COPPE
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
UFRJ

CONSTRUÇÃO DE TAXONOMIAS SOBRE INFORMAÇÕES COMPOSTAS POR DESCRIÇÕES AMBÍGUAS COM ENRIQUECIMENTO POR MEIO DE UTILIZAÇÃO DE DICIONÁRIOS *ON-LINE*

Edival Ponciano de Carvalho Filho

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro

Março de 2015

CONSTRUÇÃO DE TAXONOMIAS SOBRE INFORMAÇÕES COMPOSTAS POR DESCRIÇÕES AMBÍGUAS COM ENRIQUECIMENTO POR MEIO DE UTILIZAÇÃO DE DICIONÁRIOS *ON-LINE*

Edival Ponciano de Carvalho Filho

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Jano Moreira de Souza, Ph. D.

Prof. Geraldo Zimbrão da Silva, D. Sc.

Profa. Priscila Machado Vieira Lima, Ph.D.

Prof. Sergio Manuel Serra da Cruz, D.Sc.

RIO DE JANEIRO – RJ - BRASIL

MARÇO DE 2015

Carvalho Filho, Edival Ponciano de

Construção de Taxonomias sobre Informações Compostas por Descrições Ambíguas com Enriquecimento por meio de Utilização de Dicionários *On-line*. / Edival Ponciano de Carvalho Filho – Rio de Janeiro: UFRJ/COPPE, 2015.

XVI, 157 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo

Tese (doutorado) – UFRJ / COPPE / Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 141-150.

1. Desambiguação e Desabreviação de descrições textuais. 2. Aprendizado de Máquina. 3. Folksonomias para enriquecimento de bases de dados. 4. Construção de Taxonomias. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Dedico este trabalho a minha esposa Marcia, aos meus filhos Felipe e Frederico e aos meus pais Marilu e Edival.

AGRADECIMENTOS

Ao finalizar esta difícil etapa, eu posso afirmar que não ninguém obtém o título de Doutor somente por esforço próprio, apesar ser imenso o esforço individual necessário. O apoio e incentivo da família, parentes e amigos foram fundamentais para chegar aqui. O título de Doutor pode ser individual, mas o mérito é de todos!

A minha esposa Marcia Marques de Queiroz Carvalho, aos meus filhos Felipe e Frederico pelo amor, atenção, paciência e pela compreensão das inúmeras vezes que estive ausente.

Aos meus pais Edival e Marilu pelo amor, incentivo e apoio incondicional durante esta e todas as outras fases da minha vida.

A minha sogra Marli, pelo contínuo apoio dado à minha família.

Ao meu orientador atual, professor Geraldo Bonorino Xexéo, pela confiança depositada, pelo incentivo, visão, dedicação, paciência e capacidade de direcionar nos momentos de dúvidas e incertezas.

Ao meu orientador inicial, professor Geraldo Zimbrão da Silva, que, apesar de sua linha de pesquisa não ser da área para onde este trabalho caminhava, apresentou valiosos conselhos e incentivou a mudança para ser orientado pelo professor Geraldo Bonorino Xexéo.

Ao professor Jano Moreira de Souza, pelo apoio e incentivos.

À professora Priscila Machado Vieira Lima e ao professor Sergio Manuel Serra da Cruz por participarem da banca e pelas contribuições ao trabalho desenvolvido.

Aos meus colegas da COPPE, pelos conselhos, apoio e incentivo.

Às secretárias da linha de Banco de dados, Ana Rabello e Patrícia Leal. Aos funcionários da secretaria do PESC, em especial, a Solange e ao Gutierrez.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CONSTRUÇÃO DE TAXONOMIAS SOBRE INFORMAÇÕES COMPOSTAS POR DESCRIÇÕES AMBÍGUAS COM ENRIQUECIMENTO POR MEIO DE UTILIZAÇÃO DE DICIONÁRIOS *ON-LINE*

Edival Ponciano de Carvalho Filho

Março/2015

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

O crescimento explosivo da quantidade e variedade de informações armazenadas em sistemas trouxe à tona o desafio da capacidade de definir uma representação textual e classificar esse volume de informações com base nas suas particularidades textuais e linguísticas. Este desafio emerge de forma mais aguda nas bases de dados que recebem informações externas oriundas de organizações diversas, agravando a questão da ambiguidade descritiva dos objetos a serem analisados e classificados. O desafio de se analisar grandes bases de dados heterogêneas está apenas começando, pois a expansão da conectividade e da variedade dos sistemas que coletam informações continua em franca expansão. Para lidar com esse desafio, diversas técnicas estão sendo desenvolvidas. Este trabalho apresenta um *framework* que utiliza algoritmos de desabreviação, agentes reconhecedores de padrões gerados pelo aprendizado de máquina e classificadores de bases textuais com a utilização de informações oriundas da *Web* a fim de identificar instâncias e gerar uma taxonomia a partir de suas respectivas descrições a partir de uma base de dados heterogênea que recebeu informações de mais de 5000 sistemas diferentes.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

TAXONOMY BUILDING OVER COMPOSITE TEXTUAL DESCRIPTIONS BY ENRICHMENT USING INFORMATION AVAILABLE THROUGH ON-LINE DICTIONARIES

Edival Ponciano de Carvalho Filho

March/2015

Advisor: Geraldo Bonorino Xexéo

Department: Computer Science and Engineering

The explosive growth, both in quantity and in variety of the information stored in systems brought up the challenge to define a textual representation for this information based on textual and linguistic peculiarities. This problem occurs more frequently in the databases that receive external information coming from different organizations, where the problem of descriptive ambiguity of the objects that should be analyzed and classified is aggravated. The challenge of analyzing large heterogeneous databases is just beginning following the expansion of connectivity and the variety of systems that collect information. To address this challenge effectively, several techniques were employed. This work presents a framework that uses desabreviation, agents-based algorithms generated by machine learning capable to recognize textual patterns classifiers, data enriching with Web information and generate a taxonomy from their respective descriptions over an heterogeneous database composed by contribution of more than 5000 different information systems.

ÍNDICE

Índice das Figuras

Índice das Tabelas

Siglas Utilizadas

Sigla	Descrição
GED	Gerência eletrônica de documentos.
SEFAZ/RJ	Secretaria de Fazenda do Estado do Rio de Janeiro
DECLAN	Declaração Econômico-Fiscal anual
GIA	Declaração Econômico-Fiscal mensal
ICMS	Imposto de Circulação de Mercadoria e Serviços
IPVA	Imposto sobre veículos automotores
ITD	Imposto sobre Transmissão Causa Mortis e Doação de Qualquer Bem ou Direito.
ECF	Emissão de cupom fiscal
MFD	Memória de Fita-detalhe
EAN	European Article Numbering
GTIN	Global Trade Item Number
PAF-ECF	Programa Aplicativo Fiscal/Emissor de cupom fiscal
PLN	Processamento de linguagem Natural
AFC	Análise de Conceitos Formais
LMF	Lexical Markup Framework
URL	Uniform resource locator
HTML	HyperText Markup Language
TF-IDF	Term Frequency – Inverse Document Frequency
WEB	Web é uma palavra inglesa que significa teia ou rede.

1 INTRODUÇÃO

Com o avanço da tecnologia de armazenamento de dados e o aumento da troca de informações absolutamente digitais entre organizações e pessoas, as informações estão sendo progressivamente cada vez mais armazenadas em formato digital. Um estudo, publicado na revista *Computerworld* calcula a quantidade de dados armazenados no mundo até 2017 em 138 *exabytes* (Computerworld , 2013).

O crescimento explosivo da quantidade de dados faz surgir o problema da limitação de recursos humanos para processá-los e transformá-los em informações úteis. Para preencher esta lacuna surgiram várias tecnologias de *Business Intelligence* como *Datawarehouses* (KIMBALL, 1998) e ferramentas de *Datamining* (HAN e KAMBER, 2006). A capacidade destas tecnologias de extrair informações relevantes depende diretamente da qualidade destes dados, ou seja, para que a informação tenha utilidade, é necessário que os dados empregados na geração destas informações sejam confiáveis, precisos e representativos. Apesar da abundância dos mesmos, a qualidade destes dados pode apresentar problemas proporcionais à quantidade disponível. Além da qualidade dos dados, a ambiguidade presente nestes é também um problema para a sua análise e classificação.

Neste contexto, contribuições para o avanço nos processos automáticos ou semi-automáticos de análise, enriquecimento e classificação dessa imensa quantidade de dados disponíveis podem trazer benefícios para a sociedade.

Esta tese apresenta um *framework* quem utiliza diversas técnicas de processamento de informações textuais como desabreviação, desambiguação, agentes reconhecedores de padrões textuais gerados por aprendizado de máquina, busca de informações na Web e geração de taxonomias para o enriquecimento e classificação das instâncias textuais de uma base de dados heterogênea com informações oriundas de mais de 5000 organizações diferentes.

1.1 Motivação

A Secretaria de Fazenda do Estado do Rio de Janeiro (SEFAZ/RJ) é o órgão estadual destinado a gerir os recursos e executar o controle fiscal do governo estadual. Os contribuintes que possuem equipamento de emissão de cupom fiscal (ECF) devem mensalmente transmitir eletronicamente para a SEFAZ os arquivos referentes aos cupons fiscais emitidos no mês

anterior (ECF-MFD), referentes a vendas ao consumidor. Dentro de cada registro de venda (cupom fiscal) são definidos os impostos recolhidos na respectiva venda, sendo que o valor calculado destes impostos depende do tipo de mercadoria vendida (tipo de produto). Como cada mercadoria vendida por meio de cupom fiscal dentro do estado do Rio de Janeiro obriga a geração e transmissão de registro de cupom fiscal para a SEFAZ/RJ, a base de dados de cupom fiscal na SEFAZ/RJ possui um crescimento de aproximadamente dois bilhões de registros por ano. Nessa base estão informações sobre as datas de venda, valores de produtos vendidos, código, descrição e também a respectiva alíquota de imposto do produto vendido. O cupom fiscal ideal deveria possuir informações padronizadas sobre o produto, com a utilização do código de produto correto e uma única descrição sem abreviações do produto por todos os contribuintes, porém, como apresentado no Apêndice C, além da falta de padronização da descrição, existe a limitação do número de caracteres a serem impressos no cupom fiscal. Estes problemas são oriundos do fato das informações serem geradas por mais de 100.000 contribuintes diferentes, sendo transmitidas e consolidadas na base de Cupom Fiscal da SEFAZ/RJ. Desta forma, para se verificar se os impostos foram corretamente aplicados na operação de venda registrada no cupom fiscal, é necessária a identificação da alíquota da mercadoria vendida no cupom fiscal. Quando não identificadas conforme os critérios estabelecidos na legislação, as mercadorias serão tributadas pela maior alíquota prevista para as operações ou prestações internas promovidas pelo estabelecimento.

Os principais problemas dos dados de cupom fiscal na SEFAZ/RJ são:

- Falta da qualidade da descrição utilizada para o produto vendido no registro do cupom fiscal.
- A alta heterogeneidade da descrição de cada produto, pois não existe padronização. Sendo esta limitada em número de caracteres pelo *layout* do cupom (raramente superior a 20 caracteres), possuindo abreviações variadas e apresentando descrições conflitantes para o mesmo código de produto.
- Existência de registros inválidos referentes a outros produtos dentro do mesmo EAN (identificador) do produto.
- Massiva quantidade de dados (3,5 bilhões de registros – cupons fiscais emitidos no ano de 2010).

No caso da SEFAZ/RJ, o cupom fiscal registra as vendas efetuadas ao consumidor no atacado. No ano de 2010 foram arrecadados quase sete bilhões de Reais com o comércio (divisão 45-47 do CNAE 2.0), conforme apresentado na Tabela 01.

Tabela 01 : Arrecadação do Estado do Rio de Janeiro por CNAE.

Divisão				Seções CNAE 2.0 - subclasses	ARREC. ACUMUL. 2011 (A)	ARREC. ACUMUL. 2010 (B)	VAR % NOM. (A/B)	VAR % REAL (A/B)
A	-	01 .. 03	-	Agricultura, pecuária, produção florestal, pesca e aqüicultura	6.369.527,47	5.538.021,17	15,01	11,36
B	-	05 .. 09	-	Indústrias extrativas	1.045.964.202,84	871.993.366,32	19,95	16,05
C	-	10 .. 33	-	Indústrias de transformação	7.307.321.281,63	6.279.080.963,66	16,38	12,49
D	-	35	-	Eletricidade e gás	3.370.607.811,35	3.233.445.871,03	4,24	0,88
E	-	36 .. 39	-	Água, esgoto, atividades de gestão de resíduos e descontaminação	12.324.842,66	7.715.057,00	59,75	55,07
F	-	41 .. 43	-	Construção	96.958.350,81	66.720.528,51	45,32	40,40
G	-	45 .. 47	-	Comércio; reparação de veículos automotores e motocicletas	7.536.414.106,57	6.699.747.209,27	12,49	8,79
H	-	49 .. 53	-	Transporte, armazenagem e correio	451.821.885,01	487.866.623,33	-7,39	-10,54
I	-	55 .. 56	-	Alojamento e alimentação	234.530.238,60	217.468.876,48	7,85	4,43
J	-	58 .. 63	-	Informação e comunicação	4.153.304.079,84	3.696.111.967,09	12,37	8,75
K	-	64 .. 66	-	Atividades financeiras, de seguros e serviços relacionados	904.323,96	650.130,82	39,10	33,86
L	-	68	-	Atividades imobiliárias	44.122,00	14.119,21	212,50	201,27
M	-	69 .. 75	-	Atividades profissionais, científicas e técnicas	43.211.022,38	48.015.305,73	-10,01	-12,98
N	-	77 .. 82	-	Atividades administrativas e serviços complementares	37.436.154,27	34.830.994,17	7,48	3,83
O	-	84	-	Administração pública, defesa e seguridade social	218.400.228,68	155.824.236,44	40,16	35,59
P	-	85	-	Educação	832.391,44	523.957,78	58,87	53,63
Q	-	86 .. 88	-	Saúde humana e serviços sociais	1.072.972,12	2.516.080,68	-57,36	-58,50
R	-	90 .. 93	-	Artes, cultura, esporte e recreação	2.204.306,65	1.058.192,96	108,31	99,24
S	-	94 .. 96	-	Outras atividades de serviços	61.911.596,02	38.635.994,29	60,24	54,92
T	-	97	-	Serviços domésticos	-	-	-	-
U	-	99	-	Organismos internacionais e outras instituições extraterritoriais	-	-	-	-
TOTAL					24.581.633.444,30	21.847.757.495,94	12,51	8,82

1.2 Definição do problema

O desenvolvimento de técnicas capazes de identificar e classificar objetos a partir de descrições textuais ambíguas com a utilização de informações oriundas da Web representa uma contribuição na área de classificação de informações e na área de folksonomia. Estas técnicas seriam aplicadas para classificar os produtos utilizados nos registros de cupom fiscal transmitidos para a SEFAZ/RJ pelos contribuintes. A classificação dos produtos gerada poderia fornecer subsídios para a validação da alíquota de ICMS utilizada no cálculo dos impostos do cupom fiscal.

A expansão da tecnologia de informação levou ao aumento do armazenamento de informações digitais sobre diversos aspectos da sociedade, provocando a necessidade de estudos que desenvolvam a capacidade de analisar, desambiguar e classificar dados textuais oriundos de fontes heterogêneas para que essa imensa quantidade de informação possa ser analisada de forma a trazer benefícios para a sociedade. Desta forma, o desenvolvimento de agentes artificiais capazes de se adaptar às diferentes características textuais e linguísticas das diversas fontes textuais pode representar uma contribuição na área do aprendizado de máquina para a construção de ontologias ou

taxonomias a partir destas informações geradas por estes agentes.

Outras organizações podem ter necessidades semelhantes, como grandes corporações do tipo de empresas aéreas, fabricantes de automóveis ou ferrovias, por exemplo. Sempre existirá a necessidade de identificação e classificação do produto a partir de seus textos associados, seja no sistema de compras de peças ou insumos para a produção ou para suprir as necessidades da área administrativa com produtos ou materiais de diversos tipos.

1.3 Objetivo do trabalho

Este trabalho tem por objetivo desenvolver um *framework* capaz de gerar uma taxonomia dos produtos vendidos no Estado do Rio de Janeiro por meio de emissão de Cupom Fiscal no ano de 2010. Este *framework* utilizará diversas técnicas de tratamento de ambiguidades e abreviações oriundas das descrições de produtos utilizadas nos registros de Cupom Fiscal, utilizará ainda técnicas de aprendizado de máquina para a extração de informações da *Web* a fim de enriquecer as relações entre os tipos de produtos e construirá uma taxonomia dos tipos de produtos identificados. O desenvolvimento das técnicas utilizadas pelo *framework* tem como base as teorias de similaridades de *strings*, aprendizado de máquina, criação e refino de taxonomias a partir de informações da *Web* (folksonomias). O desenvolvimento deste *framework* poderá representar um avanço nas técnicas construção de taxonomias e aplicação de aprendizado de máquina na extração da informação em bases textuais oriundas da *Web*.

Este trabalho tem os seguintes objetivos específicos:

- Adaptar e utilizar um algoritmo de semelhança de *strings* para desenvolver uma função de similaridade eficiente para identificar as abreviações dos termos oriundos das descrições dos produtos presentes nos registros de cupom fiscal.
- Desenvolver um agente artificial a partir do aprendizado de máquina para identificar relações em textos retornados de dicionários *online* na Língua Portuguesa.
- Avaliar o resultado da implementação deste agente artificial na identificação destas relações extraídas dos dicionários *online*.
- Utilizar as relações identificadas pelo agente artificial para a construção de uma taxonomia para classificar dados descritivos dos produtos vendidos por meio da emissão de cupons fiscais a partir dos dados da SEFAZ/RJ.

- Desenvolver processos capazes de resolver ambiguidades estruturais durante a construção da taxonomia a partir das relações extraídas da *Web*.

Devido à massiva quantidade de produtos identificados (5.472.209 Produtos com 13.725.722 de descrições diferentes), será utilizado um subconjunto destes produtos selecionando-se os produtos vendidos em um supermercado de grande porte localizado no Estado do Rio de Janeiro. As descrições utilizadas por todos os contribuintes foram associadas a este subconjunto de produtos, de forma a enriquecer textualmente cada produto. Isto em uma base de dados composta de 9.493 produtos e 1.092.397 descrições, a qual será a base utilizada pelo *framework* proposto para enriquecimento e classificação das informações dos produtos.

1.4 Premissas

Este trabalho está baseado nas seguintes premissas:

- O código de EAN (código do produto utilizado no registro de venda) representa um código identificador único de produto vendido no mundo real que pode ser classificado em função de suas respetivas descrições armazenadas nos registros de cupom fiscal;
- Os termos oriundos das descrições do produto nos registros de cupom fiscal são descritivos do respectivo produto vendido;
- A ordem em que os termos são posicionados dentro das descrições dos produtos é diretamente proporcional à sua importância na identificação da classe de cada produto vendido;
- Os dicionários *online* da Língua Portuguesa utilizados apresentam descrições consistentes sobre o termo buscado.

1.5 Necessidades específicas

O desenvolvimento de técnicas de busca e extração de informações específicas na Web permite o avanço na pesquisa de processos que utilizem estas informações como:

- Construção de taxonomias e ontologias a partir de dados oriundos da *Web*;

- Enriquecimento e classificação de informações textuais;

Desta forma, as informações extraídas da *Web* serão utilizadas para o enriquecimento e classificação das descrições textuais dos produtos registrados na base de cupons fiscais da SEFAZ/RJ. Por meio da identificação do produto vendido no cupom fiscal será possível a verificação se a alíquota do imposto foi corretamente aplicada.

1.6 Etapas específicas da implementação:

Especificamente, serão usados como base de trabalho os dados de emissão de cupom fiscal transmitidos para a SEFAZ/RJ pelos contribuintes, do ano de 2010. Para a identificação correta de cada produto do cupom fiscal será necessário:

1) Relacionar cada código de produto (EAN) a uma descrição única, consistente, com os termos desabreviados, por meio das seguintes ações:
    a. Desabreviar os termos presentes nas descrições dos produtos dos registros de cupom fiscal;
    b. Selecionar entre os termos desabreviados das descrições de cada produto, os termos que melhor o representam;
2) Utilizar o agente de aprendizado de máquina para buscar na *Web* as relações de hiperonímia dos termos selecionados das descrições dos produtos;
3) Criar automaticamente um grafo de classificação dos produtos a partir do enriquecimento dos dados originais com informações extraídas da *Web*;
4) Remover as ambiguidades do grafo de classificação de produto, de forma a se obter a taxonomia de produto.

1.7 Organização do Trabalho

Além deste capítulo introdutório, este trabalho apresenta mais seis capítulos organizados da seguinte forma:

- Capítulo 2 faz a revisão da literatura abordada nesta proposta;
- Capítulo 3 apresenta diversos trabalhos relacionados com desenvolvimento de ontologias, classificação hierárquica, Processamento de Linguagem Natural (PLN), e classificação hierárquica*;*
- Capítulo 4 descreve os principais aspectos do modelo CDDW (Classificação

de Dados com informações de Dicionários na Web) ;

- Capítulo 5 apresenta o modelo de implementação do *framework CDDW*, no qual são apresentados resultados dos experimentos executados e os aspectos teóricos do modelo;
- Capítulo 6 apresenta as conclusões e trabalho futuros.

2 REVISÃO DE LITERATURA

2.1 Ontologias

O conhecimento é o entendimento sobre uma determinada área. A inteligência artificial é o desenvolvimento de programas que possam representar, codificar e processar o conhecimento sobre problemas (fatos, regras e estruturas).

Na área de inteligência artificial, o armazenamento de conhecimento é o processo de guardar o conhecimento, codificado em um formato manipulável, ou seja, em formato computacional. A recuperação de conhecimento é o processo inverso, obtendo o conhecimento armazenado quando este é necessário. O raciocínio (de inteligência artificial) consiste em utilizar o conhecimento e estratégias de resolução de problemas para validar, inferir e assim obter conclusões e explicações. Um importante pré-requisito para os processos anteriormente descritos é a aquisição de conhecimento, que consiste na coleta, organização e estruturação do conhecimento em tópicos, domínios ou áreas para serem gravados no sistema (ESCHENBACH e GRÜNINGER, 2008), (RUSSEL e NORVIG, 2009).

Existem diferentes formas de se representar o conhecimento, cada forma possui vantagens e desvantagens de capturar determinados tipos particulares do conhecimento humano. Uma categoria especial destas técnicas, utilizada para representar o conhecimento humano inexato e incerto é denominada ontologia Originária na filosofia, ontologia é um conceito usado para designar uma visão geral do mundo e a organização dos seres. Uma ontologia permite descrever e organizar a informação a partir de um grupo de conceitos, tornando-a livre de ambiguidade e passível de formalismo (GUARINO, 1998). Na tentativa de auxiliar o processamento e a representação da informação na *Web*, as ontologias vêm sendo empregadas como um mecanismo que permite oferecer consistência na sua representação em um ambiente aberto, heterogêneo e ubíquo como a Internet. (BORTH 2011)

Na ciência da computação, uma das definições mais comuns encontradas na literatura é dada por Gruber (GRUBER, 1993). O autor define ontologia como "*Uma especificação formal explícita de uma conceitualização compartilhada*". Há outras definições importantes, como a postulada por Chandrasekaran (CHANDRASEKARAN *et al*, 1999), que afirma que uma ontologia constitui uma visão de um artefato designado a usos específicos que permite capturar o domínio de conhecimento

de uma forma genérica, fornecendo um entendimento sobre o que está sendo explorado.

Os conceitos a seguir foram sumarizados a partir de Gruber (GRUBER, 1993):

- Classes: geralmente são organizadas em forma de taxonomia e representam algum tipo de interação da ontologia com o domínio.
- Relações: representam os tipos de interações entre as classes (elementos) do domínio.
- Instâncias: representam elementos específicos, os próprios dados das ontologias (geralmente estão ligadas a uma classe, como instância de uma classe).
- Funções: eventos que podem ocorrer no contexto da ontologia.
- Axiomas: são utilizados para modelar sentenças verdadeiras.

As ontologias são divididas em vários tipos de acordo com o seu grau de generalidade. Guarino (GUARINO 1998) menciona alguns desses tipos, conforme apresentado na Figura 01:

- **Ontologias gerais** (*top-level ontology*): possuem definições abstratas para a compreensão de aspectos do mundo, como, por exemplo, processos, espaços, tempo, coisas, seres, etc.
- **Ontologias de domínio** (*domain ontology*): tratam de um domínio específico de uma área genérica, como, por exemplo, uma ontologia sobre família.
- **Ontologias de tarefa** (*task ontology*): tratam de tarefas genéricas ou atividades (como diagnosticar ou vender).
- **Ontologias de aplicação** *(application ontology*): têm como objetivo solucionar um problema específico de um domínio, normalmente referenciando termos de uma ontologia de domínio.

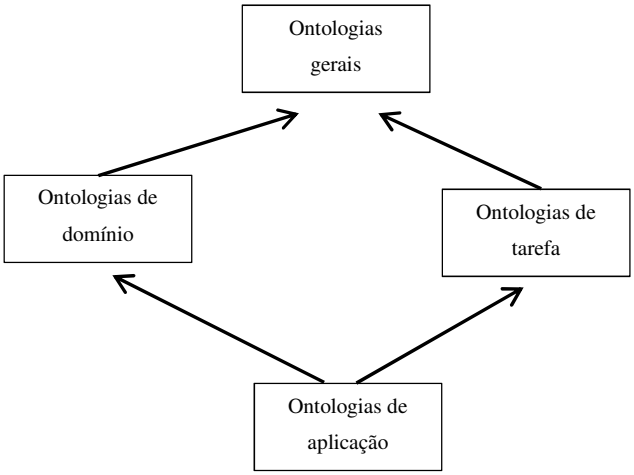


Figura 01 : Tipos de ontologia. Adaptado de Guarino 1998.

Com o uso das ontologias em sistemas, pode-se representar o domínio de conhecimento com o uso do vocabulário de forma simples. Conforme apresentado na Figura 02, esses conceitos presentes no vocabulário podem ser relacionados por relações do tipo: "*é um*", "*possui*", etc. O trabalho de Mithun Balakrishna, Dan Moldovan, Marta Tatu e Marian Olteanu (BALAKRISHNA *et al*, 2010), listou 26 tipos de relações semânticas. O vocabulário utilizado possui, como sua sustentação, os conceitos, evitando que sejam interpretados de maneira ambígua pelas aplicações semânticas (BREITMAN, HOWARD e CASANOVA, 2005). Além do mais, as ontologias podem ser estruturas eficientes para auxiliar a compreensão do usuário, podendo fornecer uma exibição diferenciada dos conceitos na recuperação de informação.

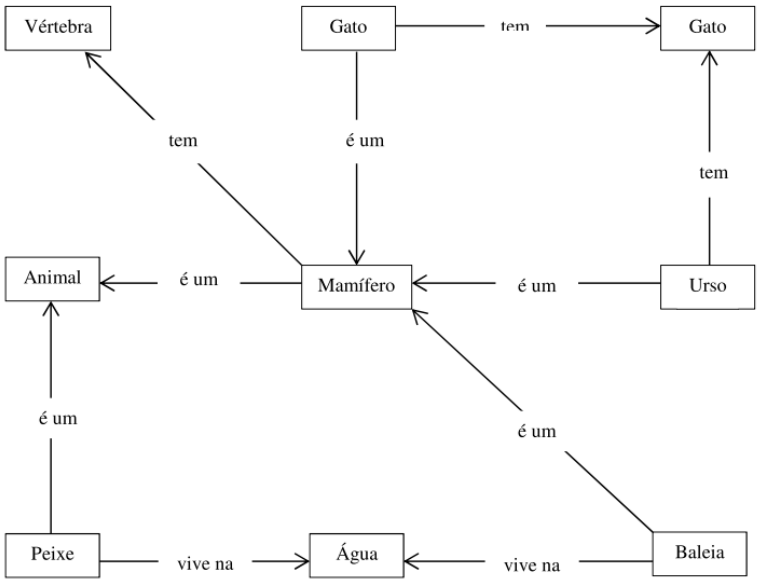


Figura 02 : Um exemplo de uma ontologia com alguns conceitos e relações (adaptado de GUARINO 1998).

2.1.1 Construção de ontologias

A construção não automática de ontologias é uma tarefa complexa, para que possa atingir resultados de qualidade, devem-se seguir metodologias e critérios bem definidos. Segundo Gruber (GRUBER, 1993), a ontologia resultante deve possuir clareza e coerência além de possuir um compromisso ontológico mínimo, o que significa que deve definir apenas os termos extremamente necessários para que as informações possam ser compartilhadas.

Como uma ontologia tem como objetivo a contextualização de um domínio de conhecimento, a primeira atividade a ser executada é a definição do domínio e do escopo (NOY E MACGUINNES, 2001).

A construção de uma ontologia deve seguir as seguintes etapas (SOFIA E MARTINS, 2004):

- **Especificação**: Identificar o propósito e o escopo da ontologia, respondendo às perguntas "*Por que a ontologia sendo construída?"* e "*Quais são as suas finalidades e usuários finais?*", para verificar o propósito e escopo.
- **Conceitualização**: Descrever, em um modelo conceitual, a ontologia a ser construída, para que ele atenda à especificação definida na etapa anterior. Diferentes metodologias propõem o uso de diferentes modelos conceituais, por exemplo: mapas mentais (*Mind Maps*) ou modelos semiformais como diagramas de relações binárias. O modelo conceitual de uma ontologia consiste de conceitos no domínio e as relações entre esses conceitos. Podem existir relacionamentos com conexões mais fortes entre grupos de conceitos. Estes grupos de conceitos altamente ligados geralmente correspondem às diferentes módulos (subontologias) em que o domínio pode ser decomposto.
- **Formalização**: Transformar a descrição conceitual para um modelo formal, ou seja, a descrição do domínio encontrado na etapa anterior é escrita de uma maneira mais formal, embora ainda não na sua forma final. Conceitos são geralmente definidos através de axiomas que restringem as interpretações possíveis para o significado desses conceitos. Conceitos são geralmente organizadas hierarquicamente através de uma relação estruturante, como “é-um” (classe superclasse, instância de classe) ou “parte de”.

- **Implementação**: Implementar a ontologia formalizada em uma linguagem de representação do conhecimento.
- **Manutenção**: atualizar e corrigir a ontologia implementada.

Depois da criação da ontologia existem as atividades que devem ser executadas durante o seu ciclo de vida:

- **Aquisição de conhecimento**: Adquirir conhecimento sobre o assunto ou usando técnicas de elicidação de especialistas de domínio ou referindo-se a bibliografia relevante. Várias técnicas podem ser utilizadas para adquirir conhecimento, tais como *brainstorming*, entrevistas, questionários, análise de texto e técnicas indutivas.
- **Avaliação**: Julgar tecnicamente a qualidade da ontologia.
- **Documentação**: Relatório que foi feito, como foi feito e por que foi feito. A documentação associada com os termos representados na ontologia é particularmente importante, não só para melhorar a sua clareza, mas também para facilitar o uso, manutenção e reutilização desta.

2.1.2 Taxonomias

Taxonomias são esquemas hierárquicos de classificação, nos quais os metadados são organizados em estruturas de árvores (DOTSIKA 2012). Uma taxonomia pode ser definida pelas seguintes regras (RESENDE e MARTINS 2006):

- Cada item da taxonomía deve possuir apenas um pai. Essa característica define o processo de generalização de cada item.
- Um item pode possuir *n* filhos. Este é um conceito básico e obrigatório para todas as taxonomias.
- Se um item possuir filhos ele não pode ser excluído da taxonomia para que não seja descartada toda a sua especialização. Para realizar a exclusão, é necessário realocar os filhos definindo seus novos pais ou excluindo-os da taxonomia.
- Os itens do nível mais alto da taxonomia (generalizarão máxima ou raiz) não possuem pai.
- Os itens do nível mais baixo da taxonomia (especialização máxima ou folha) não possuem filhos.

A construção de taxonomias pode ser feita de várias formas (USCHOLD e GRUNINGER 1996):

- ***Top-down*.** Na qual os elementos são montados a partir dos níveis superiores sendo progressivamente são adicionados à estrutura os elementos dos níveis inferiores. Apesar de se poder controlar melhor o nível de detalhe da taxonomia resultante, determinados elementos de níveis inferiores pode ser erroneamente posicionados em níveis superiores.
- ***Bottom-up***. A estrutura é montada a partir dos níveis compostos pelos elementos inferiores, sendo os elementos intermediários e superiores adicionados progressivamente à estrutura. Este método resulta em uma taxonomia com um alto nível de detalhe.
- ***Middle-out*.** A estrutura é montada a partir dos níveis intermediários, acrescentando-se elementos nos níveis inferiores e superiores. Esta abordagem permite conciliar os benefícios do detalhamento com os níveis superiores melhor posicionados hierarquicamente.

2.3 Processamento de Linguagem Natural

O processamento de linguagem natural (PLN) é uma técnica que abrange um conjunto de métodos formais para analisar textos e gerar frases em idioma humano. Estes métodos formais situam-se em diversos níveis de entendimento, conforme listados abaixo:

- Fonético e fonológico: pelo relacionamento das palavras com os sons;
- Morfológico: pela construção das palavras a partir de primitivas;
- Sintático: pelo relacionamento das palavras entre si;
- Semântico: pelo relacionamento das palavras com seus significados e de como são combinados para formar os significados das sentenças;
- Pragmático: do uso de frases e sentenças em diferentes contextos, afetando o significado.

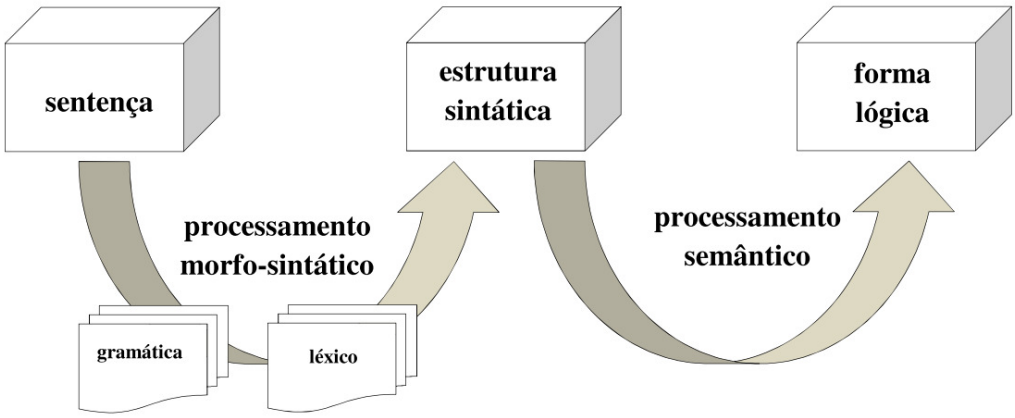


Figura 03 : Transformações da sentença na estrutura sintática e na forma lógica. Adaptado de Gonzalez e Lima, 2003

Conforme a representação da Figura 03, a análise morfológica e a análise sintática tratam da constituição das palavras e de grupos de palavras que formam os elementos de expressão de uma língua. Enquanto o analisador léxico-morfológico lida com a estrutura das palavras e com a classificação das mesmas em diferentes categorias, o analisador sintático trabalha em nível de agrupamento de palavras, analisando a constituição das frases. (GONZALEZ e LIMA, 2003).

A análise sintática (*parsing*) é o procedimento que avalia o texto, verificando as respectivas regras gramaticais aplicadas com a finalidade de gerar uma estrutura de árvore que represente a estrutura sintática da sentença analisada. Se a sentença for ambígua, o analisador sintático (*parser*) deverá obter todas as possíveis estruturas sintáticas que a representam.

A análise semântica está relacionada ao significado do conjunto resultante formado pelo conjunto de palavras. O processamento semântico é um dos maiores desafios do processamento de linguagem natural.

As palavras podem se associar através de dois tipos de relações: relações paradigmáticas e relações sintagmáticas ou colocações. As relações paradigmáticas principais são: sinonímia (é a mesmo sentido de significados entre duas palavras ou mais), antonímia (palavras que formam o oposto), hiponímia (palavra que indica cada parte ou cada item de um todo), hiperonímia (palavra que dá idéia de um todo, do qual se originavam várias partes ou ramificações), meronímia (relação entre um holônimo, que representa o todo, e um merônimo, que representa a parte), holonímia (relação de hierarquia semântica entre palavras em que uma — holónimo — refere a unidade e a outra — merónimo — refere uma parte dessa mesma unidade ou todo), implicatura (uma expressão ou um enunciado, exprimir, incluir um significado adicional ao significado literal do que diz) e

pressuposição (Relação de sentido entre duas proposições em que se P é verdadeira, Q também é verdadeira, mas se P for falsa, Q mantém-se verdadeira) (YULE, 1998).

Quanto ao nível de processamento do PLN, temos os seguintes tipos de ambiguidade:

- Sintática, quando a ambiguidade é encontrada já em nível sintático;
- Semântica, quando a ambiguidade aparece somente em nível semântico.

Diversas estratégias podem ser utilizadas na implementação da PLN, a seguir são apresentadas algumas estratégias de implementação mais comuns:

- Etiquetagem gramatical: Consiste em identificar a categoria gramatical de cada componente do texto (uma palavra pode ser associada a mais de uma categoria). Existem duas categorias principais:
  - Palavras funcionais: por exemplo: artigo, adjetivo, preposição, substantivo e etc.
  - Palavras de conteúdo: diz respeito a nomes, verbos, adjetivos e etc.
- Etiquetagem sintática: Indica a função sintática das palavras no texto, como por exemplo: sujeito, objeto direto, objeto indireto e etc.
- Padrões gramaticais: Seria o reconhecimento de determinados padrões sintáticos fortes como “através de”, “de acordo com”, “faz parte” e etc.
- Normalização morfológica: Consiste na redução de itens lexicais com o objetivo de representar classes de conceitos. A seguir dois processos comuns de normalização morfológica:
  - *Stemming*: Consiste na redução de todas as palavras ao seu radical.
  - Redução para a forma canônica (*lemmatization*), na qual os verbos são reduzidos ao infinitivo e adjetivos e substantivos são reduzidos à forma original (masculino e singular).
- Eliminação das *stopwords*: *stopwords* são palavras funcionais como: artigos, conectivos e preposições. A eliminação destes tipos de palavras para se aplicar o “*Bag of words*” pode gerar o risco de perda da estrutura composicional das frases.
- Reconhecimento de entidades nomeadas (*Named Entity Recognition*). Este processo efetua a identificação das entidades mencionadas no texto, iniciando com a seleção de palavras ou sentenças candidatas a entidades nomeadas.
- Classificação de entidades nomeadas (*Named entity classification*). Consiste na

identificação das classes ontológicas das entidades mencionadas. Conforme as classes apresentadas na Figura 04.

- Análise de constituintes: tem por objetivo analisar as ligações entre as palavras e identificar as relações entre as mesmas para a identificação do significado sintático das frases, normalmente representado por árvores sintáticas.

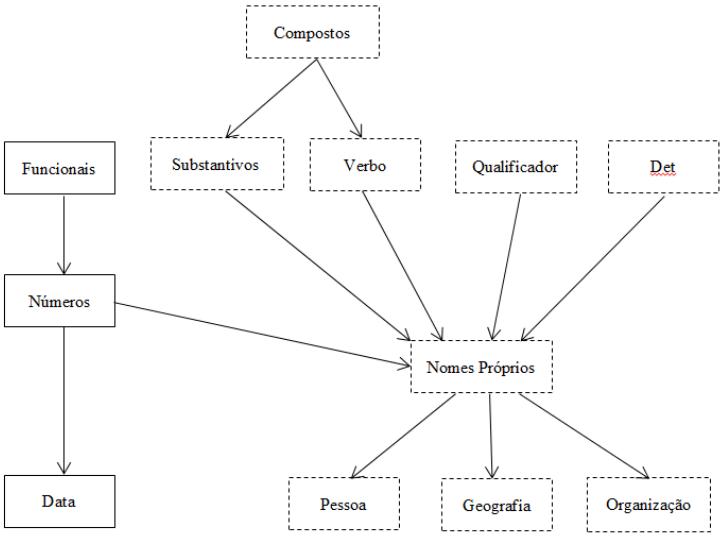


Figura 04 : Classes ontológicas das entidades nomeadas. Adaptado de Aranha 2007.

Tabela 02 : Estruturas sintagmáticas (Souza, 2005).

Sintagma	Descrição
SN = O	O sintagma nominal equivale à oração
SN = N	O sintagma nominal é um nome
SN = Det + N	O sintagma nominal é formado por um determinante mais um nome
SN = SN + O	Um novo sintagma nominal é formado com a junção de um sintagma nominal e uma oração
SV = V	O sintagma verbal é formado pelo verbo
SV = V + SN	O sintagma verbal é formado pelo verbo mais um sintagma nominal

O modelo sintagmático é baseado no estudo das hierarquias de componentes, denominados sintagmas, que foram uma oração. Sintagmas são certos grupos de unidades que fazem parte de sequências maiores, mas que mostram certo grau de coesão entre eles, conforme apresentado na Tabela 02 e na Figura 05. (SOUZA, 2005).

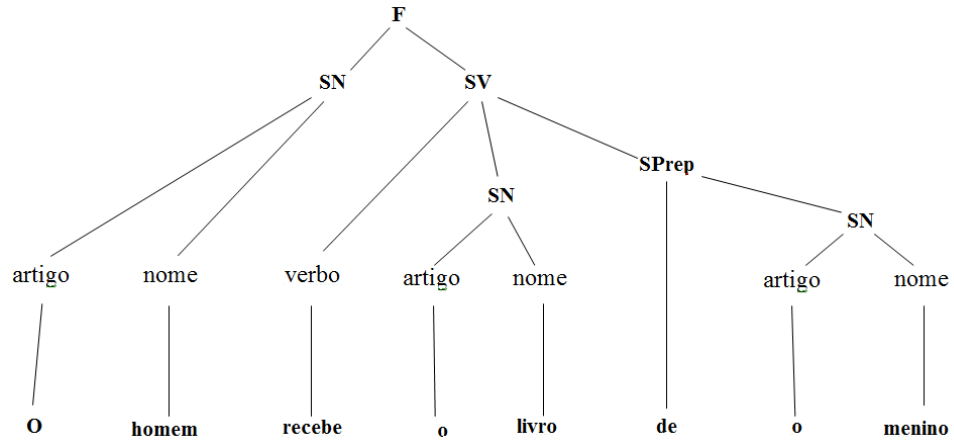


Figura 05 : Árvore sintática da frase "o homem recebe o livro do menino". Adaptado de Souza, 2005.

2.4 Princípio da incerteza

O trabalho de Jousselme, Maupin e Bissé, publicado em 2003 (JOUSSELME, 2003), propõe uma discussão sobre o papel da incerteza, apresentando uma visão geral das principais tipologias de incerteza encontradas na literatura recente. Esta grande variedade de concepções de incerteza é um consequência da riqueza intrínseca e ambiguidade de linguagem natural. Com o objetivo de esclarecer conceitos e delimitação do escopo, são apresentadas definições encontradas na literatura

recente:

- ***Situation Awareness*** (SAW): Seria uma situação de consciência ou um estado de conhecimento que envolve a percepção, a compreensão e a projeção dos elementos da situação.
- ***Situation Analysis*** (SA): A Análise de Situação é "um processo, o exame de uma situação, os seus elementos e suas relações, para fornecer e manter um produto, ou seja, um estado de SAW para o tomador de decisão". Para uma dada situação do processo SA cria e mantém um modelo mental da representação da situação.
- ***Situation Model*** (SM): A Análise de Situação carece de integração interna e externa da situação real. O Modelo da Situação representa os elementos da situação e seus relacionamentos com a finalidade de ajudar na compreensão e na resolução do problema em questão. Para atingir este objetivo deve-se levar em conta as seguintes tarefas: Uma situação de percepção composta da aquisição de elementos da situação, referências comuns, percepção da origem da Incerteza, gestão e refinamento da percepção da situação. Os métodos para a modelagem e processamento de incerteza naturalmente diferem de uma comunidade científica para outra. Por exemplo, sobre a percepção, teoria da probabilidade é adotada pela maioria dos engenheiros elétricos. No caso do raciocínio, as abordagens lógicas são bastante utilizados pela comunidade de Inteligência Artificial (AI) e filósofos.

