRelacionamentos
1: **for** `to1 ∈ listaTo1` **do**
2: **for** `to2 ∈ listaTo2` **do**
3: $sentecaMarcada = montaSentenca(to1_i, V, to2_k, sentenca)$
4: $listaRelacionamentos \leftarrow listaRelacionamentos \cup (to1_i, V, to2_k, sentecaMarcada)$
5: **end for**
6: **end for**
7: **return** *listaRelacionamentos*

Fonte: elaboração própria.

No contexto atual, considerando que há elementos contidos em “listaSn1”, “listaSn2” e Sv, o produto cartesiano é representado por listaSn1 x listaSn2 x Sv. No Algoritmo 11, a linha 1 representa o laço pelo qual é possível percorrer todos os elementos de “listaSn1”. Já a linha 2 indica um novo laço, em que cada iteração representa a quantidade de elementos de “listaSn2”. A linha 3 é responsável pela montagem da sentença, destacando nela o termo ontológico.[30] O par ordenado é montado na linha 4, onde sn_{1i} e sn_{2k} são decorrentes das iterações sucedidas nas linhas 1 e 2. O par ordenado é constituído de ($to1_i$, v, $to2_k$), entretanto é adicionada a sentença marcada, à qual a relação extraída pertence. Por fim, a linha 7 representa o conjunto dos relacionamentos extraídos.

Temos, a seguir, um exemplo de execução:

SN1: {Aluno, Professor}.
SN2: {Sala}.
SV: {Frequentar}.
Sentença: “Os alunos e professores frequentam a sala.”.

O produto cartesiano obtido é:

{(Aluno, Frequentar, Sala, “alunos e professores frequentam a sala”),
(Professor, Frequentar, Sala, “alunos e professores frequentam a sala.”)}

[30] Esse processo será detalhado mais adiante, na seção “Destaque dos relacionamentos semânticos na sentença”, do subcapítulo 4.6 (Capítulo 4).

Após o processo de extração dos relacionamentos semânticos, é necessário identificar os relacionamentos que contêm termos sinônimos e substituí-los pelos termos originais da ontologia. Para esse processo, é necessária a utilização do dicionário de sinônimos construído na etapa de pré-processamento. Cada termo do relançamento extraído (to1 e to2) é consultado no dicionário que, por sua vez, retorna o valor referente à chave informada; a chave representa os valores de to_1 e to_2. Caso o valor informado seja diferente das chaves (to_1 e to_2), o termo é substituído pelo conteúdo obtido do dicionário. No final do processo, são obtidos somente relacionamentos formados pelos termos ontológicos, ou seja, sem a presença dos termos sinônimos.

Para a sentença lematizada “Os Alunos e professores frequentam a sala.”, são demostrados os seguintes fluxos executados pelo autômato:

- [q0]o aluno e professor frequentar o sala. (‘o’ é artigo, desconsiderado).
- o [q0]aluno e professor frequentar o sala. (‘aluno’ é termo ontológico, adiciona-se em listaTo1).
- o aluno [q1]e professor frequentar o sala. (‘e’ corresponde a conj-c).
- o aluno e [q3]professor frequentar o sala. (‘professor’ é termo ontológico, adiciona-se em “listaTo1”).
- o aluno e professor [q1]frequentar o sala. (‘frequentar’ é verbo, adiciona-se em “v”).
- o aluno e professor frequentar [q4]o sala. (o artigo ‘o’ é desconsiderado).
- o aluno e professor frequentar o [q4]sala. (‘sala’ é termo ontológico, adiciona-se em listaTo2).
- o aluno e professor frequentar o sala[q5]. (‘pontuação’ finaliza a lista de relacionamentos).
- o aluno e professor frequentar o sala[q5]. (fim do processamento).

4.5.3 Ponderação da relação

A ponderação da relação consiste em atribuir pesos aos relacionamentos extraídos, em que o peso de cada relacionamento está diretamente relacionado à hierarquia das classes da taxonomia inicial.

Essa etapa de atribuição de pesos é utilizada em uma das formas de consulta dos relacionamentos extraídos na etapa de pós-processamento, em que será possível encontrar os relacionamentos formados nos quais o segundo termo (to_2) esteja a uma distância x do primeiro termo (to_1).

A primeira etapa da ponderação consiste em mapear a taxonomia, ou seja, para cada classe da taxonomia é atribuído um valor. Esse valor é referente ao nível de hierarquia da classe que está na ontologia. A Figura 25 apresenta uma hierarquia da taxonomia, onde "*Thing*" representa a raiz (e o seu valor é 0), os filhos de "Thing" (ou o próximo nível da taxonomia) referem-se às classes "Formação", "Pessoa", entre outras, e essas classes possuem valor 1. Ou seja, o valor de cada classe corresponde à sua posição na hierarquia.

A atribuição dos valores das classes foi obtida de forma dinâmica, por meio de uma adaptação de um algoritmo de grafos, buscando por profundidade. Nela, o algoritmo percorre a taxonomia tendo como nó inicial a raiz, explorando cada um dos seus nós.

Formalmente, o algoritmo percorre a taxonomia que progride por meio da expansão do primeiro nó da taxonomia e se aprofunda até que se depare com um nó que não possua filhos.

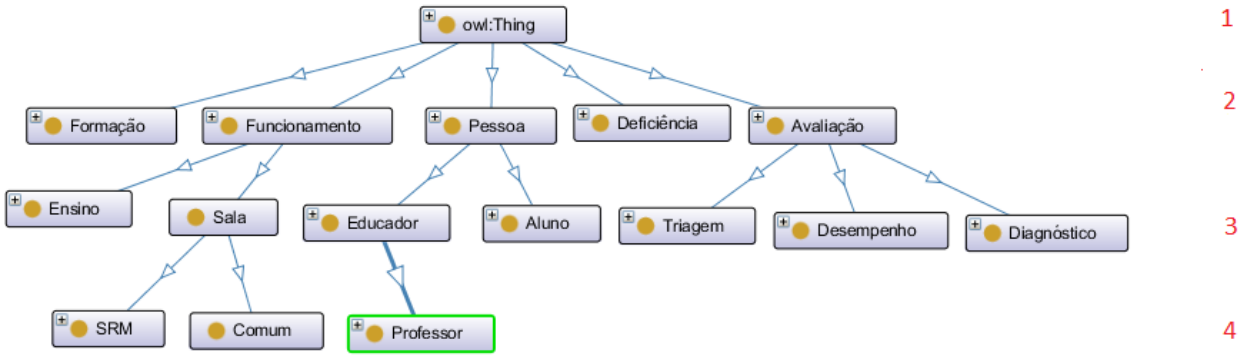


Figura 25 Hierarquia da taxonomia.
Fonte: elaboração própria.

Após identificar cada valor para cada uma das classes da taxonomia, os dados são armazenados em uma estrutura *hash*. A tabela *hash* possui como chave

as classes da taxonomia, e o seu valor se refere ao valor obtido pelo percurso na taxonomia.

Tendo determinado o valor de cada classe da ontologia, o próximo passo destina-se a calcular o valor de cada relacionamento obtido pelo autômato. O valor de cada relacionamento é definido pela distância absoluta entre os dois termos ontológicos do relacionamento.

A distância absoluta é calculada da seguinte forma: dado o relacionamento (to1, v, to2), é necessário buscar os valores dos pesos to1 e to2 na tabela *hash*. Assume-se que os pesos de to1 e to2 sejam, respectivamente, p_{to1} e p_{to2}. Logo, a diferença absoluta é representada na Equação IV.

Equação IV Distância absoluta.

$$distância = Abs\ (p_{to1} - p_{to2})$$

O algoritmo pode ser melhor representado na Figura 26. Em resumo, a linha 1 cria a tabela *hash* percorrendo a taxonomia. A linha 2 representa o laço para o cálculo da ponderação para todos os relacionamentos extraídos. Já a linha 3 define a diferença absoluta para o relacionamento processado. Na linha 4, o algoritmo insere os valores dos pesos e a diferença absoluta de cada relacionamento. Por fim, a linha 6 representa o conjunto dos relacionamentos extraídos e ponderados.

Input: RE lista de relacionamentos, G taxonomia
Output: RP relacionamentos ponderados
1: $T \leftarrow criar\_hash(G)$
2: **for** s $\in$ RE **do**
3: $abs \leftarrow Math.abs(T.key(s.to1_i) - T.key(s.to2_i))$
4: $RP \leftarrow R \cup (T.key(s.to1_i), abs, T.key(s.to2_i))$ {concatena relacionamento ponderado}
5: **end for**
6: **return** RP

Figura 26 Ponderação do relacionamento extraído.
Fonte: elaboração própria.

Dados os relacionamentos extraídos na etapa anterior {(Aluno, Frequentar, Sala) e (Professor, Frequentar, Sala)} e com base na Figura 25, demonstrada anteriormente, pode-se assumir os pesos de Aluno, Sala e Professor

respectivamente por 2, 2 e 3. Logo, a diferença absoluta dos relacionamentos são, respectivamente, {(2, 0, 2) e (3, 1, 2)}.

4.5.4 Salvar os relacionamentos extraídos no banco de dados

A última etapa do processamento consiste em armazenar os dados gerados em um banco de dados. O modelo criado baseia-se em três unidades (*Relacionamento_ponderado, Relacionamento* e *Sentenca*). A Figura 27 representa o diagrama do banco de dados utilizado para representar e armazenar os relacionamentos extraídos da etapa de processamento.

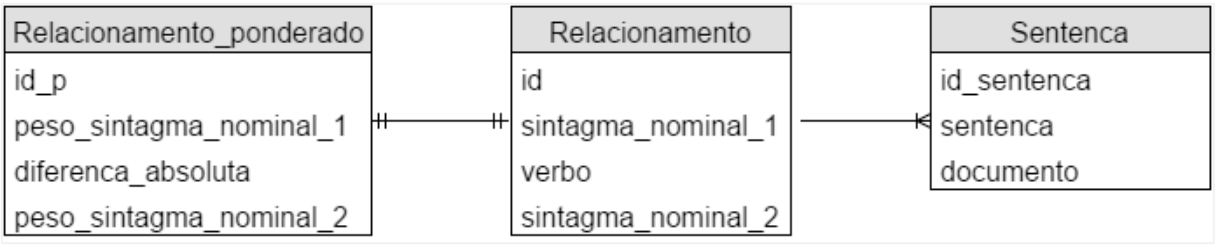


Figura 27 Diagrama do banco de dados.
Fonte: elaboração própria.

O quadro *Relacionamento* armazena os relacionamentos extraídos pelo autômato. Ele contempla as informações de cada relacionamento extraído, e seus dados são formados por ambos os termos ontológicos e o verbo que realiza a ligação entre os termos.

Já o quadro *Sentenca* é responsável por armazenar as sentenças que compõem os relacionamentos extraídos. Nele também está contido o nome do documento ao qual a sentença pertence. A relação entre as tabelas *Relacionamento* e *Sentenca* é de 1 para N, onde um relacionamento pode estar presente em mais de uma sentença.

Já a tabela *Relacionamento_ponderado* armazena as informações relacionadas à ponderação da relação. Ela guarda os valores atribuídos de cada termo ontológico da relação e, por fim, provisiona o valor da diferença absoluta entre ambos os termos ontológicos de cada relação armazenada.

4.6 Pós-processamento

O pós-processamento corresponde à terceira e última etapa do método Embed$_{NT}$RelOnto. Esta etapa é totalmente dependente de um especialista de domínio, já que ele é o responsável por avaliar se o relacionamento extraído é válido e se possui relevância para ser integrado à ontologia.

A Figura 28, a seguir, demostra as etapas envolvidas no pós-processamento. A primeira etapa consiste no desenvolvimento de uma aplicação, onde é possível realizar consultas de relacionamentos extraídos, mostrando algumas informações relevantes (tais como o relacionamento e a sentença da qual o relacionamento foi extraído). Já a etapa de representação dos relacionamentos consiste em criar um FCA com os relacionamentos selecionados pelo especialista de domínio. O FCA proporciona uma representação gráfica dos relacionamentos selecionados, e o gráfico FCA gerado é decorrente da ferramenta *Concept Explorer*. A última etapa consiste no enriquecimento da ontologia: depois de obter os relacionamentos selecionados pela etapa anterior, eles são incorporados à ontologia por meio da API Jena.