Definição de incerteza: A incerteza tem dois significados principais na maioria dos dicionários clássicos (JOUSSELME, 2003):

I - Incerteza como um estado de espírito;

II - A incerteza como uma propriedade física da informação.

O primeiro significado (I) refere-se ao estado de espírito de um agente que não possui a informação necessária ou o conhecimento para tomar uma decisão, o agente está em um estado de incerteza: "Eu não tenho certeza de que esse objeto é uma tabela".

O segundo sentido (II) refere-se a uma propriedade física, o que representa a limitação dos sistemas de percepção: "O comprimento desta tabela é incerto ".

Krause e Clark (KRAUSE e CLARK, 1993) distinguem dois aspectos de incerteza: unários (incerteza aplicada a proposições individuais) e incerteza teórica (incerteza aplicada a conjuntos de proposições). Ambas as categorias podem levar tanto ao conflito (conhecimento conflitantes) quanto à ignorância (falta de conhecimento). Como subcategorias, encontramos imprecisão, confiança, propensão, equívoco, ambiguidade, anomalia, incoerência incompletude, e irrelevância. Este modelo está reproduzido na Figura 06:

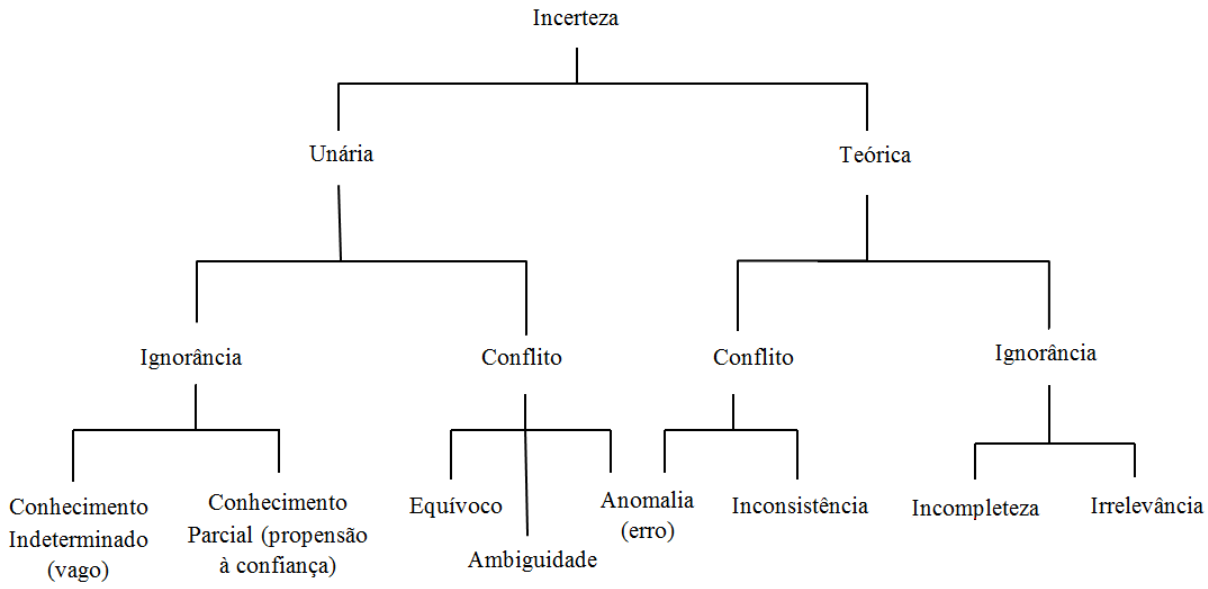


Figura 06 : Tipos de incerteza. Adaptado de Krause e Clark, 1993.

A princípio, as diferentes facetas de incerteza são misturadas e devem ser discutidas separadamente. A seguir, são enumerados quatro itens de interesse sobre incerteza:

1. Definições de incerteza: Estado de espírito (I) ou propriedade física de um pedaço de informação (II);
2. Interpretações epistêmicas de incerteza (objetivo versus subjetiva), ou seja, formas de obtenção de informações sobre a incerteza ou a medida da incerteza na situação;
3. Tipos de incerteza (imprecisão, ausência de informação e conflito);
4. Teorias matemáticas de incerteza (matemática, representação, formalização, medidas *fuzzy* e axiomização).

2.5 Inteligência Artifical

A AI (Inteligência Artifical) engloba um vasto campo de conhecimento, do mais genérico (aprendizado e percepção) até o mais específico como jogos de Xadrez ou dirigir um carro. (RUSSEL e NORVIG, 2009). A AI é organizada em quatro categorias:

Raciocinando Humanamente: As máquinas com consciência e capacidade de avaliação poderiam executar tarefas como tomar decisões, resolver problemas, aprender e *etc.* **Agindo Humanamente**: Seria a arte de criar máquinas que executem funções que requeiram inteligência quando executadas por pessoas;

Raciocinando Racionalmente: Seria o estudo computacional que torna possível perceber, raciocinar e agir;

Agindo Racionalmente: Seria o estudo direcionado no comportamento inteligente de agentes artificiais.

Define-se como **agente** algo que age ou executa uma ação, como todos os computadores executam ações, mas os agentes computacionais podem ir além: operar autonomicamente, perceber o ambiente, persistir após um período longo de tempo, se adaptar às mudanças e criar e perseguir objetivos. Um **agente racional** age para atingir o melhor resultado ou, quando existe incerteza, o melhor resultado esperado. O agente racional tem duas vantagens sobre as outras abordagens: Primeiro, é mais abrangente do que as "regras de raciocínio" porque a inferência correta é um dos vários mecanismos possíveis para se obter a racionalidade. Segundo, é mais fácil desenvolver cientificamente o raciocínio do que tentar copiar o comportamento ou raciocínio humano. Para que um agente racional execute as ações esperadas, deve-se definir **métrica de performance e objetivos**, para que este possa executar e monitorar as suas ações de forma a atender os objetivos esperados (RUSSEL e NORVIG, 2009). Desta forma, podemos destacar quatro coisas que o agente racional necessita:

- A métrica de performance que definirá o critério de sucesso;
- O conhecimento prévio do agente sobre o ambiente;
- As ações que o agente pode executar;
- A percepção de sequência e tempo pelo agente.

Os agente podem ser classificados em quatro tipos:

Single reflex Agents: É o tipo mais simples de agente. Seu comportamento é baseado na sua

percepção imediata do ambiente, ignorando o histórico percebido anteriormente, apresentado na Figura 07:

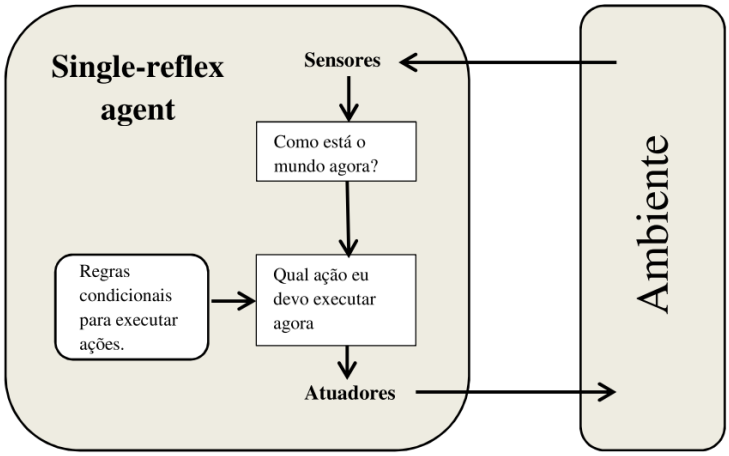


Figura 07: ***Single-reflex agent*****. Adaptado de RUSSEL e NORVIG, 2009.**

Model-based reflex agents: Este tipo mantém um status interno baseado no histórico percebido que é utilizado junto com as percepções imediatas para orientar o seu comportamento. Para atualizar o status interno é necessário dois tipos de conhecimento a serem codificados no agente: Primeiro, é necessário alguma informação sobre o ambiente se comporta independentemente do agente. Segundo, é necessário saber como as ações do agente afetam o ambiente. Este conhecimento simples sobre "Como o mundo funciona" é denominado de **modelo** do mundo. A Figura 08 apresenta uma representação de um *Model-based reflex agent*.

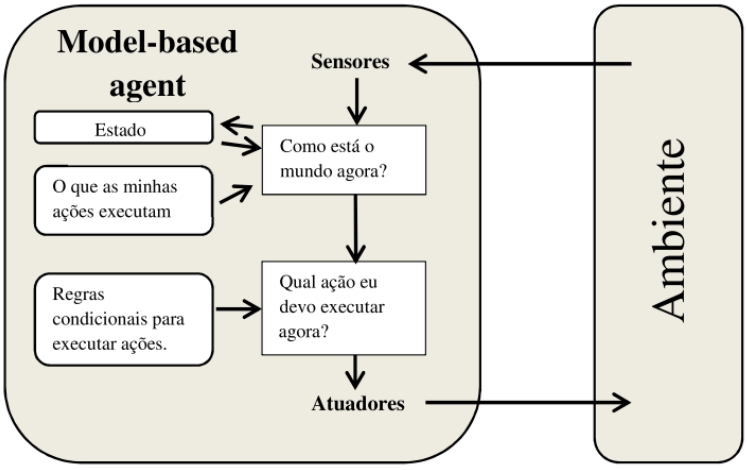


Figura 08: ***Model-based agent*****. Adaptado de RUSSEL e NORVIG, 2009.**

Goal-based agents: O conhecimento sobre o status atual do ambiente nem sempre é

suficiente para se decidir quais ações devem ser executadas. Como, por exemplo, em uma bifurcação, o veículo deverá escolher o caminho da esquerda ou o da direita. Desta forma, os agentes necessitam de um certo tipo de **objetivo** (*goal*). O agente deve possui um **modelo combinado com o objetivo** para decidir quais ações devem ser tomadas em determinado momento, conforme apresentado na Figura 09.

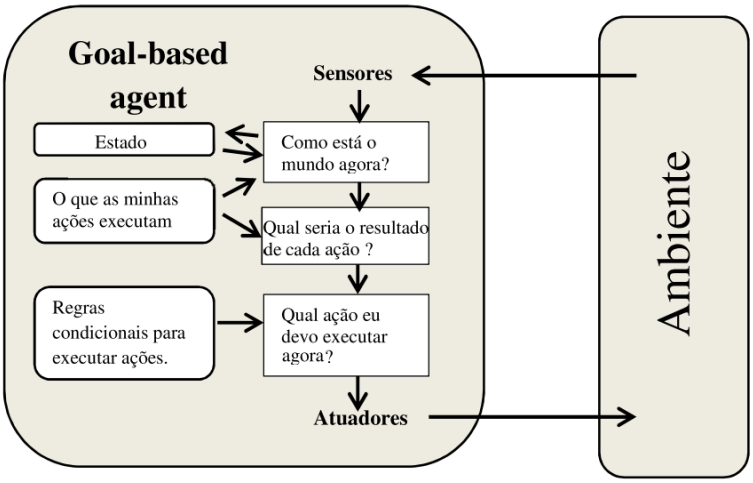


Figura 09: ***Goal-based agent*****. Adaptado de RUSSEL e NORVIG, 2009.**

Utility-based agents: Nem sempre a definição de um objetivo direciona as ações do agente para alcançá-lo por podem existir diversas maneiras de se atingir o objetivo, algumas maneiras bem objetivas e simples e outras bastante dispendiosas de recursos. O conceito de **utilidade** (*utility*) apresentado por Russel e Norvig (RUSSEL e NORVIG 2009) se baseia na escolha das ações que maximizem as sua medida externa de performance, conforme a representação de um *Utility-based Agent* na Figura 10.

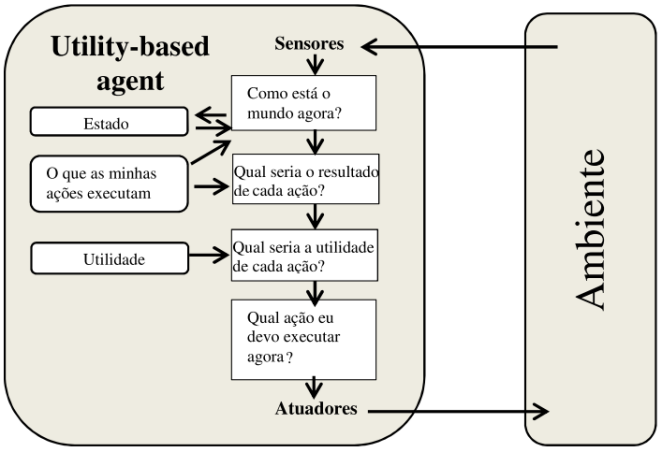


Figura 10: ***Utility-based agent*****. Adaptado de RUSSEL e NORVIG, 2009.**

Turing (TURING, 1950), considerou a ideia de programar a inteligência computacional manualmente, mas estimou o volume de trabalho necessário, considerando inviável a geração manual da inteligência computacional, concluindo que a solução para este problema seria a criação das máquinas capazes de aprender (*learning machines*). O aprendizado de maquina tem a vantagem do agente começar a operar sem conhecimento nenhum e cada vez mais ficar mais competente à medida que adquire conhecimento. O agente de aprendizado pode ser divido em quatro componentes conceituais. O componente mais importante é o módulo **de aprendizado**, responsável pelo aperfeiçoamento do agente. O módulo **de performance** é o responsável pela seleção das ações externas, que percebe, toma decisões e age. O elemento de aprendizado utiliza o retorno do **elemento crítica** sobre como o agente está agindo e determina como o elemento de performance deve ser modificado para que o agente possa agir melhor no futuro. O último módulo de um agente de aprendizado é o **gerador de problemas**, que é responsável por sugerir ações para a obtenção de novas experiências. A Figura 11 representa os componentes de um agente de aprendizado:

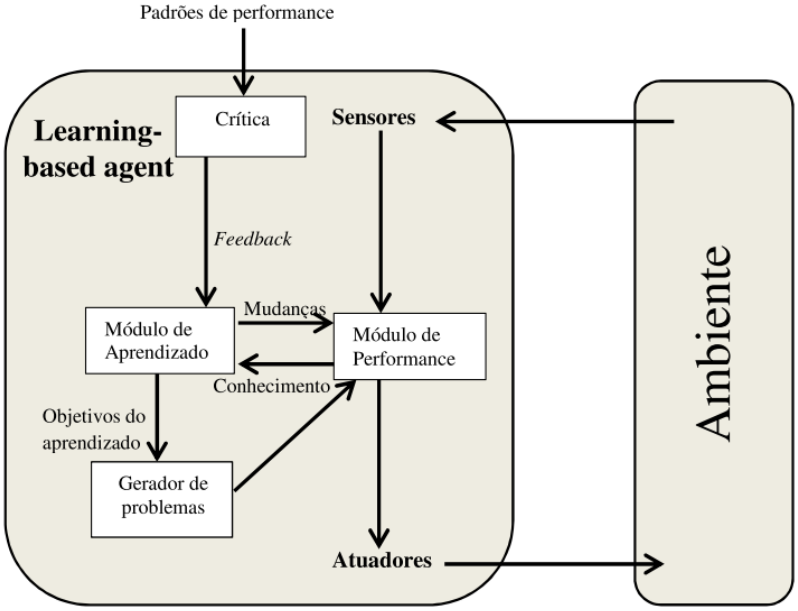


Figura 11: Agente de aprendizado. Adaptado de RUSSEL e NORVIG, 2009.

2.6 Aprendizado de Máquina

Aprendizado de Máquina (*Machine Learning* em inglês) é a área de Inteligência Artificial

que tem por objetivo o desenvolvimento de técnicas computacionais que utilizem o processo de aprendizado para a execução das tarefas (BISHOP, 2007).

O principal princípio da teoria do aprendizado é que "*Qualquer hipótese que for completamente errada terá grande probabilidade se ser identificada a partir dos resultados de um pequeno número de exemplos. Do outro lado, uma hipótese que for consistente com uma qrande quantidade de exemplos terá pequena probabilidade de ser considerada errada, tendo uma maior probabilidade de ser considerara correta*". (RUSSEL e NORVIG 2009).

Um tipo particular de aprendizado de máquina é o aprendizado indutivo, que consiste em uma técnica na qual o aprendizado de uma função é adquirido a partir de exemplos (pares de entrada e saída fornecidos para o processo de aprendizado) .

Um tipo especial de aprendizado indutivo é o processo de inferência indutiva realizada com o fornecimento de exemplos (MICHALSKI, CARBONELL e MITCHELL. 1983), no qual são gerados os padrões gerais por meio de exemplos.

Pode-se formalizar o problema de aprendizado indutivo utilizando exemplos da seguinte forma (BRATKO *apud* BATISTA, 2003):

> *Seja U o conjunto universal dos objetos, isto é, todos os objetos que aprendiz pode encontrar. Não existe limite, a princípio, para a cardinalidade de U. Um conceito C pode ser formalizado como sendo um subconjunto de objetos em U, assim:*
>
> $$C \subset U$$
>
> *Aprender um conceito C significa aprender a reconhecer objetos em C , ou seja, uma vez que o conceito C é aprendido, para qualquer objeto* $x \in U$, *o sistema é capaz de reconhecer se* $x \in C$.

Para que o processo proposto seja capaz de reconhecer padrões similares, mas não completamente iguais aos padrões dos exemplos fornecidos, os padrões identificados a partir dos exemplos deverão passar por generalizações e especializações para serem capazes de reconhecer padrões similares mas não completamente iguais aos exemplos fornecidos, (SHAW e GENTRY 1990). Desta forma, com a utilização de generalizações, o processo proposto será capaz de possível reconhecer $x \in U$, mesmo que $x \notin C$, ou seja, localizar reconhecer padrões em dados que não pertençam ao conjunto formado pelos exemplos utilizados no aprendizado indutivo.

O aprendizado indutivo por exemplos é dividido em supervisionado e não supervisionado. No supervisionado são utilizados exemplos contendo a entrada e a saída do algoritmo. No

aprendizado não supervisionado são utilizados exemplos contendo apenas a entrada, sendo a saída gerada pelo algoritmo. Para o aprendizado supervisionado não ficar limitado aos exemplos fornecidos, será necessário aplicar a generalização e especialização aos conceitos oriundos dos padrões identificados nos exemplos de forma que padrões similares, mas não iguais, possam ser identificados com sucesso (SHAW e GENTRY 1990).

2.7 O Aprendizado a Partir de Exemplos

Um agente é um agente de aprendizado se for capaz de melhorar a sua performance no futuro com base. Este aprendizado do agente pode ser trivial como um aspirador de pó aspirando todo o chão da casa ou profundo como o exibido por Albert Einstein. A razão de um agente aprender a partir de exemplos é o fato de se muito custoso ou até impossível prever todas as situações e suas respectivas ações a serem tomadas (RUSSEL e NORVIG, 2009).

Qualquer componente do agente pode ser melhorado a partir de dados. Estes melhoramentos e técnicas usadas para isso dependem de quatro fatores principais:

- Qual componente será melhorado?
- Qual é o conhecimento prévio do agente?
- Qual é a representação utilizada para os dados e os componentes?
- Qual *feedback* é disponibilizado para o aprendizado?

O aprendizado de regras a partir de pares de informação formados pela entrada-saída é denominado aprendizado indutivo. Existem três tipos de *feedback* que determinam três principais tipos de aprendizado:

Aprendizado não supervisionado: Quando o agente aprende padrões a partir do *input* sem dispor do respectivo *feedback*. A forma mais simples do aprendizado não supervisionado é a geração de clusters.

Aprendizado baseado em recompensa/punição (*reinforcement learning*), neste caso o agente aprende a partir de um retorno (positivo ou negativo) aoo final de sua execução, para poder avaliar se as ações executadas foram bem sucedidas ou não.

Apendizado supervisionado: Neste caso, os agentes recebem pares de exemplos formado por *input-output* e executam as ações a partir de padrões gerados por este conjunto de pares de

exemplos.

Aprendizado semi-supervisionado: Neste caso o aprendizado é obtido com um pequeno conjunto de exemplos formados por pares de *input-output* e um grande conjunto de exemplos sem o devido *output*.

Segundo (RUSSEL e NORVIG, 2009), os exemplos utilizados para o aprendizado de máquina podem ser classificados em **exemplos positivos** e **exemplos negativos** de forma que o agente possa identificar condições do ambiente (verdadeiras ou falsas).

3 TRABALHOS RELACIONADOS

3.1 Classificação Hierárquica

A classificação automática de textos pode ser sem a definição de relacionamentos entre as classes, na qual as categorias são geradas sem a definição de relacionamento entre elas, denominadas *flat categories* (SUN e LIM, 2001) ou classificação não-hierárquica. Já a classificação hierárquica consiste, além da definição das classes, na definição destes relacionamentos entre estas classes.

3.1.1 Classificação Hierárquica de Textos

Langie e Lima apresentaram em 2003 (LANGIE e LIMA, 2003), no trabalho "*Classificação hierárquica de documentos textuais digitais usando o algoritmo kNN*" os princípios básicos da classificação automática de textos, a saber :

Base de treino e base de teste. A primeira é usada para identificar as características das categorias do domínio. A segunda é utilizada para testar o desempenho do classificador.

Representação de documentos. Atribuir pesos a um documento é uma forma a diferenciar os termos mais relevantes daqueles termos e menor importância. Estes pesos podem ser calculados a partir da frequência do termo no documento e na distribuição de termos na coleção.

Seleção de atributos. Consiste em eliminar os termos não representativos para o melhor desempenho do processo de classificação.

O processo proposto por Langie e Lima (LANGIE e LIMA, 2003) consiste na criação de uma árvore de categorias, na qual cada documento é classificado em relação à árvore de categorias e progressivamente em relação às subárvores até ser classificado como alguma categoria-folha.

3.1.2 Clusterização Hierárquica

Jean Metz e Maria Carolina Monard (METZ e MONARD, 2006) publicaram em 2006 um artigo de nome "*Estudo e Análise das Diversas Representações e Estruturas de Dados Utilizadas nos Algoritmos de Clustering Hierárquico*" que apresenta os conceitos, algoritmos e ferramentas relacionados ao *clustering*.

Algoritmos de *clustering* agrupam exemplos de dados baseados em índices de proximidade (similaridade) entre pares. O conjunto desses exemplos pode ser descrito por meio de duas estruturas: tabela de exemplos e matriz de similaridade. No caso da tabela de exemplos, as células da tabela, ou valores dos atributos X_1 , X_2,... X_M , podem assumir tipos diferentes, tais como: binário, discreto ou contínuo. Além dos tipos dos atributos, o *clustering* é influenciado pela escala, que indica a significância relativa dos valores dos atributos. Assim, a escala dos atributos pode ser qualitativa (nominal ou ordinal) ou quantitativa (intervalo ou proporção).

O *clustering* hierárquico, assim como as outras abordagens de *clustering*, constrói os agrupamentos de modo que exemplos pertencentes ao mesmo *cluster* possuem alta similaridade e exemplos pertencentes a *clusters* diferentes possuem baixa similaridade.

Entretanto, uma distinção entre essa abordagem e as demais é que o resultado obtido não é constituído apenas de uma partição do conjunto de dados inicial, mas sim de uma hierarquia que descreve um particionamento diferente a cada nível analisado. Um conjunto de dados, geralmente, contém diversos *clusters* e esses *clusters*, por sua vez, contém *subclusters*. Os *subclusters* podem ainda ser formados a partir do agrupamento de outros *clusters* menores (*sub-subclusters*), e assim sucessivamente.

Nesses casos, torna-se necessária a utilização de uma representação formal para a hierarquia de *clusters* obtida a partir dos dados. O dendograma, apresentado na figura 12 é a estrutura mais frequentemente utilizada para representar essa hierarquia, que consiste de um tipo especial de estrutura de árvore, na qual os nós pais agrupam os exemplos representados pelos nós filhos. Outras estruturas são também utilizadas para representar a hierarquia de agrupamentos, tais como Diagramas de *Venn*, apresentado na Figura 13, *IciclePlo*t e *Banner*.

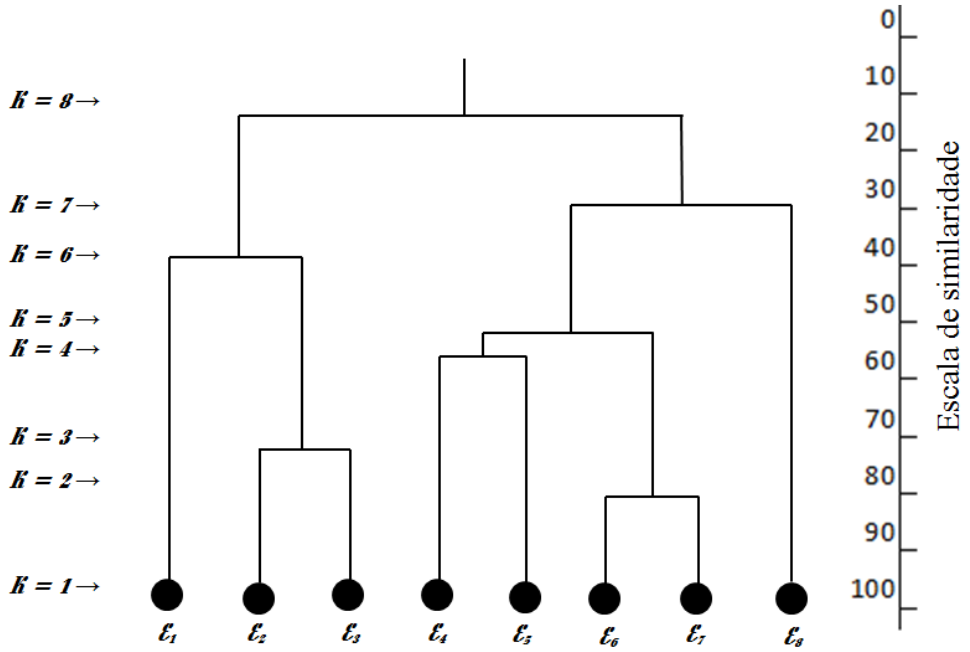


Figura 12: Hierarquia Representada por Dendograma.

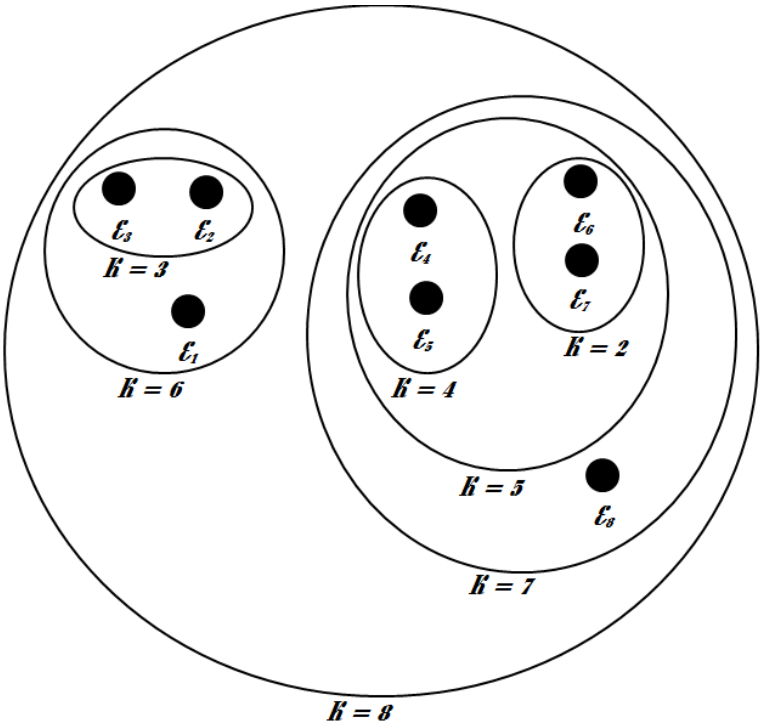


Figura 13 : Hierarquia Representada por Diagramas de *Venn*.

Duas estratégias podem ser utilizadas com o *clustering* hierárquico:

1. Aglomerativa (*Botton-Up*);
2. Divisiva (*Top-down*).

Na primeira, cada exemplo é considerado um *cluster* unitário. Em seguida, pares desses *clusters* são iterativamente agrupados considerando alguma medida de similaridade.

3.1.3 Definindo a hierarquia de classes por meio de uma ontologia

Noy e McGuinness (NOY e MCGUINNESS, 2001) apresentaram uma metodologia para criação de ontologias. Nesse trabalho é apresentada uma metodologia para a construção de uma ontologia com os seguintes passos:

1 – Determinar o domínio e escopo da ontologia;

2 – Considerar a reutilização das ontologias existentes;

3 – Enumerar os termos importantes para a ontologia;

4 – Definir as classes e a hierarquia das classes;

5 – Definir as propriedades das classes;

6 – Definir as características das propriedades;

7 – Criar instâncias;

O ponto interessante do artigo é o passo 4, no qual são definidas as classes e a hierarquia das classes pelo modelo *top-down* a partir dos termos obtidos no passo anterior (NOY e MCGUINNESS, 2001):

> "No passo 4, a definição da hierarquia de classes pode ser *top-down*, *bottom-up* ou a combinação das duas anteriores. Nenhum dos três métodos é definitivamente superior ao outro, dependendo-se da visão do desenvolvedor sobre o domínio pesquisado."

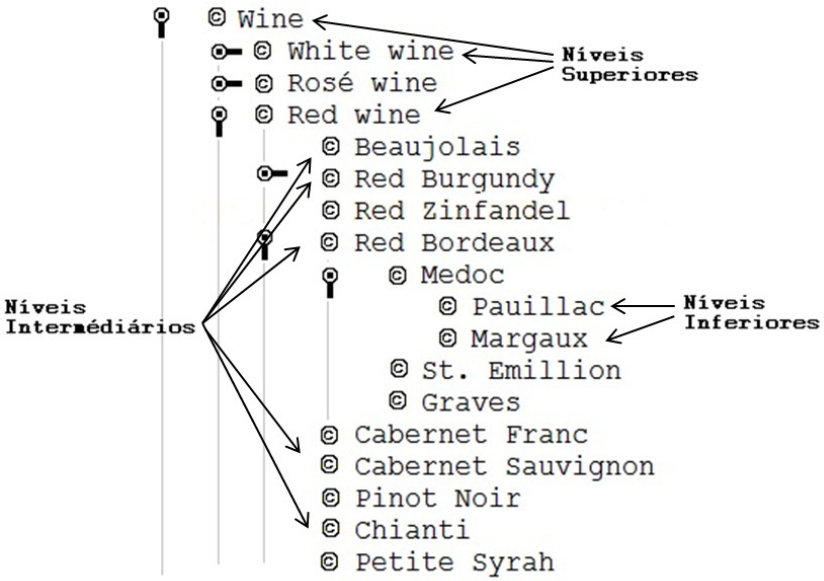


Figura 14 : Níveis diferentes da taxionomia de Vinhos, adaptado de NOY e MCGUINNESS, 2001.

A metodologia apresentada por NOY e MCGUINNESS foi utilizada para a construção de uma taxonomia sobre vinhos, conforme eram executadas as etapas eram apresentados os resultados sobre a taxonomia de vinhos, conforme o exemplo apresentado na Figura 14.

Em 2008, Sergio Andrade e Manoel Neto (ANDRADE e NETO, 2008) apresentaram o artigo "*Uma abordagem de modelagem multidimensional para datamart de compras públicas, usando taxonomia*". O trabalho consiste na definição das dimensões e fatos do modelo estrela. A partir do resultado da ontologia. Deste contexto, são retiradas as entidades que formam as tabelas de dimensões e fatos que compõem a arquitetura no processo de *data warehousing*. Um exemplo da taxonomia gerada está na Figura 15.

Figura 15 : Hierarquia de classes da ontologia, adaptado de Andrade e Neto, 2008 .

3.2 Construção semiautomática de taxonomias.

O trabalho de Camila Martins (MARTINS, 2006) apresentou uma metodologia denominada SACT (*Semi-automatic Construction of Taxonomies*). Esta metodologia inclui procedimentos automáticos e interativos, utilizando o conhecimento do especialista e também auxiliando a construção das taxonomias. A SACT foi utilizada para gerar taxonomias de regras de associação tanto em (MARTINS, 2006) quanto em (RESENDE e MARTINS, 2006). Os passos da SACT estão descritos na Tabela 03:

Tabela 03: Processos de geração de taxonomias da metodologia SAC T.

Processo de geração de taxonomias da metodologia SACT
Lê a tabela inserida pelo usuário e salva a frequência das palavras nas descrições dos registros.
Lê a tabela inserida pelo usuário e salva as descrições dos registros e os seus atributos.
Retorna as informações dos itens frequentes de determinado processo.
Retorna as informações dos itens.
Exclui os itens frequentes com frequência menor ou igual a um número determinado pelo usuário.
Exclui os itens frequentes selecionados pelo usuário.
Gera as taxonomías iniciais comparando os itens frequentes com as descrições dos elementos dos registros e criando as taxonomías de níveis 1 e 2 de acordo com a posição da palavra frequente nas descrições dos elementos.
Seleciona os itens da taxonomia c os organiza de uma maneira compreensível ao usuário.
Move ou exclui taxonomías.
Altera o nome dos itens da taxonomia.
Redistribui um item do nível mais baixo da taxonomia de um processo de acordo com um atributo selecionado pelo usuário, colocando-o no grupo onde existem mais produtos com o mesmo valor para esse atributo.
Insere uma nova informação (nome de atributo e valor de atributo) para determinado item do nível mais alto da taxonomia. A informação inserida também é salva para todos os filhos daquele elemento.
Agrupa itens de um processo segundo um atributo criado pelo usuário e gera um item de nível superior, que será o pai dos itens agrupados.
Gera um arquivo no formato padrão do algoritmo SACT,

A utilização de taxonomias para generalizar regras de associação na etapa de extração de padrões do processo de mineração de dados, apesar de não proporcionar redução no volume de regras, permite que sejam identificadas as regras de maior importância nos níveis superiores da taxonomia gerada, onde se localizam as regras generalizadas.

3.2 Construção semiautomática de ontologias.

O trabalho de Noriko Tomuro e Andriy Shepitsen (TOMURO e SHEPITSEN, 2009) apresentado em 2009 com o nome "*Construction of Disambiguated Folksonomy Ontologies Using Wikipedia"* apresenta um *framework* para a construção semiautomática de uma ontologia a partir dos *tags* do site *Delicious* com a extração de informações a partir a *Wikipedi*a. Tomuro desenvolveu em trabalhos anteriores o algoritmo DSCBC (*Domain Similarity Clustering By Committee*) para a desambiguação dos clusters que utiliza um método que utiliza termos em forma de vetores de forma semelhante ao processo *TF-IDF (Term Frequency – Inverse Document Frequency)* para agrupar *tags* com significados similares em grupos denominados *committees*. Desta forma quando uma *tag* é identificada pertencendo a mais de uma *commitee* é detectada uma ambiguidade. Despois é aplicado um algoritmo de *clustering* hierárquico para montar uma ontológica destas *tags*, sendo as ambiguidades encontradas anteriormente resolvidas durante a montagem desta ontologia.

A desambiguação dos termos na ontologia é feita por meio da identificação do "*core meaning*" com a utilização de uma função que procura, entre os termos que apresentam ambiguidade, o termo que está mais próximo dos seus pares, sendo este selecionado e os termos que apresentam ambiguidades em relação ao termo selecionado são descartados, conforme apresentado na Figura 16.

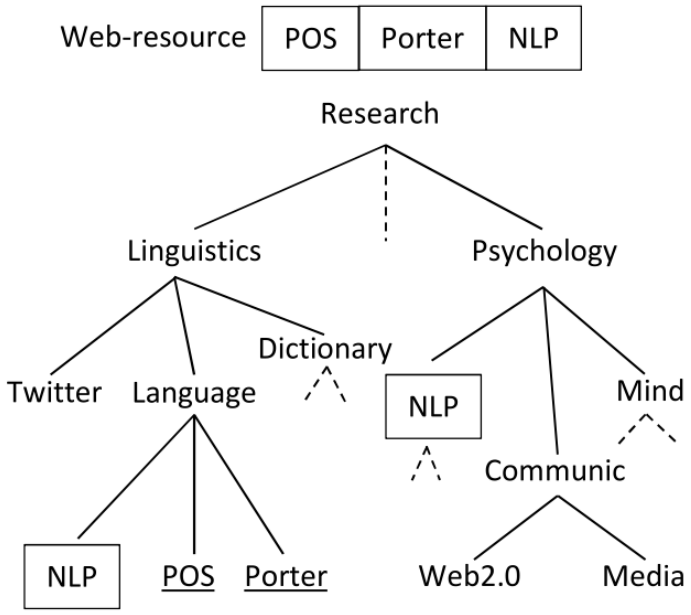


Figura 16 : Ambiguidade de termos na ontologia.

Cada ontologia é avaliada pela função de similaridade denominada *Ontology Density* (*Dens(T,R)*) que avalia o quanto os termos de cada ontologia estão próximos, conforme apresentada na Figura 17:

$$Dens(T,R) = \frac{1}{|R|} \sum_{r \in R} density(r,T)$$

$$density(r,T) = \frac{nTags(r) - 1}{\text{argmin}_{i,j}\, dist(node(i,T), node(j,T))}$$

Figura 17: Fórmula de densidade para avaliar a ontologia gerada.

A fórmula acima calcula a densidade de uma ontologia onde *density*(*r,T*) é o número de *tags* associadas a *r*, *argmin* (*x,y*) é o menor valor da função *f*(*x,y*) utilizada, *node(k,T)* é um nó em *T* para a $k^{\text{ésima}}$ *tag* (associada a *r*) e *dist*(*n*1, *n*2) é o número de vértices entre os nós *n*1 e *n*2 em *T*. Desta forma a densidade é essencialmente o inverso da distância mínima entre as *tags* associadas.

3.2.1 Criação semiautomática de ontologias a partir de bases textuais.

Mithun Balakrishna, Dan Moldovan, Marta Tatu e Marian Olteanu apresentaram em 2010 no trabalho denominado "*Semi-Automatic Domain Ontology Creation from Text Resources*" um

framework capaz de extrair informações semânticas de bases textuais e criar ontologias com o mínimo de intervenção manual. Este trabalho apresenta um *framework* formado por dois artefatos, o *Polaris* e o *Jaguar*. O *Polaris* é um *parser* semântico que extrai informações a partir de 26 relações semânticas pré-definidas, conforme a Tabela 4 mostrada abaixo:

Tabela 04 : Conjunto de relações semânticas utilizadas no Polaris

.Relation	Definition	Code	Relation	Definition	Code
Agent X, Y)	X is the agent of Y; X is protolypically a person	AGT	Association(X,Y)	Person X is associated with Person Y; the relation is not necessarily kinship	ASO
At-Location(X,Y)	X is at-location Y or where X takes place	AT-L.	At-Time(X,Y)	X is at-time Y or when X takes place	AT-T
Cause(X.Y)	X causes Y	CAU	Experiencer(X,Y)	X is an experiencer of Y; involves cognition and senses	EXP
lnfluence(X,Y)	X caused something to happen to Y	IFL.	Instrument(X,Y)	X is an instrument in Y	INS
Intent(X.Y)	X is the intent/goal/reason of Y	INT	IS-A(X,Y)	X is a (kind of) Y	ISA
Justification(X.Y)	X is the reason or motivation or justification for Y	JST	Kinship(X,Y)	X is a kin of Y: X is related to Y by blood or by marriage	KIN
Make(X.Y)	X makes Y	MAK	Manner(X,Y)	X is the manner in which Y happens	MNR
Part-Whole(X.Y)	X is a part of Y	PW	Possession(X,Y)	X is a possession of Y; Y owns/has X	POS
Property(X.Y)	X is a property/attribute/value of Y	PRO	Purpose(X,Y)	X is the purpose for Y	PRP
Quantification(X, Y)	X is a quantification of Y; Y can be an entity or event	QNT	Recipient(X,Y)	X is the recipient of Y; X is an animated entity.	RCP
Source(X.Y)	X is the source, origin or previous location of Y	SRC	Stimulus(X,Y)	X is the stimulus of Y; Perceived through senses	STI
Synonymy(X,Y)	X is a synonym/name/equal for/to Y	SYN	Theme(X,Y)	X is the theme of Y	THM
Topic(X.Y)	X is the topic/focus of cognitive communication Y	TPC	Value(X,Y)	X is the value of Y	VAL

Existem seis tipos de padrões primários buscados pelo *Polaris* dentro de frases nominais: NN e Adj-N (que compreendem compostos nominais), padrões genitivo, frases adjetivo e as cláusulas adjetivo. Os primeiros cinco estão subdivididos em ocorrências nominais e não nominais, resultando em um total de 11 padrões buscados no texto.

O artefato *Jaguar* processa recursos textuais e constrói ontologias específicas de domínio com o uso de uma base de conhecimento que inclui os seguintes componentes:

- Ontologia de Conceitos: blocos de construção básicos de ontologias.
- Hierarquia: estrutura que captura o conhecimento universal em certos conceitos ontológicos via relações transitivas (por exemplo, ISA, Parte-todo, Locativo, etc).
- Conhecimento Contextual: *Clusters* de conhecimento para capturar conhecimento

não universal e contextual através de todas as relações semânticas descobertas pelo analisador semântico no texto que está sendo analisado.

- Axiomas: Captura de afirmações de interesse sobre conceitos gerado a partir do conhecimento disponível.

O processamento executado pelo Jaguar segue passos definidos: Definição dos conceitos sementes da ontologia; Definição das sentenças a serem buscadas na coleção de documentos; Identificação das entidades nomeadas; Identificação dos adjetivos descritivos. O Jaguar possui algoritmos de resolução de conflitos para o processo de manutenção ou fusão de ontologias.

Os seguintes passos são executados quando é necessária a fusão de subconjuntos de ontologias:

Conforme mostrado na Figura 18, primeiramente o conjunto de conceitos de O2 é unido ao conjunto de conceitos O1; Se um conceito c1 de O1 existe em O2 com a mesma assinatura lexical, então o conceito oriundo de O1 é ignorado. O *WordNet Synsets* é utilizado como referência para identificar as diferentes assinaturas lexicais dos conceitos, no caso de *Stock Market* e *Stock Exchange* representam o mesmo conceito, mas com diferentes assinaturas lexicais. Quando são adicionados elementos não hierárquicos de O2 para O1, uma vez que as hierarquias são independentes, a montagem da hierarquia resultante é simples e direta.

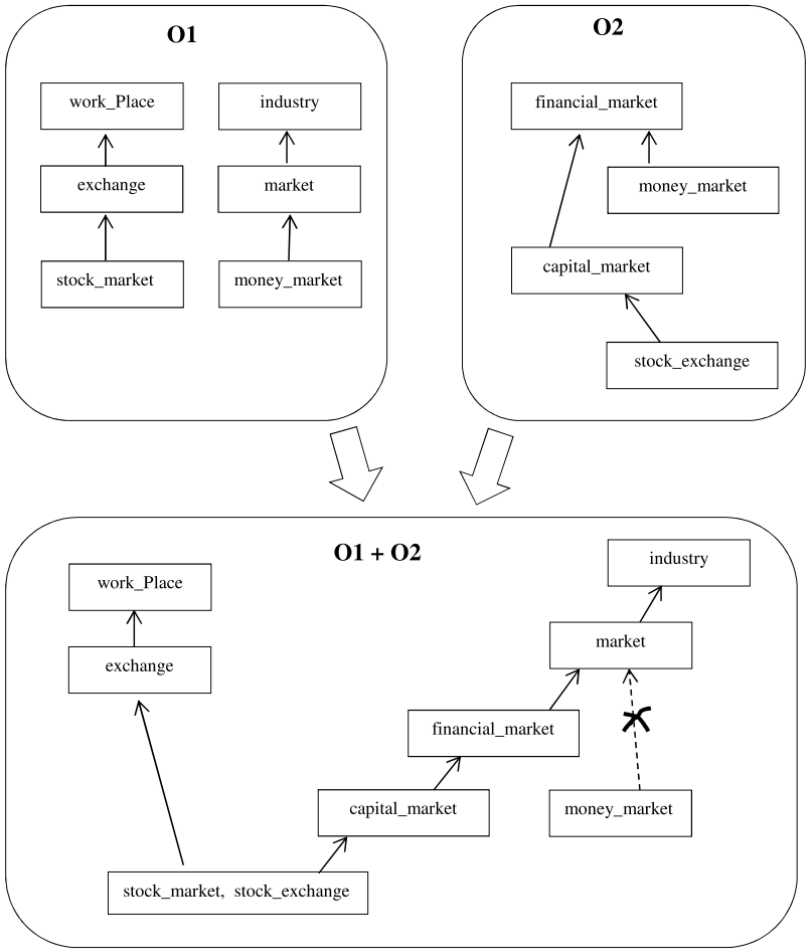


Figura 18 : Exemplo de resolução de conflitos entre componentes da ontologia.

3.3 Enriquecimento de dados com o conhecimento recuperado na *Web* (folksonomia)

A folksonomia é considerada como um conceito relativamente novo, criado em 2004 por Thomas Vander Wal (VANDER, 2007), resultado da união das palavras "*folk*" (povo, pessoas) com "taxonomia" (atribuição de nomear as coisas). A principal vantagem dessa modalidade está no cunho colaborativo, onde as pessoas podem participar livremente do processo de indexação dessas coisas, sem que para isso se faça necessário utilizar regras. A folksonomia não faz distinção dos usuários, quanto ao nível cultural e intelectual de cada um. A maior desvantagem está exatamente na falta de um vocabulário controlado e na ausência de uma normatização que defina parâmetros para a indexação desses itens.

Os principais trabalhos apresentados nesta seção utilizam a folksonomia pela análise dos *tags* (palavras-chave) atribuídos pelos usuários a *sites*. David M. Chen e Jian Qin publicaram em 2008 o artigo de nome "*Deriving ontology from folksonomy and controlled vocabular"* (CHEN e QIN 2008) abordando o fato que folksonomias por si só não são suficientes para a construção de uma ontologia abrangente e de alta qualidade. Chen e Qin utilizaram outras fontes, como *WordNet* e *Wikipedia* para auxiliar a construção de ontologias a partir de folksonomias.

O vocabulário controlado é caracterizado por estruturas rígidas e capacidade de resposta lenta a novas terminologias, ao contrário das folksonomias. Mas a organização sistemática e cuidadosa formulação de termos e relacionamentos do vocabulário controlado poderiam ser aplicados para compensar as desvantagens das folksonomias. A utilização de recursos lexicais *online* e dicionários como instâncias de vocabulário controlado parecem ser abordagens promissoras no uso folksonomias para a geração de ontologias. Para o estudo piloto, foram escolhidas as *tags* do subconjunto de "paisagens" (*landscape*) das imagens do site "*Flickr*" e a biblioteca digital ADL (*Alexander Digital Library*), a qual é uma enciclopédia geográfica para se utilizada como fonte do vocabulário controlado.

No exemplo apresentado na Figura 19, o conjunto de *tags* é construído sobre as *tag* relacionadas com "Lake" (lago), recolhendo-se as *tags* relacionadas; as *tags* relacionadas das *tags* relacionadas anteriormente e assim por diante, até que o número de *tags* chegue a um limite pré-definido (foi utilizado o limite de 100). As relações de hierarquia entre as *tags* são obtidas no *ADL Thesaurus* e o conjunto de *tags* e relações é associado à ontologia que está sendo construída.

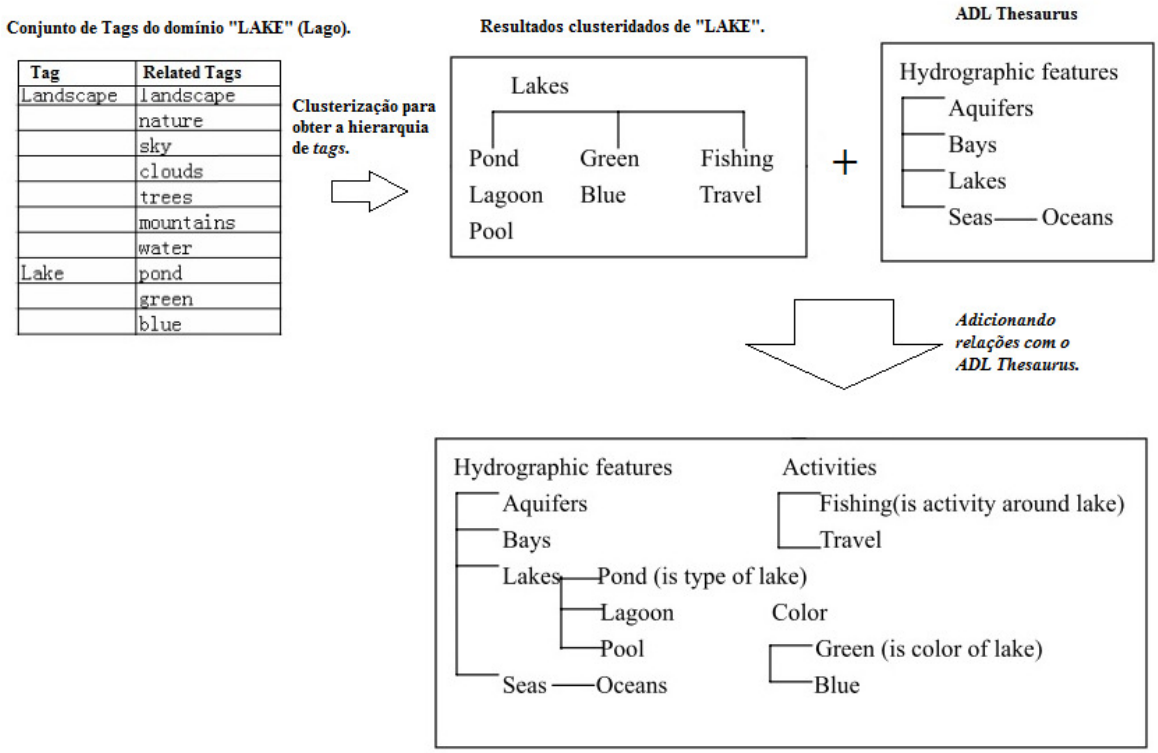


Figura 19 : Rascunho da geração de ontologia (CHEN e QIN 2008).

Sofia Angeletou, Marta Sabou, e Enrico Motta (ANGELETOU, SABOU, MOTTA 2008) apresentaram um trabalho que apresenta um enriquecimento automático de *tags* com a utilização da folksonomia por meio de associações entre os conceitos relevantes definidos em ontologias *online*. A metodologia apresentada neste trabalho é denomiana de FLOR (FLOR – ***F**ol**L**ksonomy **O**ntology en**R**ichment*), na qual consiste das seguintes etapas de processamento das tags: Processamento léxico; Definição do sentido e desambiguação; Expansão semântica (por meio de acesso ao *WordNet) e enriquecimento semântico.* A Figura 20 apresenta um modelo das etapas da metodologia utilizada pelo FLOR.

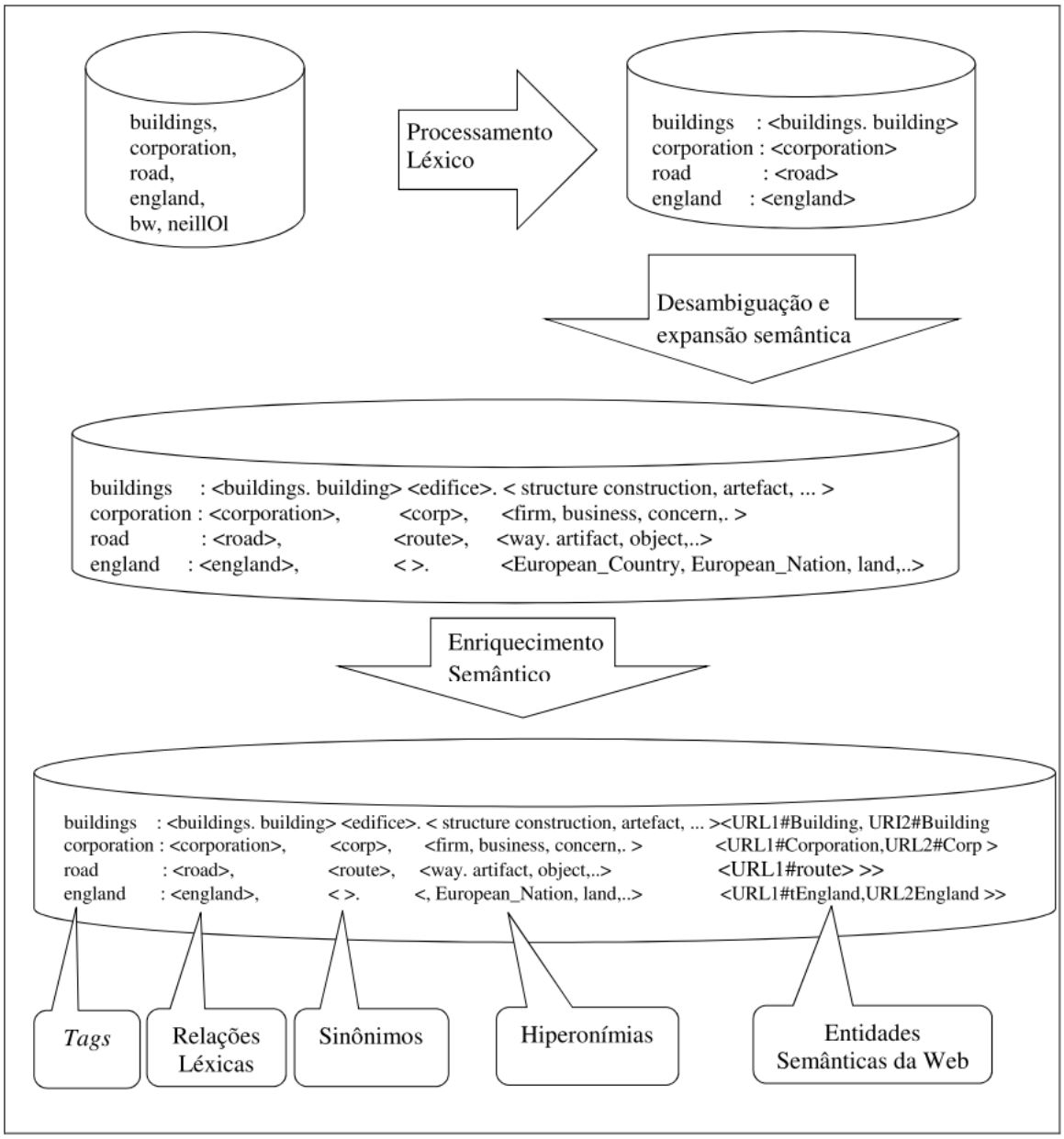


Figura 20 : Metodologia do processo FLOR (*FoLksonomy Ontology enRichment*) adaptado de (ANGELETOU, SABOU e MOTTA, 2008).

Baccar, Gargouri e Ben Hamadou (BACCAR, GARGOURI e HAMADOU 2010) apresentaram na *International Conference on Intelligent Semantic Web Services and Applications* (ISWSA 2010) o artigo de nome "*Semantically Enriching Folksonomies from LMF Standardized Dictionaries*". A sigla LMF significa *Lexical Markup Framework* (ISO-24613) (LMF, 2008), que consiste em uma padronização das representações léxicas para o processamento de linguagem natural e MRD (*Machine Readable Dictionaries*). Foi desenvolvida por Gild Francopoulo, Monte

George e Nicoletta Calzolari. O trabalho desenvolvido por Baccar, Gargouri e Ben Hamadou consiste na geração de uma ontologia de domínio a partir de um dicionário padronizado em LMF. A geração da ontologia seguiu os seguintes passos, conforme ilustrado na Figura 21:

1. Extração de um fragmento do dicionário (de acordo com o domínio).
2. Identificação do domínio, pelo *designer*.
3. Identificação dos tipos léxicos de informação.
4. Identificação das entradas léxicas.
5. Construção do núcleo da ontologia.
6. Identificação dos conceitos.
7. Identificação das relações.
8. Identificação das instâncias.
9. Enriquecimento do núcleo da ontologia com a exploração dos textos disponíveis.

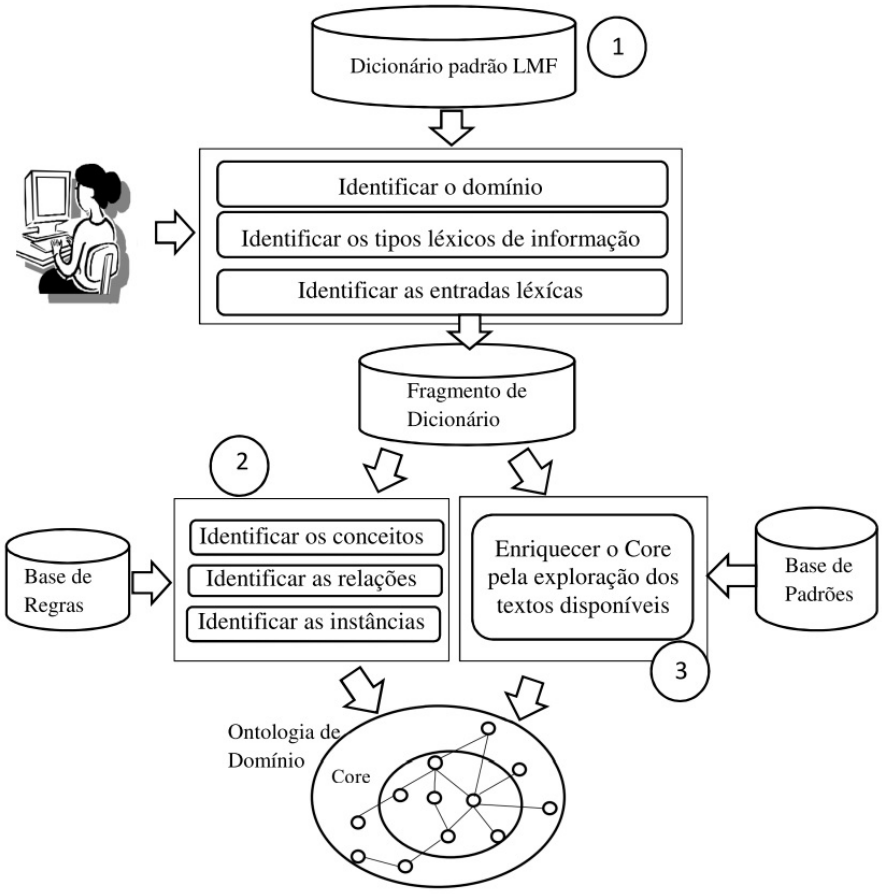


Figura 21 : Geração da ontologia a partir do dicionário LMF. Adaptado de BACCAR, GARGOURI e HAMADOU 2010

3.4 Evolução do enriquecimento de dados com informações oriundas da *Web*:

Com a popularização de *sites* provedores de informação, a quantidade e a variedade de informações disponíveis para serem consultadas ou pesquisadas aumentou exponencialmente. Estes *sites* podem ser de cunho colaborativo, como a *Wikipedia* e *blogs* ou possuir informações próprias como as ferramentas de busca, dicionários, jornais e revistas *online*. Esta oferta de informação, apesar de ser positiva, traz à tona alguns desafios para o usuário comum. Um destes desafios é a dificuldade de se lidar com esta imensa quantidade de informação disponível, mesmo que seja sobre um assunto específico. Outra dificuldade encontrada é a ambiguidade das informações recuperadas da *Web*.

O desenvolvimento de agentes automáticos para pesquisar informações na *Web* enfrenta também desafios análogos. A imensa quantidade de informações, à primeira vista, não representa um desafio para um agente automático, mas aprender a localizar a informação desejada no texto do *site* e lidar com a ambiguidade presente nestas informações tem sido os principais desafios dos desenvolvedores de agentes automáticos de extração de informações a partir da *Web*.

Os trabalhos de Roberto Navigli tem abordado a extração e a desambiguação de informações oriundas da *Web*. Em 2004, Roberto Navigli (NAVIGLI e VELARD 2004) propôs um modelo denominado de *OntoLearn*, composto de três fases: extração dos termos do domínio, análise semântica na busca de relações taxonômicas e por último construção de ontologia de domínio.

Em 2005, Massaki Murata desenvolveu um método de localização de sinônimos em textos retornados de dicionários baseado na quantidade e frequência de palavras, do texto retornado, posicionadas nas proximidades do termo buscado e seu sinônimo. Este método permitiu a seleção dos termos com maior possibilidade de serem sinônimos do termo pesquisado a partir da análise do texto retornado de vários dicionários diferentes (MURATA, KANAMARU e ISAHARA 2005).

.

Anon Plangprasopchok, em 2008, com seu trabalho intitulado "*Constructing Folksonomies from User-specified Relations on Flickr*", utilizou a abordagem "*bag-of-words*", selecionado os *tags* do *site Flickr* para construir uma folksonomia. Neste trabalho, as ambiguidades encontradas foram resolvidas com a utilização de uma métrica baseada na quantidade de usuários a que cada *tag* estava relacionada. Os *tags* insignificantes ou falhos foram eliminados previamente com a aplicação de um *threshold* baseado na frequência destes *tags*. (PLANGPRASOPCHOK, LERMAN e GETOOR

2008).

Em 2008 Navigli utilisou *part-of speech tagging* sobre o texto retornado do *WordNet* para a extração de relações semânticas, sendo estas adicionadas a uma *core ontology* previamente definida. (NAVIGLI e VELARD 2008)

Em 2009 Navigli (NAVIGLI 2009) publicou um trabalho intitulado "*Word Sense Desambiguation: A Survey*" o qual apresenta as principais técnicas utilizadas para lidar com a desambiguação como listas de decisão, árvores de decisão, redes bayesianas, redes reurais, métodos semi-supervisionados e métodos não supervisionados.

Em 2009, Felfie Dotsika (DOTSIKA 2009), com o trabalho "*Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies*" abordou o *gap* entre a construção de taxonomias e a construção de ontologias, mencionando os problemas de ambiguidade, de qualidade dos dados oriundos da *Web*, apresentando propostas e métricas para se lidar com estes problemas como: granularidade dos termos, identificação das palavras descritivas para compor as sementes (*seeds*) da ontologia a ser construída, uso de um vocabulário controlado e uma abordagem mista *bottom-up* e *up-down*. Neste mesmo ano, Jie Tang (TANG et al 2009) com outros pesquisadores da Universidade de Hong Kong apresentaram um trabalho no qual a construção das ontologias é feita com o uso de quatro métricas matemáticas: *Tag-divergence, Hypernym-divergence, Merging-divergence e Keep-divergence* que irão definir como um novo *tag* deverá ser adicionado à ontologia em construção. De forma similar, também na Universidade de Hong Kong, um trabalho de Raymond Y. K. Lau e outros pesquisadores (RAYMOND et al 2009) utilizou lógica *fuzzy* para a extração, avaliação e montagem de conceitos em uma ontologia de domínio.

Em 2010, diversos trabalhos foram desenvolvidos na construção de taxonomias, por exemplo: Zornitsa Kozareva e Eduard Hovy (KOZAREVA e HOVY 2010) abordaram o problema da construção de taxonomias a partir do zero (*from scratch*), pois o conteúdo heterogêneo coletado a partir da *Web* deverá ser utilizado para a construção de taxonomias sem a utilização de uma taxonomia base previamente fornecida (*seeds*). Foram utilizadas diversas técnicas para as operações de construção da taxonomia (*Taxonomy Induction*) como avaliação do menor caminho entre conceitos similares, reposicionamento de conceitos e a busca e eliminação de ciclos. Anon Plangprasopchok, Kristina Lerman e Lise Getoor (PLANGPRASOPCHOK et al 2010) apresentaram uma métrica de similaridade estrutural para avaliar se subárvores oriundas da extração de conceitos da *Web* podem ser fundidas (*Root-to-Root* → *merge*) ou colocadas em uma hierarquia (*Root-to-Leaf* ou *Leaf-to-Root*) na folksonomia.

Em 2011 foram apresentados os seguintes trabalhos: Iván Cantador, Ioannis Konstas e Joemon M. Jose (CANTADOR et al 2011) apresentam um mecanismo não supervisionado para a identificação e categorização de *tags* a partir de um conjunto de categorias pré-definido, que avalia quatro fatores: conteúdo, contexto, subjetividade e estrutura; Nesse mesmo ano, Alexa Breuing, Ulli Waltinger, Ipke Wachsmuth (BREUING et al 2011) apresentaram um processo automático para a identificação do tópico a partir da análise da linguagem dos diálogos da *Wikipédia* formado de seis passos: detecção do conceito; identificação do conceito local nas proximidades onde o conceito foi identificado; identificação do conceito (na hierarquia) que melhor se categoriza o conceito detectado; definição dos *labels* de cada tópico (termina somente após o processamento de todos os dados); detecção das referências a tópicos anteriores; segmentação, no qual a lista de tópicos é sumarizada e agregada em função das hierarquias entre os tópicos; Roberto Navigli, Paola Velardi e Stefano Faralli (NAVIGLI et al 2011) utilizaram em um trabalho a métrica *Edge Weighting,* na qual são avaliadas a quantidade de nós entre os conceitos e a conectividade entre os nós transversos para a resolução dos conflitos estruturais na construção da taxonomia a partir do grafo de conceitos.

Em 2012, um estudo de autoria de Ashwin Ittoo e Gosse Bouma (ITTOO e BOUMA 2012) apresentou um algoritmo capaz de extrair relações "*part-whole*" da *Wikipédia* com foco em um domínio específico. Este algoritmo tem como base o aprendizado de máquina (*minimally-supervised algorithm*), no qual diversos padrões detectados nos dados de treinamento são avaliados, selecionando-se um conjunto de padrões mais confiáveis para se utilizar na identificação das relações "*part-whole*" sobre o texto da *Wikipedia.*

Podemos destacar algumas observações do estado atual das técnicas de extração de informações a partir de texto oriundo da *Web*:

- Há uma dependência das técnicas sobre a língua na qual o texto foi escrito, sendo necessários ajustes nos padrões buscados, caso seja utilizada a busca por padrões *part-of-speech* para se localizar as relações no texto. Caso seja utilizado PLN, a gramática utilizada, bem como o *Thesauro*, deverão ser ajustados em função da língua original na qual a fonte de informação foi escrita.
- Os métodos que utilizam apenas a técnica de *bag-of-words* permitem a detecção do domínio do texto retornado, mas não apresentam bons resultados na extração de relações semânticas no texto.
- As técnicas que buscam padrões no texto são dependentes da aderência entre os padrões

buscados e o tipo de site que retorna o texto. Sites de notícias tem uma forma de escrita diferente de dicionários *online.* Os textos oriundos de jornais eletrônicos tem uma forma de escrita do texto diferente de *blogs*. Palavras como abreviações, rótulos e títulos são posicionadas em sequências e padrões variados conforme o tipo de site no qual as informações estão sendo extraídas.

3.5 Considerações Finais do capítulo

Este capítulo discutiu as diversas técnicas de construção de ontologias, processamento de linguagem natural, análise formal de conceitos e folksonomia. Foram apresentados diversos aspectos de ontologias como: a metodologia de geração de ontologias, a definição dos termos generalizadores da ontologia e a geração de uma hierarquia a partir desta. Também foram apresentadas as principais teorias sobre processamento de linguagem natural.

Ao observar as metodologias de construção de ontologias, passamos a ter uma ideia da importância da etapa de identificação dos termos generalizadores para geração da classificação hierárquica, pois os termos generalizadores são a matéria-prima deste processo.

4 O MODELO CDDW

Neste capítulo será apresentado o modelo **CDDW (Classificação de Dados utilizando Dicionários da *Web*)** de enriquecimento e classificação de dados com o uso de informações recuperadas de dicionários *online*.

Este trabalho pode ser classificado como uma pesquisa aplicada pela sua natureza, já que apresenta teorias e as aplica na solução de um problema específico. Do ponto de vista da forma de abordagem é classificado como uma pesquisa exploratória, na qual o modelo é aplicado na resolução de um problema específico e são analisados os resultados, validando propostas e identificando pontos em que as técnicas deveriam ser aperfeiçoadas (LAKATOS e MARCONI, 1991). Será utilizado o método indutivo, proposto por Popper, no qual o modelo é aplicado sobre um problema real e os resultados obtidos permitem que este seja avaliado e validado (POPPER, 1993).

Neste capítulo serão apresentadas as etapas conceituais da solução do problema em questão. Em cada etapa serão utilizados os conceitos definidos anteriormente e descritos os processos formais da solução. A Figura 22 apresenta as etapas da solução formal do problema:

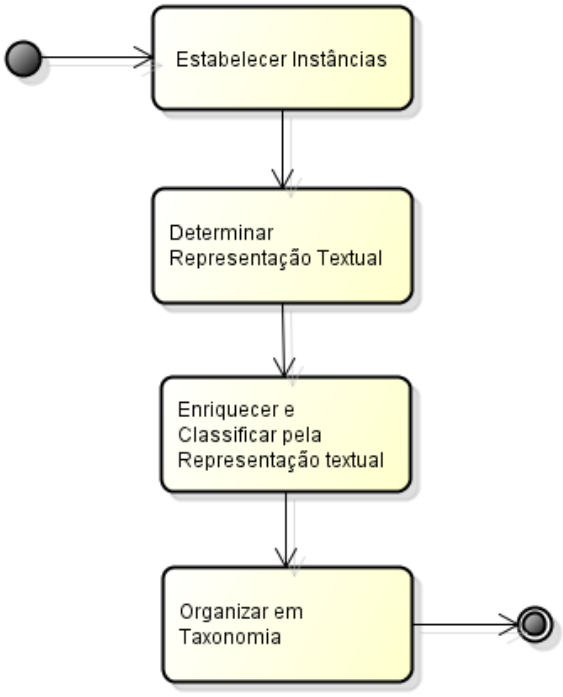


Figura 22 : Etapas do modelo CDDW.

Este trabalho apresenta o *framework* CDDW que utiliza diversas técnicas para refinar, enriquecer e classificar produtos a partir da representação textual dos registros de venda de produtos (cupom fiscal). Este *framework* é formado por quatro principais processos:

- O processo de **Estabelecer Instâncias**: Consiste em varrer um conjunto representativo de operações de venda e selecionar por código de produto as diversas descrições utilizadas nestas operações.
- O processo de **Determinar a Representação Textual**: Consiste na seleção dos termos descritivos do produto a partir dos termos desabreviados e selecionados das diversas descrições utilizadas nos registros de venda do respectivo produto.
- O processo de **Enriquecer e Classificar pela Descrição Textual**: Consiste na utilização da *Web* para se obter os termos classificadores (hiperonímias – termo que possui uma relação de classificação lexical com outro termo) dos termos descritivos dos produtos.
- O processo de **Organizar em Taxonomia**: A partir das relações anteriormente obtidas é construída a taxonomia resolvendo-se as ambiguidades e conflitos entre os termos classificadores.

4.1 Descrição formal do problema

A seguir serão apresentados conceitos essenciais para se descrever formalmente o modelo conceitual. (BEZERRA, 2007)

> *"Uma classe é uma descrição dos atributos e serviços comuns a um grupo de objetos. Sendo assim, pode-se entender uma classe como sendo um molde a partir do qual objetos são construídos. Ainda sobre terminologia, diz-se que um objeto é uma instância de uma classe."*

No modelo formal, cada *instância* deverá representar somente um produto, sendo associada a um identificador único (o código de EAN – *European Article Number*) e possuir uma única descrição.

As **descrições de cada instância**. Cada instância, inicialmente, possui diversas descrições diferentes devido ao fato desta informação ser oriunda de várias fontes diferentes (diferentes sistemas de emissão de cupom fiscal). Cada descrição é formada por uma *sequência de termos* na

qual a ordem destes termos na respectiva descrição é determinante para o significado desta. Outro atributo importante de cada instância é o frequência de utilização desta na venda dos respectivos produtos registrados nos cupons fiscais descrição. Considerar uma determinada descrição como mais representativa devido ao fato desta ser mais utilizada (quantidade de cupons fiscais que utilizam esta descrição de produto) foi uma premissa para se selecionar as descrições mais representativas.

Desta forma, o que se possui inicialmente é um modelo com instâncias, descrições destas instâncias e das sequencias de termos que compõem estas descrições, conforme apresentado a seguir na Figura 23:

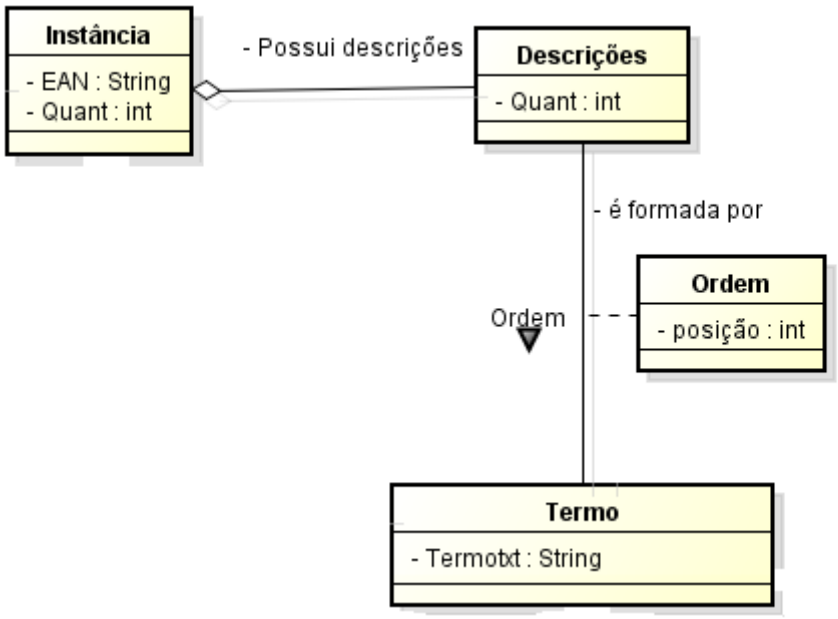


Figura 23 : Modelo de Classe para as informações iniciais.

As informações contidas no modelo apresentado na Figura 23 apresentam as seguintes características:

- Diversos termos são abreviados, desta forma, o termo “LEITE” e “LTE” tem o mesmo significado, apesar de serem termos distintos. Para representar a abreviação (relação de abreviação entre dois termos) acrescentamos a relação “*é abreviação de*” ao modelo.
- Cada instância possui um identificador único do produto (o código de EAN) e diversas descrições para o produto em questão (oriunda de diferentes emissores de cupom fiscal), esta instância, como se refere a um único produto, pode ser descrita

por uma descrição única também. Desta forma é possível se selecionar dentre o conjunto de termos das descrições de uma determinada instância os termos que compõem a descrição única (descrição representativa da instância). Para representar esta informação adicionamos a relação "*é descrita por*" ao modelo.

Desta forma o modelo conceitual inicial pode ser evoluído com a identificação das relações de abreviação entre os termos e da seleção dos termos representativos de cada instância. É importante ressaltar que estas informações já existiam no modelo inicial, sendo identificadas e selecionadas por procedimentos que estão descritos neste trabalho. Na Figura 24 está apresentado o modelo conceitual evoluído:

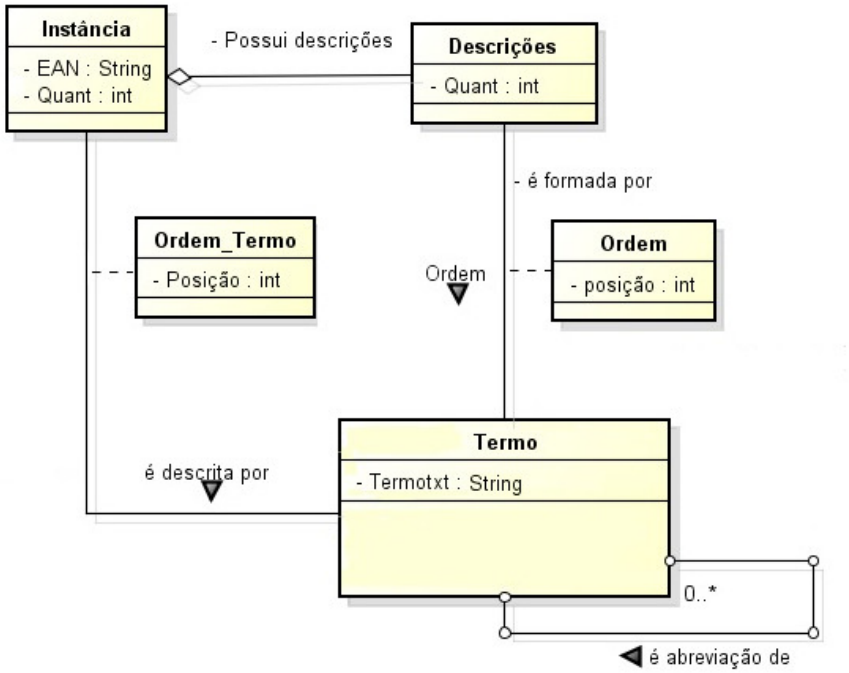


Figura 24 : Modelo de Classe evoluído a partir das informações iniciais.

Inicialmente, pode-se considerar como descrição das instâncias os termos, localizados na primeira posição da descrição, como uma classe (por exemplo: "*Queijo Prato Itambé*" e "*Queijo Parmezão Boa Nata*") como pertentes a uma mesma classe (por exemplo: a classe "Queijo"). Esta premissa é fundamental para obter as primeiras classes da taxonomia dos produtos, mas não suficiente para construção da taxonomia, pois é necessário se definir qual a classe que possui a classe cujo termo é "Queijo" (neste caso provavelmente será "Laticínio"). Neste estágio, há a necessidade de se obter a partir de fontes externas informações para se classificar cada produto, desta forma, esta hierarquia é obtida pela obtenção do termo classificador (hiperonímia) do termo

representativo da respectiva classe. A Figura 25 apresenta o modelo conceitual evoluído com a hierarquia de classes que será utilizada para a construção da taxonomia:

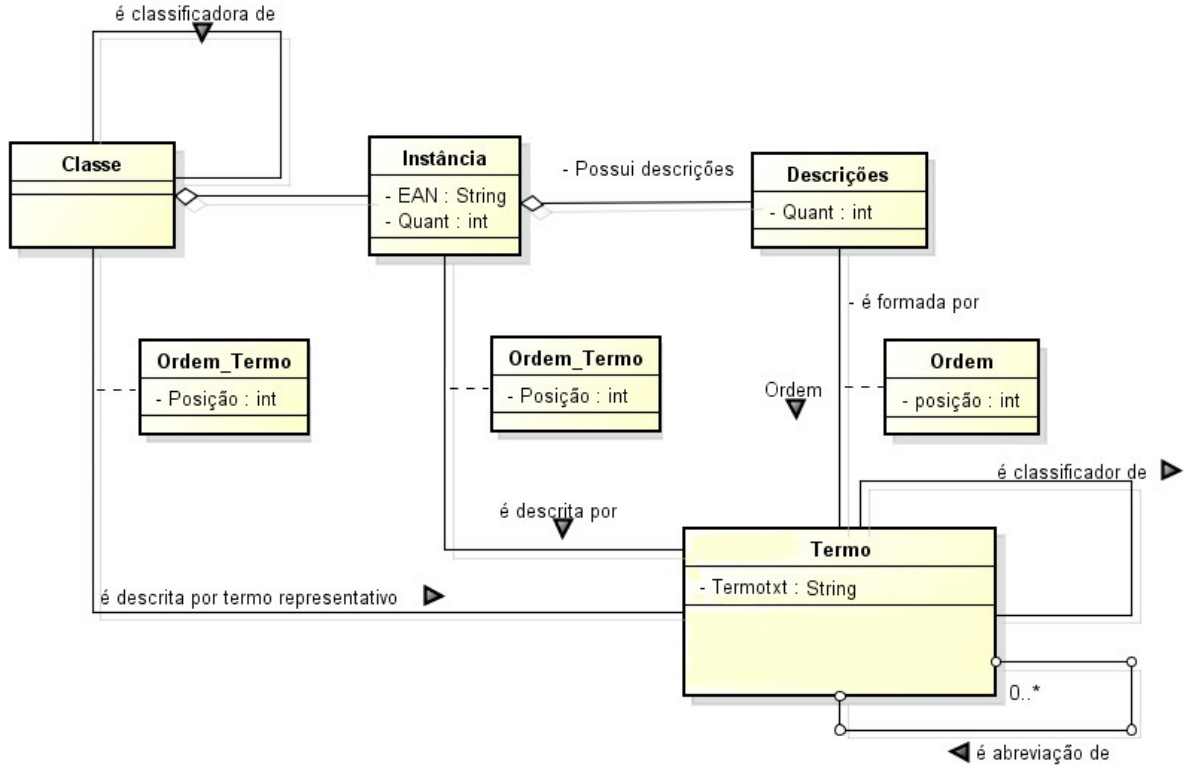


Figura 25 : Modelo de Classe evoluído com a hierarquia de classes.

Para se definir esta hierarquia de classes, será utilizado o conceito de hiperonímia de um termo, a seguir a definição de hiperonímia obtida em (Gomes e Sant'Ana, 2009):

> *"A hiperonímia e a hiponímia são fenômenos derivados das disposições hierárquicas de classificações próprias do sistema lexical. Isto significa que há significados que, pelo seu domínio semântico, englobam outros significados menos abrangentes. Exemplo: Na taxionomia animal, mamífero engloba felino, canídeo, roedor, primata etc."*.

Para se obter a hiperonímia de uma palavra na língua Portuguesa, diante da escassez de bases de dados que fornecessem exatamente a hiperonímia de uma palavra, foi desenvolvido para este trabalho um processo computacional pesquisou em determinados dicionários *online* de língua Portuguesa. Como o texto fornecido por estes dicionários possuía a descrição do termo pesquisado e não exatamente a hiperonímia surgiu o problema de se localizar a hiperonímia no texto retornado pelo respectivo dicionário *online*. Diversos autores como Balakrishna em 2010 (BALAKRISHNA *et al*, 2010), utilizaram sequencias de "*part-of-speech*" pré-definidas ou Processamento de

Linguagem Natural para se identificar relações em textos. No caso dos dicionários *online*, por estes possuírem uma escrita particular com determinadas abreviações e *tags*, a utilização de uma tabela fornecida de sequencias "*part-of-speech*" e a utilização da PLN (Processamento de Linguagem Natural) não obteve os resultados esperados. Desta forma, para se localizar a hiperonímia de uma palavra, foi construído um processo computacional capaz de aprender a partir de exemplos compostos de localizações feitas pelo especialista em domínio em textos retornados pelo respectivo dicionário *online*. A partir destes exemplos, o processo cria regras de localização e as refina, testando-as sobre os exemplos, de forma a maximizar a capacidade de localização de termos diferentes dos exemplos fornecidos. A Figura 26 apresenta a interação do especialista em domínio com as funções que constroem as regras de localização a partir dos exemplos fornecidos:

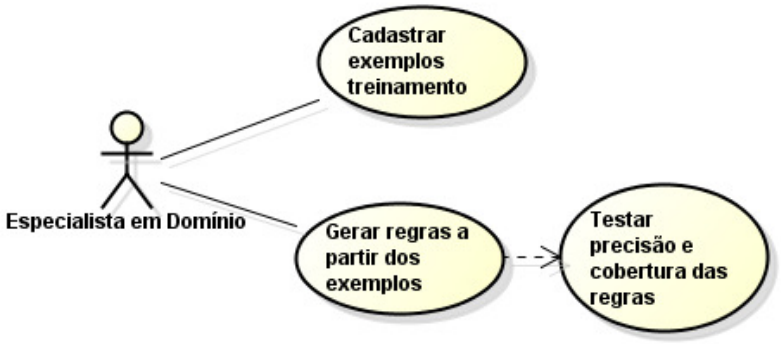


Figura 26 : Diagrama de Caso de Uso das funções que constroem as regras de localização das hiperonímias.

Na verdade, ao se identificar a hiperonímia de um termo, é identificada a relação "*é classificador de*" entre os termos no modelo apresentado anteriormente. Caso os termos sejam representativos de classes, então é identificada a relação "*é classificadora de*" entre as classes. Outro ponto importante a ser ressaltado é que a hiperonímia de um determinado termo não é única, podendo existir várias. Por exemplo: As hiperonímias de "Leite" podem ser "Laticínio", "Alimento" e "Bebida". Desta forma o processo de construção da taxonomia deverá tratar estas ambiguidades durante a construção dos níveis da taxonomia. A relação *"é classificador de*" entre os termos permitirá que um termo seja classificado por mais de um termo classificador. Já as instâncias e classes deverão ser colocadas na taxonomia com a utilização da relação "*é classificadora de*" que melhor as classifica, com a restrição de que cada instância ou classe deverá ser classificada por somente uma classe.

4.2 Descrição formal das etapas da solução do problema

Nas próximas seções serão apresentados os conceitos formais essenciais da solução e as

respectivas etapas da solução abstrata e do modelo de implementação do *framework* CDDW desenvolvido nesta tese de forma a validar os objetivos apresentados anteriormente.

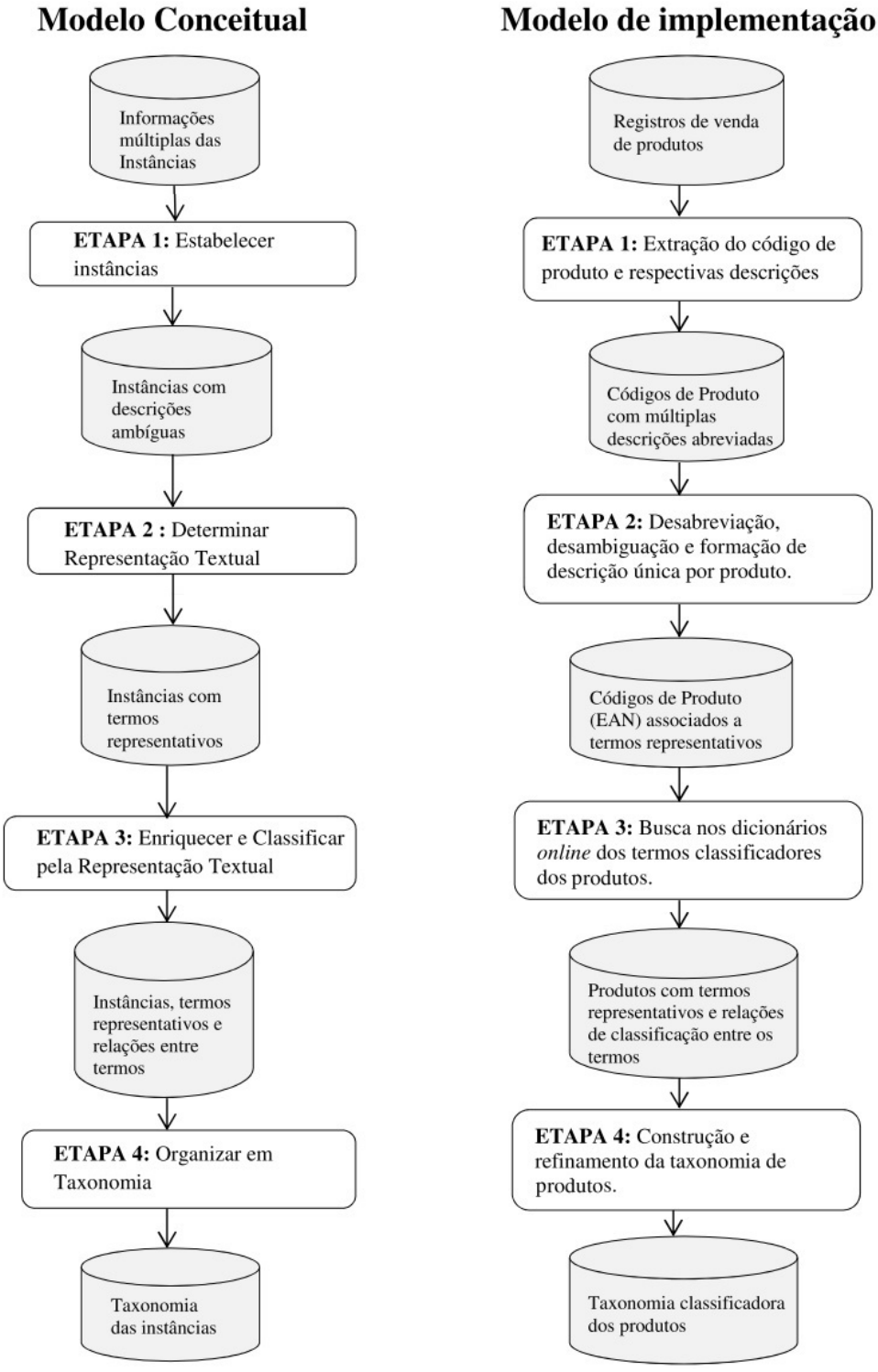


Figura 27 : Modelo conceitual e modelo de implementação de cada etapa do CDDW.