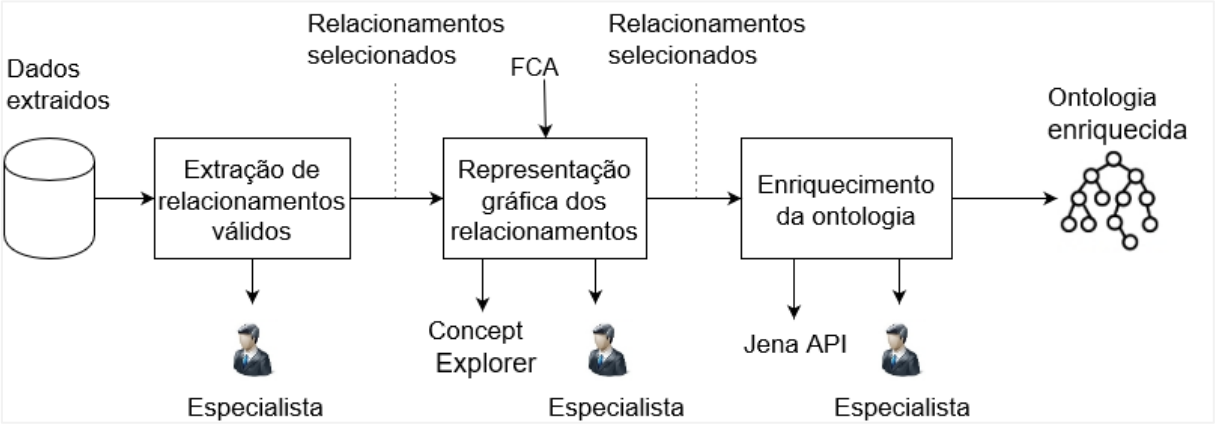


Figura 28 Pós-processamento.
Fonte: elaboração própria.

4.6.1 Extração de relacionamentos válidos

Este procedimento consiste na validação dos relacionamentos extraídos, por meio de uma ferramenta denominada “Buscador de sentenças por relacionamentos

semânticos". Nela, o especialista realiza buscas por relacionamentos, obtendo como resultado as sentenças pertinentes ao relacionamento informado. Diante das sentenças exibidas, o especialista avalia se o relacionamento em questão é válido.

A aplicação desenvolvida possui três tipos de buscas, os quais apresentam a mesma estrutura para a exibição dos resultados: eles aparecem em uma tabela, na qual se informa o relacionamento (primeira coluna) e a sentença (na segunda coluna).

A primeira consulta consiste em fornecer o relacionamento, o qual é formado por uma tripla, dois termos ontológicos e um verbo, seguindo o formato: <termo ontológico><verbo>< termo ontológico>.

A segunda consulta consiste em fornecer apenas termos ontológicos. Logo, a aplicação retorna os relacionamentos pertencentes aos nomes informados e, por conseguinte, são exibidas as sentenças pertencentes ao relacionamento. Essa busca pode ser informada numa lista de termos. No entanto, os relacionamentos obtidos não contemplaram essencialmente um único relacionamento, mas sim cada termo pertencente ao relacionamento.

A Figura 29 representa um exemplo de consulta por relacionamento, na qual se informa o relacionamento, como exemplo, o relacionamento (aluno, ir, sala). Ao realizar a busca, a aplicação retorna à relação juntamente com a sentença. O resultado obtido foi uma única sentença: "**Oferecem apoio pra esse aluno, esse aluno vai lá na sala comum, participa das atividades e não há uma relação estreita com a**".

Figura 29 Seleção de relacionamentos no buscador**.**
Fonte: elaboração própria.

A terceira consulta refere-se à ponderação do relacionamento. Nessa consulta, tem-se como entrada um termo ontológico, o campo diferença e, por fim, o *checkbox* marcado como “Busca por palavras”. A Figura 30 expressa um exemplo de busca. O resultado da busca cujos termos foram “professor”, “aluno” e “diagnóstico” mostra todos os relacionamentos em que há dois termos ontológicos e que estão no mesmo nível da hierarquia da ontologia, ou seja, o campo diferença marcado com 0 (zero) – o que é comprovado na Figura 25, já mencionada.

4.6.1.1 Destaque dos relacionamentos semânticos na sentença

Perante os resultados das buscas demonstradas nas Figuras 29 eFigura 30, nota-se que as sentenças apresentam os termos do relacionamento em destaque (grafados em vermelho). Essa funcionalidade foi concebida como forma de apoio ao usuário , para que se identifique, com mais facilidade, o relacionamento presente na sentença. Em resumo, os termos são encapsulados em *tags* html, logo o navegador as interpreta e atribui as marcações aos termos do relacionamento. A lógica é refletida no Algoritmo 12, que aparece a seguir.

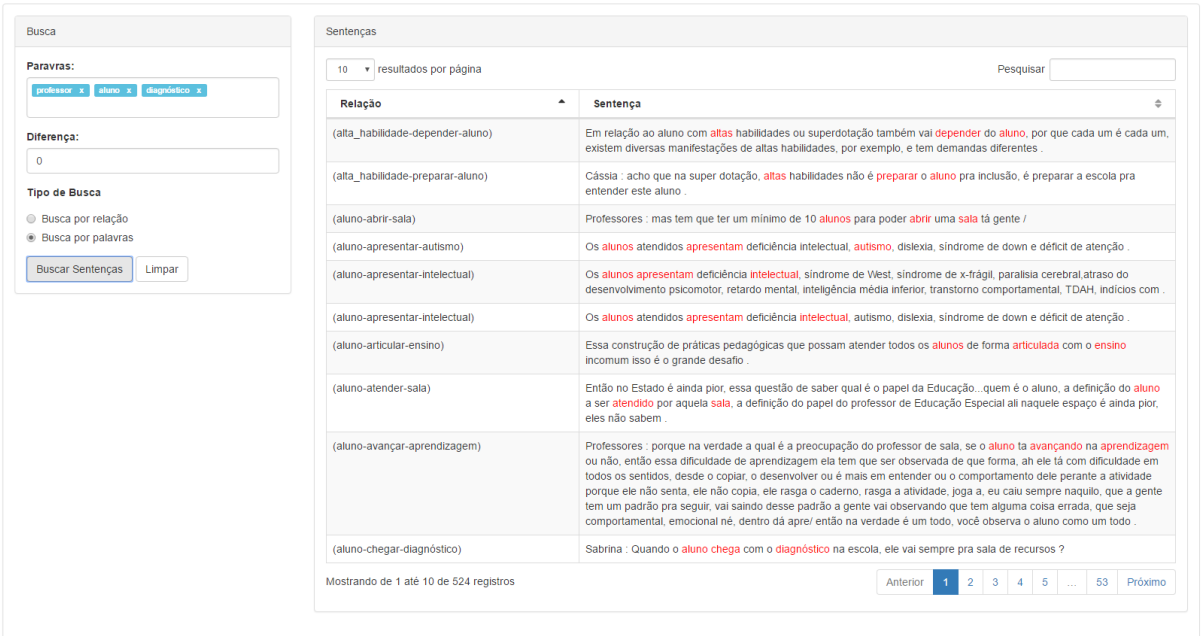


Figura 30 Busca ponderada.
Fonte: elaboração própria.

Algoritmo 12 Marcação de sentença.

Input: indiceTo1, indiceVerbo, indiceTo2, sentenca
Output: Sentença marcada

```
1: listaTokens ← Tokenizacao(sentenca)
2: newSent ←'< html >'
3: for i = 0; i < listaTokens.length; i++ do
4:    if i contém (indiceTo1, indiceVerbo, indiceTo2) then
5:       newSent ← newSent+' < font color =' red' >' +listaTokens[i]+' < /font >'
6:    else
7:       newSent ← newSent + listaTokens[i]
8:    end if
9: end for
10: newSent ← newSent+' < /html >'
11: return newSent
```

Fonte: elaboração própria.

O algoritmo tem como entrada o índice de cada termo do relacionamento e a sentença referente ao relacionamento extraído. Essa sentença é novamente tokenizada. Primeiramente, a sentença é iniciada com a *tag* html e, então, a lista de *tokens* é percorrida, em busca pelos índices dos termos do relacionamento. Ao encontrá-los, o *token* é incorporado à *tag* <*font color*> com o valor "*red*", sendo concatenado na nova sentença. Caso contrário, o termo é concatenado na nova sentença.

4.6.2 Representação gráfica dos relacionamentos

O objetivo desta etapa é combinar o método FCA com os relacionamentos semânticos selecionados, obtendo como resultado uma estrutura de apoio ao especialista, na decisão de escolher o melhor relacionamento a ser integrado na ontologia. Por meio do agrupamento do FCA é possível visualizar os relacionamentos, facilitando, assim, a escolha de um ou mais relacionamentos semânticos.

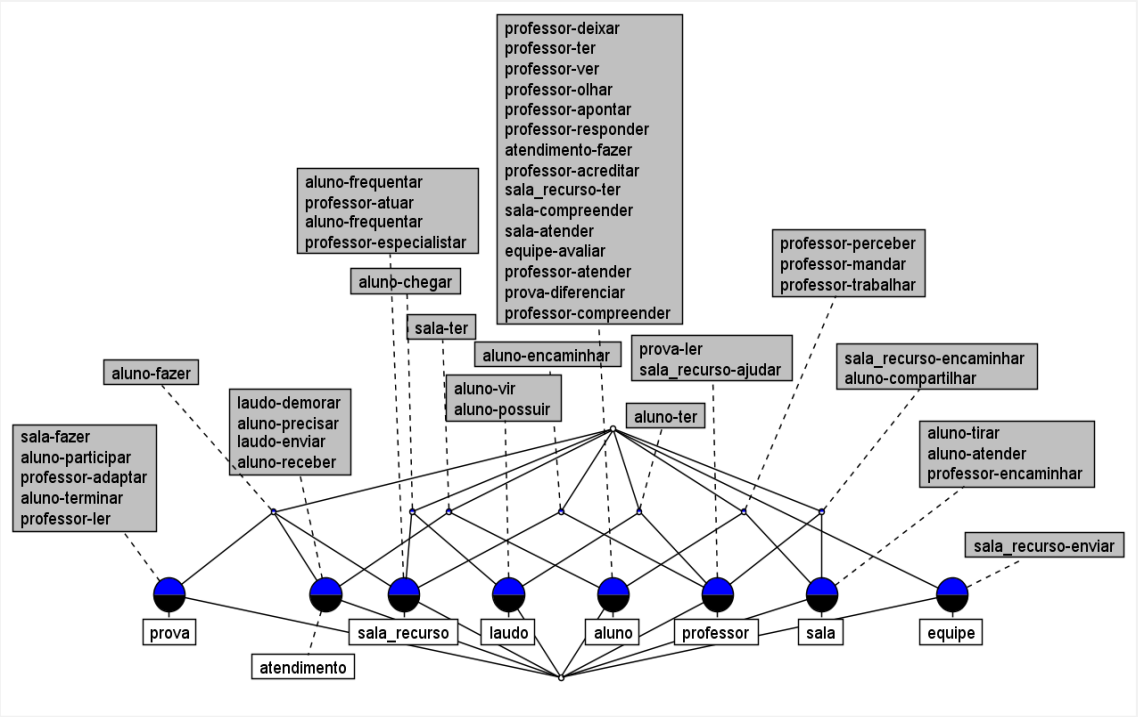


Figura 31 Representação do um.
Fonte: elaboração própria.

A visualização do FCA é viabilizada pela ferramenta *Concept Explorer*, que apresenta como entrada um arquivo do tipo planilha. Para gerar essa planilha foi desenvolvido um algoritmo que possui, como entrada, uma lista de relacionamentos semânticos, os quais são convertidos para o formado aceito pela *Concept Explorer*.

A princípio, a geração da planilha desmembra o relacionamento em duas partes, gerando objetos formais e atributos formais. Diante de um relacionamento semântico representado por (to_1, v, to_2), este é convertido para o conceito formal, o qual é definido pela tupla (atributo formal, objeto formal), representados, respectivamente, por “to1-v” e “to2”. Por exemplo, a partir do relacionamento “aluno-fazer-prova”, são gerados os respectivos atributo e objeto formal: “aluno-fazer” e “prova”.

A Figura 31, demonstrada anteriormente, representa um FCA gerado a partir dos relacionamentos semânticos extraídos de forma automática pelo SemanticExtr. Os dados utilizados para a geração do FCA serão discutidos mais adiante, no Experimento 2 (subcapítulo 5.3, Capítulo 5).

4.6.3 Enriquecimento da ontologia

A partir dos relacionamentos validados pelo especialista de domínio é possível realizar a inclusão do relacionamento na ontologia preliminar, de forma simplificada, informando somente o relacionamento em questão.

Por meio do *plug-in* Jena desenvolveu-se a ferramenta "Gerador de relacionamentos ontológicos", a partir da qual é possível informar dois termos ontológicos e um verbo, verificando se os termos ontológicos estão presentes na ontologia; em caso afirmativo, o relacionamento é incorporado na ontologia. Para que isso ocorra, é expressamente importante que a ontologia esteja presente na aplicação.

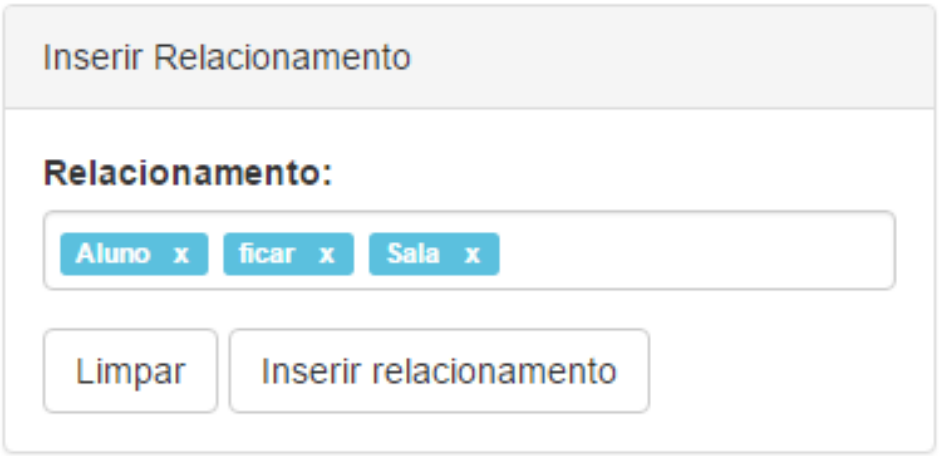


Figura 32 Inserir relacionamento.
Fonte: elaboração própria.

A Figura 32 representa a inclusão do relacionamento na ontologia, em que se tem um campo para informar os dados na respectiva ordem: termo ontológico, verbo e termo ontológico. Ao clicar no botão "Inserir relacionamento", a aplicação se encarrega de fazer a validação dos termos ontológicos e, por fim, de inserir o relacionamento na ontologia. O relacionamento inserido pode ser visto na Figura 33, a seguir.

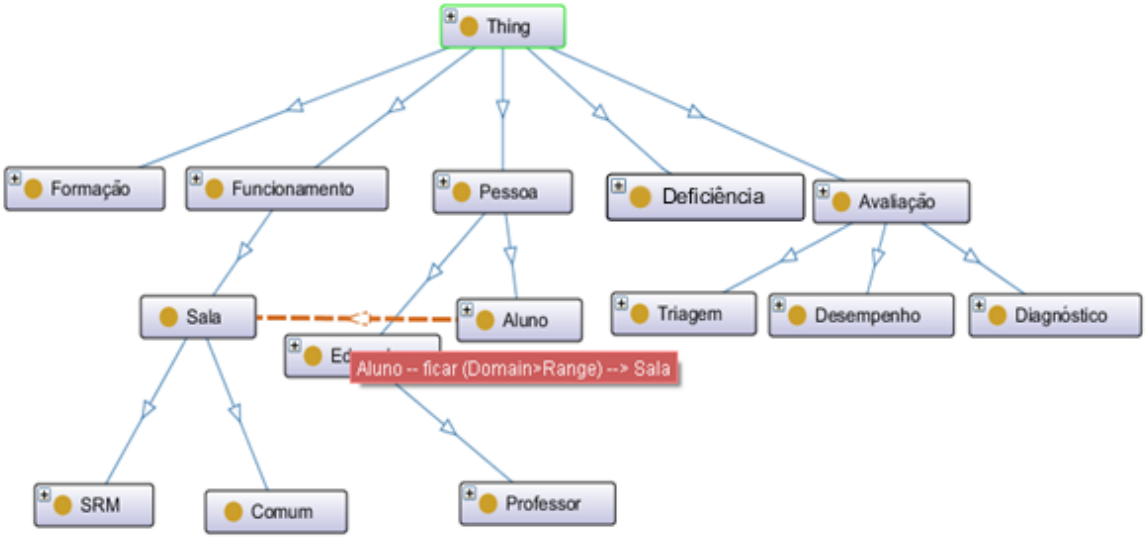


Figura 33 Relacionamento incorporado na ontologia.

Fonte: elaboração própria.

4.7 Otimização do SemanticExtr

Processadores *multicore* permitem que aplicações explorem paralelismos no nível de *threads*, a fim de habilitar melhorias no tempo de conclusão da execução.

Com a pretensão de obter mais velocidade na extração de relacionamentos semânticos, a abordagem SemanticExtr foi construída pensando na escalabilidade. O SemanticExtr consegue ser executado utilizando 100% do uso do processador, sendo possível inserir uma quantidade x de *threads*.

A aplicação possui a capacidade de identificar a quantidades de núcleos de um processador e, assim, instanciar a mesma quantidade de *threads*. De maneira geral, a aplicação faz o uso de *Future*, *Callable* e *ExecutorService*.

A *Future* é uma classe responsável por encapsular uma chamada feita em paralelo, sendo possível cancelar a execução de uma tarefa e descobrir se a execução já terminou com sucesso ou se apresentou erro.

Callable é uma interface para a implementação de uma execução em paralelo. Ela é parecida com a interface *Runnable*; a diferença é que a *Callable* retorna um valor ao final da execução

ExecutorService é uma classe para o gerenciamento de execuções em paralelo, já que cria um *pool* de *threads*, iniciando e cancelando as execuções. Também é possível cancelar o *pool*, evitando assim a criação de novas tarefas.

No contexto do presente trabalho, cada documento é atribuído a um *thread*, ou seja, cada núcleo do processador fica encarregado de processar um documento. Ao término do processamento do documento, é atribuído um novo documento a um *thread*.

4.8 Considerações finais

Ao longo deste Capítulo foi possível aprestar a abordagem semântica para incorporação de relacionamentos não taxonômicos em ontologias com documentos informais do domínio da educação especial.

Na seção 4.2 foram apresentadas algumas definições necessárias para o entendimento do trabalho, tais como *termos*, *termos ontológicos*, *token* e *relacionamento semântico*.

Na seção seguinte, foi apresentado o corpus utilizado tanto para o desenvolvimento quanto para os testes (que serão apresentados em breve, no Capítulo 5) executados como método proposto. Além disso, apresentou-se a taxonomia da qual foram extraídos os termos ontológicos.

Em seguida, a seção 4.4 trouxe uma visão geral do processo de extração de relacionamentos, dividido em três etapas: a de pré-processamento, de processamento e de pós-processamento. A etapa de pré-processamento é fundamental, sendo responsável pela preparação dos documentos e na qual são aplicadas as técnicas de processamento de texto.

Na seção 4.6 foi abordada a etapa de processamento, na qual se apresentou o método SemanticExtr, responsável por extrair relacionamentos semânticos entre termos não taxonômicos por meio de um autômato finito. Na seção seguinte, discutiu-se a etapa de pós-processamento, que prescinde o apoio de especialistas de domínio.

Por fim, a seção 4.8 apresentou a otimização do código presente no método SemanticExtr, discutindo-se o uso de *futures* para múltiplos processamentos dos documentos, para a extração dos relacionamentos.

Serão apresentados, no Capítulo que segue, os métodos de validação, bem como os testes da proposta e os resultados conquistados.

CAPÍTULO 5

Validação da proposta

5.1 Considerações iniciais

O presente Capítulo apresenta o processo de validação do método proposto, bem como a validação das hipóteses.

A validação do método de extração de relacionamentos semânticos entre termos ontológicos em textos informais do domínio Educação Especial escritos em Português variante brasileira foi realizada em dois experimentos.

O primeiro compreende a anotação manual, ou seja, os especialistas realizaram a extração dos relacionamentos semânticos manualmente. Os resultados obtidos pela aplicação do método proposto foram comparados aos resultados obtidos de forma manual.

Já o segundo experimento representa a validação manual do reticulado gerado pelo FCA, formado por um conjunto de relacionamentos extraídos pela aplicação do método proposto.

5.2 Experimento 1 – Avaliação automática do SemanticExtr

O experimento primário consistiu em fornecer uma amostra do corpus e um conjunto de termos ontológicos para os especialistas. Diante dos artefatos, os

especialistas realizaram a extração dos relacionamentos semânticos formando, assim, um *Gold Standard* com os relacionamentos extraídos.

Gold Standard é um conjunto de normas/diretivas usadas pelos profissionais para avaliar processos automáticos. No contexto desta pesquisa de mestrado, o *Gold Standard* é formado por uma lista de relacionamentos comuns entre termos ontológicos identificados pelos especialistas no escopo do trabalho realizado.

A descoberta manual de relacionamentos semânticos entre termos ontológicos de uma amostra do corpus (*Gold Standard)* serve de base para validar o método SemanticExtr, desenvolvido nesta pesquisa.

Para apoiar o processo de criação do *Gold Standard*, foi realizado um *brainstorming* por meio de um processo de anotação manual nos documentos por anotadores do domínio para, assim, verificar se o *Gold Standard* conseguiria expressar adequadamente o domínio discutido.

Tabela 5 Dados dos documentos.

Documentos	Número total de páginas
AVA_DESC_BA_Feira de Santana.S02.doc	49
AVA_DESC_ES_Linhares.S06.doc	21
AVA_DESC_ES_São Mateus.S03.doc	15
AVAL_DESC_SP.SãoCarlos.S01.01.docx	67
FORM_DESC_SP.SãoCarlos.S01.01.docx	31

Fonte: elaboração própria.

A criação do *Gold Standard* foi realizada por dois especialistas do domínio. Os especialistas selecionados para o experimento foram os mesmos que participaram do projeto ONEESP, ou seja, já estavam familiarizados com os documentos. Para cada anotador foi fornecido um protocolo, juntamente com o corpus, formado por cinco documentos (como demonstra a Tabela 5), e uma relação de 16 termos ontológicos.[31] O protocolo indicou aos anotadores os passos para a realização da anotação, quais sejam:

1. Iniciar a contagem do tempo;

[31] A relação dos termos e seus termos sinônimos pode ser consultada no Apêndice A deste trabalho.

2. Ler os documentos com atenção, identificando os termos ontológicos (termos fornecidos) e destacando-os com a cor vermelha;
3. Ler novamente os documentos e identificar os relacionamentos formados entre os termos identificados em vermelho e, por fim, destacar os relacionamentos com a cor verde;
4. Extrair o relacionamento formado;
5. Finalizar a contagem do tempo e anotar o tempo total gasto durante o experimento.

Após o experimento, os especialistas responderam a seguinte pergunta: *Qual foi o grau de cansaço (de 0 a 5) que você sentiu ao identificar os relacionamentos?* No caso, 0 representou "pouco cansaço" e 5 "muito cansaço".

De acordo com o protocolo, os anotadores são orientados a marcar, inicialmente, os termos ontológicos. Posteriormente, deve-se anotar os relacionamentos formados entre os termos ontológicos identificados compondo, assim, o *Gold Standard*.

Após o processo de extração dos relacionamentos por parte dos anotadores, os *Gold Standards* são interseccionados para compor o *Gold Standard* final. Esta foi a versão usada como base para avaliar a proposta desta dissertação de Mestrado.

O *Gold Standard* final obtido pela anotação manual compreende a intersecção entre as sentenças anotadas pelos especialistas, em que há concordância entre o relacionamento extraído nas sentenças comuns entre ambos os especialistas participantes. Obteve-se um total de 53 sentenças, contendo 35 relacionamentos distintos. Os tempos gastos e os graus de cansaço informados pelos especialistas no processo de anotação são apresentados na Tabela 6, a seguir.

Tabela 6 Tempo gasto e grau de cansaço por especialista.

Documentos	Especialista 1		Especialista 2		
	Tempo	Cansaço	Tempo	Cansaço	
AVA_DESC_BA_Feira de Santana.S02.doc	4h e 30 min	5	2 a 4h	1	4 a 5
AVA_DESC_ES_Linhares.S06.doc	1h e 45 min	3			
AVA_DESC_ES_São Mateus.S03.doc	2h	5			
AVAL_DESC_SP.SãoCarlos.S01.01.docx	3h e 50min	4			
FORM_DESC_SP.SãoCarlos.S01.01.docx	2h	3			

Fonte: elaboração própria.