Conforme apresentado na Figura 27, cada etapa possui problemas específicos a serem resolvidos para que os objetivos propostos neste trabalho sejam alcançados. A Tabela 05 apresenta

um resumo dos problemas a serem resolvidos e uma breve descrição das técnicas utilizadas, juntamente com o respectivo embasamento teórico.

Tabela 05 : Etapas, problemas e respectivas técnicas aplicadas para a solução.

Etapa Conceitual	Etapa nesta tese	Problema da informação	Exemplo	Técnica a ser utilizada
4.3 Estabelecer Instâncias	**4.3.1. Seleção dos produtos a partir dos cupons fiscais**	Massiva quantidade de cupons fiscais	O total de cupons fiscais do estado do Rio de janeiro no ano de 2010 é de 3.5 bilhões de itens (registro de produtos vendidos)	O arquivo de itens de cupons fiscais será lido para se montar a dimensão produto, agrupando-se para cada EAN as respectivas descrições utilizadas pelos diversos emissores de cupom fiscal.
4.4 Determinar Representação Textual	**4.4.1 Desabreviação dos termos**	Abreviação dos termos	EAN 5601237219224 Qtd: 1977 Qtd Descrição 1531V JP MOSCATEL 750 420 *V JP MOSCATEL 750 7 VINHO BRANCOP.J.M.S.750 5 MOSCATEL DE SETUBAL DOC 01 750ML BACALHOA 3 VINHO MOSCATEL DE SETUBAL DOC 2 VNH MOSC PTG SETUBAL DOC 750ML 2 moscatel de setubal q bac 1 VINHO M.SETUBAL 750M	Processo de desabreviação com base na seleção da palavra com maior semelhança da palavra abreviada que esteja presente em um thesauro dentre um conjunto de palavras utilizadas nas descrições do EAN. Um processo derivado de Covington's distance metric foi utilizado para medir a semelhança entre as palavras abreviadas.
	4.4.2. Seleção dos termos pela frequência das descrições	Descrições conflitantes	PRODUTO 7791206016027 Qtd26 Qtd Descrição 18 GOLFINHO 5 ORCA JUJUBA UN 2 BAL GRIN BALEI FR 40G 1 BALA ORCA/GOLFINHO	Os termos desabreviados do grupo representativo serão pontuados em função da quantidade de ocorrências destes em cada posição nas descrições e pela quantidade de ocorrências de cada descrição. Esta pontuação definirá quais termos deverão ser selecionados e em que ordem irão compor a descrição única do produto.
	4.4.3. Descarte dos termos conflitantes pela frequência nas descrições	Ambiguidades	EAN 0048001006430 Qtd Descrição 1327 PASTA AMEN TRAB 462G 806 *PASTA AMEN TRAB 462G 19 MANTEIGA AMENDOIM CREAMY 13 PASTA AMENDOIM SKIPP	Os termos oriundos das descrições conflitantes serão descartados pois somente são considerados para a seleção dos termos representativos os termos que estiverem presentes em pelo menos 10% das descrições utilizadas para descrever o produto no cupom fiscal.
4.5 Utilizar Folksonomia para Enriquecer a Representação textual	**4.5.1 Busca dos termos classificadores em dicionários *online***	Falta dos termos generalizadores para criação da ontologia.	Termos representativos: VINHO MOSCATEL SETUBAL BALA JUJUBA ORCA GOLFINHO MANTEIGA PASTA AMENDOIM CREAMY SKIPP	A obtenção da generalização (hiperonímia) dos termos representativos do produto será feita a partir de consultas a dicionários *online*. A identificação da hiperonímia do termo consultado no texto retornado pelo dicionário será feita por meio de um agente de aprendizado de máquina.
4.6 Classificar e Organizar em Taxonomia	**4.6.1. Criação do grafo com base no primeiro termo representativo dos produtos.**	Ambiguidades nas relações de classificação.	Ambiguidade nos termos generalizadores recuperados da ontologia: ORCA GOLFINHO -> ANIMAL, MAR. PASTA -> MALETA.	Técnicas utilizadas nos trabalhos de Tomuro e Shepitsen em 2009 e Balakrishna em 2010, para a geração de uma ontologia enxuta e confiável (Guarino, 1998).
	4.6.2. Remoção das ambiguidades da taxonomia	Ambiguidades estruturais da taxonomia	JUJUBA -> DOCE -> ALIMENTO VINHO ->BEBIDA ALCOOLICA -> BEBIDA	Técnicas desenvolvidas para o refinamento das relações de classificação para gerar a taxonomia classificadora dos produtos. (DOMINGUES e REZENDE 2011), (MARTINS 2006)

4.3 Etapa: Estabelecer Instâncias

Esta etapa tem por objetivo a geração das instâncias (de produtos) com as suas respectivas descrições a partir dos respectivos registros de venda de produtos. A massiva quantidade de dados nos registros de venda (3.5 bilhões de registros) que deveria ser lida para se poder identificar cada produto e todas as respectivas descrições utilizadas nas vendas, juntamente com grande quantidade de produtos (aproximadamente cinco milhões) e descrições (13 milhões) representaram um obstáculo computacional para o processo de geração das instâncias com suas respectivas representações textuais (produtos com as diversas descrições utilizadas nas vendas). Para superar este obstáculo foram utilizadas algumas técnicas como busca binária e merge de conjuntos ordenados para que estas informações sejam obtidas em tempo exequível.

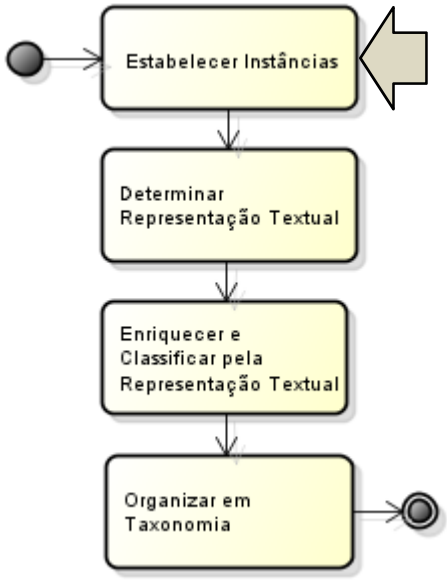


Figura 28 : Etapa 4.3

Não há novidades teóricas nesta etapa, sendo apenas um desafio de se utilizar técnicas computacionais já existentes para se obter os produtos com as respectivas descrições (Instâncias com as respectivas descrições) em um tempo de processamento exequível. Os detalhes desta implementação estão descritos na seção 5.1.1 do capítulo 5.

4.4 Etapa: Determinar representação textual

Nesta etapa serão identificados os termos representativos de cada instância. Estes termos representativos serão selecionados do conjunto de termos utilizados nas diversas descrições associadas a esta instância. Ao final desta etapa, cada instância será associada a um conjunto de termos representativos.

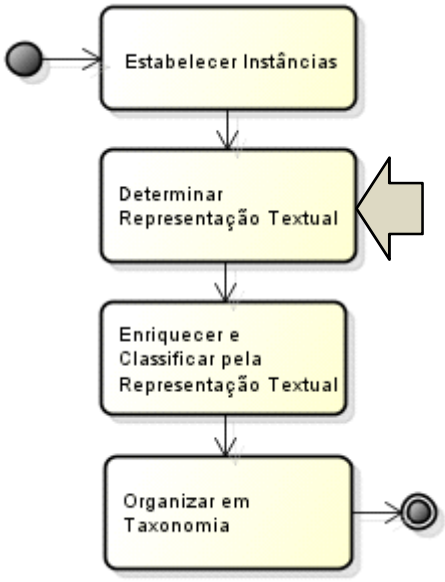


Figura 29 : Etapa 4.4

4.4.1 Objetivo desta etapa

O objetivo desta etapa é definir uma descrição única (sequência de termos) para cada produto (instância) a partir das diversas descrições que cada instância possui.

Cada instância possui diversas descrições e estas descrições podem possuir termos abreviados e símbolos. Conforme apresentado na Tabela 06, o processo proposto deverá definir uma sequência de termos representativa para cada EAN (código de produto), definindo uma sequencia de termos representativa para os produtos 7891038103503, 7891038106504, 7891038127400, 7891038195904 e 77900800143.

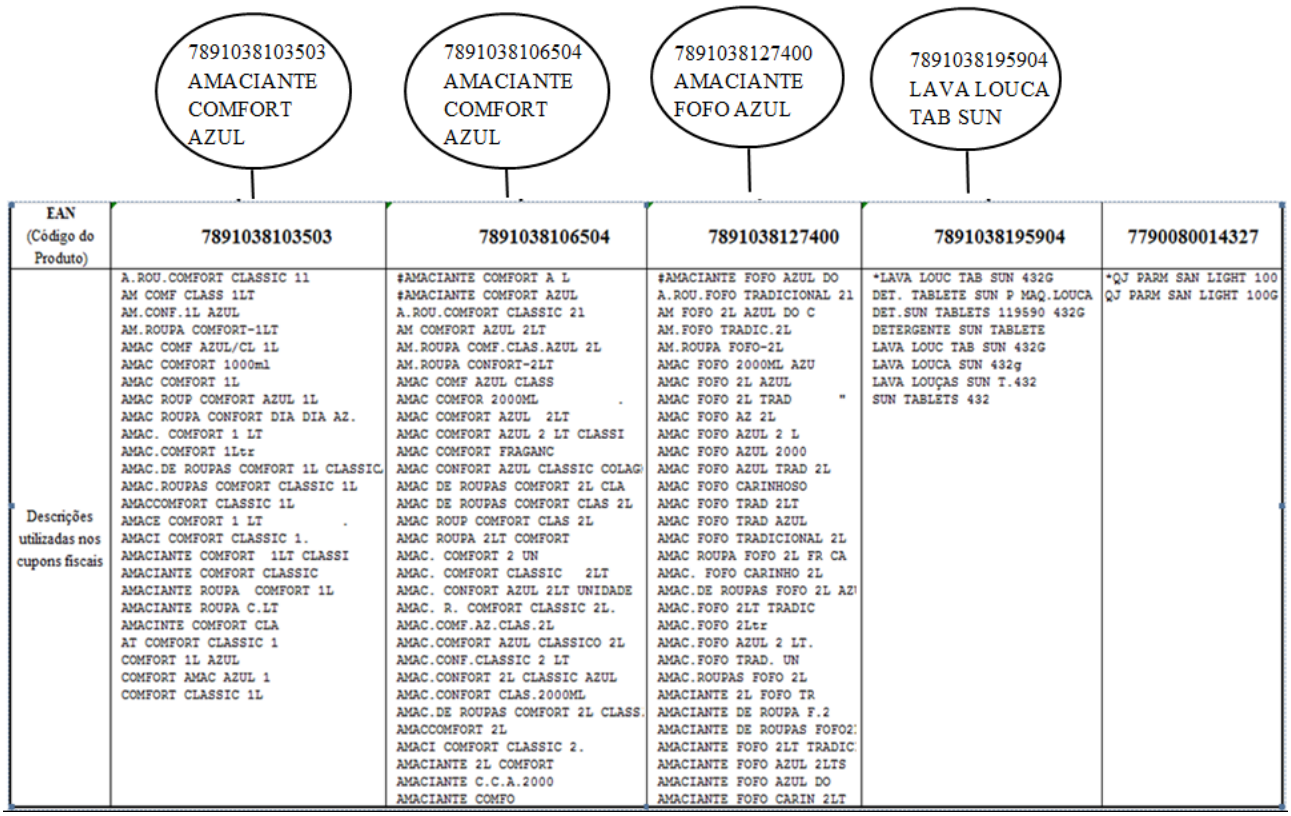
Tabela 06 : Exemplo de produtos com suas respectivas descrições oriundas dos registros de cupons fiscais.

EAN (Código do Produto)	7891038103503	7891038106504	7891038127400	7891038195904	7790080014327
Descrições utilizadas nos cupons fiscais	A.ROU.COMFORT CLASSIC 1l AM COMF CLASS 1LT AM.CONF.1L AZUL AM.ROUPA COMFORT-1LT AMAC COMF AZUL/CL 1L AMAC COMFORT 1000ml AMAC COMFORT 1L AMAC ROUP COMFORT AZUL 1L AMAC ROUPA CONFORT DIA DIA AZ. AMAC. COMFORT 1 LT AMAC.COMFORT 1Ltr AMAC.DE ROUPAS COMFORT 1L CLASSIC AMAC.ROUPAS COMFORT CLASSIC 1L AMACCOMFORT CLASSIC 1L AMACE COMFORT 1 LT AMACI COMFORT CLASSIC 1. AMACIANTE COMFORT 1LT CLASSI AMACIANTE COMFORT CLASSIC AMACIANTE ROUPA COMFORT 1L AMACIANTE ROUPA C.LT AMACINTE COMFORT CLA AT COMFORT CLASSIC 1 COMFORT 1L AZUL COMFORT AMAC AZUL 1 COMFORT CLASSIC 1L	#AMACIANTE COMFORT A L #AMACIANTE COMFORT AZUL A.ROU.COMFORT CLASSIC 2l AM COMFORT AZUL 2LT AM.ROUPA COMF.CLAS.AZUL 2L AM.ROUPA CONFORT-2LT AMAC COMF AZUL CLASS AMAC COMFOR 2000ML AMAC COMFORT AZUL 2LT AMAC COMFORT AZUL 2 LT CLASSI AMAC COMFORT FRAGANC AMAC CONFORT AZUL CLASSIC COLAG AMAC DE ROUPAS COMFORT 2L CLA AMAC DE ROUPAS COMFORT CLAS 2L AMAC ROUP COMFORT CLAS 2L AMAC ROUPA 2LT COMFORT AMAC. COMFORT 2 UN AMAC. COMFORT CLASSIC 2LT AMAC. CONFORT AZUL 2LT UNIDADE AMAC. R. COMFORT CLASSIC 2L. AMAC.COMF.AZ.CLAS.2L AMAC.COMFORT AZUL CLASSICO 2L AMAC.CONF.CLASSIC 2 LT AMAC.CONFORT 2L CLASSIC AZUL AMAC.CONFORT CLAS.2000ML AMAC.DE ROUPAS COMFORT 2L CLASS. AMACCOMFORT 2L AMACI COMFORT CLASSIC 2. AMACIANTE 2L COMFORT AMACIANTE C.C.A.2000 AMACIANTE COMFO	#AMACIANTE FOFO AZUL DO A.ROU.FOFO TRADICIONAL 2l AM FOFO 2L AZUL DO C AM.FOFO TRADIC.2L AM.ROUPA FOFO-2L AMAC FOFO 2000ML AZU AMAC FOFO 2L AZUL AMAC FOFO 2L TRAD AMAC FOFO AZ 2L AMAC FOFO AZUL 2 L AMAC FOFO AZUL 2000 AMAC FOFO AZUL TRAD 2L AMAC FOFO CARINHOSO AMAC FOFO TRAD 2LT AMAC FOFO TRAD AZUL AMAC FOFO TRADICIONAL 2L AMAC ROUPA FOFO 2L FR CA AMAC. FOFO CARINHO 2L AMAC.DE ROUPAS FOFO 2L AZ AMAC.FOFO 2LT TRADIC AMAC.FOFO 2Ltr AMAC.FOFO AZUL 2 LT. AMAC.FOFO TRAD. UN AMAC.ROUPAS FOFO 2L AMACIANTE 2L FOFO TR AMACIANTE DE ROUPA F.2 AMACIANTE DE ROUPAS FOFO2 AMACIANTE FOFO 2LT TRADIC AMACIANTE FOFO AZUL 2LTS AMACIANTE FOFO AZUL DO AMACIANTE FOFO CARIN 2LT	*LAVA LOUC TAB SUN 432G DET. TABLETE SUN P MAQ.LOUCA DET.SUN TABLETS 119590 432G DETERGENTE SUN TABLETE LAVA LOUC TAB SUN 432G LAVA LOUCA SUN 432g LAVA LOUÇAS SUN T.432 SUN TABLETS 432	*QJ PARM SAN LIGHT 100 QJ PARM SAN LIGHT 100G

Cada instância possui um código único identificador (EAN) e diversas descrições diferentes, sendo estas formadas por sequencias de termos e estes termos podem ser palavras abreviadas ou não. Desta forma, esta etapa terá de executar os seguintes passos para se definir a respectiva descrição representativa de cada instância:

1. Identificar qual termo é abreviação de outro termo (relacionamento "*é abreviação de*") no conjunto de termos que compõe cada descrição de cada instância e substituir o termo abreviado pelo termo não abreviado na respectiva descrição na qual o termo abreviado é utilizado.
2. Identificação dos termos representativos de cada instância em cada posição (ordem nas descrições) com o objetivo selecionar uma sequência de termos representativos de cada instância.
3. Descarte dos termos menos representativos de cada instância por meio der um critério que avalia a frequência e o posicionamento destes nas descrições. Pois determinados códigos de produtos podem ter sido utilizados erroneamente por contribuintes, levando a presença termos de outros produtos em determinadas instâncias. Desta forma, os termos que não são representativos do produto serão descartados com a aplicação deste critério.

O resultado desta etapa está representado pela Figura 30, apresentada a seguir. Pode-se observar que cada código de produto é associado a uma sequência representativa de termos. O código de produto 7790080014327 não obteve termos representativos, pois as suas descrições associadas não possuíam termos que estivessem dentro dos critérios de qualidade definidos para os processos de seleção.



EAN (Código do Produto)	7891038103503	7891038106504	7891038127400	7891038195904	7790080014327
Descrições utilizadas nos cupons fiscais	A.ROU.COMFORT CLASSIC 1l AM COMF CLASS 1LT AM.CONF.1L AZUL AM.ROUPA COMFORT-1LT AMAC COMF AZUL/CL 1L AMAC COMFORT 1000ml AMAC COMFORT 1L AMAC ROUP COMFORT AZUL 1L AMAC ROUPA CONFORT DIA DIA AZ. AMAC. COMFORT 1 LT AMAC.COMFORT 1Ltr AMAC.DE ROUPAS COMFORT 1L CLASSIC AMAC.ROUPAS COMFORT CLASSIC 1L AMACCOMFORT CLASSIC 1L AMACE COMFORT 1 LT AMACI COMFORT CLASSIC 1. AMACIANTE COMFORT 1LT CLASSI AMACIANTE COMFORT CLASSIC AMACIANTE ROUPA COMFORT 1L AMACIANTE ROUPA C.LT AMACINTE COMFORT CLA AT COMFORT CLASSIC 1 COMFORT 1L AZUL COMFORT AMAC AZUL 1 COMFORT CLASSIC 1L	#AMACIANTE COMFORT A L #AMACIANTE COMFORT AZUL A.ROU.COMFORT CLASSIC 2l AM COMFORT AZUL 2LT AM.ROUPA COMF.CLAS.AZUL 2L AM.ROUPA CONFORT-2LT AMAC COMF AZUL CLASS AMAC COMFOR 2000ML AMAC COMFORT AZUL 2LT AMAC COMFORT AZUL 2 LT CLASSI AMAC COMFORT FRAGANC AMAC CONFORT AZUL CLASSIC COLAG AMAC DE ROUPAS COMFORT 2L CLA AMAC DE ROUPAS COMFORT CLAS 2L AMAC ROUP COMFORT CLAS 2L AMAC ROUPA 2LT COMFORT AMAC. COMFORT 2 UN AMAC. COMFORT CLASSIC 2LT AMAC. CONFORT AZUL 2LT UNIDADE AMAC. R. COMFORT CLASSIC 2L. AMAC.COMF.AZ.CLAS.2L AMAC.COMFORT AZUL CLASSICO 2L AMAC.CONF.CLASSIC 2 LT AMAC.CONFORT 2L CLASSIC AZUL AMAC.CONFORT CLAS.2000ML AMAC.DE ROUPAS COMFORT 2L CLASS. AMACCOMFORT 2L AMACI COMFORT CLASSIC 2. AMACIANTE 2L COMFORT AMACIANTE C.C.A.2000 AMACIANTE COMFO	#AMACIANTE FOFO AZUL DO A.ROU.FOFO TRADICIONAL 2l AM FOFO 2L AZUL DO C AM.FOFO TRADIC.2L AM.ROUPA FOFO-2L AMAC FOFO 2000ML AZU AMAC FOFO 2L AZUL AMAC FOFO 2L TRAD AMAC FOFO AZ 2L AMAC FOFO AZUL 2 L AMAC FOFO AZUL 2000 AMAC FOFO AZUL TRAD 2L AMAC FOFO CARINHOSO AMAC FOFO TRAD 2LT AMAC FOFO TRAD AZUL AMAC FOFO TRADICIONAL 2L AMAC ROUPA FOFO 2L FR CA AMAC. FOFO CARINHO 2L AMAC.DE ROUPAS FOFO 2L AZ AMAC.FOFO 2LT TRADIC AMAC.FOFO 2Ltr AMAC.FOFO AZUL 2 LT. AMAC.FOFO TRAD. UN AMAC.ROUPAS FOFO 2L AMACIANTE 2L FOFO TR AMACIANTE DE ROUPA F.2 AMACIANTE DE ROUPAS FOFO2 AMACIANTE FOFO 2LT TRADIC AMACIANTE FOFO AZUL 2LTS AMACIANTE FOFO AZUL DO AMACIANTE FOFO CARIN 2LT	*LAVA LOUC TAB SUN 432G DET. TABLETE SUN P MAQ.LOUCA DET.SUN TABLETS 119590 432G DETERGENTE SUN TABLETE LAVA LOUC TAB SUN 432G LAVA LOUCA SUN 432g LAVA LOUÇAS SUN T.432 SUN TABLETS 432	*QJ PARM SAN LIGHT 100 QJ PARM SAN LIGHT 100G

Figura 30: Representação do resultado da definição das descrições das instâncias.

Para se obter o resultado representado pela Figura 30, foram definidas as seguintes etapas, as quais estão descritas a seguir:

4.4.2 Desabreviação dos termos;

4.4.3 Identificação dos termos representativos de cada instância;

4.4.4. Descarte dos termos conflitantes pela frequência nas descrições;

4.4.2 Desabreviação dos termos

Para se identificar os termos que irão compor a descrição final do produto (por EAN), será necessário primeiro desabreviar os termos das descrições de cada instância de produto (código de EAN), pois, o conjunto de palavras oriundo das descrições de um mesmo código de produto deverá

possuir palavras e abreviações relacionadas somente a ao produto em questão, salvo exceções como a utilização de um código errado de produto. Mais uma vez, a desabreviação permitirá a seleção dos termos identificadores desabreviados.

Deve-se ressaltar que as funções de comparação de *strings* atuais (Ver Tabela 07) não são específicas para se verificar a semelhança entre a palavra original e a palavra abreviada, desta forma considera-se os seguintes requisitos na comparação entre palavras desabreviadas e palavras abreviadas: A abreviação busca preservar as primeiras letras da palavra original e a abreviação busca remover prioritariamente as letras vogais, depois letras consoantes.

A Tabela 07 apresenta características dos algoritmos de semelhanças de *strings* disponíveis (GONDIM, 2006):

Tabela 07: Algoritmos de comparação de *strings* mais comuns (Gondim, 2006).

Algoritmo	Características	Uso para desabreviação
Levenshtein (OLIVEIRA 2009)	Pontua as inserções, remoções e substituições das letras para igualar as palavras. Considera o alinhamento da menor *string* na maior *string*.	Pontua a semelhança globalmente, sem prioridade para o prefixo das palavras.
Smith-Waterman (SMITH, T.F 1981)	Pontua as inserções, remoções e substituições das letras para igualar as palavras. Considera o alinhamento de *substrings* das palavras.	Pontua a semelhança globalmente, sem prioridade para o prefixo das palavras.
Stochastic Model (RISTAD and YANILOS 1996)	Alinha strings globalmente, podendo atribuir baixos escores para duas *strings* com sufixos semelhantes	Pontua a semelhança globalmente, sem prioridade para o prefixo das palavras.
Jaro Metric (BILENKO et al 2003)	Baseia no número de caracteres comuns entre duas *strings*, e na semelhança da ordem na qual estas duas cadeias de caracteres se apresentam.	Pontua a semelhança globalmente, sem prioridade para o prefixo das palavras.
Hamming Distance (NAVARRO 2001)	Permite apenas substituições não permitindo nem inserções, nem remoções.	Não adequado para abreviação.
Soundex Distance Metric (HALL e DOWLING 1980)	O objetivo deste método é transformar um nome em um código de quatro dígitos de forma tal que sons similares possuam estes quatro caracteres.	Ele só deve ser aplicado para verificar erros fonéticos, pois ele pode ter grandes erros quando usado para outra aplicação.
Covington's distance function (KONDRAK, 2003)	Este método também é utilizado para realizar comparações que levam em conta se o termo comparado é vogal ou consoante	O seu desempenho é muito ruim, pois ele utiliza a técnica de construir uma árvore usando o *depth-first search*.

O processo de semelhança de *strings* de Covington's é o processo que melhor se adequa às características citadas anteriormente, pois pontua de forma diferente as vogais e as consoantes. O algoritmo de Covington's (KONDRAK, 2003) baseia-se em penalidades para vogais e consoantes divergentes entre as *strings*. As *strings* são alinhadas globalmente, podendo atribuir baixos escores para duas *strings* com sufixos semelhantes.

A seguir estão algumas características adicionadas ao algoritmo de Covington para verificar a semelhança entre *strings* com base na abreviação:

- A semelhança de letras não repetidas tem um peso maior que as letras repetidas na semelhança (esta condição previne similaridades entre palavras diferentes que possuem letras comuns, como por exemplo: BATATA e BANANA). Esta característica substitui a pontuação diferenciada para vogais e consoantes.
- A semelhança entre as primeiras letras de cada *string* tem um peso maior que as letras posicionadas no final de cada *string*.
- A semelhança entre cadeias de letras contínuas tem maior pontuação do que a semelhança de várias cadeias descontínuas de letras, mesmo que o conjunto de letras semelhantes seja igual.
- Utiliza-se uma penalidade maior para a presença de letras na *string* menor que estejam ausentes na *string* maior, pois a abreviação consiste na remoção de letras de uma palavra (para gerar a palavra menor – abreviada) e não a inclusão de letras novas na palavra abreviada.
- Deve-se considerar uma penalidade reduzida para as letras restantes no final da *string* maior, após o término de todas as equivalências de letras entre as *strings*.

O total de pontuação gerado das comparações será dividido por um fator calculado a partir dos tamanhos das duas *strings* comparadas para que o fator de semelhança resultante possa se usado para mensurar a semelhança abreviada de *strings* de diferentes tamanhos.

FatorLetraRepetida (i) = { 0.40 se letra(i) for repetida na palavra | 1.10 se letra(i) não for repetida}

FatorPosição (i) = FatoPosicao (i-1) - 2/ TamanhoString

L1 = TamanhoMaiorStr

L2 = TamanhoMenorStr

P1 = (L1 + L2) / L1

P2 = (L1 + L2) / L2

$$PontosS1 = \sum_{1}^{L2} i \;\; P1 * \; LetraPuladaS1 * FatorPosição(i) * FatorLetraRepetida(i)$$

$$PontosS2 = \sum_{1}^{L2} i \;\; P2 * \; LetraPuladaS2 * FatorPosição(i) * FatorLetraRepetida(i)$$

$$PontosResto = \sum_{L2}^{L1} i \;\; P1 * \; LetrasRestanteS1 * \; FatorPosição(i) * FatorLetraRepetida(i) \; /4$$

FatorCadeiasContínuas: É um fator que é proporcional ao tamanho das cadeias contínuas de caracteres iguais. O Algoritmo apresentado a seguir demonstra com mais clareza o uso deste fator.

PontosS1 = PontosS1/L1 + PontosResto/L1

PontosS2 = PontosS2/L2

Semelhança_Abreviada = 1 - (PontosS1 + PontosS2) / 2

O algoritmo 01 apresentado a seguir contém o pseudocódigo da função de similaridade abreviada utilizado neste trabalho. **É importante ressaltar que as constantes utilizadas pelo algoritmo foram definidas por um processo interativo no qual se utilizou um conjunto de termos abreviados e não-abreviados oriundo das descrições dos produtos. Foram realizadas várias execuções com a respectiva análise dos resultados (fatores de similaridade) gerados e os devidos ajustes das constantes**. **Foram executadas 15 interações com aproximadamente 6000 termos.** O resultado da similaridade entre os termos do conjunto foi colocado em uma planilha e esta foi ordenada pelos fatores de similaridade. Desta forma pode-se validar e ajustar o algoritmo de calculo de similaridade e calibrar os fatores referentes a cada característica de similaridade (ordem das letras, cadeias contínuas semelhantes, falta de letras na *string* de maior tamanho e etc) da função.

Algoritmo 01: Calcula o fator de similaridade entre *strings* considerando abreviações.

Entrada:
S1-Maior String
S2-Menor String
Saída:
FS – Fator de similaridade abreviada
Constantes:
_FDiagS = 1.10; - Fator para sequencias contínuas de letras
_FDiagNS = 0.60; - Fator de penalidade para semelhanças de sequências de letras não contínuas
_FSeq = 1.30; - Fator de similaridade
_FPos = 0.85; - Fator de penalidade para semelhança para posições finais das *strings*
_FInic = 1.40; - Fator de similaridade inicial
_FRepete = 0.90; - Fator de degradação sobre semelhança baseada em caracteres repetidos.
Variáveis:
S1, S2 : *strings* a serem comparadas
L1 : Numero de caracteres de S1
L2 : Numero de caracteres de S2
M : Vetor [0..L1, 0..L2] de caracteres – Armazena as respectivas posições com caracteres iguais
Total, Fator, FDiag, FDist : Real - Fatores usados para o cálculo da similaridade
Colseq - Última coluna semelhante. Usado para verificar se é uma sequencia contínua de letras semelhantes
Iguais : Conjunto de caracteres iguais entre as strings
Pseudocódigo:

```
1:      L1 = Numero de caracteres de S1
2:      L2 = Numero de caracteres de S1
3:         { calcula o tamanho médio das duas strings comparadas }
4:      MaxTam ← (L2*0.9) + 0.30 * (L1-L2)
5:        { zera a matriz quadrada de semelhança }
6:      for i ← 0 to _Tam_Str
7:        for j ← 0 to _Tam_Str
8:          M[i,j] ← '.'
```

```
9:        end for
10:     end for
11:     M0,0 ← 'X'
12:           { zera o conjunto de letras iguais }
13:     Iguais = <Conjunto Vazio>
14:       { inicializa a matriz quadrada de semelhança }
15:     for cada Elemento Si pertencente a S1
16:       for cada Elemento Sj pertencente a S2
17:         if (Si = Sj) AND (i >= J)
18:           if (Si IN Iguais)      { Letras repetidas contam menos }
19:           then Mi,j ← 'X'
20:           else Mi,j ← 'N'
21:           incluir Si em Iguais
22:         end if
23:       end for
24:     end for
25:        { incializa os fatores que compõe o calculo da semelhança }
26:     if S1_1 = S2_1     { letra inicial de cada string tem uma pontuação maior }
27:     then Total ← 0.0
28:     else Total ← -1.5
29:     end if
30:     Fator ← _FInic
31:     ColSeq ← 0
32:     FDiag ← _FDiagS
33:     for cada Li pertencente a S1
34:       FDist ← Fator
35:       LI     ? 1
36:       for CI pertencente a S1
37:         if (MCI,LI = 'X') OR (MCI,LI = 'N')   then
38:           if ColSeq >= CI then            { Verifica se a semelhança é de uma sequencia contínua de letras }
39:             FDiag ← _FDiagNS
40:           end if
41:           FDiag ← FDiag * _FSeq
42:           if MCI,LI = 'X'
43:           then Total ← Total + Fator * FDiag * FDist
44:           else Total ← Total + Fator * FDiag * FDist * _FRepete   { Penaliza as semelhanças de letras repetidas }
45:           end if
46:           Colseq ← CI                                          { marca o início da cadeia de letras semelhantes }
47:         else
48:           FDiag ← _FDiagS
49:         end if
50:         LI ← LI + 1
51:         FDist ← FDist * _FPos     { Decrementa o fator à medida que se compara as letras do meio para o final }
52:       end for
53:       Fator ← Fator * _FPos
54:     end for
55:     SemelhancaAbreviada ← Total / MaxTam  { torna a pontuação independente do tamanho das strings }
56:   end
```

Deve-se destacar um instrumento que foi fundamental para a realização deste trabalho, o ***Thesauro***. Foi utilizado um *Thesauro* denominado *LABEL-LEX (LABEL-LEX)*, contendo aproximadamente um milhão de palavras inflexionadas, com os devidos *lemmas*, classificações

gramaticais e atributos morfológicos.

O processo de desabreviação dos termos é feito de forma progressiva, em função do conjunto dos termos utilizados para a seleção:

Em primeiro lugar são feitas desabreviações nos conjuntos de termos de cada código de produto, verificando-se a semelhança abreviada entre os termos oriundos das descrições de uma única instância. Desta forma, amplia-se a possibilidade de se identificar um termo desabreviado que seja representativo da instância em questão, pois o conjunto de termos utilizado na seleção é oriundo das descrições da instância.A desabreviação dos termos é feita em dois passos, em função do termo desabreviado existir ou não no *Thesauro* utilizado:

Desabreviação para termo existente no *Thesauro*: No conjunto de termos das descrições de um mesmo prefixo de EAN (produtos de um fabricante), pares de palavras com fator de semelhança superior a 0.80 serão considerados semelhantes se uma constar no *Thesauro* e a outra não. A palavra que não constar no *Thesauro* será substituída pela outra palavra semelhante, independentemente do tamanho ou acentuação de cada uma destas palavras.

Desabreviação para termo não existente no *Thesauro*: No conjunto de termos das descrições de um mesmo prefixo de EAN (produtos de um fabricante), pares de palavras com fator de semelhança superior a 0.95 serão considerados semelhantes se nenhuma das duas palavras estiver presente no *Thesauro*. A menor palavra ou a menos acentuada (caso tenham tamanhos iguais) será substituída pela outra palavra.

Os fatores 0.80 e 0.95 foram definidos a partir de diversas interações com a respectiva análise do resultado das comparações de um conjunto dos termos oriundos das descrições de produtos. Este conjunto de comparações foi colocado em uma planilha, ordenado pelo fator de semelhança e analisado. Durante a análise destes resultados observou-se que o processo de desabreviação deveria considerar, além do fator de semelhança calculado, o fato dos termos semelhantes existirem ou não no *Thesauro*.

Após a desabreviação dos termos em cada instância, os termos desabreviados são inseridos em uma tabela de termos, na qual todos os termos desabreviados de cada instância são inseridos. No momento desta inserção são executadas as seguintes operações:

- Verificado se o termo já existe na tabela de termos;
- Verificado se já existe na tabela de termos um termo semelhante (que exista no *Thesauro*) ao termo a ser inserido. Neste caso, o termo da tabela de termos será considerado como a desabreviação do termo a ser inserido, sendo o termo a ser inserido

substituído pelo termo da tabela de termos.

- Verificado se já existe na tabela de termos um termo semelhante (que NÃO exista no *Thesauro*) ao termo a ser inserido (que exista no *Thesauro*). Neste caso, o termo a ser inserido será considerado como a desabreviação do termo da tabela, modificando o termo da tabela e as descrições que o utilizam (Neste caso é utilizada uma semelhança igual ou superior a 0.95).

Conforme apresentado no diagrama da Figura 31, o processo de desabreviação é aplicado com um grau de semelhança mais abrangente no conjunto restrito de descrições de um prefixo EAN (produtos de um fabricante), pois os produtos de um mesmo fabricante possuem características comuns, o que leva a suas respectivas descrições a possuírem palavras e abreviações restritas a determinados tipos de produtos. Esta característica pode ser observada na Figura 32 nas amostras de desabreviação dos produtos da BATAVO, prefixo de EAN 78910980.

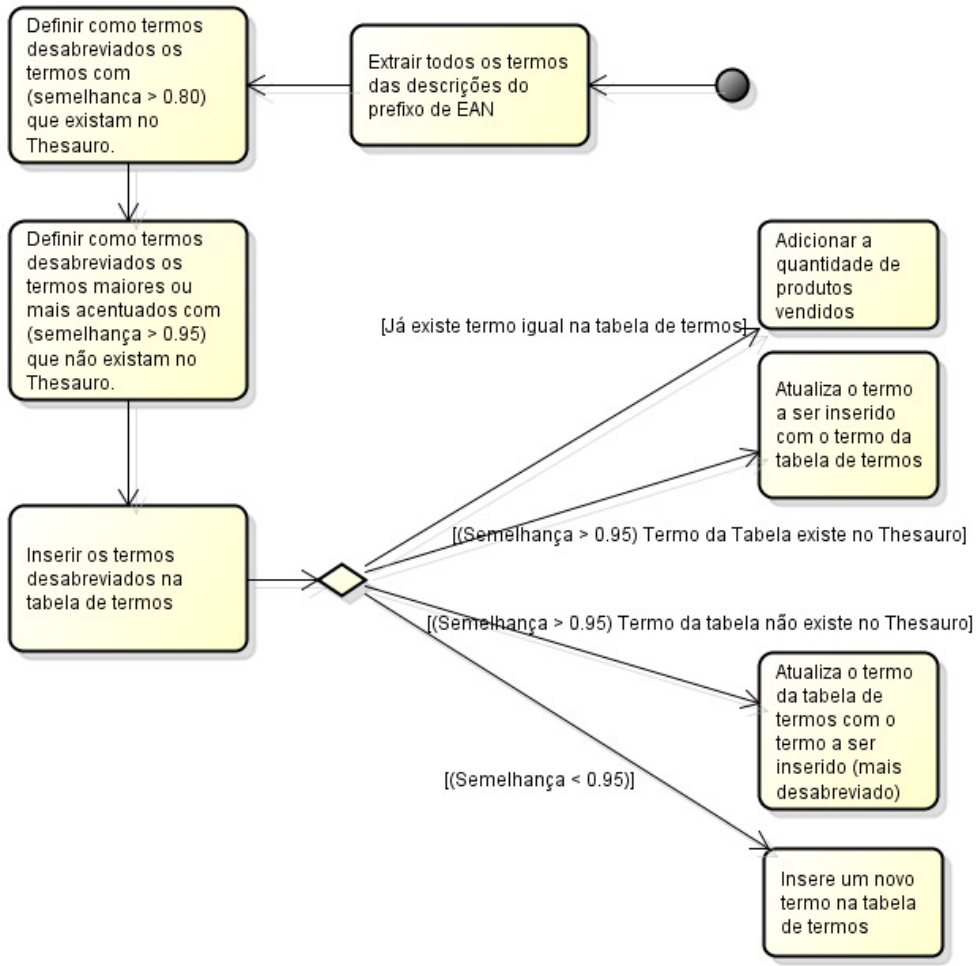


Figura 31: Processo de desabreviação dos termos de um prefixo de EAN.

Como podemos observar na Figura 32, que apresenta alguns exemplos da aplicação da desabreviação no prefixo 78910980 (fábricas da BATAVO), composto de 32 Produtos (EANs) diferentes, a desabreviação para palavras existentes no *Thesauro*, executada prioritariamente, ajusta diversas abreviações para termos significativos na identificação do produto. Já a desabreviação para palavras não existentes no *Thesauro* atinge, na sua maioria, termos de marca ou nomes de fabricantes dos produtos (nomes próprios), o fato destes termos não existirem no *Thesauro* impede que o processo identifique quando um termo não necessita mais ser desabreviado. Este problema é minimizado com aplicação de um critério de semelhança mais rígido para definir semelhança entre termos que não existem no *Thesauro* :

Desabreviação de pares de palavras para palavras presentes no *Thesauro*

```
MORANGO <- MORANG Fator_semelhança:0,9929
EXTRA <- EXTR Fator_semelhança:0,9900
CREME <- CREM Fator_semelhança:0,9900
LITRO <- LITR Fator_semelhança:0,9900
FERRO <- FERR Fator_semelhança:0,9900
FIBRA <- FIBR Fator_semelhança:0,9900
SUCO <- SCO Fator_semelhança:0,9250
SEMIDESNATADO <- SEME Fator_semelhança:0,8937
INTEGRAL <- INEGRAL Fator_semelhança:0,8883
LEITE <- LTE Fator_semelhança:0,8880
PENSE <- PES Fator_semelhança:0,8580
FRUTAS <- FTAS Fator_semelhança:0,8125
FRUTAS <- FTA Fator_semelhança:0,8104
```

Desabreviação de pares de palavras para palavras NÃO presentes no *Thesauro*

```
PESSEGO <- PESSEG Fator_semelhança:0,9929
BATAVO <- BATAV Fator_semelhança:0,9917
LACTEA <- LACT Fator_semelhança:0,9847
NATURIS <- NATUR Fator_semelhança:0,9760
ACHOCOL <- ACHOC Fator_semelhança:0,9760
LIGHT <- LIGH Fator_semelhança:0,9725
SENSY <- SENS Fator_semelhança:0,9725
PESSEGO <- PESS Fator_semelhança:0,9708
ACTIV <- ACT Fator_semelhança:0,9680
LACTEA <- LAC Fator_semelhança:0,9644
NATURIS <- NAT Fator_semelhança:0,9584
```

Desabreviação de termos na tabela de termos

```
---TERMO AJUSTADO DE TAB_TERMO LISO <- LISOS  Semelhanca: 0,9450
---TERMO AJUSTADO DE TAB_TERMO CONDI <- CONDIC       Semelhanca: 0,9542
---TAB_TERMO AJUSTADO APERITIVO <- APER       Semelhanca: 0,9246
---TERMO AJUSTADO DE TAB_TERMO LICOR <- LIC   Semelhanca: 0,9360
---TAB_TERMO AJUSTADO SILVESTRE <- SILVEST    Semelhanca: 0,9596
---TERMO AJUSTADO DE TAB_TERMO ORIGINAL <- ORIGIN     Semelhanca: 0,9574
---TERMO AJUSTADO DE TAB_TERMO TRADICAO <- TRADI      Semelhanca: 0,9451
```

Figura 32: Desabreviação de termos presentes nas descrições dos produtos do prefixo 78910980

4.4.3 Identificação dos termos representativos de cada instância

Após o processo de desabreviação dos termos das descrições do produto foi possível definir

quais termos desabreviados irão compor a descrição única para o respectivo produto. Esta descrição única do produto será composta por uma sequência de termos representativos selecionados dentre os termos posicionados na respectiva ordem das descrições desabreviadas. Desta forma, foram consideradas as seguintes premissas neste processo:

- O primeiro termo (de cada descrição) possui uma importância singular na identificação do produto.
- A ordem dos termos é considerada. O termo mais à esquerda é o que identifica o produto, o termo seguinte complementa a identificação e assim sucessivamente. Por exemplo: “feijão preto”, “queijo provolone” e “Fanta Uva Diet”.
- Termos que possuam menos que três letras não devem ser considerados para o processo de desabreviação e seleção dos termos representativos. Apenas deve-se preservar a ordem dos outros termos da respectiva descrição para que sejam analisados corretamente pelo processo que seleciona a sequencia de termos representativos do produto.
- Em determinadas descrições existem termos identificadores do fabricante do produto, como por exemplo: “DESODORANTE **NIVEA** AERO DRY”, “LASANHA **SADIA** BOLONHESA” e “TEMPERO **AJINIOMOTO** VIDRO”. Normalmente estes termos não constam no *Thesauro*.