O especialista 2 não realizou marcações individuais de tempo, mas informou que a anotação variou entre 4 (para os documentos maiores) e 1h/2h (para os demais documentos). Com relação ao nível de cansaço, ele apontou a nota em duas perspectivas. A primeira nota informada foi 1, referindo-se ao grau de cansação para a identificação dos termos ontológicos. Já a variação 4 a 5 foi atribuída para a identificação dos relacionamentos. Uma observação feita pelo especialista foi a de que o processo de identificação dos termos foi prazeroso, ao contrário da identificação dos relacionamentos, etapa mais cansativa.

Na avaliação por meio da comparação com a anotação manual do corpus (*Gold Standard* final) são utilizadas as métricas de precisão, cobertura e medida F (apresentadas na seção 2.3.1.4 do Capítulo 2). Perante o experimento foram identificados os seguintes dados: verdadeiros positivos (VP), falsos positivos (FP) e falsos negativos (FN). Obteve-se, em detalhes:

- Total de relacionamentos semânticos encontrados pelo algoritmo (VP): 58;
- Total de relacionamentos semânticos não anotados manualmente e que foram encontrados pelo algoritmo (FP): 5;
- Total de relacionamentos anotados manualmente e que não foram encontrados pelo algoritmo (FN): 3.

Diante dos dados apresentados para “VP”, “FP” e “FN”, foi possível lograr as medidas de precisão, cobertura e medida F. Os valores estão representados na Tabela 7.

Tabela 7 Resultado extração automática.

Precisão	Cobertura	Medida F
0,92	0,95	0,93

Fonte: elaboração própria.

Em posse dos relacionamentos anotados manualmente pelos especialistas no experimento 1, foram selecionados alguns deles para que fossem inseridos na ontologia utilizando a ferramenta “Gerador de relacionamentos ontológicos”. A Figura 34 contém alguns dos relacionamentos introduzidos.

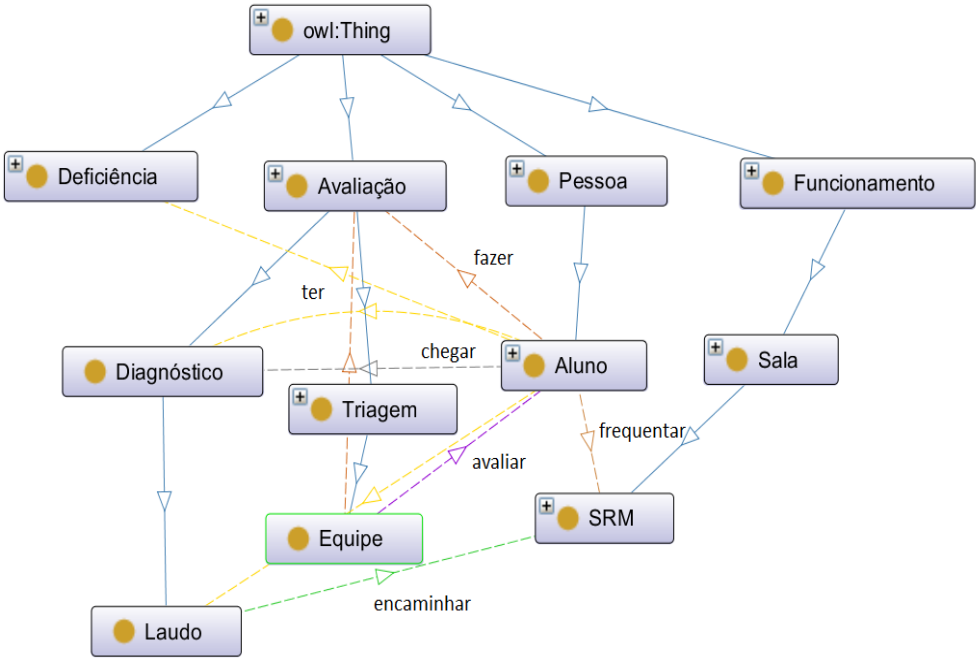


Figura 34 Relacionamentos inseridos.
Fonte: elaboração própria.

A seguir, será abordado o experimento 2, em que foram fornecidos os relacionamentos semânticos extraídos pela ferramenta e dispostos no método FCA.

5.3 Experimento 2 – Identificação de relacionamentos com o auxílio do FCA

O experimento 2 consistiu em disponibilizar os relacionamentos extraídos dispostos em um FCA para que os especialistas realizassem o processo de validação de relacionamentos perante a proposta apresentada no presente trabalho. Este experimento teve como objetivo avaliar relacionamentos extraídos pela abordagem da pesquisa.

Foram extraídos, no total, 57 relacionamentos, a partir de cinco documentos (demonstrados na Tabela 5), com uma seleção de oito termos ontológicos: “aluno”, “atendimento”, “ensino”, “equipe”, “laudo”, “professor”, “prova” e “sala de recurso”.

O experimento foi executado por dois especialistas de domínio. Ao contrário dos profissionais do experimento 1, os especialistas do experimento 2 não

participaram do projeto ONEESP, ou seja, não haviam tido contato com os documentos dos quais foram extraídos os relacionamentos.

Para cada anotador foi fornecido um protocolo, juntamente com a imagem do FCA.[32] O protocolo foi dividido em duas etapas: na primeira, o especialista avaliou os relacionamentos extraídos representados no FCA. Os passos contidos da primeira etapa no protocolo foram os seguintes:

1. Iniciar a contagem do tempo;
2. Olhando os relacionamentos pelo FCA, marcar com um asterisco (*) quais dos relacionamentos devem ser integrados à ontologia;
3. Olhando os relacionamentos pelo FCA, marcar com um traço (–) quais dos relacionamentos não fazem sentido (ou se não conseguiu identificar o contexto no qual ele é empregado);
4. Olhando os relacionamentos pelo FCA, marcar com o símbolo de "mais" (+) quais dos relacionamentos estão corretos, mas que não são importantes para compor a ontologia;
5. Listar os relacionamentos semânticos relevantes;
6. Finalizar a contagem do tempo e notar o tempo gasto.
7. Responder a seguinte pergunta: *Informe um número de 0 a 5 (onde 0 representa "pouco confortável" e 5 "muito confortável"): qual o grau de conforto que você sentiu ao identificar os relacionamentos relevantes? Se você tivesse acesso às sentenças das quais os relacionamentos foram extraídos, qual seria seu grau de conforto?*

Após a primeira etapa do experimento em análise, o especialista 1 identificou: 31 relacionamentos como relevantes e que deveriam compor a ontologia (o que corresponde a 54%); 6 relacionamentos corretos, mas irrelevantes (o que corresponde a 11%); e 20 relacionamentos identificados como descontextualizados (correspondente a 35%). O tempo gasto nessa etapa foi de 5 minutos e 50 segundos. O especialista considerado indicou, como grau de conforto, 2 e 5, respectivamente. O gráfico gerado com base nestas informações pode ser visualizado na Figura 35, a seguir.

[32] O FCA é representado no Apêndice B deste trabalho.

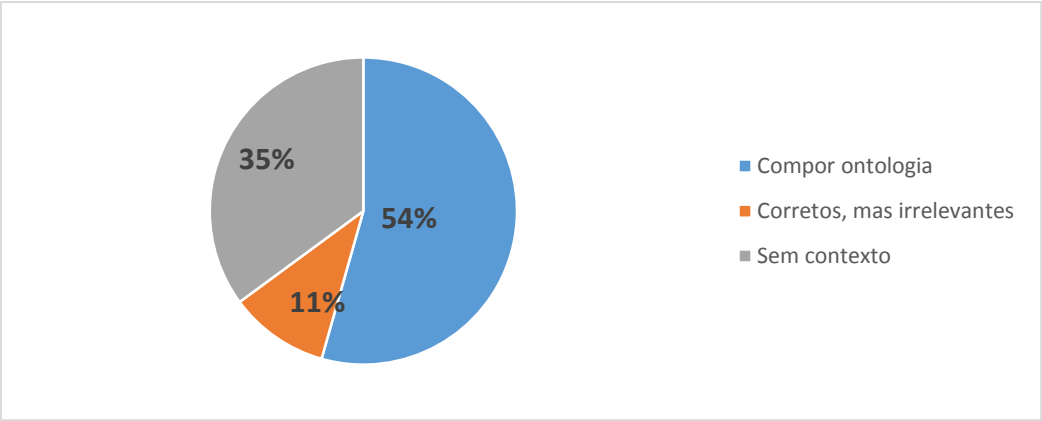


Figura 35 Gráfico da Etapa 1, Especialista 1.
Fonte: elaboração própria.

O segundo especialista identificou: 39 relacionamentos como relevantes e que deveriam compor a ontologia (68%); 10 relacionamentos corretos, mas irrelevantes (18%); e 8 relacionamentos identificados como "sem contexto" (14%). O tempo gasto por ele nesse processo foi de 24 minutos e 10 segundos. O gráfico gerado pode ser visualizado na Figura 36. Foi atribuído, como grau de cansaço, os níveis 3 e 5, respectivamente.

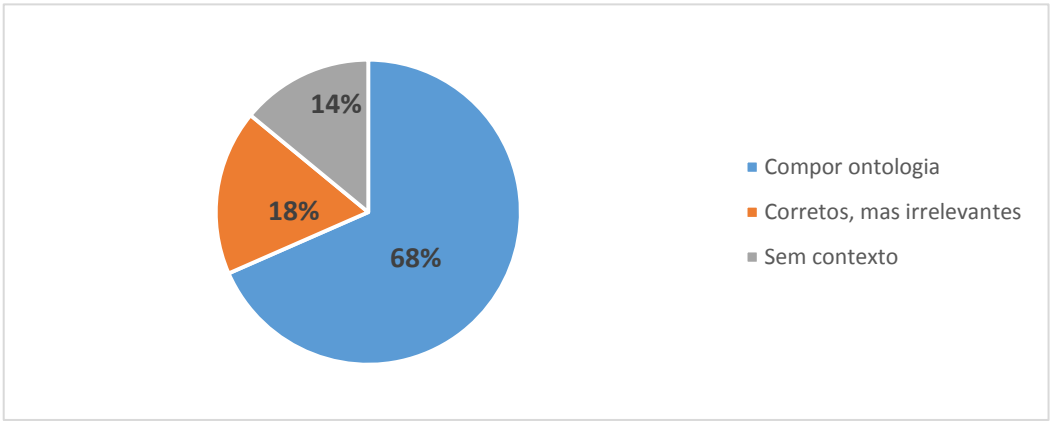


Figura 36 Gráfico da Etapa 1, Especialista 2.
Fonte: elaboração própria.

A Figura 37 apresenta o gráfico de comparação entre os especialistas na identificação dos relacionamentos referentes à primeira etapa. É possível observar uma diferença entre os apontamentos indicados pelos especialistas.

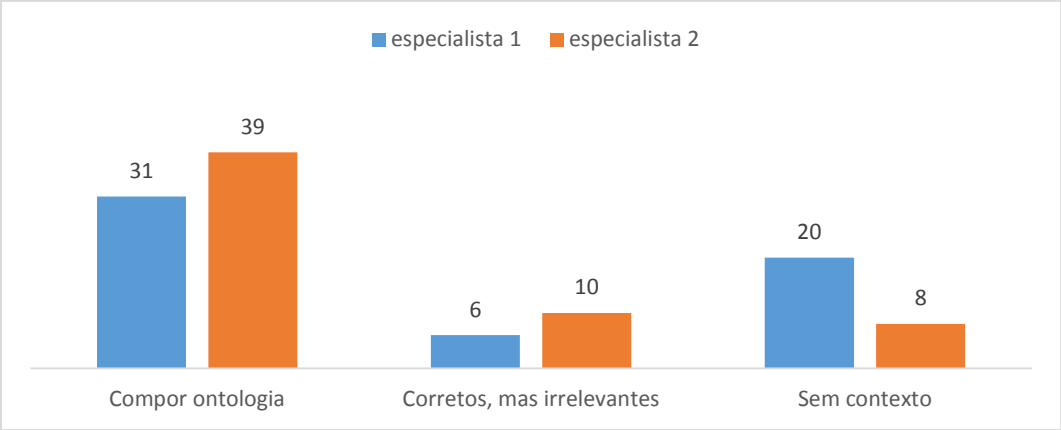


Figura 37 Comparação da primeira etapa.
Fonte: elaboração própria.

A segunda etapa do processo compreendeu uma reavaliação dos relacionamentos marcados com "–" e "+". No entanto, os especialistas tiveram acesso às sentenças das quais os relacionamentos foram extraídos. Lhes foi fornecida a ferramenta de busca por sentenças perante o relacionamento informado, denominada "Buscador de sentenças por relacionamentos semânticos". Por meio dela, os especialistas informaram o relacionamento e analisaram as retornadas de acordo com o parâmetro de busca. Ou seja, eles informaram os relacionamentos de interesse e a ferramenta retornou a(s) sentença(s) da(s) qual(is) o relacionamento foi extraído. Os passos contidos nesta segunda etapa no protocolo foram os seguintes:

1. Iniciar a contagem do tempo;
2. Realizar a busca pelas sentenças cujos relacionamentos estão marcados com "–" e "+";
3. Perante as sentenças que compõem o relacionamento, realizar uma nova marcação no relacionamento marcado:
    a. "*" (asterisco) para relacionamentos relevantes e que devem compor a ontologia;
    b. "–" (traço) para os relacionamentos sem contexto;
    c. "+" (símbolo de "mais") para relacionamentos válidos, mas não relevantes.
4. Listar os relacionamentos semânticos relevantes;
5. Finalizar a contagem do tempo e notar o tempo gasto;
6. Com o apoio da ferramenta de busca de sentenças e do FCA, informar um número de 0 a 5 (onde 0 representa "pouco confortável" e 5 "muito

confortável": qual o grau de conforto que você sentiu ao identificar os relacionamentos relevantes?

Após a segunda etapa do experimento, o especialista 1 identificou: 43 relacionamentos como relevantes e que deveriam compor a ontologia (75%); 2 relacionamentos corretos, mas irrelevantes (4%); e 12 relacionamentos identificados como "sem contexto" (21%). O tempo gasto por ele nesse processo foi de 20 minutos e 32 segundos. Foi indicado, como grau de conforto, o número 5. O gráfico gerado a partir das informações coletadas pode ser visualizado na Figura 38.

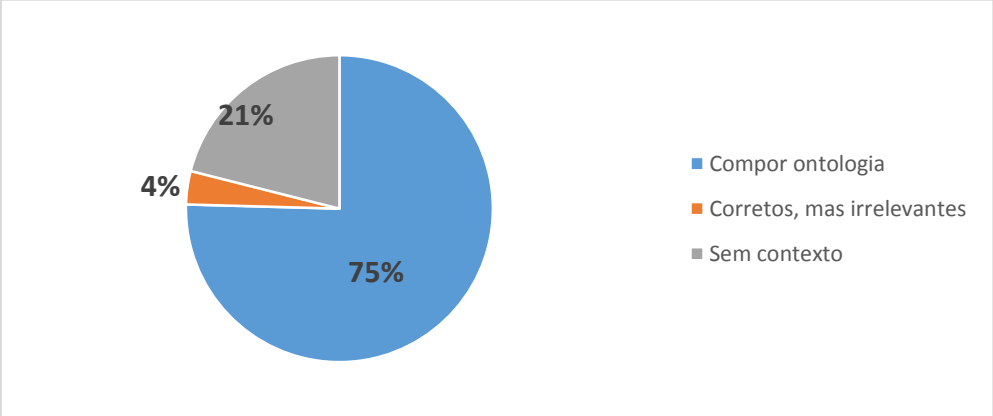


Figura 38 Gráfico da Etapa 2, Especialista 1.
Fonte: elaboração própria.

O segundo especialista identificou: 44 relacionamentos como relevantes e que deveriam compor a ontologia (77%); 10 relacionamentos corretos, mas irrelevantes (18%); e 3 relacionamentos identificados como sem contexto (5%). O tempo gasto por ele neste processo foi de uma hora de 29 minutos, tendo sido atribuído o número 4 para o grau de conforto. O gráfico gerado pode ser visualizado na Figura 39.

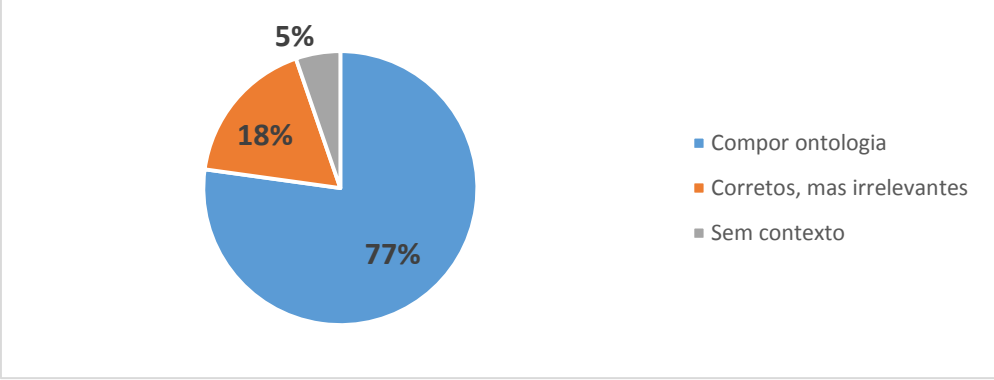


Figura 39 Gráfico da Etapa 2, Especialista 2.
Fonte: elaboração própria.

Diante da observação do gráfico da Figura 40, nota-se maior uniformidade na comparação feita entre os especialistas da segunda etapa com relação à quantidade de relacionamentos extraídos que devem compor a ontologia, retomando a perspectiva da comparação realizada anteriormente e demonstrada na Figura 37. No entanto, somente 34 relacionamentos foram selecionados em comum, por ambos os especialistas, na categoria dos relacionamentos que devem compor a ontologia.

Figura 40 Gráfico da comparação entre especialistas, Etapa 2.
Fonte: elaboração própria.

Os relacionamentos analisados nas três categorias por ambos os especialistas do experimento 2 estão descritos em uma tabela que pode ser consultada no Apêndice C deste trabalho. Considerando-se, como relacionamentos válidos, as seguintes categorias: relacionamentos que devem ser inseridos na ontologia e os relacionamentos corretos, mas irrelevantes, é possível obter boas medidas de precisão.

A Tabela 8 contém as medidas de precisão de ambos os especialistas. A média obtida entre as suas medidas é de 86,5%.

Tabela 8 Precisão, experimento 2.

	Precisão
Especialista 1	79%
Especialista 2	94%

Fonte: elaboração própria.

A Tabela 9 abrange, por sua vez, a medida de tempo de execução tomado pelos especialistas durante as etapas do experimento 2. Nela também são informados os graus de conforto atribuídos como nota para cada uma das etapas.

Tabela 9 Tabela de tempo no experimento 2.

	Etapa 1		Etapa 2	
	Tempo	Nota	Tempo	Nota
Especialista 1	5min e 50s	2	20min e 32s	5
Especialista 2	24min e 10s	3	1h e 28min	4

Fonte: elaboração própria.

5.4 Experimento de performance

Os experimentos de performance tem como objetivo demonstrar o tempo de execução do método SemanticExtr utilizando multi *threads* e concorrência. Para tal, são considerados todos os documentos disponíveis (cerca de 300 documentos) e todos os termos disponíveis na ontologia.

Em ambos os experimentos (experimento 1 e experimento 2) foram extraídos cerca de 2.700 relacionamentos, os quais não foram validados por especialistas de domínio. O tempo de execução utilizando uma única *thread* foi de 1h e 40min. Realizando o mesmo procedimento empregando quatro *threads*, o tempo de extração foi de 27 minutos.

Os experimentos foram executados em um computador da marca *Dell*, com processador Intel Core i5-6300HQ, dois núcleos de 2:30GHz cada, 8Gb de memória RAM, com sistema operacional *Windows 10 Pro*, de 64 bits. No desenvolvimento das ferramentas foi utilizada linguagem de programação Java, versão 8, 64 bits e a IDE Eclipse Neon.

5.5 Discussão sobre os resultados

A partir dos resultados apresentados nos dois experimentos realizados, é possível promover algumas reflexões e tirar conclusões. No experimento 1 foi utilizada como base de comparação a anotação manual do corpus, criando, assim, um *Gold Sandard*.

O *Gold Standard* criado é considerado uma base de comparação, pois executa a tarefa proposta pelo trabalho – neste caso a identificação de relacionamentos não taxonômicos entre termos ontológicos em documentos informais – sob a forma mais precisa possível, de modo que o especialista do domínio consiga extrair os relacionamentos existentes nos documentos. Porém, devido à grande quantidade de documentos, a extração manual necessita de um longo período de tempo para ser executada.

Para auxiliar o trabalho de extração de relacionamentos a partir dos documentos, foi proposta, para a presente pesquisa, uma solução utilizando técnicas de Extração de Informação. Essa solução utiliza o artifício da extração baseado em regras (padrões textuais), por meio de um autômato finito. Segundo os experimentos realizados, a solução obteve um bom rendimento – com aproximadamente 91% de precisão, 92% de cobertura e 91% de medida F, quando aplicada em um conjunto de sentenças selecionadas dentro do domínio da Educação Especial. Segundo os dados informados pelos especialistas do experimento, constatou-se que o processo de extração é muito cansativo, dispendendo horas de análise dos documentos. Diante dos relatos dos profissionais envolvidos, são somadas novas evidências de que o processo de extração manual é oneroso.

Com os bons resultados obtidos pelo processo de extração automático, é possível afirmar que a abordagem proposta consegue obter bons resultados na etapa de extração de relacionamentos, considerada cansativa e onerosa pelos especialistas.

Na avaliação do experimento 1, detectou-se um problema, o qual não foi levantado durante a elaboração do método. Segundo o especialista, o relacionamento “aluno-encaminhar-professor” foi extraído em uma sentença na voz passiva; logo, o relacionamento na voz ativa ficaria como “professor-encaminhar-aluno”. De acordo com o especialista, o relacionamento presente na voz passiva é

um relacionamento válido, contudo o relacionamento transformado em voz ativa apresentaria melhor definição da realidade.

O experimento 2 consistiu na geração de uma estrutura gráfica por meio de 57 relacionamentos semânticos, extraídos automaticamente pelo método SemanticExtr. Por meio da imagem do FCA, os dois especialistas do domínio realizaram a validação dos relacionamentos, obtendo como medidas de precisão, respectivamente, 81% e 96%. Ambos os profissionais, ao analisarem os relacionamentos por meio do FCA juntamente com as sentenças extraídas, informaram como grau de conforto da tarefa realizada, os números 4 e 5, respectivamente.

Com relação aos tempos informados por ambos os especialistas, é possível notar que há uma diferença considerável entre eles – o que nos conduz a levantar alguns questionamentos, tais como: o especialista 2 demonstrou maior dificuldade em realizar o experimento, ou o especialista 1 possui maior domínio do tema.

Com relação ao experimento 2, foi observado que o relacionamento “aluno-encaminhar-professor” foi considerado pelo especialista 1 como sendo “sem contexto”; já o especialista 2 apontou tal relacionamento como sendo válido. Como mencionado no experimento 1, o relacionamento em questão é decorrente de uma sentença em voz passiva. Tal sentença é a seguinte: “P8 – Quando eu recebo também aluno encaminhado pelo professor da sala da regular, eu faço o que P1 e P4 falaram.”. Os relacionamentos “professor-encaminhar-sala” e “sala_recurso-enviar-equipe” foram apontados como incorretos ou descontextualizados pelo especialista 1. Entretanto, o especialista 2 classificou-os como sendo relacionamentos relevantes.

Comparando os experimentos 1 e 2, é possível notar que o experimento 2 foi realizado em um tempo bem menor, e ambos os especialistas se sentiram confortáveis ao avaliar os relacionamentos extraídos. Já o experimento 1 foi desenvolvido em um tempo maior, e o processo de identificação de relacionamentos foi apontado como causador de um alto grau de cansaço.

Com o encerramento dos experimentos, é possível afirmar que a abordagem $Embed_{NT}$RelOnto obteve resultados promissores, e é capaz de apoiar os especialistas no processo de enriquecimento de ontologias, dado que o processo realiza a extração de relacionamentos em um tempo bem menor se comparado ao processo manual, obtendo boas métricas de avaliação.