Durante a desabreviação dos termos que compõem as descrições dos produtos de um mesmo EAN, são descartados os termos que tenham menos de três letras, mas preserva-se a ordem dos termos da respectiva descrição, para que a posição real destes na descrição seja considerada para se determinar a sequência de termos representativos do produto em questão. A Figura 33 apresenta a seleção dos termos do produto de código EAN igual a 7896050505212.

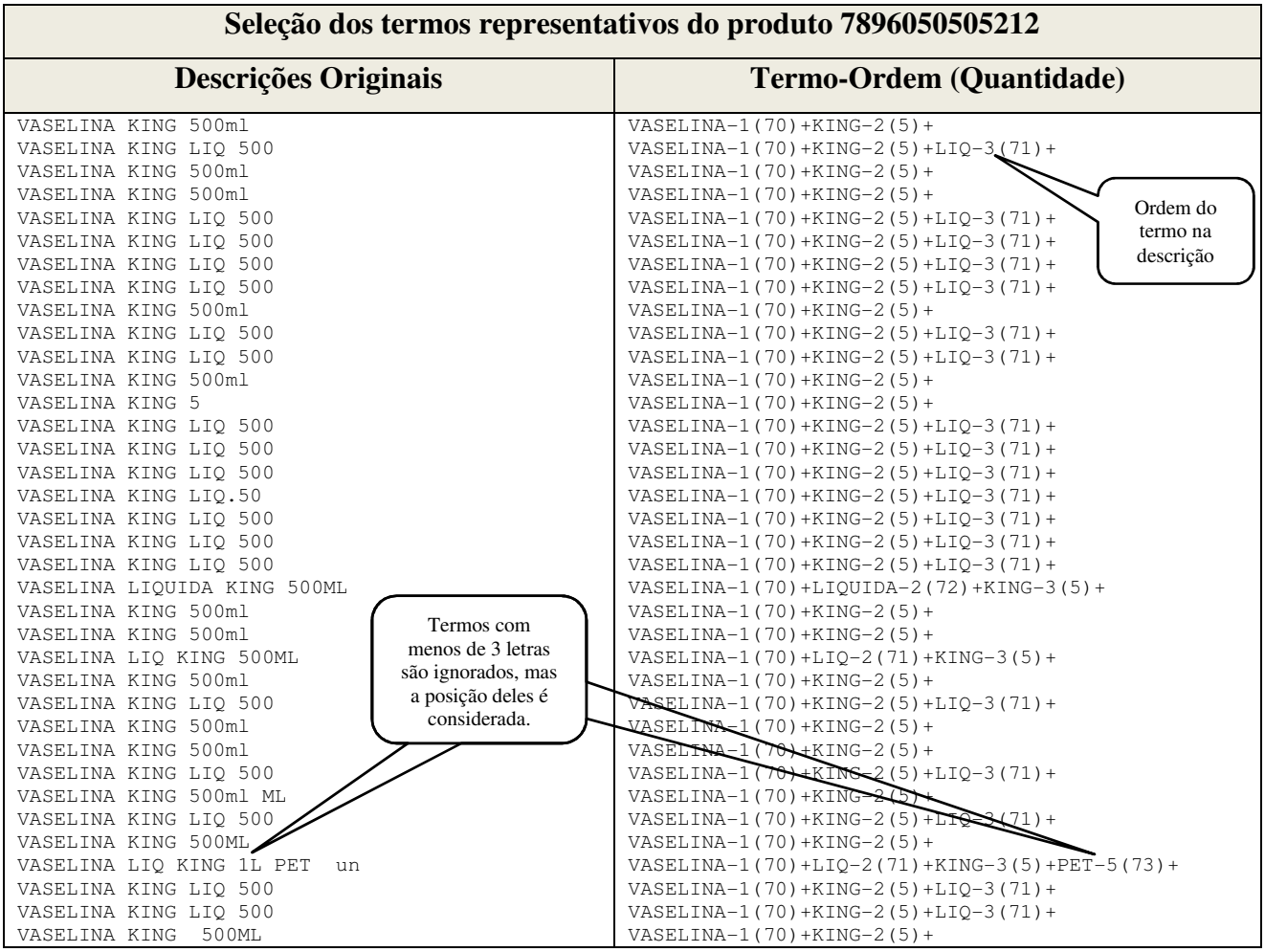


Figura 33: Seleção dos termos para compor a descrição representativa do produto.

Os termos são pontuados por um fator formado pelo somatório da quantidade vendida das respectivas descrições que utilizam este termo na respectiva ordem (ordem do termo em cada descrição) que está sendo avaliada, multiplicado pelo número de descrições (diferentes) que utilizam este termo nesta respectiva ordem (posição), conforme a fórmula apresentada a seguir:

$$QVT\ (Ordem) = \sum Quantidade\ Vendida\ por\ descrições\ com\ o\ termo\ na\ posição\ (Ordem)$$

$$QDT(Ordem) = \ Quantidade\ de\ Descrições\ com\ o\ termo\ na\ posição\ (Ordem)$$

$$\textbf{\textit{Representatividade Termo na ordem}} = \textbf{\textit{RT}}\ (Ordem) = QVT\ (Ordem) * QDT\ (Ordem)$$

A seleção dos termos da descrição identificadora do produto tem como base a identificação

dos termos de maior frequência dentre o conjunto de termos de cada ordem (posição na descrição) multiplicado pela quantidade de descrições diferentes (dentro deste EAN) que possuem o respectivo termo. No caso dos dois primeiros termos de cada instância, esta seleção é feita em dois passos:

1. Primeiro seleciona-se o termo representativo (de cada posição) dentre os termos que existam no *Thesaur*o (BREITMAN, HOWARD e CASANOVA, 2005). Caso a pontuação do termo selecionado seja superior a um limite mínimo de pontuação, será considerado representativo para a respectiva posição.
2. Caso a pontuação do termo selecionado não seja superior ao limite mínimo de pontuação definido a seguir na etapa 4.4.4, serão utilizados os termos que não constam no *Thesaur*o para se selecionar o termo com maior pontuação.

O processo se inicia com o seleção do primeiro termo (posição 1) dentre o conjunto de termos de posição 1 das respectivas descrições. Como um termo pode estar presente em diversas posições das várias descrições do produto, não será permitida a repetição do termo caso este termo já esteja selecionado como termo identificador em uma posição mais à esquerda. Neste caso, serão selecionados os termos de maior frequência das próximas posições que não estejam no conjunto de termos já selecionados. Conforme apresentado pela Tabela 08, está o valor da representatividade de cada termo em cada posição (**RT**(ordem) de cada termo).

Tabela 08: Representatividade por posição de cada termo do produto de EAN 7896050505212.

NOME	**Existe no *Thesauro***	**Quantidade Vendida**	**Quantidade Descrições**	**RT (1)**	**RT (2)**	**RT (3)**	**RT (4)**	**RT (5)**	**RT (6)**
VASELINA(8305)	Sim	3453	36	**3453*36**	0*36	0*36	0*36	0*36	0*36
KING(357)	Sim	3453	36	0*36	3261*36	192*36	0*36	0*36	0*36
LIQUIDA(3014)	Sim	2007	22	0*22	192*22	1815*22	0*22	0*22	0*22
PET(3150)	Não	53	1	0*1	0*1	0*1	0*1	53*1	0*1

Conforme o critério de Representatividade do Termo na Ordem (RT), a sequência de termos selecionada seria formada por: VASELINA, KING, LIQUIDA e PET.

4.4.4 Descarte dos termos menos representativos de cada instância

Conforme os dados apresentados na Tabela 08, os termos representativos selecionados foram VASELINA, KING e LIQUIDA. O próximo termo candidato seria PET, mas como a quantidade vendida deste na posição quatro é igual a zero (0*1), o que fez com que este termo seja

considerado. Mesmo que existam termos com alguma pontuação para as próximas posições, somente serão considerados os termos nos quais a respectiva quantidade vendida (das descrições que o utilizam) seja superior a 10% da quantidade total vendida do respectivo produto (DOTSIKA, 2009), (PLANGPRASOPCHOK, LERMAN e GETOOR 2008). Como o total de produtos vendidos é 5319, não serão considerados os termos com quantidade vendida inferior a 530. Vale ressaltar que não é permitida a repetição de termos representativos em um mesmo produto. Desta forma, somente serão selecionados os termos representativos da instância, mesmo que sejam poucos ou existam produtos nos quais não foram identificados termos representativos.

Tabela 09: Definição dos termos representativos do produto de EAN 7891097019692.

Descrições Originais	Termo-Ordem (Quantidade)
ALIM NATURIS 1L UVA	ALIM-1(17)+NATURIS-2(20)+UVA-4(101)+
BEBIDA SOJA B.1.N.U.	BEBIDA-1(21)+SOJA-2(15)+
SCO BATAVO SOJA1L	SCO-1(85)+BATAVO-2(2)+
SOJA+SUCO BATAVO 1L	SOJA-1(15)+SUCO-2(78)+BATAVO-3(2)+
BEBIDSOJA BATAVO 1LT	BEBIDSOJA-1(86)+BATAVO-2(2)+
SUCO SOJA BATAVO UVA	SUCO-1(78)+SOJA-2(15)+BATAVO-3(2)+UVA-4(101)+
SUCO SOJA BATAVO UVA	SUCO-1(78)+SOJA-2(15)+BATAVO-3(2)+UVA-4(101)+
SC BATAVO UVA/SJ 1L	BATAVO-2(2)+UVA-3(101)+
Suco Batavo Uva Tp 1l	SUCO-1(78)+BATAVO-2(2)+UVA-3(101)+
BEB DE SOJA BAT 1L	BEB-1(19)+SOJA-3(15)+BAT-4(4)+
SUCO BATAVO 1LT SOJA	SUCO-1(78)+BATAVO-2(2)+SOJA-4(15)+

Pode-se observar na Tabela 09 que a aplicação do *Thesauro* na desabreviação não pode ser total, sob o risco de perda das palavras que representam nomes próprios. A desabreviação do EAN 7891097019692, por exemplo, possui diversos termos representativos do fabricante do produto como BATAVO e outros como BAT (ambos não existem no *Thesauro*). Se aplicarmos a desabreviação somente para termos presentes no *Thesauro* geraríamos a desabreviação errônea de BAT (Batavo) para BATATA ao adicionar BAT na tabela final de termos, pois BATAVO não existe no *Thesauro* e existe BATATA neste. Desta forma, a prioridade para termos que existam no *Thesauro* é aplicada apenas para a seleção dos dois primeiros termos de cada instância, não podendo ser usada de forma excessivamente abrangente sob o risco de haver perda dos nomes próprios das descrições como: BATAVO, PARMALAT, KIBON, COLGATE etc.

Tabela 10: Representatividade por posição de cada termo do produto de EAN 7891097019692.

NOME	Existe no *Thesauro*	Quantidade Vendida	Quantidade Descrições	RT (1)	RT (2)	RT (3)	RT (4)	RT (5)	RT (6)
ALIMENTO(2174)	Sim	527	1	527*1	0*1	0*1	0*1	0*1	0*1
NATURIS(4887)	Não	527	1	0*1	527*1	0*1	0*1	0*1	0*1
PESS(3079)	Não	594	2	0*2	0*2	67*2	527*2	0*2	0*2
BEBIDA(557)	Sim	430	2	430*2	0*2	0*2	0*2	0*2	0*2
SOJA(2175)	Sim	1412	11	186*11	367*11	63*11	796*11	0*11	0*11
SUCO(17)	Sim	1237	10	1051*10	186*10	0*10	0*10	0*10	0*10
BATAVO(3943)	Não	1504	12	0*12	1318*12	186*12	0*12	0*12	0*12
BEBIDSOJA(4897)	Não	200	1	200*1	0*1	0*1	0*1	0*1	0*1
PES(1989)	Não	714	7	0*7	0*7	0*7	0*7	714*7	0*7

Pode-se observar na Tabela 10 na seleção dos termos representativos do EAN que os termos selecionados não correspondem aos termos originários da descrição que teve a maior quantidade vendida, pois a fórmula levou em consideração a quantidade de descrições diferentes na seleção dos termos representativos do produto. Sendo selecionada a sequencia de termos: "SUCO + SOJA + BATAVO + PESS". O Próximo termo candidato seria PES, o qual foi descartado por não possuir pontuação mínima na posição cinco (ordem 5). Observa-se que o termo ALIMENTO, presente no início da descrição, não foi selecionado como representativo, pois possuía pontuação adequada apenas na primeira posição, mas sua pontuação era inferior ao termo SUCO. **Desta forma, o processo de selecionar a melhor pontuação em cada posição minimiza a seleção de termos sinônimos ou semelhantes como representativos de cada instância**, priorizando a seleção de termos classificadores, subclassificadores e termos identificadores de marca, pois cada termo em cada posição nas descrições tem um determinado papel no significado da descrição. Outro ponto importante deste processo é o **descarte dos termos das descrições erradas oriundas da vendas de produtos com a utilização de um código de EAN errado.** Neste caso, essas descrições tem os termos descartados pela quantidade de ocorrências (bem inferior à frequência de corte).

4.4.5 Conclusões da etapa

Concluindo, nesta etapa, a aplicação da função de similaridade abreviada pelo processo de desabreviação em passos com o uso do *Thesauro*, seguindo-se da definição da sequencia de termos

representativos do produto, selecionados com base na frequência de cada termo em cada posição de cada descrição, o que permitiu a geração de uma descrição única e representativa para cada instância processada.

Desta forma, esta etapa apresenta como propostas de contribuição as seguintes premissas:

- Para utilizar a similaridade abreviada as seguintes informações devem ser consideraradas:
    - A posição das letras nas *strings* comparadas, priorizando-se as letras iniciais de cada *string*;
    - Que a similaridade entre letras não repetidas nas palavras tem maior prioridade que a similaridade entre letras repetidas;
    - A similaridade é proporcional ao tamanho das sequências contínuas de letras iguais.
- Para o processo de desabreviação de grupos de descrições, deve-se processar com prioridade os termos presentes no *Thesauro*, mas deve-se considerar também os termos que não estão presentes no *Thesauro*, pois estes podem ser nomes próprios ou nomes de marca.
- A seleção de termos representativos de uma instância a partir de múltiplas descrições deve considerar se o termo existe no *Thesauro* utilizado, se a frequência de cada termo em cada posição das descrições de forma que seja selecionada uma sequencia de termos capaz de qualificar de forma global e específica a respectiva instância.
- Diferentes termos situados na mesma posição, em diferentes descrições de uma mesma instância, tem grande possibilidade de serem sinônimos.

4.5 Etapa: Enriquecimento e classificação pela representação textual

Todas as representações textuais das instâncias determinadas na etapa anterior foram geradas a partir de informações já existentes, que estavam associadas a cada uma das instâncias. Diversos termos identificados como representativos de determinadas instâncias podem ser considerados como termos classificadores destas instâncias, dependendo da frequência no conjunto total de termos representativos das instâncias. Mas este conjunto de informações não é suficiente para a se determinar a generalização de cada instância para a construção da taxonomia destas instâncias. Esta etapa propõe como solução deste problema buscar em fontes externas as informações para classificar as instâncias. Estas fontes externas são os dicionários *online* da Língua Portuguesa.

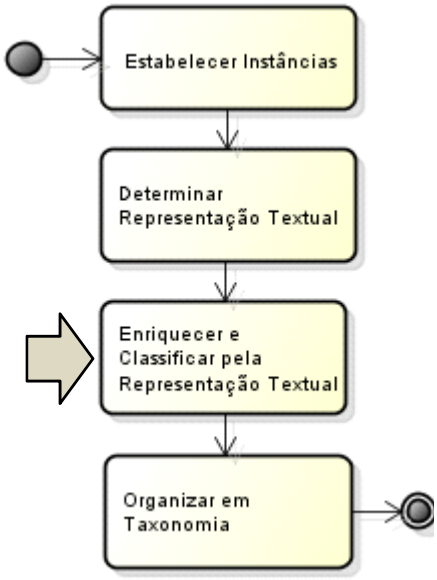


Figura 34 : Etapa 4.5

4.5.1 Objetivos desta etapa

Esta etapa propõe a desenvolver um processo que utilize os dicionários *online* da Língua Portuguesa para recuperar relações sobre determinado conjunto de termos. A seguir, na tabela 11, está uma lista dos dicionários *online* utilizados neste trabalho:

Tabela 11: Dicionários *online* utilizados para recuperar as relações entre termos.

Sites de Dicionários *Online*
http://www.tiosam.org/
http://www.priberam.pt/
http://michaelis.uol.com.br/
http://www.dicionarioweb.com.br/
http://dicionario-online.com/
http://bemfalar.com/

Como cada dicionário *online* tem um *layout* particular e o texto apresentado possui padrões textuais particulares, específicos de cada *site*, o processo de extração das relações se adaptou às características particulares de cada dicionário *online* utilizado e foi capaz de identificar a palavra que possui relação com cada termo pesquisado por meio de um agente gerado pelo aprendizado de máquina. A Figura 34 apresentada a seguir representa o processo nesta etapa.

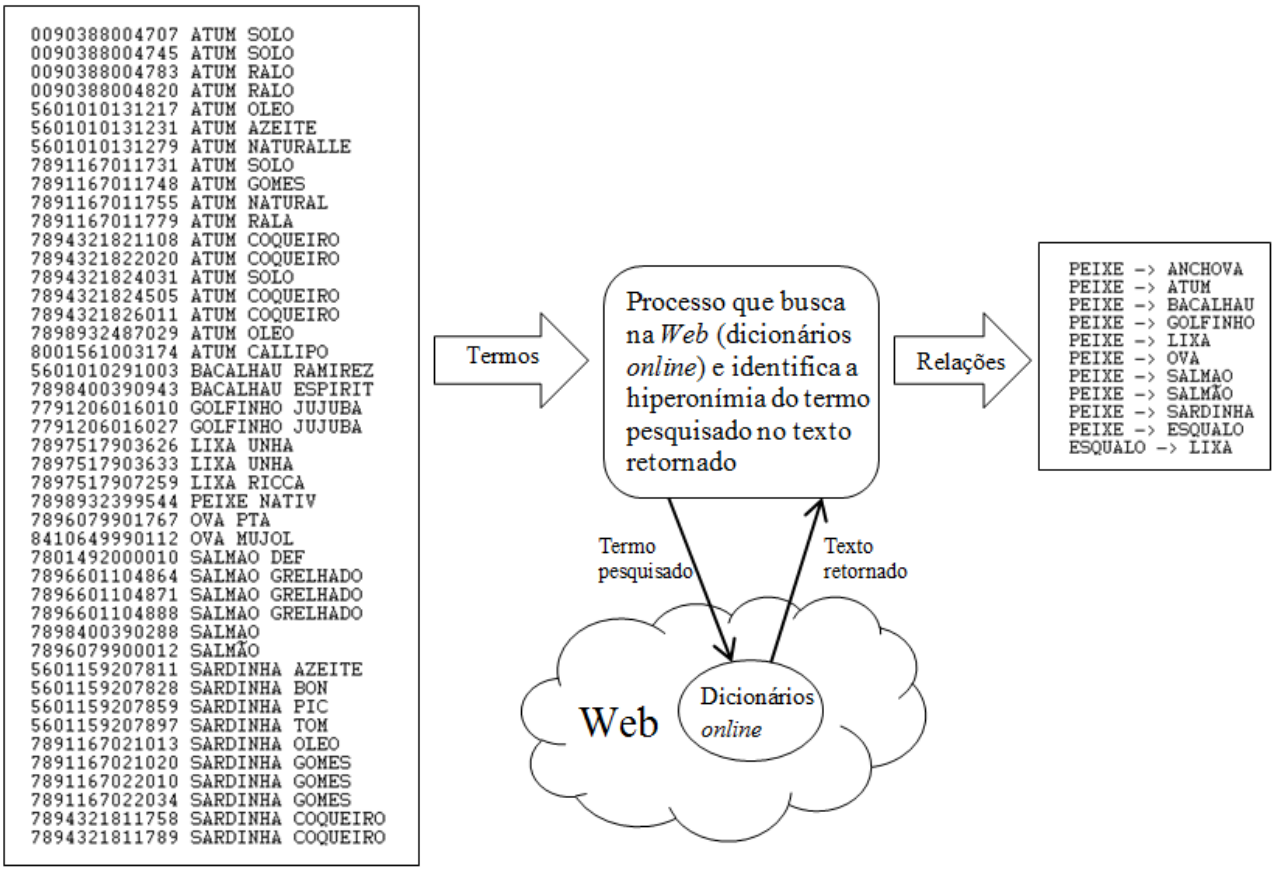


Figura 35: Processo de enriquecer e classificar pela descrição textual.

Conforme apresentado na Figura 35, o processo extrai do texto resultante da pesquisa do termo, retornado pelo dicionário *online,* a relação do termo pesquisado. Todos os dicionários *online* retornam um texto na linguagem *HTML*, a partir deste é extraído o texto descritivo do termo. Pode-se observar na Figura 36 que existem padrões textuais particulares em cada dicionário *online* no texto anterior à descrição do termo pesquisado.

Figura 36: Exemplo das características dos textos retornados pelos dicionários *online*.

O padrão textual apresentado na Figura 36 não está estruturado conforme a gramática da língua portuguesa, pois a apresentação do texto na *Web* normalmente é posicional, separada em campos e o texto destes campos nem sempre contém sentenças gramaticais completas. No caso dos dicionários *online*, além do texto ser apresentado em campos separados, também existem símbolos e abreviações particulares no texto retornado por cada dicionário *online*. Devido à existência destes padrões textuais particulares, o processo descrito a seguir utilizou aprendizado de máquina para identificar a relação desejada no texto retornado por cada dicionário *online*.

4.5.2 Busca das relações de classificação em dicionários *online*.

.

Para classificar as instâncias pela respectiva representação textual foram obtidas as relações de generalização (hiperonímia) dos termos da representação textual. Esta relação de generalização é buscada em dicionários *online* da *Web* por um processo que solicita as informações de uma determinada palavra (oriunda da representação textual da instância) e analisa o texto retornado pelo respectivo dicionário *online*, buscando neste texto determinados padrões de palavras que representem relações de generalização (hiperonímias) do termo pesquisado.

Conforme apresentado na Figura 37, os padrões textuais das relações hiponímias buscadas nos textos retornados por estes dicionários são gerados a partir de exemplos fornecidos por um especialista a partir do texto retornado por cada dicionário *online*, de forma a que este padrão textual seja coerente com as particularidades de cada fonte de informação particular (dicionário *online*). O especialista analisa o retorno da aplicação dos padrões na localização das relações e ajusta os exemplos de forma a maximizar os resultados. Quando os padrões alcançarem a capacidade de localização desejada, podem ser utilizados para localizar as relações hiperonímias dos termos das descrições textuais das instâncias.

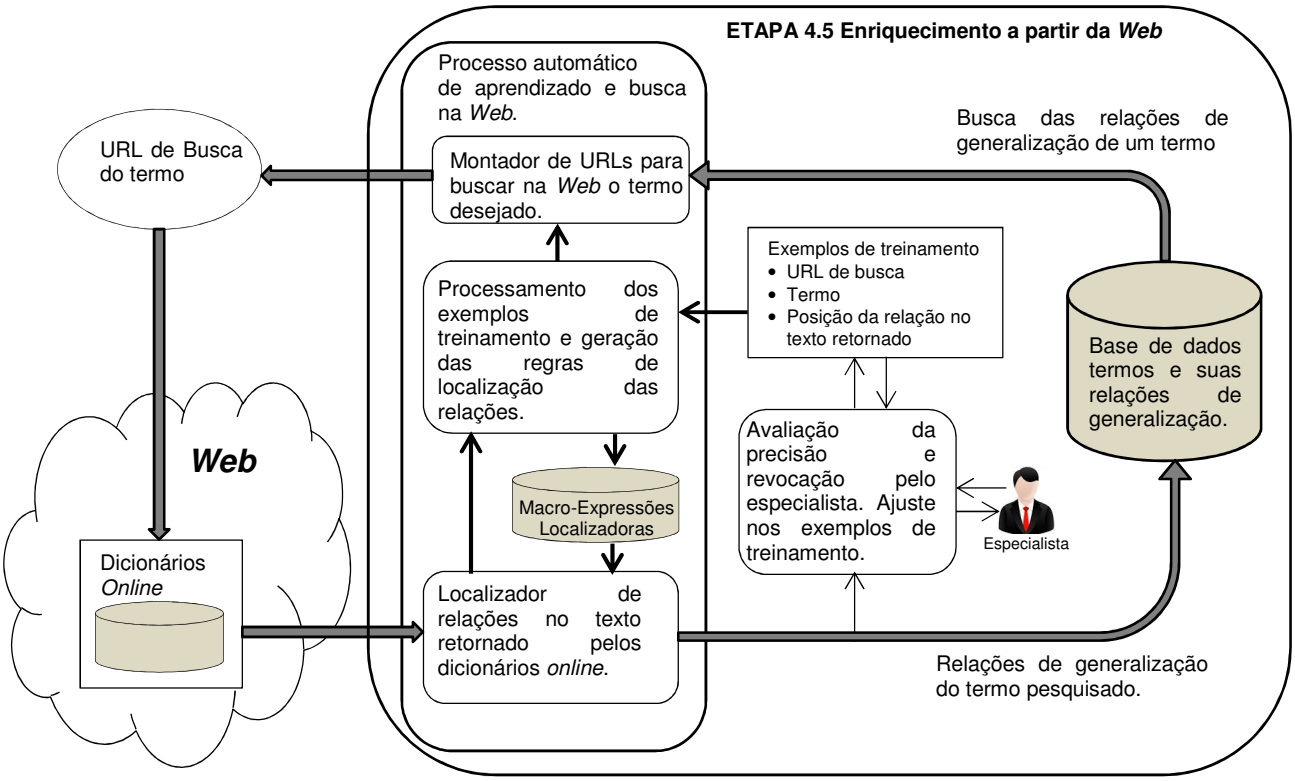


Figura 37: Ciclo do aprendizado de máquina para criação dos padrões de extração das informações na *WEB*.

Diversos dicionários *online* são acessados e cada um destes dicionários possui uma forma distinta de apresentar as informações sobre o termo pesquisado entre as diversas outras informações apresentadas na página *Web*. Diante desta variedade de informações, particulares de cada dicionário *online*, foi criado um processo que aprenda e se adapte à forma que cada dicionário apresenta estas informações para poder se localizar com sucesso o termo classificador (hiperonímia) do termo pesquisado.

Diversos pesquisadores como Navigli (NAVIGLI *et al* 2011), Balakrishna (BALAKRISHNA *et al* 2010), Dotsika (DOTSIKA 2009), Cantador (CANTADOR *et al* 2011) e outros utilizaram a busca por meio de padrões *part-of-speech* para localizar os termos das relações hiperonímias nos textos retornados da *Web*. Apesar de eficiente, este método exige um grande esforço na definição destes padrões *part-of-speech*, os quais são dependentes da linguagem na qual o texto foi escrito. A utilização do processamento de linguagem natural (PLN) produz bons resultados dependendo da ambiguidade (pouca) da linguagem. No caso da língua Portuguesa, existem palavras que possuem diferentes classificações gramaticais, sendo assim, o resultado da PLN pode apresentar diversas ambiguidades. No caso dos dicionários *online*, o texto apresentado nem sempre forma sentenças gramaticais completas para se processado pela PLN. Desta forma, a utilização da PLN não daria bons resultados na extração de informações do texto retornado pelos dicionários *online*.

Outro ponto a ser ressaltado é que a forma com que o texto é apresentado na *Web* depende do tipo de *site* na qual as informações estão sendo coletadas. Textos oriundos de jornais eletrônicos tem uma forma de escrita do texto diferente de *blogs* e cada tipo de site possui palavras como abreviações, rótulos e títulos posicionados em sequências e padrões particulares.

O principal motivo da utilização do aprendizado indutivo é a existência de padrões de escrita na definição do termo que nem sempre seguem as regras gramaticais formais. Anteriormente foi utilizado um processo de PLN com o algoritmo de *Earley* (JURAFSK, 2008) utilizando uma gramática para a língua Portuguesa adaptada de Othero (OTHERO, 2009), mas a excessiva quantidade de árvores sintáticas geradas criou um outro problema: qual das árvores deveria ser utilizada? A origem deste problema é a presença de ambiguidades na língua Portuguesa e pelo fato do texto descritivo retornado pelos dicionários *online* nem sempre estar escrito na gramática formal.

Ashwin Ittoo e Gosse Bouma (ITTOO, BOUMA e GOSSE 2013) apresentaram em 2013 um trabalho intitulado "*Minimally-supervised learning of domain-specific causal relations using an open-domain corpus as knowledge-base*" o qual utiliza PLN com aprendizado indutivo, na qual os

padrões não ambíguos gerados pela PLN são selecionados para se recuperar as relações "*part-whole*" na *Wikipédi*a.

A partir dos termos identificadores de cada produto, serão buscados os termos generalizadores destes em dicionários *online*. Como cada dicionário *online* retorna uma definição textual do termo pesquisado, o processo de aprendizado de máquina gerará padrões para localizar no texto retornado os padrões textuais que antecedem às descrições (que contém as hiperonímias) para se obter os termos generalizadores do termo pesquisado.

O processo de aprendizado de máquina é baseado em um agente (*Goal-based Agent*), apresentado na Figura 38, no qual o agente fará um ciclo interativo, partindo dos exemplos iniciais fornecidos pelo especialista, no qual serão gerados padrões iniciais, os quais serão generalizados (actions) e avaliados pelo agente, com o objetivo (*Goal*) de se obter o conjunto de padrões que melhor localize as informações desejadas no texto retornado por cada dicionário *online*. (RUSSEL e NORVIG, 2009).

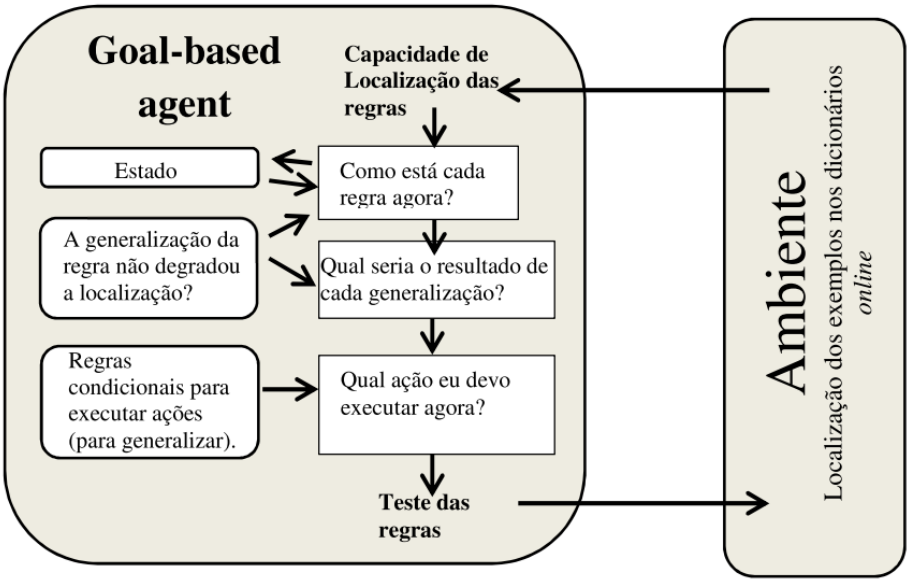


Figura 38: Agente baseado em objetivos destinado a refinar as macro-expressões (RUSSEL e NORVIG, 2009)

Desta forma, a busca das relações hiperonímias dos termos identificadores dos produtos foi feita por um agente baseado no aprendizado de máquina. O especialista em domínio deverá fornecer um conjunto de termos a ser utilizado como exemplos para consultas ao dicionário *online* e selecionar no texto retornado por estes o termo generalizador de cada termo consultado ou o termo imediatamente posterior à mensagem de ausência do termo pesquisado. Estes exemplos são classificados da seguinte forma:

Exemplos Positivos: São exemplos de treinamento de retornos bem sucedidos de consultas ao dicionário *online*, possuem a correta localização do termo pesquisado para que o processo analise o texto imediatamente anterior a este termo pesquisado e para definir um ou mais padrões de localização para o termo desejado.

Exemplos Negativos: São exemplos de treinamento de retornos malsucedidos de consultas ao dicionário *online*, estes possuem a localização de um termo posicionado imediatamente após uma mensagem informativa sobre a ausência de informações sobre o termo pesquisado na base de dados do respetivo dicionário *online*. O texto desta mensagem informativa será analisado para a definição de padrões que poderão identificar situações nas quais o dicionário *online* não possui tal informação.

Tanto os exemplos positivos quanto os exemplos negativos são analisados pelo processo de aprendizado de máquina, sendo progressivamente generalizados e testados sobre os exemplos, de forma a serem capazes de localizar as hiperonímias de outros termos, termos diferentes dos exemplos (no caso dos exemplos de positivos) ou situações nas quais o dicionário não possui informações sobre o termo pesquisado (no caso dos exemplos negativos). A geração destes padrões é feita por um processo que refina e avalia por meio de um ciclo evolutivo de aprendizado assistido por um especialista, que fará a análise dos casos não bem sucedidos de localização das relações hiperonímias, podendo criar ou modificar os exemplos enquanto o percentual de sucesso nas localizações não atingir um limite aceitável, conforme apresentado na Figura 39:

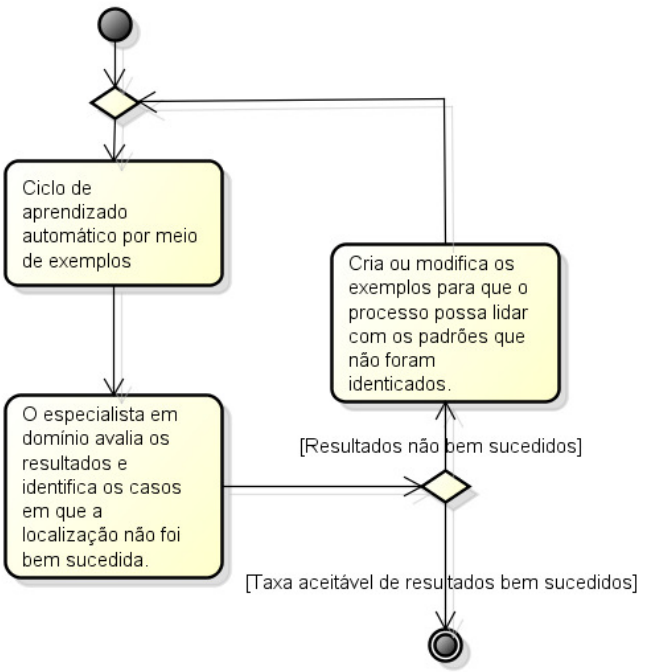


Figura 39: Ciclo de aprendizado de máquina assistido por especialista em domínio.

Cada padrão de localização é avaliado por dois fatores: Precisão e Revocação. Segundo MANNING, RAGHAVAN e SCHÜTZE (2008):

> ***Precisão (Precision**) é a fração de documentos relevantes (#documentos relevantes dividido por #documentos retornados) e **Revocação (Recall)** é a fração de #documentos relevantes retornados dividido por #documentos corretos.*
>
> ***Precisão (Precision)** é a fração de documentos corretos retornados (#documentos relevantes retornados dividido por #documentos retornados) e **Revocação (Recall)** é a fração de documentos relevantes retornados (#documentos relevantes retornados dividido por #documentos relevantes).*

No processo "Ciclo de aprendizado automático por meio de exemplos", apresentado a seguir na Figura 40, os padrões identificados no texto são generalizados até atingirem um ponto ideal de equilíbrio entre a precisão e a revocação (SHAW & GENTRY 1990). Durante cada passo desta generalização (remoção de termos à esquerda), o padrão em questão é re-testado sobre os exemplos de forma a validar a este passo de generalização degrada ou não a precisão, tendo por objetivo prevenir os casos de falsos positivos. Ao final, os padrões são ordenados de forma decrescente pela sua respectiva Precisão final.

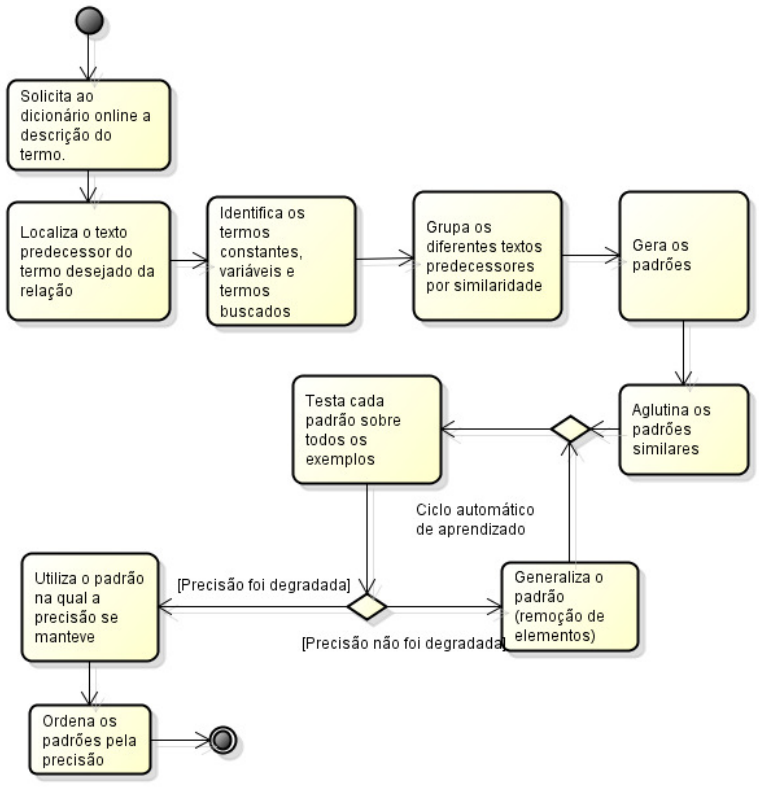


Figura 40: Ciclo de aprendizado automático baseado nos exemplos.

A precisão de cada padrão definirá a prioridade de utilização deste sobre o texto retornado dos dicionários *online*, pois serão utilizados prioritariamente os padrões com maior precisão, pois estes tem a menor possibilidade de apresentar falsos positivos. Caso o padrão em questão não seja localizado, o próximo padrão será utilizado e assim sucessivamente, até um resultado verdadeiro (positivo ou negativo). Caso a aplicação de todos os padrões não apresente nenhum resultado positivo, o resultado deste acesso será considerado verdadeiro negativo. Esta ordem de utilização dos padrões baseada na precisão tem por objetivo maximizar a precisão do processo como um todo.

Como o objetivo desta etapa é localizar a relação dentro do texto retornado pelo dicionário *online* será considerado válido qualquer termo (que seja substantivo ou nome) que atenda a uma das expressões regulares específicas para localizar a descrição de um determinado termo em determinado dicionário *online*. O texto retornado pelo dicionário será rejeitado se não atender a nenhum dos padrões de localização (positivos) definidos pelas expressões regulares específicas do termo buscado ou se uma regra de localização negativa for identificada.

Como cada expressão regular é específica para cada termo buscado no dicionário *online*, serão definidas duas variações de classes de elementos de expressões regulares para que a expressão formada por estes elementos possa ser utilizada na localização da descrição de qualquer termo em um dicionário *online* específico (ver Tabela 12). Pode-se considerar esta expressão como uma macro-expressão na qual será transformada em expressão regular no momento em que se definir o termo que está sendo consultado no dicionário *online*:

Tabela 12: Variações de classes de palavras para as macro-expressões:

Variação de classe	Exemplo	Função na macro-expressão
[:termo]	[:termo]	Será definida como o símbolo [:termo] pois a macro-expressão reconhecedora deverá ser independente do termo consultado.
Constantes	DUPLO CLIQUE EXPERIMENTE	Estas palavras serão consideradas constantes na macro-expressão.
[:outrapalavra]	[:outrapalavra](1,n)	As palavras não comuns serão consideradas como sequência de palavras (não constantes e diferentes do termo consultado). Cada sequência será associada a um número máximo de ocorrências (n).

Para se definir se uma palavra dos exemplos de treinamento seja considerada constante ou variável, as seguintes considerações são aplicadas:

- **Palavras repetidas**. São palavras que ocorrem pelo menos duas vezes no conjunto

de palavras dos exemplos de treinamento. Significa que estas palavras influenciam na similaridade entre os exemplos de treinamento.

- **Palavras não repetidas**. São palavras que ocorrem uma vez somente no conjunto de palavras dos exemplos de treinamento. Significa que são palavras que não influenciam na similaridade entre os exemplos de treinamento.

No caso dos exemplos positivos, ou seja, nos casos nos quais o dicionário *online* possui a definição do termo e retornou estas informações em forma de HTML para o agente, podem existir diversos padrões de sequências de palavras antecedentes ao termo desejado. Nestes padrões, o termo buscado é substituído pelo elemento [:termo], codificado por meio da palavra "$$", para que faça parte do conjunto de palavras repetidas, apesar de variar em função de cada exemplo de treinamento. A separação das palavras constantes das palavras variáveis é feita pela ordenação das palavras pela sua respectiva frequência nos exemplos de treinamento e separação das palavras de maior frequência até que o conjunto destas atinja 90% do somatório da frequência de todas as palavras nos exemplos de treinamento. Caso uma palavra não fizer parte deste conjunto, será considerada como uma palavra variável. Conforme o histograma de frequência das palavras dos exemplos positivos, apresentado na Figura 41:

$$Total\ Frequência\ Positiva\ = 0.9 * \sum Frequência\ Palavras\ Exemplos\ Positivos$$

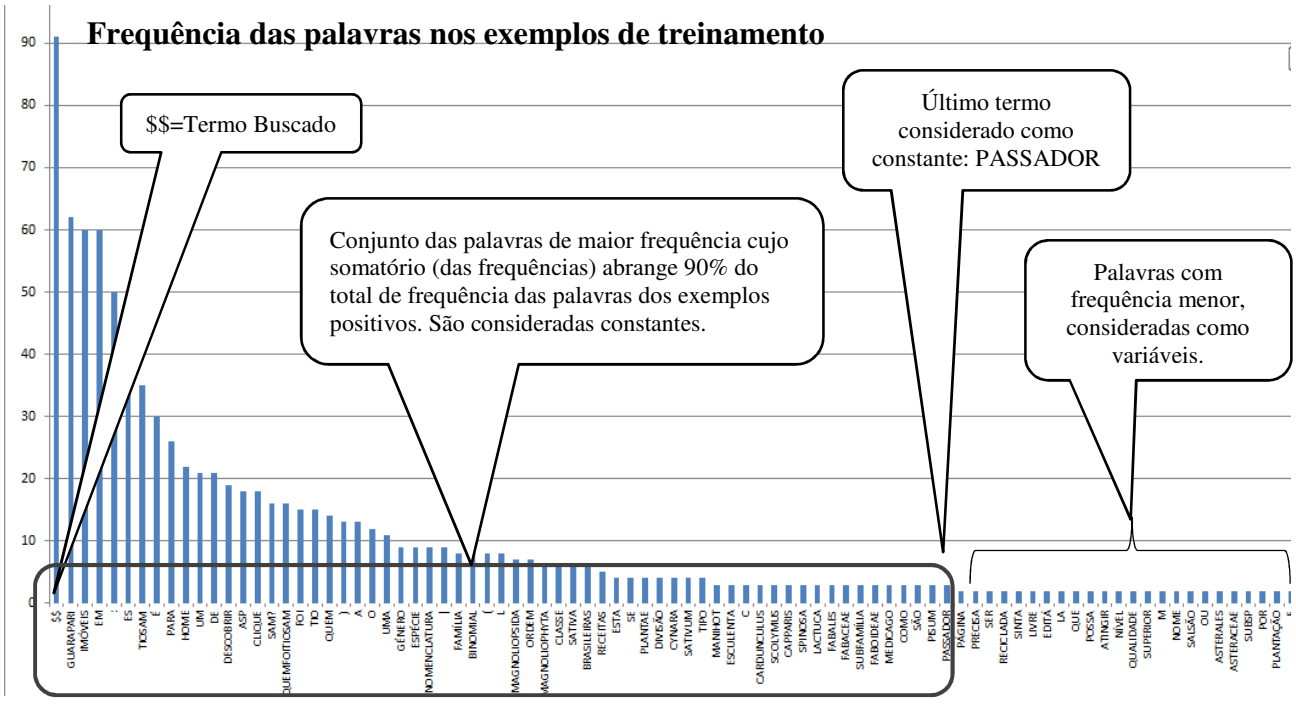


Figura 41: Histograma com a frequência das palavras dos exemplos positivos de treinamento.