5.6 Validação das hipóteses

Após a realização dos testes e a apresentação dos resultados, pode-se validar as hipóteses enumeradas ao início do presente trabalho:

- **Hipótese 1**: *é possível realizar a extração de relacionamentos semânticos não taxonômicos entre termos ontológicos a partir de textos informais no idioma Português variante brasileira*.
- ✓ **Resposta**: Sim. Diante do experimento 1, foi possível obter boas medidas de precisão, cobertura e medida F. De acordo com os resultados obtidos, a abordagem auxilia o especialista na extração dos relacionamentos.

- Hipótese 2: *o FCA auxilia os especialistas de domínio na identificação dos relacionamentos semânticos.*
- ✓ Resposta: sim. Pautando-se no experimento 2, foi possível obter boas medidas de precisão entre os especialistas. Ambos confortáveis em realizar a validação dos relacionamentos por meio do FCA em conjunto das sentenças que pertencem ao relacionamento extraído, e o tempo de execução foi razoavelmente baixo.

5.7 Considerações finais

Neste Capítulo, foram apresentados dois experimentos que possuíam, como objetivo principal, a validação do processo de extração de relacionamentos semânticos entre termos ontológicos não taxonômicos em documentos informais do domínio da Educação Especial.

O experimento 1 consistiu em comparar o processo de extração manual de relacionamentos semânticos e o processo automático de extração de relacionamentos semânticos. Ao concluí-lo, foram obtidos bons resultados de precisão, revocação e medida F.

Em seguida, foi apresentado o experimento 2, que consistiu em apresentar, de forma gráfica, os relacionamentos extraídos automaticamente. Neste experimento, os especialistas dispunham, como suporte, de uma ferramenta para consultar as sentenças das quais foram extraídos os relacionamentos. Este experimento obteve boa medida de precisão.

Por fim, foram discutidos os resultados obtidos em ambos os experimentos e, consequentemente, foi realizada uma análise frente aos trabalhos correlatos. Diante dos resultados positivos em ambos os experimentos, as hipóteses foram consideradas válidas.

CAPÍTULO 6

Conclusões e trabalhos futuros

Nesta dissertação apresentamos uma abordagem para a extração de relacionamentos semânticos não taxonômicos entre termos ontológicos, por meio de documentos escritos em linguagem informal sobre a Educação Especial, para o enriquecimento de ontologias. Este processo é considerado trabalhoso pelos profissionais da área.

A revisão da literatura mostrou alguns trabalhos que abordam processos para a construção e o enriquecimento de ontologias de forma automática. No entanto, estes processos são mais comuns em documentos formais e voltados para a língua inglesa.

Na literatura foram apresentados vários processos voltados para a extração de relacionamentos semânticos; dentre eles, se destacam as abordagens baseadas em dicionários, em Aprendizado de Máquina (AM) e, por fim, em padrões textuais.

Este trabalho teve como base uma ontologia preliminar, na qual foi possível identificar os termos ontológicos. Por meio desses termos e da construção de um autômato finito, foi construída uma ferramenta que realiza a extração de relacionamentos semânticos e, posteriormente, foi desenvolvida uma funcionalidade de apoio ao especialista na fase de avaliação e inclusão do relacionamento semântico na ontologia.

Nas próximas seções serão descritas, resumidamente, as contribuições juntamente com os resultados da avaliação e as limitações do trabalho apresentado. Por fim, uma discussão dos possíveis trabalhos futuros proporcionados pelo desenvolvimento da abordagem $Embed_{NT}$RelOnto encerrará o presente trabalho.

6.1 Síntese dos resultados e contribuições

Este trabalho prestou-se à construção de uma abordagem de apoio ao especialista para o enriquecimento de ontologias, denominada Embed$_{NT}$RelOnto. A abordagem realiza a extração de relacionamentos em documentos e auxilia o especialista na inclusão dos relacionamentos na ontologia.

A abordagem foi avaliada por meio de dois experimentos: no primeiro, foi construído pelos especialistas um *Gold Standard* de forma manual. Os resultados encontrados a partir de uma análise comparativa entre o *Gold Standard* e os relacionamentos provenientes da abordagem proposta apresentaram altos índices de coincidência, comprovando que o processo configura uma boa ferramenta de apoio ao especialista de domínio.

No segundo experimento, foi fornecida uma imagem do FCA, gerando-o com os relacionamentos extraídos diante do método SemanticExtr. Os especialistas tiveram acesso à aplicação que realiza a busca das sentenças diante do relacionamento informado. A avalição do experimento 2 obteve boas medidas de precisão, e os especialistas se sentiram confortáveis em avaliar os relacionamentos extraídos em consulta com as sentenças pertencentes ao relacionamento.

Diante do trabalho apresentado, é possível listar as principais contribuições do método Embed$_{NT}$RelOnto, como por exemplo a extração de relacionamentos semânticos não taxonômicos entre temos ontológicos decorrentes do método SemanticExtr.

Outra contribuição da ferramenta de apoio ao especialista de domínio para o enriquecimento da ontologia é a disponibilização de meios para a validação dos relacionamentos semânticos encontrados, facilitando, por fim, a inclusão do relacionamento na ontologia, sem a necessidade de um editor de ontologias.

Como contribuições secundárias, destacam-se as seguintes: o fato de que o método proposto realiza a anotação semântica de sentenças, uma vez que os relacionamentos extraídos são marcados nas sentenças de origem; o desenvolvimento do método SemanticExtr utilizando múltiplas *threads*, sendo possível obter 100% de processamento do computador no qual a aplicação desenvolvida é executada.

6.2 Dificuldades encontradas

Uma das dificuldades encontradas refere-se à informalidade presente nos textos analisados, contendo sentenças enormes e que carecem de pontuação e revisão gramatical. A informalidade dos textos fez com que as ferramentas de PLN se perdessem no processo. Ou seja, o processo de etiquetação realizou marcações incorretas, persistindo o erro para o processo de lematização. Podemos citar como exemplo o termo “sala”, que foi reconhecido como verbo e, consequentemente, lematizado para “*salar*”. Com relação à informalidade dos textos, houve uma dificuldade em moldar o autômato para que ele buscasse por relacionamentos genéricos.

Outra dificuldade encontrada disse respeito ao pequeno número de especialistas para a realização dos experimentos, o que proporcionou uma pequena amostra de dados para a realização dos testes. Com a disponibilidade de mais especialistas seria possível obter uma maior uniformidade dos resultados apresentados no experimento 2, por exemplo, no qual ocorreu uma divergência entre os dois especialistas e os temps de execução do experimento.

Uma terceira dificuldade refletiu o fato de alguns relacionamentos extraídos pertencerem a sentenças escritas em voz passiva, o que pode ter levado ao erro de validação por parte dos especialistas.

6.3 Limitações da abordagem

De maneira geral, a abordagem $Embed_{NT}$RelOnto tem um enfoque na extração de relacionamentos semânticos entre termos ontológicos não taxonômicos. Contudo, ela não realiza a identificação e criação de axiomas referentes aos relacionamentos extraídos.

Embora a avaliação da abordagem tenha estabelecido um bom nível significativo, ela não identifica sentenças em voz passiva, conduzindo a uma interpretação equivocada dos especialistas, uma vez que a leitura do relacionamento é realizada de forma ativa.

Por fim, com relação à inclusão de relacionamentos, a abordagem não valida se o relacionamento a ser inserido está presente na ontologia. Outra limitação refere-se à extração dos relacionamentos presentes na ontologia inicial, no sentido de compará-los aos relacionamentos extraídos, ou seja, de realizar uma pré-avaliação dos relacionamentos extraídos.

O último subcapítulo do trabalho traz algumas propostas para estudos futuros que poderão auxiliar na solução das limitações mencionadas e proporcionar uma melhor avaliação da proposta da presente pesquisa.

6.4 Trabalhos futuros

Diante do trabalho apresentado, observa-se que há uma necessidade de expandir o processo de avaliação da abordagem $Embed_{NT}$RelOnto. Esta expansão consiste em aumentar a gama de documentos de domínio analisados e, consequentemente, aumentar o número de especialistas. Com um maior número de profissionais envolvidos, é possível promover uma revisão gramatical e ortográfica dos documentos em análise, afastando as chances de se enfrentar dificuldades ao longo do processo.

Pretende-se tornar a abordagem compatível com outros idiomas, como por exemplo o inglês. Para tal, será necessária a incorporação de ferramentas de PLN com o suporte ao novo idioma. Também será preciso realizar alterações no autômato, para que ele consiga interpretar os rótulos do *POS tagger* referentes ao novo idioma. É necessária, ainda, uma análise na estrutura gramatical do novo idioma, para que seja identificada a necessidade de reorganização dos estados do autômato e, por fim, uma ontologia no mesmo idioma. Além disso, há a pretensão de se realizar experimentos da abordagem em outros domínios, como o domínio médico, pois há uma grande variedade de documentos, bem como uma disponibilidade diversa de ontologias válidas neste domínio.

Diante das limitações da abordagem, pretende-se incluir novos recursos na abordagem, como por exemplo a identificação de axiomas entre os relacionamentos extraídos, obtendo-se como resultado final uma ontologia mais completa. Outro

recurso a ser incorporado na abordagem é a capacidade de identificar sentenças na voz passiva e, consequentemente, realizar a extração do relacionamento na voz ativa.

Como trabalhos secundários, considera-se a necessidade de realizar a minimização do autômato, ou seja, diminuir a quantidade de estados sem alterar o resultado final da abordagem.

REFERÊNCIAS

ANANIADOU, S.; MCNAUGHT, J. *Text Mining for Biology and Biomedicine*. Norwood: Artech House, 2005. Disponível em: <http://www.amazon.com/Text-Mining-Biology-And-Biomedicine/dp/158053984X>. Acesso em: 01 jun. 2017.

ARANHA, C. N. *Uma abordagem de Pré-Processamento Automático para Mineração de Textos em Português*: sob o enfoque da Inteligência Computacional. Tese (Doutorado em Engenharia Elétrica). Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007. 144 p. Disponível em: < https://www.maxwell.vrac.puc-rio.br/10081/10081_1.PDF>. Acesso em: 01 jun. 2017.

AZEVEDO, R. R. et al. An approach for learning and construction of Expressive Ontology from Text in Natural Language. *IEEE Computer Society Washington*, n. 14, ago. 2014, p. 149-156. Disponível em: <http://dl.acm.org/citation.cfm?id=2682647.2682714>. Acesso em: 09 mar. 2016.

BERLAND, M.; CHARNIAK, E. Finding parts in very large corpora. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS ON COMPUTATIONAL LINGUISTICS, 37. *Anais…*, v. 1910, n. c, jul. 1999, p. 57-64. Disponível em: <http://portal.acm.org/citation.cfm?doid=1034678.1034697>. Acesso em: 01 jun. 2017.

BICK, E. *The parsing system "Palavras"*: automatic grammatical analysis of portuguese in a constraint grammar framework. Aarhus: Aarhus University Press, 2000.

BREITMAN, K. K. *Web Semântica*: a internet do futuro. 1. ed. Rio de Janeiro: Sociedade Brasileira de Computação, 2005. Disponível em: <https://goo.gl/XZP5j9>. Acesso em: 11 mar. 2015.

CASELI, H. D. M. *Indução de léxicos bilíngues e regras para a tradução automática*. Tese (Doutorado em Ciências de Computação e Matemática Computacional), Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos, 2007. 186 p. Disponível em: <https://goo.gl/cBnk2R>. Acesso em: 01 jun. 2017.

CIMIANO, P. *Ontology learning and population from text*. Nova York: Springer US, 2006. Disponível em: <http://link.springer.com/10.1007/978-0-387-39252-3>. Acesso em: 01 jun. 2017.

CIMIANO, P.; HOTHO, A.; STAAB, S. Learning concept hierarchies from text corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, v. 24, 2005, p. 305-339. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.228>. Acesso em: 11 mar. 2015.

COHEN, K. B.; HUNTER, L. Getting started in text mining. *PLoS Computational Biology*, v. 4, n. 1, p. 1-3, 2008.

DOMINGUE, J. Tadzebao and WebOnto. In: KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP, 11. *Anais…*, 1998, p. 1-20. Disponível em: <https://goo.gl/9CTtMM>. Acesso em: 01 jun. 2017.

DONGEN, S. VAN. A cluster algorithm for graphs. *Information Systems [INS]*, n. R0010, jan. 2000, p. 1-40. Disponível em: <https://goo.gl/hlMxW1>. Acesso em: 01 jun. 2017.