No caso dos exemplos negativos, o mesmo processo é aplicado às palavras oriundas dos exemplos negativos de treinamento. É utilizado também o percentual de 90% para definir o conjunto de palavras constantes. **É importante observar que foram feitas experimentações com outros percentuais como 75% e 80%, mas o percentual de 90% foi o que gerou melhores padrões positivos e negativos para localizar as informações**.

O Algoritmo 02 a seguir descreve o processo de geração das macro-expressões iniciais (MRE) a partir da descrição (DS) do termo T no texto retornado por um dicionário:

Algoritmo 02: Cria as macro-expressões iniciais a partir dos exemplos.

Entrada:
T - Termo buscado
DS - Descrição do termo T buscado oriunda do dicionário *online*.
Saída:
MRE - Conjunto de macro-expressões localizadoras de descrições.

Variáveis:
PS - Sequência de k palavras anteriores à descrição do termo T. (Utilizou-se k = 32).
PSC - Agrupamento de Sequencias de PS pela quantidade de palavras constantes em comum.

01: **for** cada DS
02: novo PS com a sequência de n palavras anteriores da DS.
03: end for
04: **for** cada PS
05: substituir o respectivo termo T por "[:termo]" em cada PS
06: **end for**
07: **for** cada PS
08: novo PSC ← grupo de PS com palavras em comum.
09: **end for**
10:**for** cada PSC
11: Identificar as palavras comuns (constantes) em cada PSC
12: substituir as sequencias outras palavras (não constantes) por "[:outrapalavra]{0,n}" em cada PSC.
13: novo MRE ← PSC depois das substituições.
14: **end for**
15: aglutinar os MRE's com mesma sequencia elementos constantes e variáveis, escolhendo-se os termos de maior n para cada elemento "[:outrapalavra]{0,n}".
16: remover os termos do tipo "[:outrapalavra]{0,n}" que estejam no início de MRE (à esquerda).
17: descartar os MRE's vazios (sem termos)

As expressões geradas pelo pseudocódigo apresentado anteriormente possuem a precisão suficiente para se localizar os termos dos exemplos (conjunto *C*), mas provavelmente não possuem a revocação necessária para se localizar qualquer termo do dicionário *online* (conjunto *U*). Desta forma as expressões geradas deverão ser generalizadas para que possam ser aplicadas com sucesso a outros termos de *U* que não pertençam ao conjunto *C* (SHAW & GENTRY 1990).

Se abordarmos esta questão pelo método Hipotético-Dedutivo de Popper (POPPER, 1993): **Cada expressão gerada pode ser considerada uma hipótese, a qual terá sua capacidade de localização (precisão e revocação) avaliada sobre os exemplos fornecidos. A generalização da expressão (hipótese) executada pelo processo automático consiste é substituir a hipótese original por outra, caso esta nova hipótese apresente uma melhor capacidade de localização que a expressão (hipótese) anterior, conforme a Teoria de Aprendizado Computacional de Russel e Norvig. (RUSSEL e NORVIG, 2009).**

A generalização de cada expressão consistirá na remoção progressiva dos termos iniciais desta enquanto a precisão se mantiver, de forma a se obter uma expressão com o conjunto mínimo de termos que localize o termo com a precisão original. O pseudocódigo está descrito a seguir:

Algoritmo 03: Algoritmo que generaliza as macro-expressões obtidas dos exemplos.

Entrada:
MRE - Conjunto de macro-expressões localizadoras do conjunto de termos T.
Saída:
MRE - Conjunto de macro-expressões localizadoras generalizadas.

```
1:    procedure GENERALIZA_MACRO_EXPRESSOES
2:    for cada MRE
3:        PMRE      ← precisão de MRE sobre exemplos de T
4:        PMRE_Aux  ← PMRE
5:        CMRE      ← revocação de MRE sobre exemplos de T
6:        repeat enquanto PTMRE >= PMRE
7:            MRE_Aux   ← MRE com o primeiro termo (inicial) removido.
8:            PMRE_Aux  ← precisão de MRE_Aux sobre exemplos de T
9:            CMRE      ← revocação de MRE_Aux sobre exemplos de T
10:           if PMRE_Aux = PMRE then
11:               MRE   ← MRE_Aux
12:           end if
13:       end repeat
14:       if (PMRE = 0) OR (MRE is VAZIO)  then
15:           descartar MRE
16:       end if
17:   end for
```

Cada dicionário *online* será associado a um determinado conjunto de macro-expressões (MREs geradas pelo algoritmo 03), as quais serão utilizadas para se localizar as relações hiperonímias do termo pesquisado no texto retornado pelo respectivo dicionário *online*. Ressaltamos que as macro-expressões foram geradas a partir de exemplos de hiperonímias, sendo assim, especializadas na busca deste tipo de relação. **Caso forem fornecidos outros tipos de exemplo, como relações de sinônimos, as macro-expressões oriundas destes exemplos seriam**

especializadas na busca de relações de sinônimos. Esta flexibilidade das macro-expressoes pode ser aplicada a sites de outras línguas que utilizem a mesma estrutura de sequência de palavras.

Ao se pesquisar determinados termos nos dicionários *online*, os textos são retornados em formato HTML, sendo necessária a execução de um processo de *parsing* para remover as *tags* e extrair o texto puro de cada *site*. Determinados textos são retornados com a acentuação em *tags* HTML e também com outros tipos de *charsets* diferentes do padrão ASCII. Um processo faz a conversão destes caracteres sobre o texto retornado para que o texto resultante a ser processado pelas macro-expressões seja padrão ASCII.

Um ponto importante deste processo é verificação se o termo da relação localizado pelas macro-expressões no texto retornado pelos dicionários *online* existe no *Thesauro*. O termo localizado que não existir no *Thesauro* ou não possuir classificação gramatical de substantivo ou pertencer a *ListaPalavrasIgnoradas* será descartado, sendo verificado o próximo termo do texto até um limite especificado de avanço no texto. A aplicação desta condição é feita sobre os cinco primeiros termos, caso nenhum termo atenda a esta condição o resultado da consulta deste termo deste dicionário *online* é descartado. A aplicação desta restrição impede que sejam localizados erroneamente como termos classificadores palavras como preposições, verbos ou abreviaturas que existem no texto retornado pelos dicionários *online*. A Tabela 13 apresenta um exemplo de aplicação da *ListaPalavrasIgnoradas.* Pode-se observar que diversas palavras como “”ESPECIE”, “DE”, “SE”, “NOME”, “COMUM”, “VARIEDADE” são ignoradas na descrição para se localizar a hiperonímia. Caso se estivesse utilizando o PLN uma parte destas palavras seriam consideradas termos de determinadas regras gramaticais.

Tabela 13: Exemplo de utilização da *ListaPalavrasIgnoradas* na localização das hiperonímias.

Termo buscado	**Trecho do texto retornado pelo http://michaelis.uol.com.br** Sublinhado = Descrição, Negrito = **Palavras Ignoradas,** Itálico = *Hiperonímia localizada* [:termo] = Termo buscado
BACALHAU	TWEETAR [:termo] BA CA LHAU SM (FR CABILLAUD COM METÁTESE) 1 **ICTIOL** *PEIXE*
MARSHMALLOW	MARSH MAL LOW (MARCHIMÉLOU) SM (INGL) **ESPÉCIE DE** *DOCE*
MOSCATEL	CA TEL ADJ SF (CAT MOSCATELL) **DIZ SE DA OU A VARIEDADE DE** *UVA*
SABÃO	ESTA PÁGINA: TWEETAR [:termo] 1 SA BÃO 1 SM (LAT SAPONE) 1 *SUBSTÂNCIA*
LENTILHA	PÁGINA: TWEETAR [:termo] LEN TI LHA SF (LAT LENTICULA) 1 **BOT PEQUENA** *ERVA*
LICOR	COMPARTILHE ESTA PÁGINA: TWEETAR [:termo] LI COR SM (LAT LIQUORE) 1 *BEBIDA*
GORGONZOLA	COMPARTILHE ESTA PÁGINA: TWEETAR [:termo] GOR GON ZO LA SM (ITAL [:termo]) *QUEIJO*
ESPAGUETE	PÁGINA: TWEETAR [:termo] ES PA GUE TE SM (ITAL SPAGHETTI) **ESPÉCIE DE** *MACARRÃO*
COLHER	TWEETAR [:termo] 1 CO LHER 1 (É) SF (LAT COCHLEARE) 1 *UTENSÍLIO*
CERVEJA	COMPARTILHE ESTA PÁGINA: TWEETAR [:termo] CER VE JA SF (LAT CEREVISIA) *BEBIDA*
CENOURA	ESTA PÁGINA: TWEETAR [:termo] CE NOU RA SF (ÁR SAFUNÂRIYA) **BOT** *PLANTA*
CIDREIRA	COMPARTILHE ESTA PÁGINA: TWEETAR [:termo] CI DREI RA SF (CIDRA EIRA) **BOT** *ÁRVORE*
COCADA	COMPARTILHE ESTA PÁGINA: TWEETAR [:termo] CO CA DA SF (COCO ADA 1) 1 *DOCE*
BAUNILHA	BAU NI LHA SF (CAST VAINILLA) 1 **BOT NOME COMUM A VÁRIAS** *TREPADEIRAS*

O Algoritmo 04 utiliza as macro-expressões para localizar o início da descrição do termo pesquisado no texto retornado pelo respectivo dicionário *online*. Após a localização da descrição, as palavras desta são verificadas a partir do inicio, na busca por um substantivo que não pertença a *ListaPalavrasIgnoradas*.

Algoritmo 04: Rotina que localiza o termo no texto por meio de uma macro-expressão.

Entrada:
Termo – Termo a ser pesquisado
DicOn – Conjunto de Dicionários *online* utilizados.
MRE - Macro-expressão que localiza relações no texto
Saída:
Termo.Classificadores -.Lista de termos generalizadores do termo pesquisado

```
1:      Procedure BuscaRelacaoWeb (Termo)
2:      Termo.Classificadores ← 0
3:      for cada DicOn
4:         TextoHTLM ← AcessaWeb (DicOn.URL + Termo)
5:         TextoDic ← ParseHTMLCharset(TextoHTML)
6:         DicOn.MRE.First
```

```
7:          pare ← false
8:          repeat
9:           for cada MRE de DicOn
10:             Palavra ← ParseMacroExpressao (DicOn.MRE, TextoDic)
11:             if DicOn.MRE.Positiva
12:               if  NOT Palavra.Vazia AND ThesauroSubstantivo(Palavra)  AND NOT ListaPalavraIgnoradas
13:                  Termo.Classificadores  ← Termo.Classificadores + Palavra
14:               else
15:                  DicOn.MRE.Next
16:               end if
17:             else if
18:               if DicOn.MRE.Negativa AND NOT Palavra.Vazia
19:                  pare   ← True
20:               end if
21:             end if
22:          until pare OR DicOn.MRE.Fim
23:       end for
```

A seguir, apresentado na Tabela 14, estão as macro-expressões geradas pelo processo correspondentes ao dicionário *online* www.tiosam.org. As macro-expressões são processadas na ordem definida pela coluna *Seq*. As macro-expressões podem ser de dois tipos:

- **VP-Verdadeiro Positivo**, que indica a localização do termo relacionado com o termo pesquisado;
- **VF-Verdadeiro Negativo**, que indica a localização de uma expressão informativa que o dicionário *online* não possui a informação sobre o termo pesquisado.

Tabela 14: Macro-expressões para acesso ao dicionário *online* www.tiosam.org

URL de acesso ao dicionário: http://www.tiosam.org/enciclopedia/index.asp?q=<termo>		
Seq	**Tipo**	**Macro-expressão**
1	VP	É UM TIPO DE
2	VN	PÁGINA DE DESAMBIGUAÇÃO
3	VP	UMA VARIEDADE DE
4	VP	([:outrapalavra]{0,2}) É UMA
5	VN	[:termo] [:outrapalavra]{0,3} NÃO [:outrapalavra]{0,2} ARTIGO [:outrapalavra]{0,3}
6	VP	SATIVA) É UMA
7	VP	[:termo] [:termo] [:termo] [:outrapalavra]{0,1} [:termo] É UMA
8	VP	UM [:outrapalavra]{0,2} DE [:termo] [:termo] É UM
9	VP	DE [:termo] O [:termo] É UMA
10	VP	DE UM [:outrapalavra]{0,4} [:termo] É [:outrapalavra]{0,1}
11	VP	[:termo] DE [:outrapalavra]{0,1} [:termo] É UM
12	VP	) [:outrapalavra]{0,2} O [:termo] É O
13	VN	[:termo] [:outrapalavra]{0,1} [:termo] [:outrapalavra]{0,2} ARTIGO :outrapalavra]{0,3}
14	VN	PARA [:outrapalavra]{0,2} DE [:outrapalavra]{0,5} ARTIGO [:outrapalavra]{0,3}
15	VP	DE [:outrapalavra]{0,2} [:termo] É UMA
16	VP	[:termo] [:termo] [:termo] É UM
17	VP	É A [:outrapalavra]{0,1} COMUM [:outrapalavra]{0,1}
18	VP	SCOLYMUS) É UMA
19	VP	SATIVUM) É UMA
20	VP	A [:termo] É A
21	VP	O [:termo] É UM
22	VN	FOI [:outrapalavra]{0,1} PARA [:outrapalavra]{0,1}
23	VP	) É UM
24	VP	COMUM DADO [:outrapalavra]{0,1}
25	VP	PASSADOR É UM
26	VP	GÉNERO DE
27	VP	[:termo] [:termo] A [:termo] É UM
28	VN	NÃO ENCONTRADO

4.5.3 Conclusões desta etapa

A utilização dos dicionários *online* como fontes de informação permite o enriquecimento de dados que estejam incompletos ou exista a necessidade de mais informações sobre os dados já disponíveis.

A geração de padrões de localizações a serem aplicados no texto retornado pelos dicionários *online* pelo processo de aprendizado de máquina se apresenta como uma alternativa promissora (em relação ao PLN) na área de extração de informações sobre bases textuais, mesmo que o texto destas bases textuais não esteja totalmente coerente com a respectiva gramática da língua utilizada no texto.

4.6 Etapa: Organizar em Taxonomia

O objetivo desta etapa é utilizar as relações dos termos oriundos das descrições dos produtos para construir uma taxonomia que permita classificar os produtos (instâncias).

Com a utilização das relações recuperadas pelo processo descrito na etapa anterior (4.5) será montado um dígrafo de classificação dos termos representativos de produto, os quais foram obtidos na segunda etapa. Sobre o dígrafo gerado, serão aplicadas operações de refinamento que removerão as suas ambiguidades estruturais, de forma que o dígrafo seja transformado em uma taxonomia coerente que represente a classificação dos produtos.

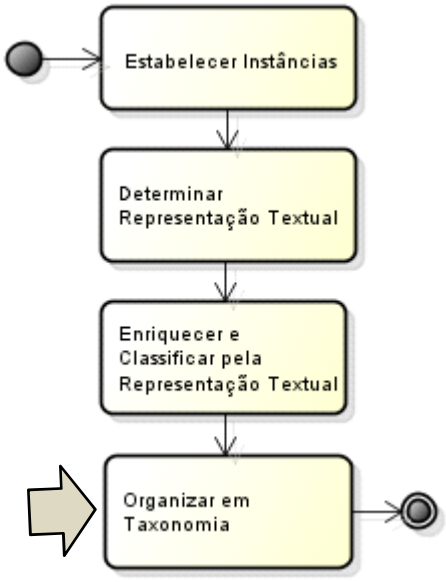


Figura 42 Etapa 4.6

Desta forma, esta etapa tem por objetivo propor um processo que construa uma taxonomia coerente e representativa capaz de classificar os produtos em questão. Neste trabalho, são apresentados algoritmos e técnicas capazes de:

- Gerar com sucesso um dígrafo a partir de informações iniciais enriquecidas com relações recuperadas em dicionários *online*.
- Remover ambiguidades e inconsistências deste dígrafo, de forma a transformá-lo em uma taxonomia classificadora dos produtos.

Concluindo, esta etapa apresenta um conjunto de processos capaz de gerar uma taxonomia classificadora coerente e sem ambiguidades a partir de informações descritivas de cada instância enriquecidas com informações recuperadas de dicionários *online*.

4.6.1 Passos da construção da taxonomia a partir dos termos classificadores

Conforme apresentado na Figura 43, as relações obtidas na etapa anterior foram utilizadas para construir um grafo direcionado, na qual cada termo pode estar relacionado a nenhum ou diversos termos generalizadores (relações hiperonímias). Uma ou mais instâncias serão associadas a elementos do grafo (termos), conforme os termos presentes nos respectivos termos representativos de cada instância.

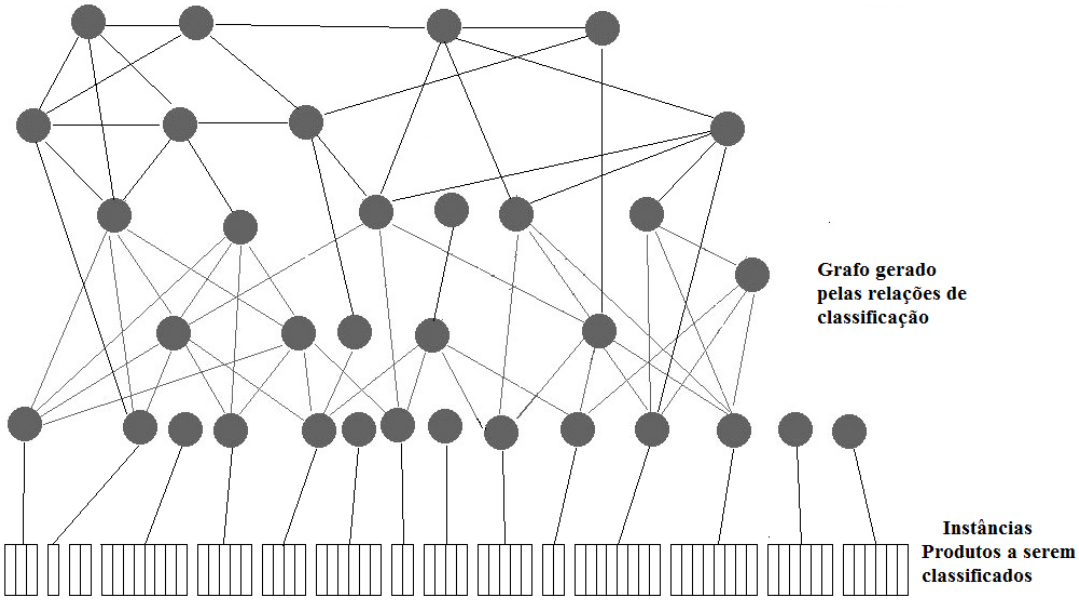


Figura 43: Subgrafo gerado a partir das relações de classificação dos termos das instâncias.

Apesar das relações hiperonímias identificadas na etapa anterior serem consideradas válidas, somente será utilizada uma relação hiperonímia por termo, pois cada termo terá somente uma generalização. Esta etapa tratará a ambiguidade destas relações para criar a taxonomia das instâncias.

A construção da taxonomia deverá tratar diversas ambiguidades (NAVIGLI e VELARD 2004) e problemas executado na etapa anterior oriundos do enriquecimento . A seguir está a lista de problemas a serem resolvidos (PLANGPRASOPCHOK *et al* 2010),:

- **Atalhos.** São relações redundantes entre termos de níveis não adjacentes.
- **Ciclos.** São formados por uma sequência de relações que retornam ao ponto inicial.
- **Relacionamento Recíprocos.** Quando dois termos possuem dois relacionamentos entre si, cada termo é classificador em um relacionamento e classificado no outro relacionamento.
- **Ruído nos dados oriundos da *Web*.** São informações errôneas de relações classificadoras recuperadas na *Web* (TANG *et al* 2009), (PLANGPRASOPCHOK *et al* 2010), oriundas de textos obtidos dos dicionários *online* que atendem a algum padrão das macro-expressões utilizadas, mas não contém o termo classificador na posição esperada.

Desta forma a construção da taxonomia foi feita em dois passos:

1. **Geração do grafo (dígrafo) formado pelos termos classificadores** dos produtos e seus respectivos termos generalizadores, oriundos a folksonomia. Este processo é descrido adiante, na seção 4.6.1.
2. **Transformação do grafo (dígrafo) em uma estrutura hierárquica**, com a resolução dos problemas apresentados anteriormente e com a seleção de um único termo generalizador para cada termo identificador de produto. Este processo está descrito adiante na seção 4.6.2.

4.6.2 Geração do grafo (dígrafo) formado pelos termos classificadores

A geração do grafo será feita pelo processo de construção *bottom-up* (USCHOLD e GRUNINGER 1996), (DOTSIKA, 2009). Os níveis serão construídos de baixo para cima, começando com os termos oriundos dos produtos. A geração do grafo é processada em etapas, um nível de cada vez, descritas a seguir:

4.6.3 Geração do Nível 1 do grafo a partir do primeiro termo de cada produto.

O conjunto de termos formado pelo primeiro termo de cada produto será utilizado para a construção do nível 1 do grafo, conforme descrito no Algoritmo 05:

Algoritmo 05: Geração do Nível 1 do dígrafo.

```
1:    Procedure GeraNivel1Grafo
2:      Termos  Nível 1← 0.
3       Relações ← 0.
4:      for cada produto
5:        if produto possuir primeiro termo representativo
6:              Termos Nivel 1 ← Termos Nivel 1  + Primeiro termo representativo de produto
7:        end if
8:      end for.
9:    end procedure
```

A aplicação do filtro do *Thesauro* nos dois primeiros termos representativos de cada produto (etapa 4.4.3) proporciona uma melhor qualidade nos termos do nível 1, os quais serão a base do

dígrado que está sendo construído e este evoluirá para a hierarquia de produto.

Tabela 15: Exemplo da construção do nível 1 a partir do primeiro termo de cada produto (N/E=Não Encontrado).

Nivel 1 do Digrafo Composto do primeiro termo representativo de cada produto	TOMATE	TOMATE	TOMATE	VAGEM	VAGEM	VAGEM	<Não Encontrado>	CEBOLA	BATATA	BATATA	BATA	CEBOLA	CEBOLA	CEBOLA	PANQUECA		ESTRAGA	BERINGELA	SUFLA	<Não Encontrado>	<Não Encontrado>	FRAMBOESA
Termos desabreviados selecionados como classificadores do produto	TOMATE CEREJA	TOMATE SALADA ROCAMBOLE	TOMATE CAQUI ROCAMBOLE	VAGEM MACA ROCAMBOLE	VAGEM MANTEIGA ROCAMBOLE	VAGEM FRANCESA ROCAMBOLE	COGATE ERYNGUI DROCA	CEBOLA PACOTE	BATATA ASTERIX	BATATA EMPACOTADO	BATA PACOTE	CEBOLA PIKCLES	CEBOLA ROXAS IMPOR	CEBOLA BRANCO IMPOR	PANQUECA SOJA	LASB SOJA	ESTRAGA SOJA	BERINGELA SOJA	SUFLA SOJA	EMPD SOJA	BLUEBERRIE	FRAMBOESA DOURO
Código do produto EAN (Fabricante-Produto)	78989344-78926	78989344-78933	78989344-78940	78989344-78957	78989344-78964	78989344-78971	78989344-79213	78989352-61053	78989352-61091	78989352-61114	78989352-61121	78989352-61138	78989352-61183	78989352-61190	78989358-27013	78989358-27020	78989358-27037	78989358-27051	78989358-27068	78989358-27075	78989358-86010	78989358-86027

Pode-se observar na Tabela 15 que determinados produtos não possuem termos representativos. A causa disso é o critério de restrição dos termos representativos do produto definido em 4.4.4, o qual não permite termos representativos abaixo de determinadas frequência e quantidade mínimas. A decisão de utilizar este critério motiva-se pelo fato de se utilizar somente termos realmente representativos para a construção da taxonomia de produtos.

4.6.4 Geração do Nível 2 do grafo a partir da recuperação do termo classificador na *Web*

Com o nível 1 definido, é feita a construção do nível 2 do grafo a partir dos termos classificadores do nível 1, pelo processo descrito na etapa 4.5.2, na qual os dicionários *online* são acessados e as macro-expressões localizam as relações dos termos pesquisados.

Tabela 16: Exemplo da construção do nível 2 a partir da busca dos classificadores nos dicionários *online*.

Nivel 2 do Digrafo Composto dos termos classificadores do nivel 1, recuperados dos dicionários *online*	FRUTO	FRUTO	FRUTO	<Não Encontrado>	<Não Encontrado>	<Não Encontrado>	<Não Encontrado>	<Não Encontrado>	TUBÉRCULO	TUBÉRCULO	<Não Encontrado>	<Não Encontrado>	<Não Encontrado>	<Não Encontrado>	MASSA RODELA	<Não Encontrado>	<Não Encontrado>	PLANTA	<Não Encontrado>	<Não Encontrado>	<Não Encontrado>	FRUTO
Nivel 1 do Digrafo Composto do primeiro termo representativo de cada produto	TOMATE	TOMATE	TOMATE	VAGEM	VAGEM	VAGEM	<Não Encontrado>	CEBOLA	BATATA	BATATA	BATA	CEBOLA	CEBOLA	CEBOLA	PANQUECA		ESTRAGA	BERINGELA	SUFLA	<Não Encontrado>	<Não Encontrado>	FRAMBOESA
Termos desabreviados selecionados como classificadores do produto	TOMATE CEREJA	TOMATE SALADA ROCAMBOLE	TOMATE CAQUI ROCAMBOLE	VAGEM MACA ROCAMBOLE	VAGEM MANTEIGA ROCAMBOLE	VAGEM FRANCESA ROCAMBOLE	COGATE ERYNGUI DROCA	CEBOLA PACOTE	BATATA ASTERIX	BATATA EMPACOTADO	BATA PACOTE	CEBOLA PIKCLES	CEBOLA ROXAS IMPOR	CEBOLA BRANCO IMPOR	PANQUECA SOJA	LASB SOJA	ESTRAGA SOJA	BERINGELA SOJA	SUFLA SOJA	EMPD SOJA	BLUEBERRIE	FRAMBOESA DOURO
Código do produto EAN (Fabricante-Produto)	78989344-78926	78989344-78933	78989344-78940	78989344-78957	78989344-78964	78989344-78971	78989344-79213	78989352-61053	78989352-61091	78989352-61114	78989352-61121	78989352-61138	78989352-61183	78989352-61190	78989358-27013	78989358-27020	78989358-27037	78989358-27051	78989358-27068	78989358-27075	78989358-86010	78989358-86027

Pode-se observar nos exemplos da Tabela 16, o produto de EAN 7898935827013, definido pelos termos “PANQUECA SOJA” associou-se ao termo do nível 1 “PANQUECA” e este termo foi associado a dois termos no nível 2: “MASSA” e “RODELA”. Esta ambiguidade será resolvida na etapa 4.6.4, durante a transformação do dígrafo em uma estrutura hierárquica. Os termos “COGATE”, “EMPD” e “BLUEBERRIE”, que nem existem no *Thesauro*, também não tiveram a respectiva hiperonímia localizada, provavelmente uma macro-expressão negativa informou que o respectivo dicionário *online* consultado não possui tal informação. Já os termos “VAGEM” e “CEBOLA não tiveram a respectiva hiperonímia localizada no texto retornado pelos dicionários *online*, apesar de existirem no *Thesauro*. Neste caso (Falso Negativo) pode-se analisar a necessidade de se adicionar mais alguns exemplos para gerar um conjunto mais abrangente de macro-expressões capaz de localizar estes termos.

Os termos classificadores recuperados dos dicionários *online* são considerados do nível 2.

Caso o respectivo termo classificador já exista no nível 1, será ajustado para pertencer ao nível 2 (promovido) ao se identificar uma relação de classificação sobre um termo do nível 1. Um caso típico é o termo MASSA, que é o primeiro termo representativo de diversos produtos (exemplo: EANs 7730430000143 e 773043000015), mas ao se verificar a relação de classificação MASSA → MACARRÃO, o termo MASSA é ajustado para pertencer ao nível 2. Este ajuste ocorre quando o produto possui termos representativos muito genéricos como, por exemplo: MASSA, ALIMENTO, BEBIDA, etc.

4.6.5 Geração do Nível 3 do grafo a partir da recuperação do termo classificador da *Web*

Com os termos do nível 2 definidos, é feita a construção do nível 3 do grafo a partir dos termos classificadores do nível 2, pelo processo descrito na etapa 4.5.2, na qual os dicionários *online* são acessados e as macro-expressões localizam os termos desejados. A Figura 44 apresentada a seguir mostra um subgrafo gerado com o termo FARINHA no nível 3:

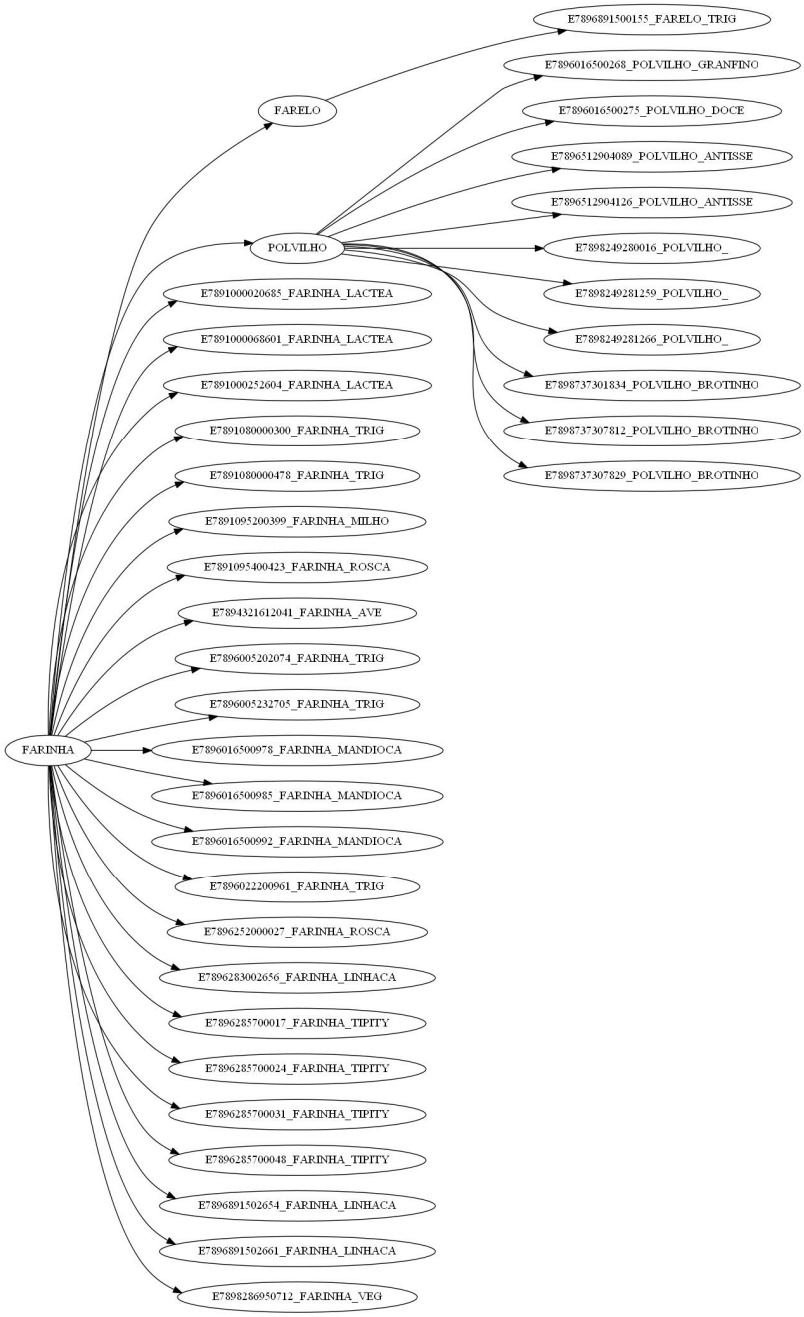


Figura 44: Subgrafo com termos no nível 3.

4.6.6 Promoção dos termos que são classificadores de termos em níveis superiores.

Apesar da busca nos dicionários *online* dos termos classificadores estar direcionada para apenas três níveis hierárquicos acima, determinados termos em um nível podem ser classificadores de termos localizados em níveis superiores e também podem existir ciclos no dígrafo, nos quais um termo é classificador de outro termo e este (ou algum termo classificador deste, sucessivamente) pode classificar o termo original. O Algoritmo 06 promove os termos com estas características, alterando o nível de determinados termos cujas relações de classificação são sobre termos de níveis iguais ou superiores ao nível do termo verificado. Pode-se observar que o algoritmo utiliza um nível limite máximo arbitrário neste processo de promoção de termos. A razão deste limite máximo de nível (neste caso o nível limite máximo é cinco) é devido à existência de ciclos, nos quais os termos seriam promovidos infinitamente.

Algoritmo 06: Ajuste nos níveis dos termos que são classificadores de termos em níveis superiores

```
1:   Procedure PromoveTermoClassificadores
2:     NivelMaximo ← 5
3:     Repeat
4:       Pare ← True
5:       for cada Relacionamento em Tabela_Relacionamentos
6:         if Relacionamento.Classificador.Nivel <= Relacionamento.Classificado.Nivel then
7:           if Relacionamento.Classificado.Nivel < NivelMaximo then
8:             Relacionamento.Classificador.Nivel ← Relacionamento.Classificado.Nivel + 1
9:             Pare ← False
10:          end if
11:        end if
12:      end for
13:    until Pare
14:  end procedure
```

4.6.7 Características do dígrafo gerado

O dígrafo gerado pelo processo de construção *bottom-up* já tem os termos posicionados em determinados níveis, conforme as relações de classificação obtidas pela etapa 4.5.1. A Figura 45 apresenta uma representação do dígrafo com seus respectivos níveis e das relações de classificação:

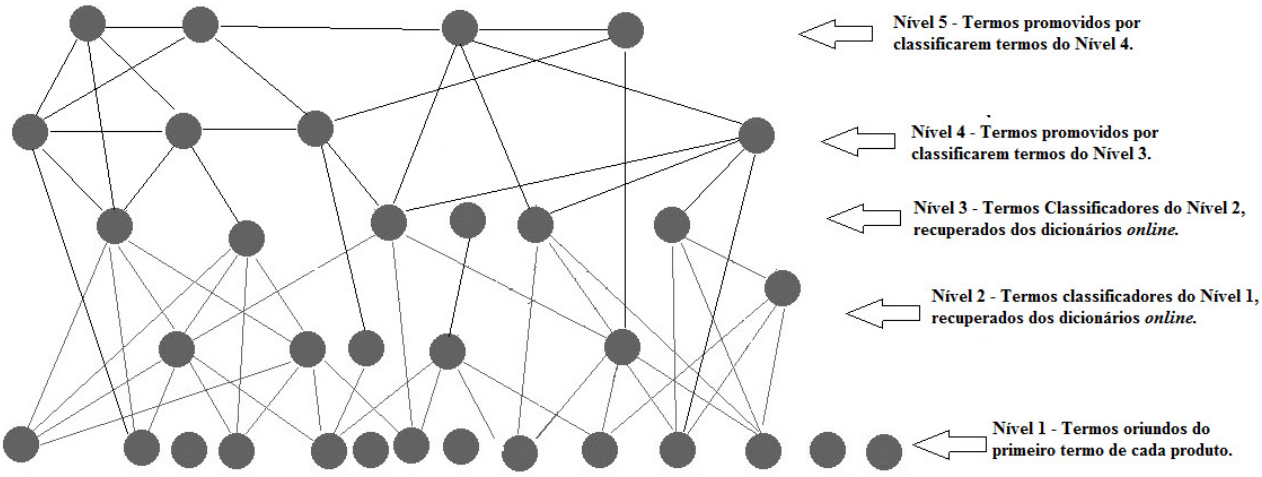


Figura 45: Dígrado gerado a partir dos termos de produto e seus termos classificadores.

Na representação do grafo apresentada na Figura 45 observa-se as seguintes inconsistências:

- **Determinados termos possuem mais de um termo generalizador**, mas as regras de consistência da taxonomia somente permitirão um termo generalizador para cada termo generalizado.
- **Relacionamentos Recíprocos**. Determinados pares de termos possuem relacionamentos entre si nos quais cada termo é generalizador em um relacionamento e generalizado no outro relacionamento. Será descartado o relacionamento no qual o respectivo termo generalizador deste possuir a menor quantidade de relacionamentos nos quais este termo é generalizador.
- **Determinados termos não possuem termo generalizador**. Este problema é oriundo da ausência de informações nos dicionários *online* ou da presença de padrões textuais não previstos nos exemplos de treinamento para a geração das macro-expressões descritos na etapa 4.5.2.
- **Existência de ciclos**. Podem existir relações circulares de generalização.

.

4.6.8 Transformação do grafo (dígrafo) em uma estrutura hierárquica

O dígrafo construído na etapa 4.6.2 já possui uma distribuição dos seus elementos por níveis, apesar desta distribuição não ser definitiva, esta distribuição torna mais fácil a resolução de ambiguidades. De foma semelhante aos métodos utilizados por Tomuro (TOMURO e SHEPITSEN, 2009) e Balakrishna (BALAKRISHNA *et al* 2010), as ambiguidades anteriormente listadas serão

resolvidas das seguintes formas:

- **Ajuste nos relacionamentos recíprocos.** Serão identificados os termos que possuem relacionamentos recíprocos, sendo descartado o relacionamento no qual o termo generalizador possuir menor quantidade de relações de generalização em relação ao outro relacionamento.
- **Remoção dos atalhos:** Será verificada a existência de atalhos, os quais seriam relações diretas de um termo para outro, sendo que este outro termo possui uma relação indireta com o primeiro termo. A proposta de solução deste problema está descrita no algoritmo 08, o qual percorre o dígrafo a partir dos níveis superiores, descartando as relações diretas (atalhos) que possuam relações indiretas.
- **Seleção de um termo classificador entre vários termos classificadores.** Após a remoção dos atalhos, será utilizada uma métrica que avalia cada um dos termos classificadores sobre um mesmo termo classificado, de forma que se selecione o termo classificador que melhor se adequa à taxonomia que está sendo construída. A métrica utilizada para selecionar o melhor termo classificador está descrita na seção **4.6.2.1**.

4.6.9 Remoção das ambiguidades de generalização de cada termo do dígrafo.

Para a resolução das ambiguidades descritas anteriormente, foram utilizados algoritmos descritos a seguir:

O Algoritmo 07 faz a remoção dos relacionamentos recíprocos. Os pares de termos com relacionamentos recíprocos são identificados e são descartados os relacionamentos recíprocos cujos termos generalizadores tiverem a menor quantidade de relacionamentos nos quais estes termos são generalizadores.

Algoritmo 07: Remoção dos relacionamentos recíprocos

Entrada:
Tab_Termos : Tabela de Termos
Tab_relacionamentos: Tabela de Relacionamentos entre os termos
Variáveis:
Tab_Termos_Generalizadores : Tabela com os termos generalizadores de um determinado termo.

```
1:   Function QtdRelacsClassificador ( t1 : TipoTermo)
2:     for cada Relacionamento r1em Tab_Relacionamentos
3:       if r1.Classificador = t1then
4:         QtdRelacsClassificador ← QtdRelacsClassificador + 1
5:       end if
6:     end for
7:     return (QtdRelacsClassificador)
8:   end function
9:
10:  Procedure RemoveRelacionamentosReciprocos ;
11:    for cada Relacionamento r1em Tab_Relacionamentos
12:      for cada Relacionamento r2em Tab_Relacionamentos
13:        if r1.Classificador = r2.Classificado AND
14:          r1.Classificado  = r2.Classificador AND
15:          r1.Status <> D-Descartado AND
16:          r2.Status <> D-Descartado then
17:          if QtdRelacsClassificacao (r1.Classificador) > QtdRelacsClassificacao (r2.Classificador)
18:          then r2.Status ← D-Descartado
19:          else r1.Status  ← D-Descartado
20:          end  if
21:        end  if
22:      end for
23:    end for
24:  end procedure
```

Como o processo de busca na *Web* (etapa 4.5.2) utiliza diversos dicionários *online* para buscar a relação de um determinado termo, podem ser recuperados mais de um termo classificador para determinados termos pesquisados. A métrica apresentada a seguir identifica o melhor termo classificador dentre vários termos classificadores candidatos de um determinado termo.

A métrica para avaliar a capacidade classificadora de um determinado termo classificador candidato foi denominada de **FC-Fator de classificação**, sendo definida a seguir:

QC = Quantidade de termos Classificados
QRCE = Quantidade de Relações Classificadoras Externas dos termos classificados

$$FC = \frac{QC}{1 + QRCE + QC}$$

O **FC-Fator de Classificação** tem por objetivo priorizar o termo classificador cujo conjunto de termos classificados possui menos termos classificadores conflitantes de seus elementos (de preferência, apenas o termo classificador em questão). A razão de se utilizar a quantidade de elementos do conjunto classificado pelo termo no numerador e no denominador se deve a tornar esta métrica independente da quantidade de termos classificados, permitindo que seja medida a qualidade de classificação do termo em questão. Esta qualidade de classificação é inversamente proporcional ao número de conflitos (1+QRCE), na classificação do conjunto de termos classificados estes conflitos são relacionados ao fator QRCE, no qual QRCE=zero é o ideal, pois indica que o conjunto de termos classificados não possui relações com outros termos, apenas com o termo classificador que está sendo avaliado, levando ao valor de FC a ser aproximar do valor 1,0 (um).

A Figura 46 apresenta um exemplo de desambiguação com a utilização do FC, o termo ***T*** possui dois termos classificadores ***Ca*** e ***Cb***. O processo de transformação do dígrafo em uma estrutura hierárquica deve escolher apenas uma relação entre as relações ***Ca→ T*** e ***Cb→ T***, pois o termo ***T*** deve possuir somente um termo classificador.

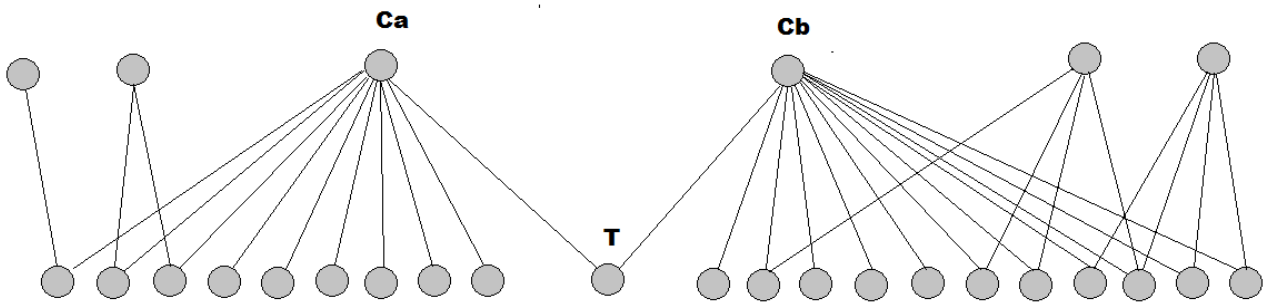


Figura 46: A escolha do termo classificador de T entre Ca e Cb pela métrica do FC.