DUQUE, J. L. et al. Um processo baseado em parágrafos para a extração de tratamentos em artigos científicos do domínio biomédico. *Brazilian Symposium in Information and Human Language Technology*, v. d, p. 124-133, 2011.

EBECKEN, N.; LOPES, M.; COSTA, M. Mineração de Textos. In: REZENDE, S. O. (Org.). *Sistemas inteligentes*: fundamentos e aplicações. 1. ed. São Carlos: Manole, 2003. p. 337-370. Disponível em: <https://goo.gl/K5jb0k>. Acesso em: 01 jun. 2017.

FELDMAN, R.; SANGER, J. *The Text Mining handbook*. Nova York: Cambridge University Press, 2006.

FERNANDES, W. *SERENDIPITY Prospecção Semântica de dados qualitativos em Educação Especial*. Tese (Doutorado em Educação Especial), Programa de Pós-Graduação em Educação Especial da Universidade Federal de São Carlos, São Carlos, 2016. 231 p. Disponível em: <https://goo.gl/zve383>. Acesso em: 01 jun. 2017.

FREITAS, M. C.; QUENTAL, V. Subsídios para a elaboração automática de taxonomias. In: CONGRESSO DA SBC - V WORKSHOP EM TECNOLOGIA DA INFORMAÇÃO E DA LINGUAGEM HUMANA (TIL), *27. Anais...,* p. 1585-1594, 2007. Disponível em: <http://www.de9.ime.eb.br/~sousamaf/cd/pdf/arq0163.pdf>. Acesso em: 01 jun. 2017. mimeo.

GANTER, B.; WILLE, R. *Formal Concept Analysis*. Heidelberg: Springer Berlin Heidelberg, 1999. Disponível em: <http://link.springer.com/10.1007/978-3-642-59830-2>. Acesso em: 01 jun. 2017.

GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. Methodologies and methods for building ontologies. *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Nova York: Springer US, 2004, p. 107-197.

GRUBER, T. Ontology. In: LIU, L.; ÖZSU, T. O. (Ed.). *Encyclopedia of Database Systems*. Springer-Verlag, 2009. Disponível em: <http://tomgruber.org/writing/ontology-definition-2007.htm>. Acesso em: 16 mar. 2015.

GUARINO, N. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies - Special issue: the role of formal ontology in the information technology*, v. 43, n. 5/6, p. 625-640, nov./dez. 1995. Disponível em: <http://dl.acm.org/citation.cfm?id=219668>. Acesso em: 01 jun. 2017.

GUARINO, N. Formal Ontology and Information Systems. *Cite Seer X*, v. 46, p. 3-15, jun. 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.1776&rep=rep1&type=pdf>. Acesso em: 01 jun. 2017. mimeo.

GUIZZARDI, G. *Desenvolvimento para e com reuso*: um estudo de caso no domínio de vídeo sob demanda. Dissertação (Mestrado em Informática), Programa de Mestrado em Informática do Centro Tecnológico da Universidade Federal do Espirito Santo, Vitória, 2000. Disponível em: <https://goo.gl/fe5kpQ>. Acesso em: 01 jun. 2017.

GUPTA, V.; LEHAL, G. S. A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, v. 1, n. 1, p. 60-76, 2009.

HEARST, M. A. Automatic Acquisition of Hyponyms ftom Large Text Corpora. *Proceedings of the 14th conference on Computational Linguistics*, v. 2, p. 23-28, 1992.

HIGUCHI, S. *Representação do conhecimento e modelagem conceitual de ontologia no domínio da História do Brasil Contemporâneo*. Dissertação (Mestrado em Ciência da Informação), Programa de Pós-Graduação em Ciência da Informação da Universidade Federal Fluminense, Niterói, 2012. 172 p. Disponível em: <http://www.ci.uff.br/ppgci/arquivos/Dissert/Dissertacao_Suemi_Higuchi.pdf>.

HOPCROFT, J. E.; ULLMAN, J. D. *Formal languages and their relation to automata*. Boston: Addison-Wesley Longman Publishing Co., 1969.

HORROCKS, I. et al. OWL: a Description Logic Based Ontology Language for the Semantic Web.In: BAADER, F. et al. (Ed.). *The Description Logic handbook*: theory, implementation and applications. Nova York: Cambridge University Press, 2007. Disponível em: <https://goo.gl/BRtPzo>. Acesso em: 01 jun. 2017.

HOU, X. et al. GRAONTO: a graph-based approach for automatic construction of domain ontology. *Expert Systems with Applications*, v. 38, n. 9, p. 11958-11975, set. 2011. Disponível em: <https://goo.gl/uw4qlH>. Acesso em: 03 mar. 2015.

JACKSON, P.; MOULINIER, I. *Natural Language Processing for online applications text retrieval, extraction and categorization*. 2 ed. Reino Unido: John Benjamins Publishing Company, 2007.

JURAFSKY, D.; MARTIN, J. H. Speech and Language Processing: an introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. *Speech and Language Processing An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, v. 21, p. 0-934, 2000. Disponível em: <https://goo.gl/imUzdF>. Acesso em: 01 jun. 2017.

KOU, Z.; COHEN, W. W.; MURPHY, R. F. High-recall protein entity recognition using a dictionary. *Bioinformatics*, v. 21, n. 1, 2005.

KRAUTHAMMER, M.; NENADIC, G. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, v. 37, n. 6, p. 512-526, 2004.

LIMA, J. C. D.; CARVALHO, C. L. D. Ontologias - OWL (Web Ontology Language). *Relatório técnico (004-05)*, Instituto de Informática da Universidade Federal de Goiás, p. 1-24, jun. 2005.

MAEDCHE, A.; PEKAR, V.; STAAB, S. Ontology Learning Part One - On Discovering Taxonomic Relations from the Web. *Journal Web Intelligence*, p. 3-25, 2002. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.1881>. Acesso em: 04 jun. 2017.

MAEDCHE, A; STAAB, S. Discovering conceptual relations from text. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 14. Anais…*, p. 1-17, 2000. Disponível em: <https://goo.gl/J2mK5c>. Acesso em: 01 jun. 2017.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. Nova York: Cambridge University Press, 2009.

MENDES, E.; CIA, F. National Observatory on Special Education: Network Study about Inclusive Education in Brazil. *Open Journal of Social Sciences*, v. 3, n. 9, p. 60-64, 2015. Disponível em: <https://goo.gl/6AXxhE>. Acesso em: 01 jun. 2017.

MENEZES, P. *Linguagens Formais e Autômatos*. 4. ed. Porto Alegre: Instituto de Informática da UFRGS, 1997.

MONARD, M. C.; BATISTA, G.; CARVALHO, A. Applying one-sided selection to unbalanced datasets. *MICAI 2000: Advances in Artificial Intelligence*, v. 1793, p. 315-325, abr. 2000. Disponível em: <http://dl.acm.org/citation.cfm?id=646401.689068>. Acesso em: 01 jun. 2017.

MORAES, S. M. W. *Construção de estruturas ontológicas a partir de textos*: um estudo baseado no método Formal Concept Analysis e em papéis semânticos. Tese (Doutorado em Ciência da Computação), Pontifícia Universidade Católica do Rio Grande do Sul, Rio Grande do Sul, 2012. 184 p. Disponível em: <http://tede2.pucrs.br/tede2/bitstream/tede/5184/1/439881.pdf>. Acesso em: 01 jun. 2017.

MORAIS, E. A. M.; AMBRÓSIO, A. P. L. Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens. *Relatório técnico (001-07)*, Instituto de Informática da Universidade Federal de Goiás, p. 1-22, 2007. Disponível em: <http://www.portal.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-07.pdf>. Acesso em: 01 jun. 2017.

NEVES, P.; CORRÊA, D.; CAVALCANTI, M. Uma análise sobre abordagens e ferramentas para Extração de Informação. *Revista Militar de Ciência e Tecnologia*, v. xxx, p. 32-58, 2013. Disponível em: <https://goo.gl/R6YDym>. Acesso em: 01 jun. 2017.

NOY, N. F.; MCGUINNESS, D. L. Ontology development 101: A guide to creating your first ontology. *Development*, v. 32, p. 1-25, 2001. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136.5085&rep=rep1

&type=pdf%5Cnhttp://liris.cnrs.fr/alain.mille/enseignements/Ecole_Centrale/What is an ontology and why we need it.htm>. Acesso em: 01 jun. 2017. mimeo.

PRISS, U. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, v. 40, n. 1, p. 521-543, set. 2007. Disponível em: <https://goo.gl/Erws6n>. Acesso em: 01 jun. 2017.

QUINLAN, J. C4. 5: programs for machine learning. São Francisco: Morgan Kaufmann Publishers Inc., 1993. (Col. *Machine Learning*, v. 240.).

RAUTENBERG, S. et al. Uma metodologia para o desenvolvimento de ontologias. *Revista Ciências Exatas e Naturais*, v. 10, p. 237-262, 2008. Disponível em: <http://200.201.10.18/index.php/RECEN/article/view/711>. Acesso em: 01 jun. 2017.

REBHOLZ-SCHUHMANN, D.; KIRSCH, H.; COUTO, F. Facts from text - Is text mining ready to deliver? *PLoS Biology*, v. 3, n. 2, p. 188-191, 2005.

REZENDE, S. O.; MARCACINI, R. M.; MOURA, M. F. O uso da Mineração de Textos para Extração e Organização não supervisionada de Conhecimento. *Revista de Sistemas de Informacao da FSMA*, v. 7, p. 7-21, 2011. Disponível em: <http://www.fsma.edu.br/si/edicao7/FSMA_SI_2011_1_Principal_3.pdf>. Acesso em: 01 jun. 2017.

RODRIGUES, R.; GONÇALO OLIVEIRA, HUGO GOMES, P. LemPORT: a High-Accuracy cross-platform lemmatizer for Portuguese. In: PEREIRA, M. J. V.; SIMÕES, A.; LEAL, J. P. (Org.). *Symposium on Languages, Applications and Technologies*. 3. ed. Dagstuhl, Germany: Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik, 2014. p. 267-274. Disponível em: <http://drops.dagstuhl.de/opus/volltexte/2014/4575/>. Acesso em: 01 jun. 2017.

ROSS, D. T. Structured analysis (sa): A language for communicating ideas. Software Engineering. *IEEE Transactions on*, IEEE, n. 1, p. 16-34, 1977.

SANCHES, M. K. *Aprendizado de máquina semi-supervisionado*: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional), Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos , 2003. 142 p. Disponível em: <https://goo.gl/NRRfwU>. Acesso em: 01 jun. 2017.

SCHEICHER, R. B. *Um método para descoberta de relacionamentos semânticos do tipo "causa e efeito" em sentenças de artigos científicos do domínio biomédico*. Dissertação (Mestrado em Ciência da Computação), Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, São Carlos, 2013. 76 p. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/591>. Acesso em: 01 jun. 2017.

SERRA, I.; GIRARDI, R.; NOVAIS, P. Reviewing the problem of learning non-taxonomic relationships of ontologies from text. *International Journal of Semantic Computing*, v. 6, p. 491-507, 2012.

SHEN, M.; LIU, D.-R.; HUANG, Y.-S. Extracting semantic relations to enrich domain ontologies. *Journal of Intelligent Information Systems*, v. 39, n. 3, p. 749-761, jun. 2012. Disponível em: <http://link.springer.com/10.1007/s10844-012-0210-y>. Acesso em: 3 mar. 2015.

SMITH, M. K.; WELTY, C.; MCGUINNESS, D. L. *OWL Web Ontology Language Guide*. *W3C Recommendation*, fev. 2004. Disponível em: <http://www.w3.org/TR/owl-guide/>. Acesso em: 16 out. 2016.

SPASIC, I. et al. Text mining and ontologies in biomedicine: Making sense of raw text. *Briefings in Bioinformatics*, v. 6, n. 3, p. 239-251, set. 2005. Disponível em: <http://bib.oxfordjournals.org/cgi/doi/10.1093/bib/6.3.239>. Acesso em: 01 jun. 2017.

SUMIDA, A.; TORISAWA, K.; SHINZATO, K. Concept-instance relation extraction from simple noun sequences using a full-text search engine. In: WEB CONTENT MINING WITH HUMAN LANGUAGE TECHNOLOGIES WORKSHOP ON THE INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC2006), 50. *Anais…*, 2006, p. 1-10. Disponível em: <https://pdfs.semanticscholar.org/dc55/46417536d4716fa3449a255da45adb423ae4.pdf>. Acesso em: 04 jun. 2017.

TABA, L. S. *Extração automática de relações semânticas a partir de textos escritos em Português do Brasil*. Dissertação (Mestrado em Ciência da Computação), Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, São Carlos, 2013. Disponível em: <https://repositorio.ufscar.br/bitstream/handle/ufscar/543/5456.pdf?sequence=1>. Acesso em: 01 jun. 2017. 98 p.

TABA, L. S.; CASELI, H. DE M. ARS – Ferramenta de anotação de relações semânticas em textos escritos em Português do Brasil. *Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional*. São Carlos: USP/UFSCar/Unesp, 2013. Disponível em: <https://goo.gl/GFX8XB>. Acesso em: 01 jun. 2017.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. Nova York: Springer-Verlag New York, Inc, 2000.

WILLE, R. Formal Concept Analysis as Applied Lattice Theory. *Concept Lattices and Their Applications*. Heidelberg: Springer Berlin Heidelberg, 1997. p. 42-67. Disponível em: <https://goo.gl/q6IYfl>. Acesso em: 01 jun. 2017.

WILLE, R. Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. *Lecture Notes in Computer Science*, v. 2636, 2005. p. 1-33. Disponível em: <http://link.springer.com/10.1007/11528784_1>. Acesso em: 01 jun. 2017.

YAN, X. Y. X.; HAN, J. H. J. gSpan: graph-based substructure pattern mining. In: IEEE INTERNATIONAL CONFERENCE ON DATA MINING. *Anais...*, n. d, p. 721-724, 2002.

APÊNDICE A
Tratamento de termos rotulados errados

Diante da informalidade dos textos em análise, a ferramenta de Processamento de Linguagem Natural cometeu alguns erros na etapa de rotulação, propagando-os na etapa de lematização. O *POS tagger* etiquetou os termos ontológicos “sala”, “escola” e “deficiência”, como sendo verbos. Diante desse equívoco, os temos foram lematizados de forma errada, gerando os valores “*salar*”, “*escolar*”, e “*deficiêncer*”.

Por se tratarem de termos ontológicos, foi necessário modificar o Algoritmo 5. A modificação está presente no Algoritmo 13, no qual é verificado se o termo possui o rótulo de verbo; em seguida, é verificado se o lema está contido na lista (“*escolar*”, “*salar*”, “*deficiêncer*”). Em caso afirmativo, o rótulo é atualizado para “n” e o *lemma* é ajustado para a forma correta (linha 10 a 14).

Algoritmo 13 isTermoOntologico alterado.

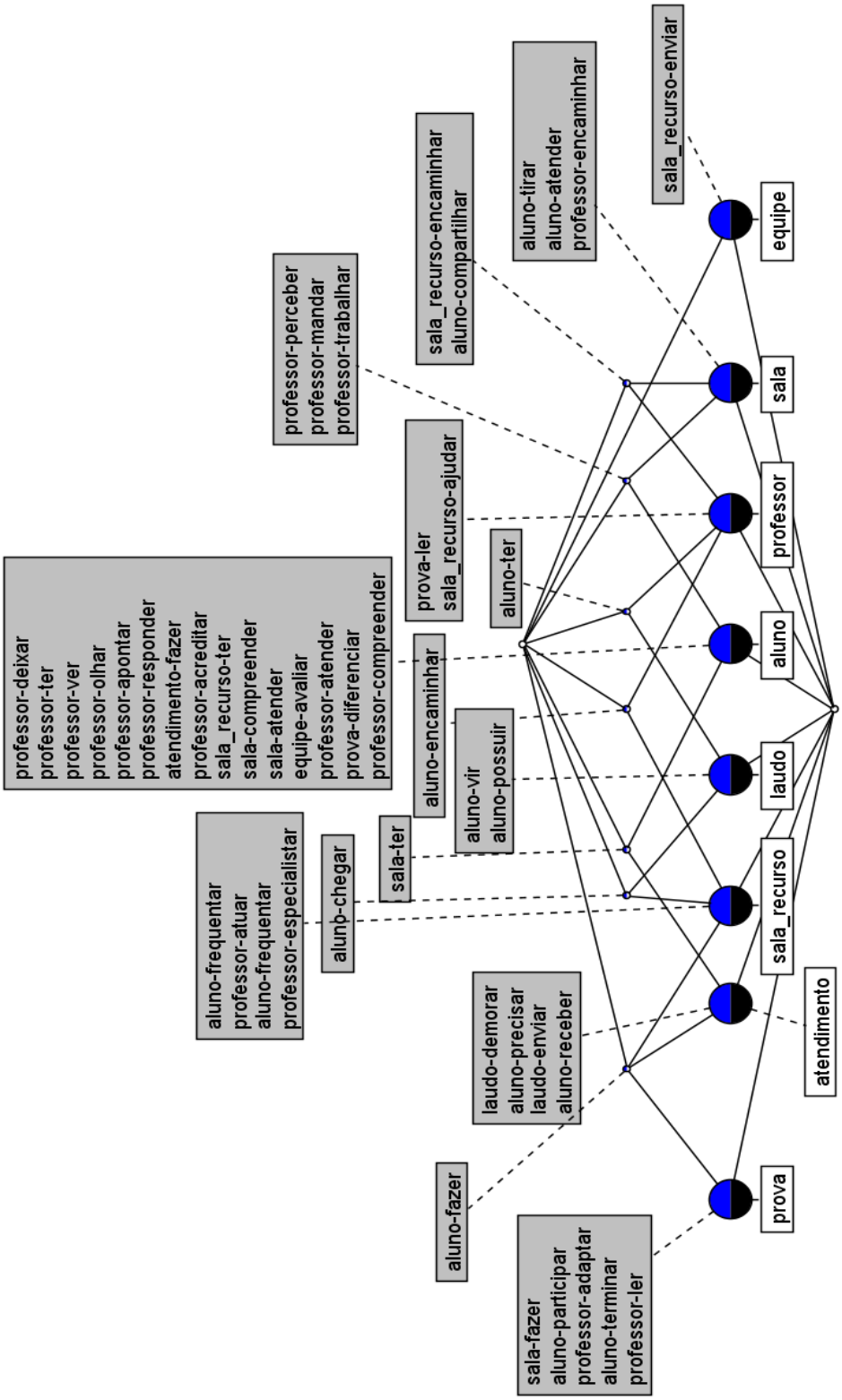
```
Input: token
Output: flag
 1: flag = false
 2: if t.rotulo igual "n" ou "adj" then
 3:    if t.termoOntologico não nulo then
 4:       flag = true
 5:    else
 6:       listasControle.limparListaControles()
 7:       contador = 0
 8:    end if
 9: end if
10: if t.rotulo contêm "v-" then
11:    if t.lemma contêm ("escolar", "salar", "deficiêncer") then
12:       t.rotulo = "n"
13:       t.lemma = ajustarLema
14:       flag = true
15:    end if
16: end if
17: return flag
```

Fonte: elaboração própria.

APÊNDICE B
Termos ontológicos selecionados para o experimento 1

- Aluno
  - Estudante
- Atendimento
- Avaliação
  - Teste
  - Prova
  - Atividade
  - Exercício
- Criança
  - Garoto/garota
  - Menino/menina
- Deficiência
  - Incapacidade
- Diagnóstico
  - Diagnose
- Educação especial
- Ensino
  - Docência
  - Educador
  - Magistério
- Equipe
- Formação
  - Instrução
- Laudo
- Orientação
- Relatório
  - Declaração
  - Relato
- Professor
  - Docente
- Prova
  - Exame
- Sala_recurso
  - Sala_de_atendimento
  - SEM

APÊNDICE C – FCA



APÊNDICE D
Tabela: experimento 2

O Apêndice D apresenta as tabelas com as marcações de ambos os especialistas.

A etapa 1 refere-se à validação dos relacionamentos existentes no FCA (Apêndice C). A avaliação é feita em três categorias: “asterisco” (*), referindo-se à identificação dos relacionamentos relevantes e que devem estar contidos na ontologia; “traço” (–), referindo-se aos relacionamentos que estão descontextualizados ou que não fazem sentido; e “símbolo de mais” (+), indicando que são relacionamentos válidos, mas não relevantes.

Na etapa 2, os especialistas têm acesso às sentenças das quais os relacionamentos foram extraídos, e ambos os profissionais reavaliaram os relacionamentos marcados como “–” e “+”.

Especialista 1	Etapa 1 – **TEMPO: 05:50**			Etapa 2 – **TEMPO: 20:32**		
	(*)	(–)	(+)	(*)	(–)	(+)
aluno-encaminhar-sala_recurso	■					
professor-adaptar-prova	■					
aluno-chegar-sala_recurso	■					
professor-mandar-aluno		■			■	
professor-perceber-aluno	■					
professor-apontar-aluno		■		■		
aluno-frequentar-sala_recurso	■					
professor-compreender-aluno	■					
sala_recurso-ajudar-professor			■	■		
aluno-freqüentar-sala_recurso	■					
sala-ter-aluno		■		■		
atendimento-fazer-aluno		■		■		
atendimento-fazer-aluno			■		■	
aluno-precisar-atendimento	■					
laudo-enviar-atendimento	■					
professor-responder-aluno	■					
professor-atender-aluno	■					
aluno-terminar-prova			■			■
aluno-chegar-laudo		■		■		
aluno-possuir-laudo	■					

sala-atender-aluno						
sala_recurso-enviar-equipe						
aluno-fazer-atendimento						
aluno-atender-sala						
aluno-encaminhar-professor						
sala_recurso-encaminhar-professor						
professor-acreditar-aluno						
professor-encaminhar-sala						
aluno-compartilhar-sala						
sala_recurso-ter-aluno						
professor-ler-prova						
professor-perceber-sala						
sala-compreender-aluno						
professor-ver-aluno						
professor-ter-aluno						
professor-atuar-sala_recurso						
aluno-vir-laudo						
sala_recurso-encaminhar-sala						
professor-especialistar-sala_recurso						
professor-olhar-aluno						
aluno-fazer-sala_recurso						
aluno-ter-laudo						
laudo-demorar-atendimento						
aluno-participar-prova						
professor-trabalhar-aluno						
equipe-avaliar-aluno						
professor-mandar-sala						
prova-diferenciar-aluno						
aluno-fazer-prova						
sala-ter-atendimento						
professor-deixar-aluno						
prova-ler-professor						
prova-ler-professor						
professor-trabalhar-sala						
aluno-receber-atendimento						
aluno-compartilhar-professor						
sala-fazer-prova						

Especialista 2	Etapa 1 – **TEMPO: 24:10**			Etapa 2 – **TEMPO: 1:28:00**		
	(*)	(–)	(+)	(*)	(–)	(+)
aluno-encaminhar-sala_recurso						
professor-adaptar-prova						
aluno-chegar-sala_recurso						
professor-mandar-aluno						
professor-perceber-aluno						
professor-apontar-aluno						

aluno-frequentar-sala_recurso						
professor-compreender-aluno						
sala_recurso-ajudar-professor						
aluno-freqüentar-sala_recurso						
sala-ter-aluno						
atendimento-fazer-aluno						
atendimento-fazer-aluno						
aluno-precisar-atendimento						
laudo-enviar-atendimento						
professor-responder-aluno						
professor-atender-aluno						
aluno-terminar-prova						
aluno-chegar-laudo						
aluno-possuir-laudo						
sala-atender-aluno						
sala_recurso-enviar-equipe						
aluno-fazer-atendimento						
aluno-atender-sala						
aluno-encaminhar-professor						
sala_recurso-encaminhar-professor						
professor-acreditar-aluno						
professor-encaminhar-sala						
aluno-compartilhar-sala						
sala_recurso-ter-aluno						
professor-ler-prova						
professor-perceber-sala						
sala-compreender-aluno						
professor-ver-aluno						
professor-ter-aluno						
professor-atuar-sala_recurso						
aluno-vir-laudo						
sala_recurso-encaminhar-sala						
professor-especialistar-sala_recurso						
professor-olhar-aluno						
aluno-fazer-sala_recurso						
aluno-ter-laudo						
laudo-demorar-atendimento						
aluno-participar-prova						
professor-trabalhar-aluno						
equipe-avaliar-aluno						
professor-mandar-sala						
prova-diferenciar-aluno						
aluno-fazer-prova						
sala-ter-atendimento						

professor-deixar-aluno						
prova-ler-professor						
prova-ler-professor						
professor-trabalhar-sala						
aluno-receber-atendimento						
aluno-compartilhar-professor						
sala-fazer-prova						