A definição do melhor termo classificador de T, entre Ca e Cb é feita pela escolha do termo classificador de maior FC, como por exemplo:

FC (Ca) = (10) / (1 + 3+10) = 10/14 = 0,714

FC (Cb) = (12) / (1 + 8+12) = 12/21 = 0,571

Desta forma, a relação Ca→ T é selecionada para a hierarquia, sendo seu status correspondente ajustado para C-Contabilizada e a relação Cb→ T é marcada como D-Descartada, mas continuará a ser considerada para o cálculo do FC de outros termos relacionados com Cb.

A transformação do dígrafo em uma taxonomia será feita com a remoção das ambiguidades descritas anteriormente, aplicando-se as operações de remoção de inconsistências começando nos termos dos termos dos níveis superiores até os termos dos níveis inferiores. O fato de se utilizar um nível limite máximo para o processo de promoção os termos (etapa 4.6.2.4) levou ao ciclos ficarem concentrados nos dois níveis superiores, desta forma, o processo de remoção das inconsistências será aplicado primeiro nestes níveis superiores e depois nos níveis abaixo, progressivamente, resolvendo as inconsistências conforme descrito a seguir:

,

- **Ciclos.** Serão resolvidos progressivamente, à medida que cada termo que possua mais de um termo classificador seja verificado e seja escolhido dentre os seus termos classificadores, o melhor termo classificador deste com base na métrica apresentada na seção 4.6.4.1.
- **Ruído nos dados oriundos da *Web*.** O processo de filtragem pelo *Thesauro,* utilizado na recuperação das informações dos dicionários *online* reduz o ruído (hiperonímias erradas) geradas na etapa 4.5.2, descartando termos localizados como generalizadores que não estejam no *Thesauro* ou não sejam substantivos. Os processos de desambiguação descritos a seguir farão o descarte das hiperonímias que possuem classificação conflitante de termos.
- **Ambiguidades na classificação de termos.** Será resolvido com a seleção do termo classificador com maior FC-Fator de Classificação.

A seguir estão apresentadas as representações gráficas da aplicação dos processos descritos neste trabalho para a remoção das inconsistências:

- **Remoção das ambiguidades na generalização dos termos.** Caso o termo possua mais de um termo generalizador, será selecionado o termo generalizador que possuir o maior valor de FC (métrica descrita na etapa 4.6.3.1), conforme o exemplo apresentado na Figura 47:

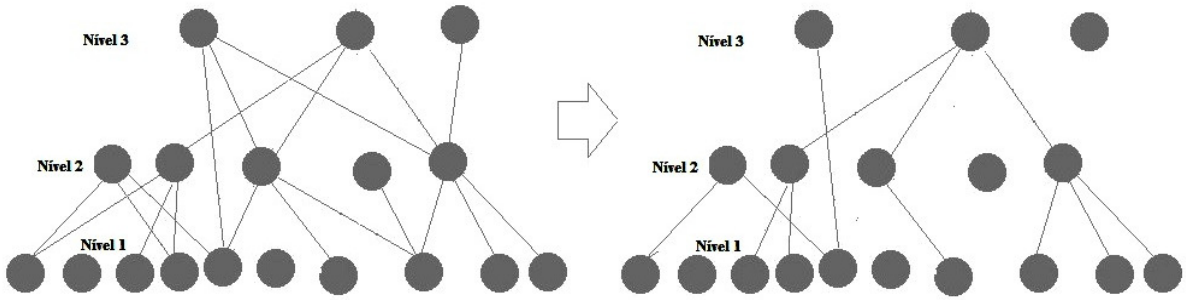


Figura 47: Representação da remoção das ambiguidades na generalização dos termos.

- **Remoção dos termos que não são generalizadores de termos e não são representativos de produtos.** Caso algum termo que não seja oriundo dos termos representativos de produto (etapa 4.6.2.1) e não pertencer como generalizador a nenhuma relação (com status C-Contabilizada) sobre alguns termos que não esteja com status D-Descartado, este será removido conforme a representação abaixo na Figura 48.

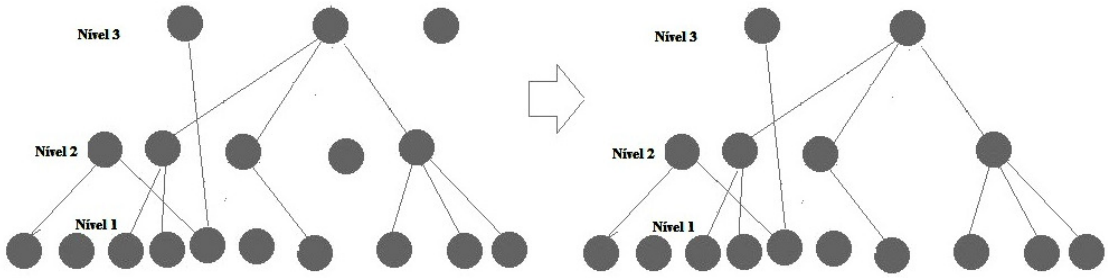


Figura 48: Representação da remoção dos termos dos Níveis 2 e 3 que não são generalizadores.

- **Ajuste nos atalhos.** Caso exista algum termo pertencente ao Nível 3 que não possua relação de generalização para com termos do Nível 2, somente com termos do Nível 1, este termo será reposicionado no Nível 2, conforme a representação apresentada na Figura 49.

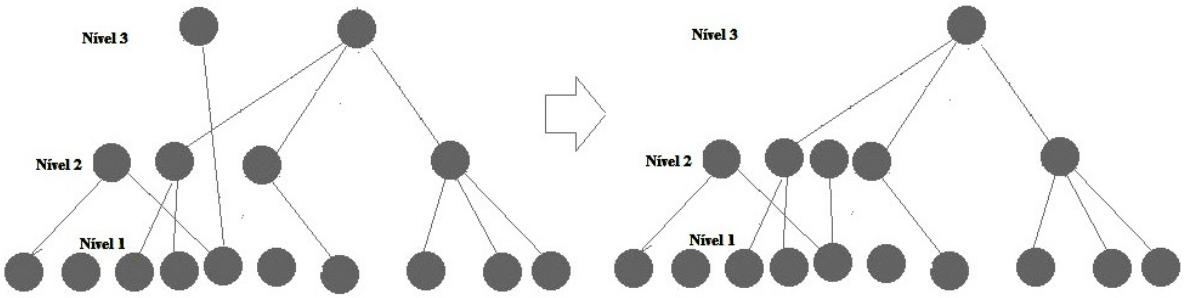


Figura 49: Ajustes nos níveis dos termos.

O processo descrito a seguir fará a seleção do melhor termo generalizador quanto existir mais de um termo generalizador e fará também a remoção dos atalhos. O algoritmo proposto resolverá as ambiguidades dos níveis superiores e progressivamente as ambiguidades dos níveis

inferiores (*top-down*). O motivo disto é o fato dos ciclos estarem posicionados nos níveis superiores devido ao processo descrito na etapa 4.6.2.4, no qual os termos são promovidos caso sejam classificadores de termos posicionados em níveis que não sejam inferiores ao nível do termo classificador. O processo de resolução destas ambiguidades de generalização está descrito a seguir no Algoritmo 08.

Algoritmo 08: Remoção dos relacionamentos ambíguos e atalhos do dígrafo

Entrada:
Tab_Termos : Tabela de Termos
Tab_relacionamentos: Tabela de Relacionamentos entre os termos
Variáveis:
Tab_Termos_Generalizadores : Tabela com os termos generalizadores de um determinado termo.

```
1:   Procedure CarregaTermosGeneralizadoresAscendentes (T1 : TipoTermo) ;
2:     for cada Termo T1 em Tab_Termos_Generalizadores
3:       Pilha_Generalizadores ← Termos Generalizadores de T1 e seus Ascendentes
4:   end procedure
5:
6:   Procedure DescartaAtalhosGeneralizadoresTermo ( T : TipoTermo)
7:     for cada Relacionamento em Tab_Relacionamentos
8:       if  Relacionamento.Generalizado = T then
9:         Insere Relacionamento.Generalizado.Termo em Tab_Termos_Generalizadores
10:      end if
11:    end for
12:    carregaTermosGeneralizadoresAscendentes
13:    for cada TermoT de Tab_Termos_Generalizadores
14:      if TermoT EXISTE em Pilha_Generalizadores then
15:        TermoT é Atalho
16:        Pilha_Generalizadores [TermoT].FC ← 0
17:      else
18:        Pilha_Generalizadores [TermoT].FC ← CalculoFC (TermoT)
19:      end if
20:    end for
21:  end procedure
22:
23:  Procedure RemoveAmbiguidadeGeneralizacao
24:    NivelProcessar <- Maior Nível de termo do dígrafo
25:    repeat
26:      for cada termo de Tab_Termos
27:        if termo.nivel = NivelProcessar
28:          if termo.NumeroGeneralizadores > 1 then
29:            DescartaAtalhosGeneralizadoresTermo (termo)
30:            seleciona termo generalizador de maior FC
31:            descarta outras relações de generalização de termo de menor FC
32:          end if
33:        end if
34:    end for
35:    NivelProcessar ← NivelProcessar – 1
36:    until NivelProcessar = 0
37:  end procedure
```

Após a remoção das ambiguidades nas relações de generalização entre os termos, deve-se

ajustar o nível destes na estrutura hierárquica. De forma diferente que o algoritmo 06, que posicionou os termos em níveis com um nível limite máximo para aplicar o processo de desambiguação no sentido *top-down*, o Algoritmo 09, descrito a seguir, somente pode ser aplicado a uma estrutura sem ambiguidades.

Algoritmo 09: Ajuste nos níveis de cada termo da taxonomia com as ambiguidades removidas.

```
1:   Procedure AjustaNivelTermosTaxonomia
2:     repeat
3:       mudou ← false
4:       for cada Termo T1 em Tabela_Termos
5:         Tab_Termos_Generalizados  ← Termos Generalizados de T1
6:         Buscar o maior Nível N1 em Tab_Termos_Generalizados
7:         if  T1.nivel <> (N1+1) then
8:            T1.Nivel ← N1+1
9:            Mudou ← True
10:        end if
11:    until not mudou
12:  end procedure
```

4.6.10 Conclusões do processo da criação da taxonomia

O processo proposto para a criação da taxonomia a partir das relações recuperadas na Web apresenta uma solução simples e de fácil implementação capaz de remover inconsistências e ambiguidades das informações iniciais.

Algumas características deste processo podem ser destacadas:

- O processo de construção do dígrafo foi *bottom-up*, desta forma os elementos dos níveis inferiores na sua maioria tem origem nos dados iniciais e os elementos dos níveis superiores tem sua origem no enriquecimento destes dados iniciais com informações recuperadas na *Web*. Devido a isso, é aconselhável que a resolução das ambiguidades seja *top-down*, pois nas partes superiores estará posicionada a maior parte das ambiguidades estruturais. Ressalta-se que, ao final da remoção das ambiguidades, os elementos desconexos que foram recuperados da *Web* podem ser removidos, mas os elementos oriundos dos dados iniciais devem continuar na taxonomia, posicionados no nível inferior.
- O processo de construção *bottom-up* de taxonomias, com a promoção de termos classificadores leva ao posicionamento dos elementos classificadores que formam ciclos para os níveis mais elevados, sendo necessária a aplicação de um nível limite máximo

para a promoção de elementos. Caso contrário o processo irá promover estes elementos indefinidamente.

- Após a finalização do dígrafo, serão resolvidas as ambiguidades oriundas dos relacionamentos recíprocos.
- O processo de resolução de ambiguidades irá resolver os conflitos que envolvam elementos entre níveis diferentes (atalhos e ciclos dos níveis superiores) para depois resolver as ambiguidades de classificação de cada elemento (do mesmo nível), pois a o **Fator de Classificação** descrito na seção **4.6.4.1** considera apenas fatores relacionados ao elemento do grafo que está sendo avaliado em relação aos seus elementos classificadores diretos.

5 IMPLEMENTAÇÃO E AVALIAÇÃO DO MODELO

A implementação do modelo CDDW foi feita em etapas, conforme eram analisados os dados gerados pelas etapas anteriores.

A primeira parte da implementação foi destinada a extrair e consolidar as informações dos produtos a partir da leitura dos registros de venda de produtos (cupons fiscais). Cada produto possui diversas descrições conforme devido ao fato deste ser vendido por diferentes emissores de cupom fiscal.

Devido à massiva quantidade de produtos identificados (5.472.209 Produtos com 13.725.722 de descrições diferentes), foi utilizado um subconjunto destes produtos selecionando-se um subconjunto de produtos vendidos em um supermercado de grande porte localizado no Estado do Rio de Janeiro. Todas as descrições utilizadas por todos os contribuintes foram associadas a este subconjunto de produtos, de forma a enriquecer textualmente cada produto, resultando em uma base de dados composta de 9.493 produtos e 1.092.397 descrições, a qual será a base utilizada pelo *framework* proposto para enriquecimento e classificação das informações dos produtos.

Desta forma, a implementação do modelo consistirá de cinco etapas, nas quais as etapas 2 em diante utilizarão este subconjunto de produtos descrito anteriormente para tornar o processamento destes dados exequível de ser executado em um computador pessoal. Estas etapas estão apresentadas a seguir:

1. Estabelecer Instâncias;
2. Determinar a Representação Textual;
3. Enriquecer e Classificar pela Representação Textual;
4. Organizar em Taxonomia;

5.1 Etapa: Estabelecer Instâncias

Para se obter os dados de cada produto (Estabelecer Instâncias) foi necessário varrer todos os cupons fiscais do ano de 2010 (3.5 bilhões de registros) e montar uma tabela com todos os produtos encontrados, associando-se a cada EAN válido identificado as suas respectivas descrições, conforme apresentado na Figura 51. Para cada registro lido é verificado se é um produto novo (novo EAN), se este for novo é acrescentado na tabela de produtos, se for um EAN já existente é verificado se esta descrição já existe associada ao produto (EAN) em questão. Caso seja uma descrição nova, esta é acrescentada ao produto em questão. Caso a descrição já exista, apenas é acrescentada uma unidade no contador de frequência da respectiva descrição do código de EAN em questão.

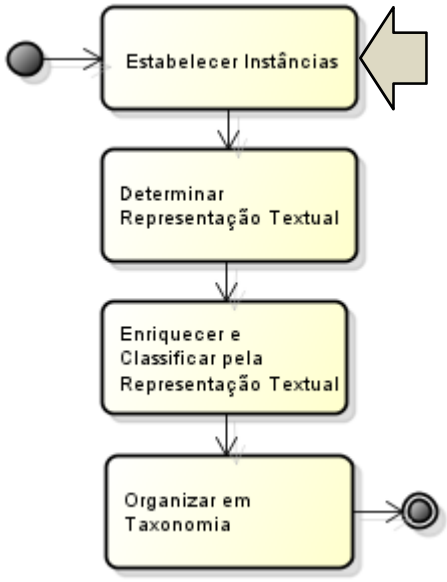


Figura 50 : Etapa 5.1

Como as informações de cada produto não existiam inicialmente, à medida que os cupons fiscais (vendas de produtos) eram processados, as informações dos produtos eram agrupadas por código de produto (código de EAN). As seguintes operações serão executadas, para cada registro de venda de produto lido, conforme apresentado na Figura 51:

1. Verificar se produto (código de EAN) já está registrado na tabela de produtos;
2. Verificar se descrição do produto vendido já está registrada na tabela de produtos;
3. Registrar produto novo com descrição nova na tabela de produtos;
4. Registrar descrição nova de produto nova em produto já registrado na tabela de produtos.

Durante o processamento dos cupons fiscais, à medida que o volume de informações na tabela de produtos aumentava, mais lento ficava o processo de verificação de cada produto lido, pois era necessário verificar se o código já existia ou não na tabela de produtos. Mesmo utilizando-se busca binária na tabela de produtos em memória ordenada por código de produto (EAN), o processo de re-ordenar a tabela com o grupo dos novos produtos encontrados consumia um tempo de processamento significativo, chegando a uma previsão dezenas de meses para se completar esta etapa.

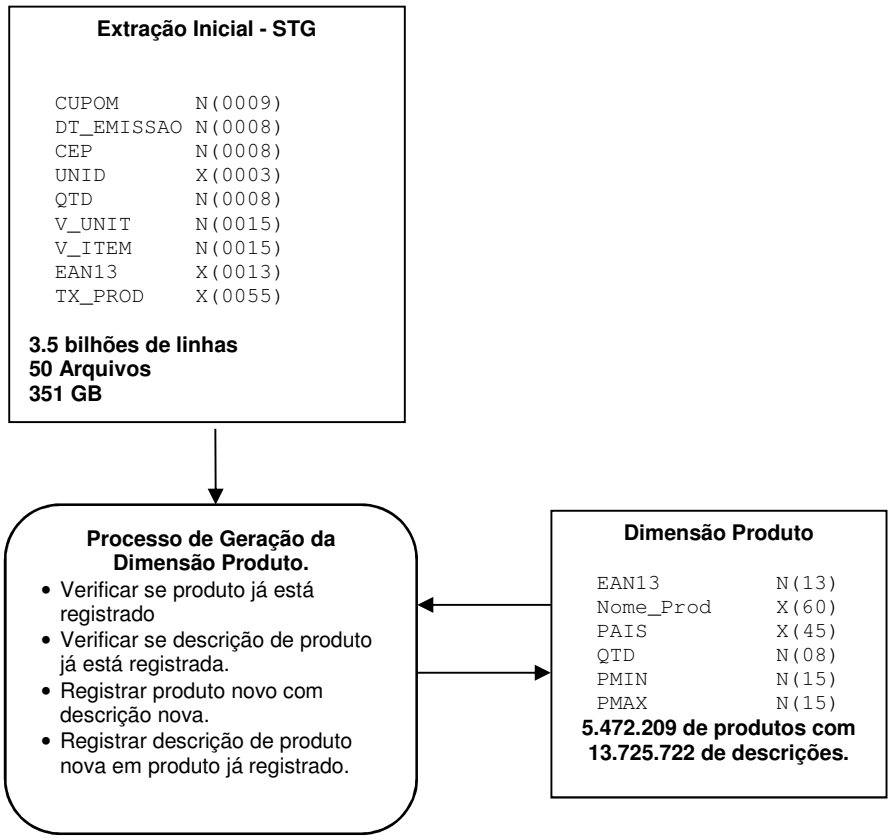


Figura 51 : Processo de seleção das descrições de cada produto.

Para que esta etapa pudesse ser completada em um prazo exequível, foram estudadas várias soluções para resolver este problema de performance. Uma das soluções analisadas era utilizar o algoritmo de *Hash* com a chave o código de EAN. Esta solução não foi utilizada de imediato devido ao fato de utilizar alocação de memória dinâmica, o que poderia acarretar erros de falta de memória durante o processamento, depois de horas ou dias de processamento. A solução inicial era definir os vetores para os dados dos produtos em memória com o maior tamanho possível e verificar se o programa conseguia carregar para ser executado. Caso o programa pudesse ser executado com um limite máximo de dados de produtos definido haveria a certeza que não ocorreriam erros de falta de memória, pois toda a memória necessária seria alocada durante a carga inicial do programa. Desta forma, o processamento poderia ser monitorado com um display informando o percentual do vetor utilizado.

Uma vez definido o vetor para comportar as informações dos produtos, o processamento dos cupons fiscais tinha a performance degradada à medida que o vetor de produtos acumulava informação. Inicialmente, para otimizar a operação "*1 – Verificar se produto (código de EAN) já*

está registrado na tabela de produtos" foi utilizada a ordenação por código de produtos com busca binária. Mas surgiu o problema dos produtos novos, que devem ser inseridos na tabela (operação 3). A inserção de um produto novo implicava na reordenação da tabela, a qual seria uma operação que era executada um número de vezes proporcional a quantidade de produtos. A previsão do tempo de duração desta etapa completar continuou em um número significativo de meses pois o processamento de menos de 1% dos cupons chegava a dezenas de horas.

A solução para não ordenar a tabela de produto para incluir os novos produtos encontrados na leitura do cupom fiscal foi utilizar uma segunda estrutura, não ordenada, de produtos novos a serem inseridos na dimensão produto. Quando esta estrutura possuir uma determinada quantidade de produtos, estes produtos serão inseridos na dimensão produto e esta será reordenada. O único detalhe desta solução é que a busca por produto será feita primeiro na dimensão ordenada de produto, caso não seja encontrado será feita a busca sequencial na estrutura não ordenada. Esta implementação permitiu uma melhora no desempenho do processamento dos cupons fiscais, mas à medida que a quantidade de produtos na dimensão produto aumentava, o processo de ordenação desta consumia mais tempo de processamento para ordená-la, pois a ordenação tem complexidade $O(n^2)$ (SHAFFER, 2012), à medida que a dimensão produto aumentava, o processo de ordenar os novos produtos encontrados consumia um tempo de processamento significativo e tornava-se mais lento, a previsão de duração desta etapa continuava em termo de meses. A solução para este problema foi trocar a ordenação da dimensão produto por um merge com a segunda estrutura, que é ordenada antes deste merge, já que a dimensão produto já está ordenada. O merge entre duas estruturas ordenadas tem complexidade $O(n)$.

Concluindo, por meio destas técnicas conseguiu-se um poder de processamento de aproximadamente 10 mil registros por segundo, levando-se aproximadamente 65 horas de processamento para processar os 3.5 bilhões de registros e gerar os dados de produto.

5.2 Etapa: Determinar representação textual

Esta etapa tem por objetivo definir uma única descrição textual para cada produto. Cada produto pode possuir diversas descrições, cada um utilizando uma descrição própria. O principal problema encontrado nesta etapa foi a qualidade dos dados de entrada. As descrições utilizadas nos registros de venda de produtos são limitadas a 28 caracteres. Desta forma, nas descrições do produto, as abreviações de palavras eram mais frequentes que a respectiva palavra original e utilização de palavras e abreviações variadas levaram a implementação de diversas técnicas nesta etapa para que a representação textual obtida tivesse qualidade suficiente para ser utilizada para a próxima etapa.

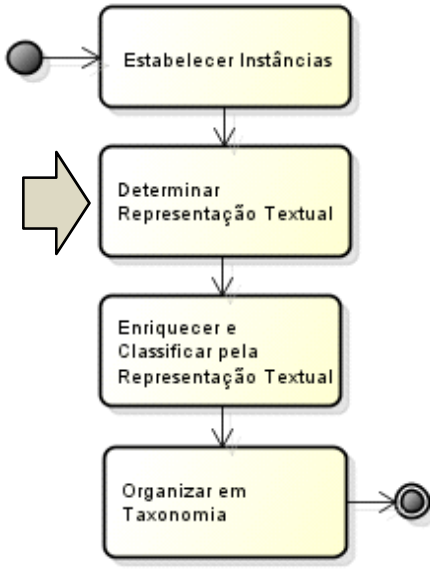


Figura 52 : Etapa 5.2

5.2.1 Desabreviação dos termos

O processo de desabreviação utilizou a métrica se semelhança descrita na etapa 4.4.2 juntamente com uma função que informava se cada palavra do par de palavras semelhantes existia no *Thesauro* ou não. Para minimizar os erros, o processo de desabreviação das palavras das descrições dos produtos é feito em dois passos:

Desabreviação dos termos de um mesmo prefixo de EAN (fabricante). Neste caso, o processo de desabreviação pode utilizar padrões de semelhança mais abrangente, pois os produtos de um mesmo prefixo de EAN (mesmo fabricante) são restritos a um determinado ramo, cada fabricante possui um ramo específico de atuação no mercado. Desta forma, as descrições dos produtos de um mesmo prefixo de EAN formam um conjunto mais restrito de termos e possuem muitos termos comuns, sem grande variedade de termos, podendo-se utilizar um padrão mais abrangente de desabreviação. Foram utilizados dois *Thresholds* para desabreviar o conjunto de termos de um mesmo EAN:

- Quando um par de termos com similaridade nos quais um dos termos existe no *Thesauro* e o outro não utilizou-se o *Thresold* de 0.80, neste caso o termo presente no *Thesauro* substitui o termo que não existe.

- Quando nenhum dos termos de um par com similaridade existe no *Thesauro*, o maior ou o mais acentuado substitui o outro, neste caso a semelhança deve ser igual ou superior ao *Thresold* de 0.90.
- Quando se encontra um par de termos com similaridade nos quais ambos os termos existam no *Thesauro* nada é feito, pois, apesar de semelhantes, cada termo possui um significado distinto.

Desabreviação dos termos de vários prefixos de EAN (fabricante). Neste caso, somente serão substituídos os termos que não existam no *Thesauro* por termos semelhantes que existam no *Thesauro* e o fator da semelhança seja superior a 0.95. A razão da aplicação de critérios mais rígidos para a substituição de termos abreviados é o fato do conjunto de termos envolvido ser bastante amplo e diversificado, oriundo de todas as descrições de produtos.

Tabela 17: Variações de classes de palavras para as macro-expressões:

Código EAN	Descrição Original (da maior quantidade vendida)	Descrição Final (Termos representativos selecionados)
7894000000268	#CALDO KNORR LEGUMES 57	CALDO KNORR LEGUMES
7894000000275	#CALDO KNORR GALINHA 57	CALDO KNORR GALINHA
7894000000282	#CALDO KNORR CARNE 57g	CALDO KNORR CARNE
7894000000299	TEMPERO KNORR FEIJAO 5	TEMPERO KNORR FEIJAO
7894000000350	#CALDO KNORR GALINHA 11	CALDO KNORR GALINHA
7894000000367	#CALDO KNORR CARNE 114g	CALDO KNORR CARNE
7894000001135	ALIM SOJA LIGHT ADES1L	ADESIVO SOJA LIGHT ZERO
7894000001142	ALIM SOJA LIGHT ADES1L	ADESIVO SOJA LIGHT
7894000001159	#BEBIDA ADES UVA 1000ml	ADESIVO UVA SOJA
7894000001180	#BEBIDA ADES MORANGO 10	ADESIVO MORANGOPESSEGO SOJA
7894000010014	AMIDO DE MILHO MAIZENA	MAIZENA MILHO AMIDO
7894000021249	MEL KARO NATURAL PET 3	KARONENBIER NATURAL PETE
7894000030548	SOPA PTO FGO/ESPINAFRE	SOPA PTO FGOBATA ESPINAFRE
7894000030555	SOPA CARNE BAT 14G	SOPA CARNE BATA
7894000033396	BEBIDA ADES PESSEGO ZE	ADESIVO SOJA PESSEGO
7894000033808	#CALDO KNORR COSTELA 57	CALDO KNORR COSTELA
7894000033860	#CALDO KNORR BACON/LOUR	CALDO KNORR BACON LOURO
7894000034638	#BEBIDA ADES MARACUJA 1	ADESIVO MAR SOJA
7894000034669	#BEBIDA ADES MANGA 1000	ADESIVO MANGA SOJA
7894000034690	ALIM SOJA LIGHT ADES1L	ADESIVO SOJA LIGHT ZERO
7894000047515	#BEBIDA ADES PESSEGO 10	ADESIVO PESSEGO SOJA
7894000047522	#BEBIDA ADES ABACAXI 10	BEBIDA ADESIVO ABACAXI
7894000050027	MAIONESE HELLMANN S 25	MAIONESE HELLMANNS
7894000050034	#MAIONESE HELLMANN S 50	MAIONESE HELLMANNS
7894000050416	MOLHO HELLMANNS SALADA	MOLHO HELLMANNS SALADA CASEIRA
7894000050423	MOLHO HELLMANNS SALADA	MOLHO HELLMANNS SALADA ITALIANO
7894000050454	MOLHO HELLMANNS SALADA	MOLHO HELLMANNS SALADA ROSE
7894000050522	MAION.HELLMANS LIMAO 250	MAIONESE HELLMANNS LIMA
7894000050720	#MAIONESE HELLMANN S LI	MAIONESE HELLMANNS LIGHT
7894000050737	MAIONESE HELLMANN S LI	MAIONESE HELLMANNS
7894000051031	COB KARO CARAMELO 340G	COBERTURA KARONENBIER CARAMELO
7894000051048	COB KARO MORANGO 340G	COBERTURA KARONENBIER MORANGOPESSEGO
7894000051055	COB KARO CHOC 340G	COBERTURA KARONENBIER CHOCOZE
7894000068718	MOLHO HELLMANNS SALAD	MOLHO HELLMANNS SALADA PARMESAO
7894000080505	SOPA KNORR DE CEBOLA 4	SOPA KNORR CEBOLA

Conforme as amostras apresentadas na Tabela 17, o processo de desabreviação foi bastante efetivo na desabreviação de substantivos como “CALDO”, “MAIZENA”, “SOPA”, “BEBIDA”, “MOLHO”, “COBERTURA” e etc, pois estes existem no *Thesauro* utilizado e não serão desabreviados pois possuem significado. No caso dos nomes próprios e nomes de marca como “ADES”,“KARO” e “CHOC” não foi possível ter um critério definido para identificar que o termo encontrado não necessita mais de desabreviação, levando a formação de termos errôneos como “KARONENBIER” e “ADESIVO”.

5.2.2 Conclusões da desabreviação dos termos

Podemos verificar que o a função de semelhança abreviada descrita na etapa 4.4.2 foi efetiva na desabreviação dos termos das descrições dos produtos. Foi fundamental a utilização do *Thesauro* para identificar que o termo desabreviado está correto e não necessita mais de desabreviação. A utilização de *Thresolds* de semelhança diferentes em função do tamanho do conjunto de termos comparados minimizou os erros de desabreviação de palavras semelhantes com significados diferentes.

Um problema detectado que não pode ser resolvido foi a desabreviação dos nomes próprios (de marca ou de fabricante), pois como estes não existem no *Thesauro*. Desta forma, estes termos são progressivamente desabreviados até a maior palavra semelhante, mesmo que seja um erro de digitação na descrição do produto. A tabela 18 apresenta a seguir algumas desabreviações e o termo desabreviado “ADESCALC” na amostra de termos desabreviados, oriundo da desabreviação dos termos do prefixo de EAN 78911500:

Tabela 18: Amostra dos termos desabreviados do prefixo de EAN 78911500

```
TERMO_DESABREVIADO: SUC --> SUCO STATUS:F
TERMO_DESABREVIADO: CEREA --> CEREAL STATUS:F
TERMO_DESABREVIADO: CER --> CEREAL STATUS:F
TERMO_DESABREVIADO: NUTR --> NUTRITIVO STATUS:F
TERMO_DESABREVIADO: CALC --> CALCIO STATUS:F
TERMO_DESABREVIADO: VITAM --> VITAMINA STATUS:F
TERMO_DESABREVIADO: FRAPE --> FRAPES STATUS:F
TERMO_DESABREVIADO: FRAP --> FRAPES STATUS:F
TERMO_DESABREVIADO: SOJ --> SOJA STATUS:F
TERMO_DESABREVIADO: ADESCALC --> ADESCALCIO STATUS:F
TERMO_DESABREVIADO: ZER --> ZERO STATUS:F
TERMO_DESABREVIADO: SCO --> SUCO STATUS:F
TERMO_DESABREVIADO: LEIT --> LEITE STATUS:F
TERMO_DESABREVIADO: NUTRITI --> NUTRITIVO STATUS:F
TERMO_DESABREVIADO: BEBID --> BEBIDA STATUS:F
TERMO_DESABREVIADO: FRA --> FRANGO STATUS:F
```

5.2.3 Identificação dos termos representativos de cada instância.

A partir das descrições clusterizadas, desabreviadas e selecionadas de cada produto (EAN), será necessário definir quais termos representarão cada produto dentre os diversos termos presentes no respectivo conjunto de descrições, conforme mostrado nas Tabelas 19 e 20.

Tabela 19 : Descrições do produto de EAN 7896043001011.

```
---PRODUTO:7896043001011->
 Desc Original: PO SOYMILKE NATURAL 30    (0-1047)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-915)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-716)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-700)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-676)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-588)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-552)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-497)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL G     (0-336)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30    (0-321)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: LEITE SOYMILKE 300G NATU(0-311) Termos:LEITE-1(6)+SOYMILKE-2(4)+NATU-4(7)+
 Desc Original: PO SOYMILKE NATURAL 30   (0-308) Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: LEITE PO SOYMILKE NATURAL(0-295) Termos:LEITE-1(6)+SOYMILKE-3(4)+NATURAL-4(5)+
 Desc Original: PO SOYMILKE NATURAL 30 (0-295)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30 (0-290)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: PO SOYMILKE NATURAL 30 0-288)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: LEITE PO SOYMILKE NATURAL (0-282) Termos:LEITE-1(6)+SOYMILKE-3(4)+NATURAL-4(5)+Desc
Original: SOYMILKE NATURAL 300G (0-278)  Termos:SOYMILKE-1(4)+NATURAL-2(5)+
 Desc Original: PO SOYMILKE NATURAL 30    0-278)  Termos:SOYMILKE-2(4)+NATURAL-3(5)+
 Desc Original: SOYMILKE NATURAL 300G     0-253)  Termos:SOYMILKE-1(4)+NATURAL-2(5)+
 Desc Original: SOYMILKE NATURAL 300GR    (0-248)  Termos:SOYMILKE-1(4)+NATURAL-2(5)+
 Desc Original: SOYMILKE NATURAL 300GR   (0-244)  Termos:SOYMILKE-1(4)+NATURAL-2(5)+Desc
```

A seleção dos termos representativos é feita a partir da seleção do termo mais representativo a partir do conjunto de termos localizados na posição 1 das descrições. Este processo se repete para as próximas posições, até a posição 5 ou até nenhum termo selecionado possuir a frequência mínima (10% das descrições).

No caso das posições 1 e 2, a seleção do termo mais representativo de uma determinada posição nas descrições é feita prioritariamente sobre o conjunto de termos que existam no *Thesauro*. Caso se selecione um termo q ue não exista no *Thesauro* e não possua a frequência mínima (10% das descrições), será feita a seleção do termo de maior frequência a partir do conjunto de todos os termos da respectiva posição nas descrições, sendo aplicado o limite de frequência mínima.

Este processo busca selecionar os termos representativos com o foco na frequência destes a partir das posições iniciais (mais à esquerda) na descrição. A razão de ser preservar a ordem dos termos é considerar que os termos representativos do produto são os termos iniciais das descrições.

Este critério é importante para a qualidade dos dados a serem utilizados nas próximas etapas.

Na Tabela 19, apresentada anteriormente, estão descrições de um produto cujo EAN é 7896043001011 e na Tabela 20 está o resultado do processo de definição dos termos representativos do respectivo produto. Os termos selecionados como representativos deste produto foram: LEITE, NATURAL E SOYMILKE , sendo que os dois primeiros existem no *Thesauro* utilizado. Podemos observar que a utilização do *Thesauro* priorizou os termos descritivos do produto, como LEITE e NATURAL no lugar de SOYMILKE, que é um termo de marca que provavelmente não será útil na busca de seu significado nas próximas etapas do *framework*.

Tabela 20 : Definição dos termos representativos do produto de EAN 7896043001011.

NOME	Existe no Thesauro	Quantidade Vendida	Quantidade Descrições	Ordem (1)	Ordem (2)	Ordem (3)	Ordem (4)	Ordem (5)	Ordem (6)
SOYMILKE(8256)	Não	19599	110	8101*110	9361*110	**2137*110**	0*110	0*110	0*110
NATURAL(1002)	Sim	19161	107	0*107	**7090*107**	10087*107	1984*107	0*107	0*107
LEITE(248)	Sim	2562	16	**2562*16**	0*16	0*16	0*16	0*16	0*16
INSTANT(5633)	Não	166	1	0*1	0*1	166*1	0*1	0*1	0*1
COMPOSTO(2568)	Sim	431	3	431*3	0*3	0*3	0*3	0*3	0*3
ALIMENTO(2174)	Sim	694	5	263*5	431*5	0*5	0*5	0*5	0*5
SOYMI(8257)	Não	630	5	0*5	0*5	630*5	0*5	0*5	0*5
SOJA(2175)	Sim	584	6	0*6	584*6	0*6	0*6	0*6	0*6
SOY(246)	Não	998	13	998*13	0*13	0*13	0*13	0*13	0*13
MILK(1018)	Não	998	13	0*13	998*13	0*13	0*13	0*13	0*13
LTE(398)	Não	79	1	79*1	0*1	0*1	0*1	0*1	0*1
TRADICIONAL(571)	Sim	76	1	0*1	76*1	0*1	0*1	0*1	0*1

Podemos observar na Tabela 20 que o processo seleciona os termos LEITE e NATURAL. Os próximos termos existentes no *Thesauro* não possuem a pontuação mínima para serem selecionados, desta forma o processo seleciona dentre o conjunto de termos total (sem filtrar pelo *Thesauro*) o último termo SOYMILKE.

5.2.4 Descarte dos termos conflitantes pela frequência nas descrições

Podemos observar na tabela 18 que o processo seleciona os termos LEITE e NATURAL. Os próximos termos existentes no *Thesauro* não possuem a pontuação mínima para serem selecionados, desta forma o processo seleciona dentre o conjunto de termos total (sem filtrar pelo *Thesauro*) o termo SOYMILKE. Como não existe nenhum termo acima da pontuação mínima para a quarta posição em diante, o processo encerra com três termos selecionados: LEITE, NATURAL E

SOYMILKE.

5.2.5 Conclusões do processo de determinação da descrição textual de cada instância.

Observa-se que os dados recebidos pela próxima etapa foram utilizados com sucesso na geração da taxonomia das instâncias, apesar de existirem abreviações e ambiguidades.

Pode-se destacar quatro técnicas que possibilitaram que o processo de geração da descrição formal fosse bem sucedido:

Função de similaridade abreviada. A função proposta na etapa 4.4.2 permitiu a identificação de semelhanças entre termos não abreviados e termos abreviados, possibilitando a substituição dos termos abreviados pelos termos desabreviados.

A utilização da função de desabreviação de forma progressiva. A desabreviação foi aplicada com critérios mais amplos sobre um conjunto de termos restrito (termos de um conjunto de instâncias – produtos de um fabricante), depois foi aplicada a desabreviação com critérios um pouco mais restritos sobre um conjunto maior de termos, oriundo de todos os termos dos produtos, conforme apresentado na Tabela 21. Desta forma, foi possível minimizar os erros de desabreviação entre palavras semelhantes que não tem o mesmo significado. Deve-se ressaltar que os fatores de semelhança utilizados, foram definidos a partir de algumas execuções do processo de desabreviação, após cada análise dos resultados do processo e do respectivo ajuste nestes fatores. Estes fatores não devem ser considerados definitivos para todos os processos de desabreviação e sim como *Thresholds* que devem ser definidos em função do conjunto de informações que devem ser desabreviadas.

Tabela 21 : Fatores de semelhança utilizados sobre cada conjunto de termos.

Tipo de Conjunto de Termos	Abrangência do conjunto a ser desabreviado	Fator de Semelhança Utilizado para definir que é abreviação a outra palavra
Termos de um conjunto de instâncias semelhante (prefixo de produtos), o termo desabreviado existe no *Thesauro*.	**Restrita**	**0.80**
Termos de um conjunto de instâncias semelhante (prefixo de produtos), nenhum existe no *Thesauro*.	**Restrita**	**0.90**
Termos do conjunto formado por todos os termos de produtos. O termo desabreviado existe no *Thesauro*.	**Ampla**	**0.95**

Utilização do *Thesauro* para definir quando um termo já atingiu a forma desabreviada. Devido ao grande volume de termos, determinados termos eram desabreviados sucessivamente, passado de sua forma padrão (*Lemma*) para plurais, diminutivos, aumentativos ou mesmo outras palavras semelhantes cujo prefixo é semelhante à palavra original. Este problema foi resolvido ao se verificar se a palavra a ser abreviada existe no *Thesauro* ou não antes de substituí-la por outra semelhante de maior tamanho. O fato de uma palavra existir no *Thesauro* indica que possui significado próprio, não necessitando mais se desabreviada.

Utilização de um critério para definir se um termo deve ser selecionado ou descartado. Devido à quantidade de termos desabreviados de cada instância, deve-se definir como representativos apenas os termos que atingirem um determinado critério (definido na etapa 4.4.4). Mesmo que nenhum termo da uma instância tenha sido selecionado por não atingir o critério, esta instância ficará sem termos representativos. A aplicação deste critério permitiu a construção de uma taxonomia enxuta e com poucos erros nas próximas etapas.

5.3 Utilizando Dicionários *Online* para Enriquecer a Representação textual

A partir da identificação dos termos representativos de cada instância (cada produto), deve-se buscar uma forma de classificar os produtos por meio destes termos. O objetivo desta etapa é buscar a generalização (hiperonímia) de cada termo na *Web*, mais especificamente em dicionários *online.*

Devido ao fato dos dicionários *online* possuírem padrões textuais particulares, foi utilizado o aprendizado de máquina para a construção de um agente capaz de localizar as relações dos termos desejadas no texto retornado por cada dicionário *online* pesquisado.

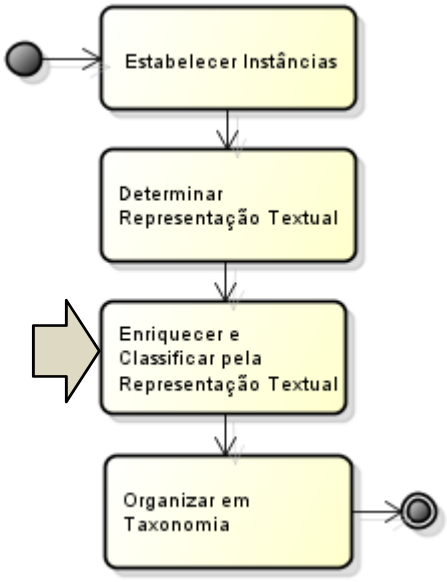


Figura 53 : Etapa 5.3

Foi desenvolvido para esta tese um software que navega na *internet* (*crawler)* e acessa diversos dicionários *online* com o objetivo de buscar o(s) termo(s) generalizador(res) do termo (hiperonímia) pesquisado em questão. A partir destes termos generalizadores será montada uma taxonomia capaz de classificar os produtos.

Desta forma, o *software* desenvolvido acessa determinados dicionários *online* passando o termo a ser pesquisado na URL, recebendo como resposta um arquivo HTML no qual é extraído o texto e processados os caracteres de *charsets* e acentos para se obter o texto a ser analisado pelas macro-expressões localizadoras da generalização do termo pesquisado.

Ao se analisar os dados retornados pelos dicionários *online*, observou-se que determinados dicionários retornavam exatamente o mesmo texto retornado por outros dicionários. Quando esta situação era encontrada, apenas um dicionário *online* era escolhido para se utilizado e o outro era descartado. A Tabela 22 apresenta a lista de dicionários *online* analisados e descartados:

Tabela 22: Dicionários *online* analisados e descartados para evitar redundâncias.

URL dos Dicionários *Online* que foram descartados
http://www.dicio.com.br/
http://www.lexico.pt
http://www.infopedia.pt/dicionarios/lingua-portuguesa/
http://pt.wikipedia.org/wiki
http://dicionariorapido.com.br/

Na Tabela 23 apresentada a seguir está a lista de dicionários *online* que foram utilizados para a recuperação das relações de classificação a serem utilizadas na construção da hierarquia de classes de produtos. Cada dicionário possui suas regras particulares de disposição das suas informações textuais nas sua respectivas páginas da *Web*. Para se recuperar estas informações são necessárias regras particulares para a localização destas. As macro-expressões geradas pelo aprendizado de máquina são compostas por dois tipos (RUSSEL e NORVIG, 2009):

- **Regras positivas** localizam a descrição do termo pesquisado;
- **Regras negativas** identificam a situação de que o site pesquisado informa que não possui informações sobre o termo pesquisado.

Tabela 23 : Dicionários *online* utilizados na busca das generalizações dos produtos.

Número	Sites de Dicionários *Online*	Quantidade de Regras Positivas	Quantidade de Regras Negativas
1	http://www.tiosam.org/	**18**	**1**
2	http://www.priberam.pt/	**19**	**4**
3	http://michaelis.uol.com.br/	**44**	**2**
4	http://www.dicionarioweb.com.br/	**15**	**0**
5	http://dicionario-online.com/	**2**	**1**
6	http://bemfalar.com/	**19**	**1**

Conforme definido na etapa 4.5.2, foram geradas macro-expressões localizadoras a partir de exemplos fornecidos por um especialista. Estas macro-expressões são particulares de cada dicionário *online,* pois cada site possui padrões particulares de apresentação, escrita do texto e utilização de abreviações. A localização fornecida pelas macro-expressões é referente ao ponto onde se inicia a descrição do termo pesquisado no texto retornado pelos dicionários *online*, com grande possibilidade do termo pesquisado estar na primeira posição. Mas pelo fato do termo procurado não estar necessariamente na primeira posição da descrição, foi incluído um processo que procura o primeiro termo que seja substantivo e não esteja na lista de palavras que devem ser ignoradas na descrição do termo retornada.

O processo de geração das macro-expressões é feito por um processo computacional que analisa os exemplos fornecidos pelo especialista, identifica os padrões comuns e generaliza e testa

estes padrões obtidos de forma a maximizar a precisão deste padrão, de forma que possam ser aplicados com sucesso sobre os outros exemplos fornecidos, conforme descrito na etapa 4.5.2. A Tabela 24 apresentada a seguir mostra a generalização de um padrão oriundo do site http://bemfalar.com/, conforme a respectiva precisão deste padrão é mantida, neste caso a Precisão é de 22,22%, resultando na macro-expressão: **"[:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO".** Neste caso, a macro-expressão gerada é especializada na localização da descrição no texto para o caso do termo ser um substantivo feminino.

Tabela 24: Generalização de um padrão oriundo do site http://bemfalar.com/.

```
REGRA_GENERALIZADA_TP: 2 NOS PALAVRA A PESQUISAR: [:termo] SIGNIFICADO DE [:outrapalavra]{0,1}
SUBSTANTIVO FEMININO
 GENERALIZAÇÃO_NOVA_TP: 2 PALAVRA A PESQUISAR: [:termo] SIGNIFICADO DE [:outrapalavra]{0,1}
SUBSTANTIVO FEMININO  (22,22% , 9,52% )  REGRA_ANTERIOR:  (22,22% , 9,52% )

 REGRA_GENERALIZADA_TP: 2 PALAVRA A PESQUISAR:  [:termo] SIGNIFICADO DE [:outrapalavra]{0,1}
SUBSTANTIVO FEMININO
 GENERALIZAÇÃO_NOVA_TP: 2 A PESQUISAR:  [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO
FEMININO  (22,22% , 9,52% )  REGRA_ANTERIOR:  (22,22% , 9,52% )

 REGRA_GENERALIZADA_TP: 2 A PESQUISAR:  [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO
FEMININO
 GENERALIZAÇÃO_NOVA_TP: 2 PESQUISAR:  [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO
FEMININO  (22,22% , 9,52% )  REGRA_ANTERIOR:  (22,22% , 9,52% )

 REGRA_GENERALIZADA_TP: 2 PESQUISAR:  [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO
FEMININO
 GENERALIZAÇÃO_NOVA_TP: 2 [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO
(22,22% , 9,52% )  REGRA_ANTERIOR:  (22,22% , 9,52% )

 REGRA_GENERALIZADA_TP: 2 [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO
 GENERALIZAÇÃO_NOVA_TP: 2 SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO  (20,00% , 9,52%
)  REGRA_ANTERIOR:  (22,22% , 9,52% )

 GENERALIZACAO_DESCARTADA_TP:SIGNIFICADO DE [:outrapalavra]{0,1}  SUBSTANTIVO FEMININO

 REGRA_FINAL_A_SER_CONSIDERADA (TP) 2 : [:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO
FEMININO  (22,22% , 9,52% )
```

O processo gerou macro-expressões para cada dicionário *online* utilizado pelo modelo a partir de uma massa de treinamento. Na Tabela 25, apresentada a seguir, estão as macro-expressões referentes ao site http://bemfalar.com. Estas macro-expressões são aplicadas sobre o texto retornado pelo dicionário sequencialmente, conforme o algoritmo 04 descrito na etapa 4.5.1, parando quando algum padrão for localizado ou quando nenhuma macro-expressão localizar o padrão no texto. Caso seja localizado um padrão, se a respectiva macro-expressão for do tipo VP (verdadeiro positivo), então o ponto localizado é o início da descrição do termo pesquisado. Caso seja localizado um padrão cuja respectiva macro-expressão do tipo VN então o termo pesquisado não existe na base de dados do dicionário *online* pesquisado.

Tabela 25: Macro-expressões geradas para o site http://bemfalar.com

URL de acesso ao dicionário: http://bemfalar.com/significado/<termo>.html		
Seq	**Tipo**	**Macro-expressão**
1	VN	PÁGINA PRINCIPAL POLÍTICA DE PRIVACIDADE CONTACTE NOS PARCERIAS PUBLICIDADE JOGUE NO NOSSO SITE OS JOGOS MAIS
2	VP	DESIGN COMUM [:outrapalavra]{0,1}
3	VP	RUBRICA: ANGIOSPERMAS DESIGN COMUM [:outrapalavra]{0,1}
4	VP	MASCULINO RUBRICA: ANGIOSPERMAS 1
5	VP	RUBRICA: [:outrapalavra]{0,1} 1 [:outrapalavra]{0,4} DE
6	VP	MASCULINO 1 RUBRICA: ANGIOSPERMAS
7	VP	DE [:outrapalavra]{0,2} 1
8	VP	MASCULINO RUBRICA: [:outrapalavra]{0,1} 1
9	VP	FEMININO 1 RUBRICA: [:outrapalavra]{0,1}
10	VP	[:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO RUBRICA: [:outrapalavra]{0,1}
11	VP	[:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO RUBRICA: ANGIOSPERMAS 1
12	VP	MASCULINO 1 [:outrapalavra]{0,3}
13	VP	FEMININO 1 RUBRICA: ANGIOSPERMAS
14	VP	1 RUBRICA: [:outrapalavra]{0,2} DE [:outrapalavra]{0,3}
15	VP	MASCULINO 1 [:outrapalavra]{0,1} DE
16	VP	MASCULINO 1
17	VP	SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO MASCULINO
18	VP	FEMININO 1
19	VP	[:termo] SIGNIFICADO DE [:outrapalavra]{0,1} SUBSTANTIVO FEMININO

Cada conjunto de macro-expressões foi avaliada com a execução de busca de 251 termos de exemplos. O resultado da busca das macro-expressoes de cada dicionário *online* era avaliado pelo especialista, que ajustava os exemplos de forma a maximizar a precisão e a revocação de cada conjunto de macro-expressões de cada dicionário *online*. A Figura 54 a seguir mostra a implementação deste processo interativo de refinamento das macro-expressões:

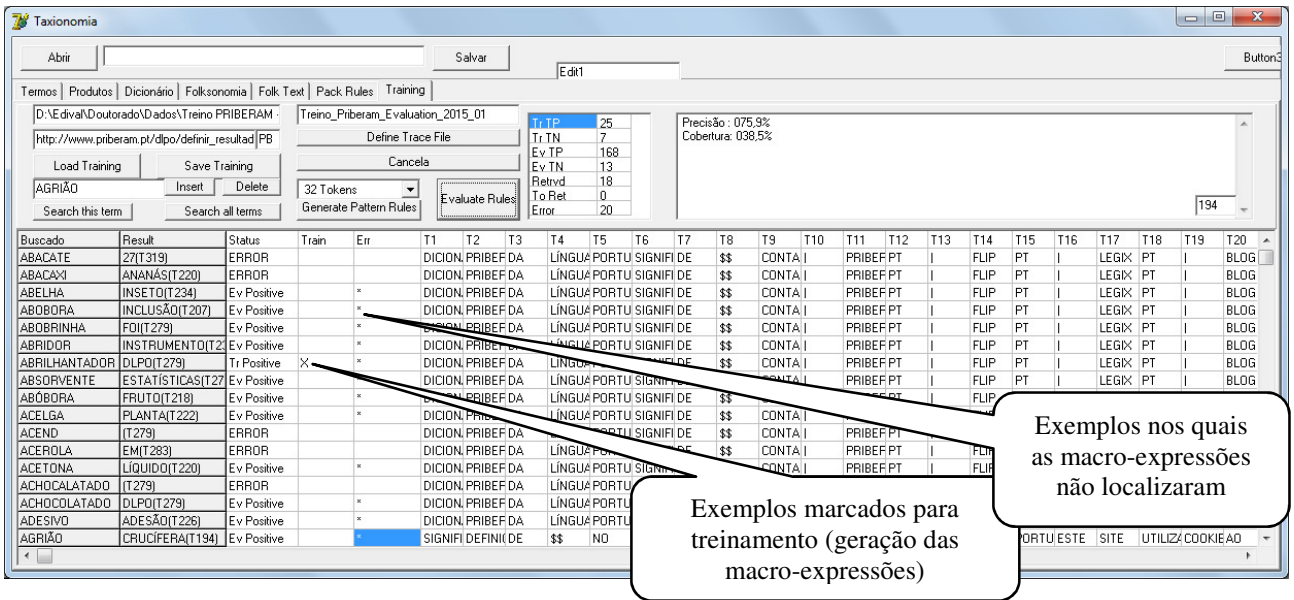


Figura 54 : Implementação do ciclo evolutivo de aprendizado assistido pelo especialista.

As macro-expressões são avaliadas sobre o conjunto de 251 termos fornecidos de exemplos, sendo calculados os percentuais de precisão e revocação. Cada exemplo pode possuir os seguintes estados (coluna *Status* da Figura 54):

- ***Error*.** Não será utilizado para gerar padrões. Sendo descartado do processo.
- ***Evaluation Positive***. Será utilizado para se avaliar as macro-expressões VP (verdadeiro positivo), cuja localização deve coincidir com o ponto selecionado no texto (coluna *Result*).
- ***Evaluation Negative***. Será utilizado para se avaliar as macro-expressões VN (verdadeiro negativo), cuja localização deve coincidir com o ponto selecionado no texto (coluna *Result*).
- ***Training Positive***. Será utilizado para gerar os padrões que formarão o conjunto de macro-expressões do tipo VP (verdadeiro positivo) e para avaliar as macro-expressões geradas do tipo VP (verdadeiro positivo), cuja localização deve coincidir com o ponto selecionado no texto (coluna *Result*).
- ***Training Negative***. Será utilizado para gerar os padrões que formarão o conjunto de macro-expressões do tipo VN (verdadeiro negativo) e para as macro-expressões geradas do tipo VN (verdadeiro negativo), cuja localização deve coincidir com o ponto selecionado no texto (coluna *Result*).

Os exemplos de treinamento podem mudam de status conforme são processados pelo software (para buscar o texto no dicionário *online*) e por comandos executados pelo especialista, conforme os resultados da localização das macro-expressões geradas (a partir dos status *Training*) são verificadas sobre os exemplos (status *Evaluation* e *Training*). Conforme o diagrama apresentado na Figura 55:

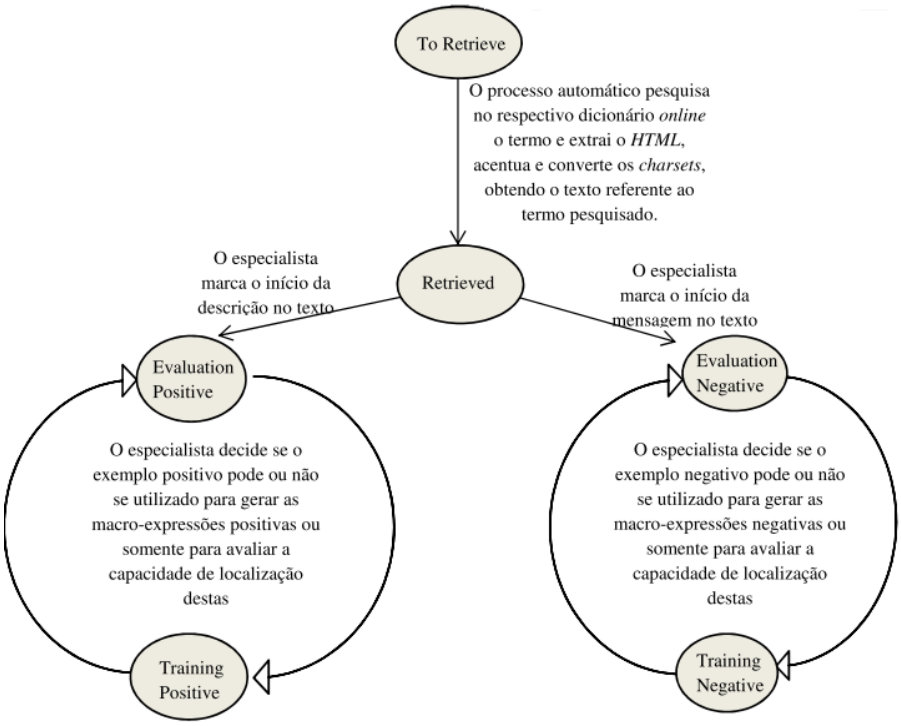


Figura 55: Diagrama de mudança de status dos exemplos de treinamento.

O processo que avalia as macro-expressões também marca os exemplos que não foram localizados com sucesso colocando um asterisco "*" na coluna *Err*. Desta forma o especialista pode incluir este termo nos exemplos de treinamento (trocar o estado de *Evaluation Positive* para *Training Positive*) ou retirar o temo dos exemplos de treinamento (mudar de *Training Positive* para *Evaluation Positive*), para não influenciar a geração de macro-expressões a partir de padrões que não são localizados corretamente. **Este processo é feito interativamente pelo especialista, mas poderá ser evoluído no futuro para ser feito automaticamente por *software*, pois pode-se identificar quais exemplos deram origem a cada macro-expressão e fazer a avaliação, ajustando-se o estado de cada exemplo envolvido conforme os percentuais de precisão e revocação da respectiva macro-expressão**. O Especialista seria necessário apenas para definir o ponto desejado do texto, de cada exemplo, que se deseja localizar. Esta proposta será colocada mais adiante na seção de trabalhos futuros.

5.3.1 Avaliação do processo de utilização das macro-expressões sobre o texto dos dicionários *online* para enriquecer a descrição do produto.

Após as avaliações e ajustes de todas as macro-expressões dos dicionários *online*, foi montado um pacote de acesso com todas estas macro-expressões geradas e evoluídas de todos os dicionários *online* para a recuperação da hiperoníma dos termos dos produtos, as quais foram utilizadas na próxima etapa. Na Tabela 26 a seguir estão os percentuais de precisão e revocação de cada conjunto de macro-expressões de cada dicionário *online*.

Tabela 26 : Dicionários *online* utilizados na busca das generalizações dos produtos.

Número	Site de Dicionário *Online*	Quantidade de Regras Positivas	Quantidade de Regras Negativas	Precisão	Revocação
1	http://www.tiosam.org/	18	1	83,90%	63,80%
2	http://www.priberam.pt/	19	4	75,93%	38,50%
3	http://michaelis.uol.com.br/	44	2	73,33%	64,71%
4	http://www.dicionarioweb.com.br/	15	0	76,67%	63,89%
5	http://dicionario-online.com/	2	1	98,08%	39,53%
6	http://bemfalar.com/	19	1	65,05%	63,81%

A Tabela 27 apresenta a seguir o percentual de retorno de informação de cada dicionário *online* utilizado. Pode-se observar que as macro-expressões de determinados dicionários *online* apresentaram um percentual de retorno bastante baixo. Neste caso deve-se avaliar as macro-expressões geradas destes dicionários *online* específicos e ajustar os exemplos, de forma a aumentar o percentual de retorno destes *sites*.

Tabela 27: Percentual de retorno de cada dicionário *online*.

Dicionário *Online*	Número de pesquisas	Retornos com informação (VP + VN)	Percentual de retorno (apenas sobre VP)
bemfalar.com	5048	2092	68,5%
dicionario-online.com	5048	6	0,1%
www.dicionarioweb.com.br	5048	1526	45,7%
michaelis.uol.com.br	5048	782	21,1%
www.priberam.pt	5048	888	24,3%
www.tiosam.org	5048	138	3,4%

Foram retornados 1443 hiperonímias dos dicionários *online* (Retornos **Verdadeiro Positivo**). Uma conferência manual nestes resultados identificou 202 retornos errados e 1241 retorno corretos, o que corresponde a **uma taxa de acerto de 86,0%.**

5.3.2 Conclusões da utilização da folksonomia para enriquecer a descrição do produto.

A recuperação de informações a partir da *Web* apresenta perspectivas promissoras para o enriquecimento de dados. O aprendizado de máquina executado pelo agente, implementado pelo processo computacional (RUSSEL e NORVIG, 2009), foi capaz de se adaptar às particularidades de cada fonte de informação tornando possível que um processo semiautomático possa analisar e adquirir a capacidade de extrair informações de um determinado *site*. O esforço do especialista em domínio para treiná-lo e avaliá-lo pode ser minimizado com a automação do "*Ciclo de aprendizado de máquina assistido por especialista em domínio*", pois este ciclo poderá ser evoluído no futuro para ser feito automaticamente por *software*, pois pode-se identificar quais exemplos deram origem a cada macro-expressão e fazer a avaliação destas macro-expressões de forma automática, ajustando-se o estado de cada exemplo envolvido conforme os percentuais de precisão e revocação da respectiva macro-expressão, este ciclo poderá ser executado diversas vezes sem intervenção humana de forma a maximizar a precisão e a revocação do conjunto de macro-expressões de cada dicionário *online* . Cabendo ao especialista apenas informar o resultado correto de cada exemplo.

Vale ressaltar que as macro-expressões podem ser utilizadas em outros línguas, bastando que os exemplos fornecidos sejam da mesma língua dos dicionários *online* utilizados. O *Thesauro* utilizado para verificar a classe gramatical das palavras também deve ser específico da língua utilizada.

As macro-expressões possuem um grande potencial para a localização de informações em bases textuais, desde que sejam geradas e avaliadas de forma cuidadosa pelo especialista em domínio para poderem lidar com as características textuais particulares de cada dicionário *online*.

5.4 Classificar e Organizar em Taxonomia

Com as macro-expressões de cada dicionário *online* geradas, pode-se utilizar as informações oriundas dos dicionários *online* para recuperar as hiperonímias dos termos representativos de cada produto. Estas hiperonímias foram utilizadas para a construção do grafo direcionado (dígrafo), o qual será refinado para que se transforme em uma hierarquia dos produtos.

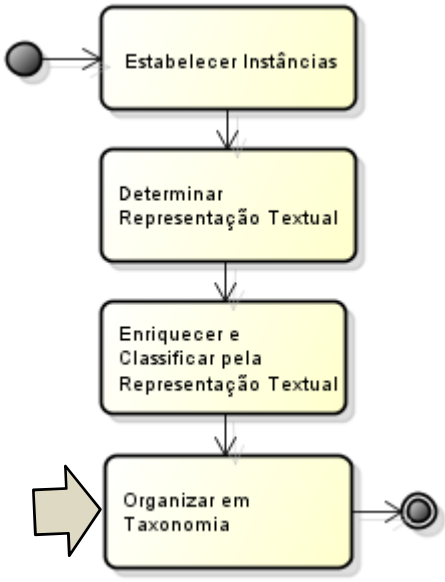


Figura 56: Etapa 5.4

A construção da taxonomia deverá tratar diversas ambiguidades e problemas oriundos do enriquecimento executado na etapa anterior. A seguir está a lista de problemas a serem tratados:

- **Atalhos.** São relações entre termos de níveis não adjacentes.
- **Ciclos.** São formados por uma sequencia de relações que retornam ao ponto inicial.
- **Ruído nos dados oriundos da *Web*.** São informações errôneas de relações classificadoras recuperadas na *Web* (TANG *et al* 2009) (PLANGPRASOPCHOK *et al* 2010), oriundas de textos obtidos dos dicionários *online* que contém algum padrão das macro-expressões, mas não contém o termo classificador na posição esperada.

Desta forma a construção da taxonomia, conforme descrito na etapa 4.6.1 foi feita em dois passos:

1. **Geração do grafo (dígrafo) formado pelos termos classificadores** dos produtos e seus respectivos termos generalizadores, oriundos a folksonomia.
2. **Transformação do grafo (dígrafo) em uma estrutura hierárquica**, com a resolução dos problemas apresentados anteriormente e com a seleção de um único termo generalizador para cada termo identificador de produto.

5.4.1 Geração do grafo formado pelos termos classificadores

O dígrafo foi construído pelo método *bottom-up* (METZ e MONARD, 2006), com a inclusão dos elementos do nível mais baixo, que seriam os produtos e os termos classificadores destes, formado pelo conjunto de termos originário do primeiro termo representativo de cada produto, conforme descrito na etapa 4.6.2. O processo de seleção dos termos representativos de produto (etapa 4.4) prioriza os termos mais frequentes que existam no *Thesauro* utilizado, consequentemente, o processo de montagem do dígrafo utilizará as relações (termos) que atingirem um *Threshold* mínimo de confiabilidade (GRUBER, 1993).

Conforme apresentado a seguir na Figura 57, o subgrafo resultante do termo PEPINO:

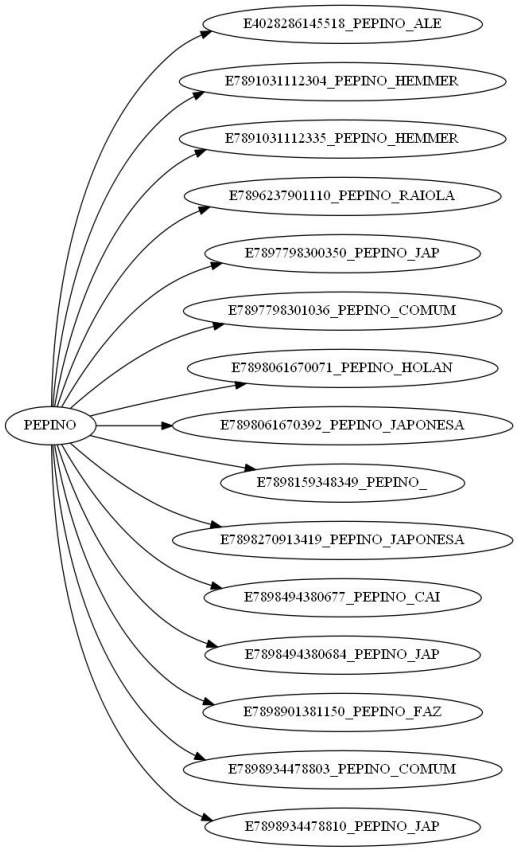


Figura 57: Subgrafo gerado pela classificação do termo PEPINO, no nível 1.

A partir do nível formado pelo conjunto dos primeiros termos de cada produto, é feita a busca na *Web* da relação hiperonímia (classificadora) de cada elemento deste nível, resultando no segundo nível deste grafo, conforme se observa no subgrafo no qual o elemento PEIXE é a raiz, apresentado na Figura 58:

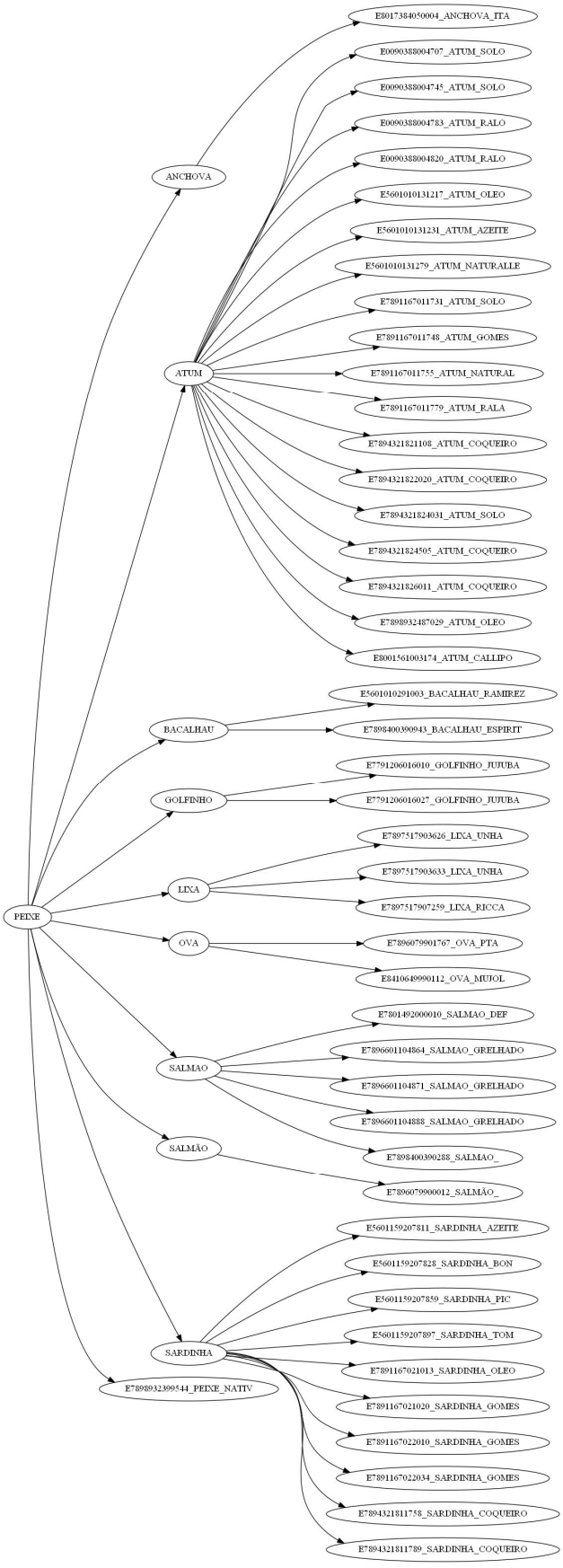


Figura 58 : Subgrafo gerado pela classificação do termo PEIXE, no nível 2.

O dígrafo montado a partir dos termos generalizadores obtidos dos dicionários *online* possui algumas ambiguidades devido à existência de diversas generalizações de determinados termos. A Figura 59 apresentada a seguir apresenta uma parte do dígrafo gerado a partir da classificação dos produtos cujo prefixo de EAN é igual a 78977983:

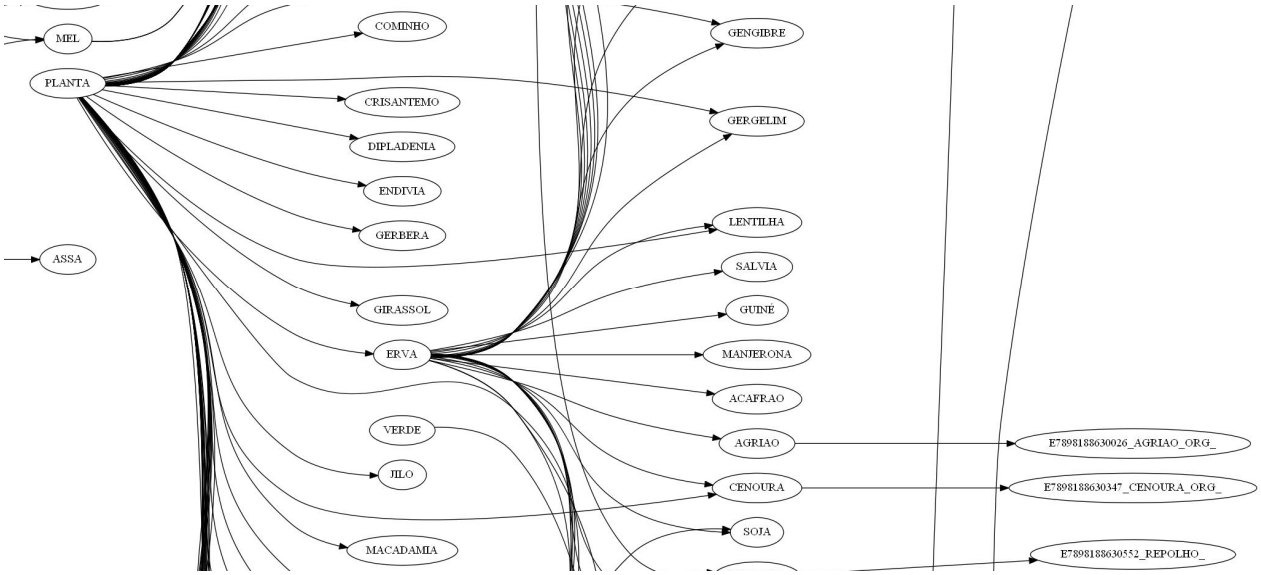


Figura 59 : Subgrafo gerado a partir dos termos de prefixo 78977983.

Pode-se observar na Figura 59 que o dígrafo apresenta algumas ambiguidades como a presença de dois termos classificadores (PLANTA e ERVA) para os termos LENTILHA, GERGELIM, GENGIBRE e CENOURA. Estas ambiguidades formam **ciclos** nos quais as relações de classificação do termo PLANTA com esses termos podem ser considerados **atalhos**, já que existe uma relação indireta por meio do termo ERVA e uma relação direta por meio do termo PLANTA. Estas ambiguidades estão resolvidas na próxima etapa.

5.4.2 Transformação do grafo (dígrafo) em uma estrutura hierárquica

Pode-se observar que existem ambiguidades no dígrafo gerado, Durante o processo de montagem da taxonomia deverão surgir ambiguidades que deverão ser resolvidas pelos algoritmos 07, 08 e 09 propostos na etapa 4.6. A Figura 60 apresentada a seguir mostra a taxonomia dos produtos do prefixo de EAN 78981886. Pode-se observar que as ambiguidades na classificação (tanto as ambiguidades na classificação como os atalhos também) dos termos LENTILHA, GERGELIM, GENGIBRE e CENOURA foram removidas pelos processos propostos na etapa 4.6.

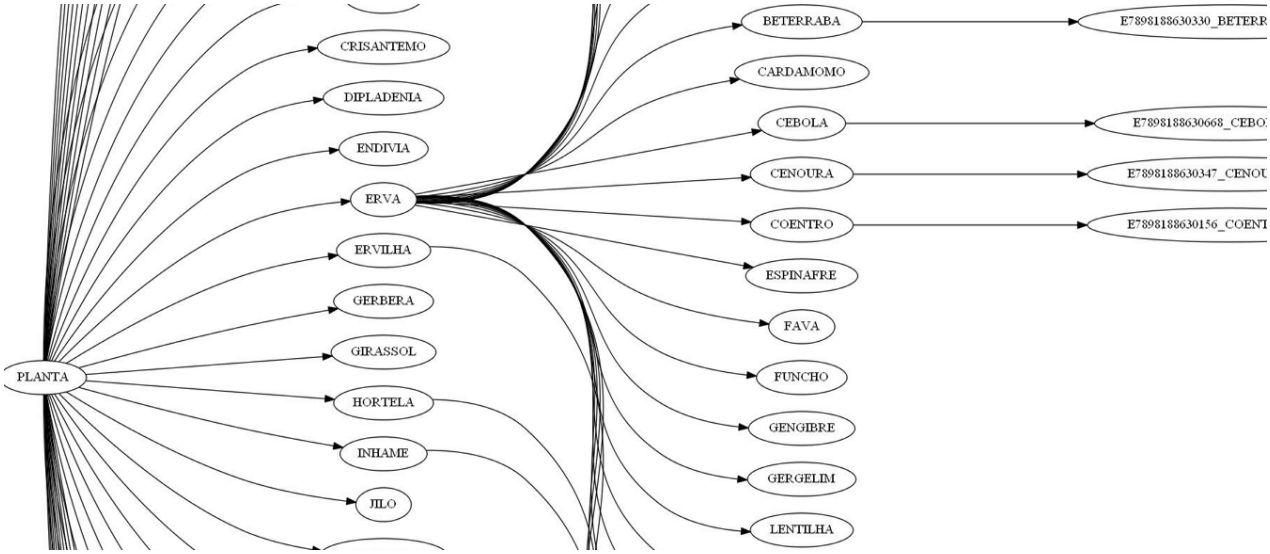


Figura 60 : Taxonomia final gerada a partir dos termos de prefixo 78977983.

5.4.3 Avaliação das taxonomias geradas.

Um ponto a ser ressaltado é que a qualidade da taxonomia gerada depende da qualidade das relações identificadas na etapa anterior (recuperação das hiperonímias dos dicionários *online*). Determinadas relações identificadas levaram a presença de determinados termos erroneamente como considerados como classificadores, como por exemplo LIXA apresentada anteriormente na Figura 58.

A Figura 61 apresenta mais uma amostra do grafo com os dados do prefixo de EAN. O termo HUMOR e MASSA foi erroneamente colocado como dependente do termo LIQUIDO. O termo FUBÁ foi classificado como BEBIDA, por um erro na localização da hiperonímia no texto do dicionário *online*.

Figura 61 : Taxonomia final gerada a partir dos termos de prefixo 78989047.

Determinados termos oriundos dos produtos foram erroneamente interpretados pelos dicionários *online*. Por exemplo: no produto de EAN 7898927561048 com os termos ESPETO e CORDEIRO, o termo ESPETO foi interpretado como um instrumento, o que ocasionou na classificação do referido produto com INSTRUMENTO, conforme apresentado na Figura 62.

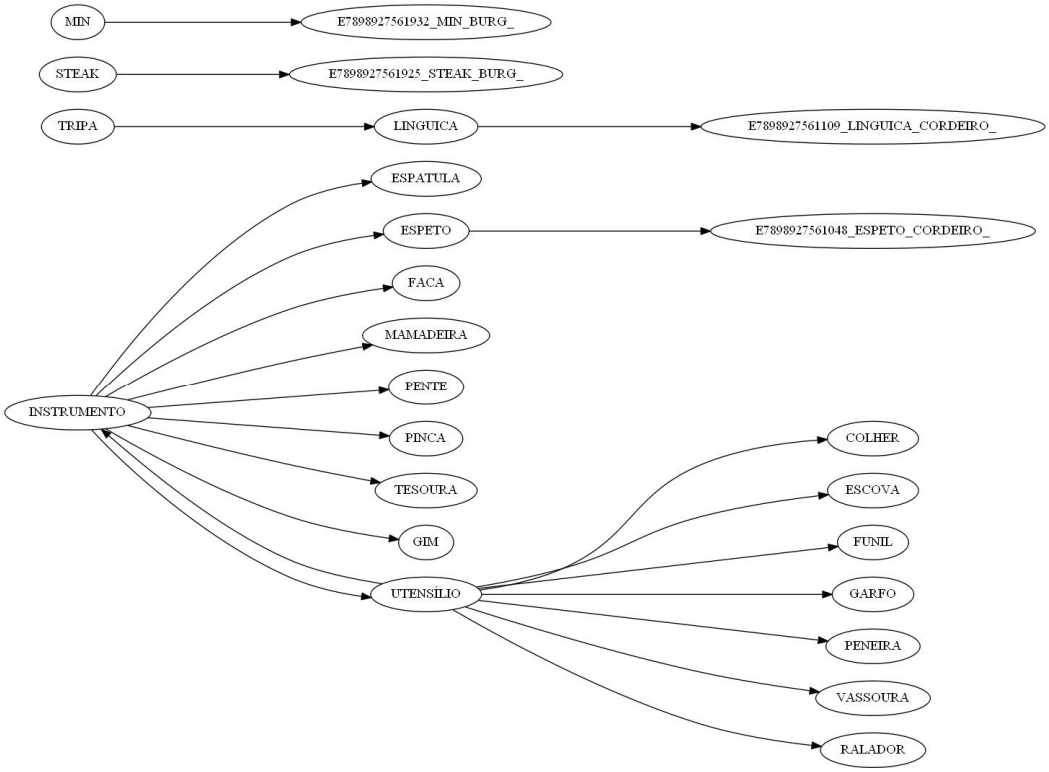


Figura 62 : Taxonomia final gerada a partir dos termos de prefixo 78989275.

Apesar de existirem problemas pontuais em alguns resultados, o processo construiu com sucesso diversas estruturas hierárquicas consistentes, como podemos verificar na Figura 63 os termos classificados pelo termo MASSA:

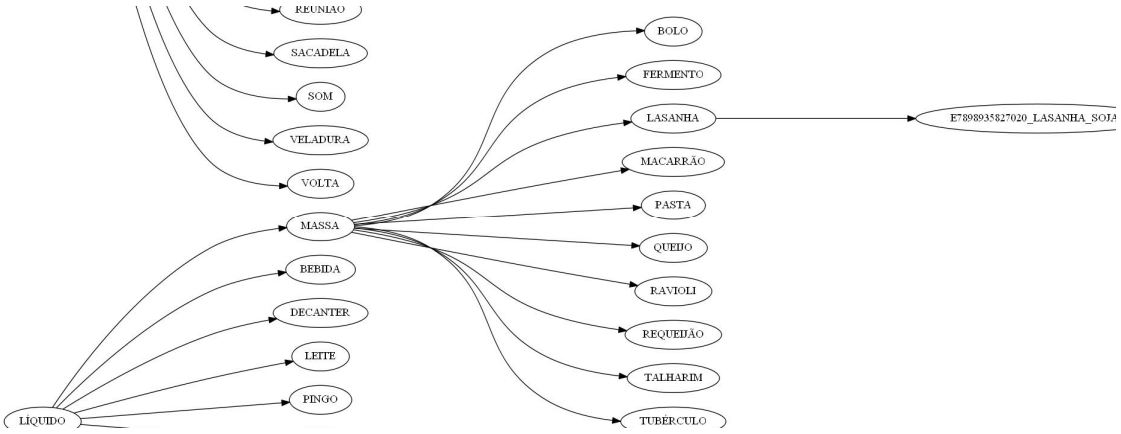


Figura 63 : Taxonomia final gerada a partir dos termos de prefixo 78989358.

5.4.4 Avaliação quantitativa da taxonomia gerada.

Foi gerada uma taxonomia com 5115 termos e 1447 relações. Não sendo possível incluir o gráfico representativo desta por inteiro neste trabalho, pois seria necessário utilizar uma escala que não permitiria a visualização das informações. A seguir, na Tabela 28 estão informações quantitativas sobre esta taxonomia gerada:

Tabela 28: Detalhes da taxonomia gerada.

Nível	Quantidade de Termos	Percentual de Termos com Classificador	Percentual de Termos que existem no *Thesauro*
9	1	0,0%	100,0%
8	1	100,0%	100,0%
7	1	100,0%	100,0%
6	3	33,3%	100,0%
5	11	27,3%	100,0%
4	26	53,8%	100,0%
3	100	42,0%	100,0%
2	252	73,0%	100,0%
1	4720	13,1%	42,9%

Pelas informações apresentadas na Tabela 28, observa-se que a taxonomia gerada, apesar de ter nove níveis de hierarquia, tem a maioria dos seus termos posicionados nos três níveis iniciais.

Dos 9493 produtos a serem classificados, 8944 produtos tiveram termos representativos selecionados, dentre estes, 5691 produtos foram associados a termos que possuíam classificadores e 3253 produtos foram associados a termos que não possuíam classificadores.

No conjunto de termos associados aos produtos que não possuem classificadores, observou-se que 74,1% destes termos não existem no *Thesauro*. Este conjunto de termos, que não existe no *Thesauro*, pode ser formado por abreviações de palavras ou nomes de marca, não sendo possível a recuperação de informações sobre estes tipos de palavras com a utilização dos dicionários *online*.

A Figura 62 apresenta um gráfico representativo da efetividade do *framework*. **Observa-se que 60% dos produtos tiveram seus respectivos termos representativos selecionados e classificados pelo *framework*.** Um trabalho semelhante é o de Ashwin Ittoo e Gosse Bouma (ITTOO, BOUMA e GOSSE, 2013) que identificou relações no texto da *Wikipedia*, atingindo uma taxa de revocação de 63% sem lidar com palavras truncadas e abreviadas. Desta forma, podemos afirmar que o processo foi efetivo na criação de uma taxonomia capaz de classificar os produtos, atingindo os seus objetivos.

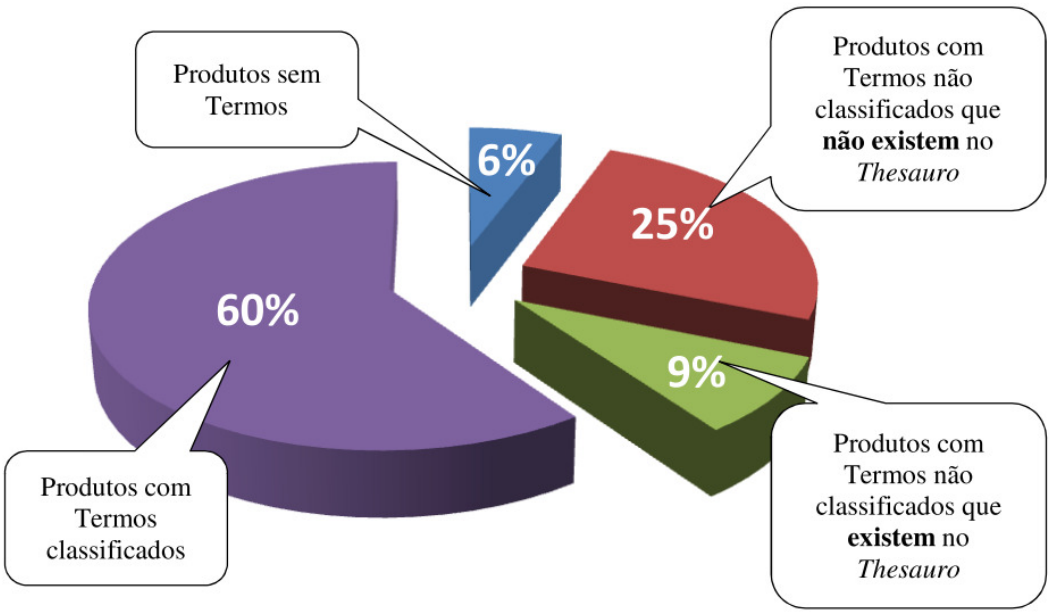


Figura 64 : Comparativo da efetividade do framework na classificação dos produtos.

Conforme observado na etapa 5.3.1, os dicionários *online* retornaram 1447 hiperonímias, sendo 202 hiperonímias erradas, identificadas por uma verificação manual. Destas 202 hiperonímias erradas, 92 foram descartadas pelos Algoritmos de remoção de ambiguidades 08 e 09 durante a

construção da taxonomia, atingindo um percentual de 7,6% de hiperonímias erradas na taxonomia final. Podemos concluir que é um percentual de erro baixo comparando com o trabalho de (CANTADOR *et al*, 2011), que apresentou um percentual de 25,0 % de erros na categorização de *tags*, o que evidencia que o *framework* desenvolvido neste trabalho apresenta bons resultados qualitativos também. **Diante da taxa de 7,6% de hiperonímias erradas, concluímos que o processo gerou uma taxonomia com 92,4% de relações corretas, o que evidencia a efetividade qualitativa do *framework*.**

5.4.5 Conclusões do processo gerador de taxonomias

O processo formado pelos algoritmos 05, 06, 07, 08 e 09 foi capaz de gerar com sucesso uma taxonomia dos produtos, tratando as seguintes ambiguidades:

- **Atalhos.** Foram removidos com a aplicação do Algoritmo 08.
- **Relacionamentos Recíprocos**. Foram removidos com a aplicação do Algoritmo 07.
- **Ciclos.** Foram removidos com a aplicação do Algoritmo 08.
- **Ruído nos dados oriundos da *Web*.** Os ruídos oriundos da *Web* e os erros oriundos da etapa de definição da representação textual dos produtos foram minimizados nesta etapa pelo fato de apresentarem poucas relações com os outros termos, sendo parcialmente removidos pelos algoritmos de desambiguação. Outro fato que contribuiu para a minimização deste ruído foi a utilização do *Thesauro*, na qual os termos localizados na etapa 4.5 tinham que existir no *Thesauro* e terem a classificação gramatical de NOME ou SUBSTANTIVO.

Concluindo, o processo apresenta uma contribuição nas técnicas atuais de construção de taxonomias a partir de descrições textuais, apresentando técnicas para tratar as ambiguidades e inconsistências, classificar e construir uma taxonomia coerente capaz de classificar as instâncias identificadas.

6 CONCLUSÕES

Este trabalho apresentou um *framework* que, por meio da recuperação de informações da *Web*, foi capaz de refinar, enriquecer, classificar e construir uma taxonomia consistente a partir de informações textuais incompletas e conflitantes oriundas de bases de dados, enriquecidas com informações recuperadas da *Web*.

O *framework* implementado está representado pela Figura 65, apresentada a seguir:

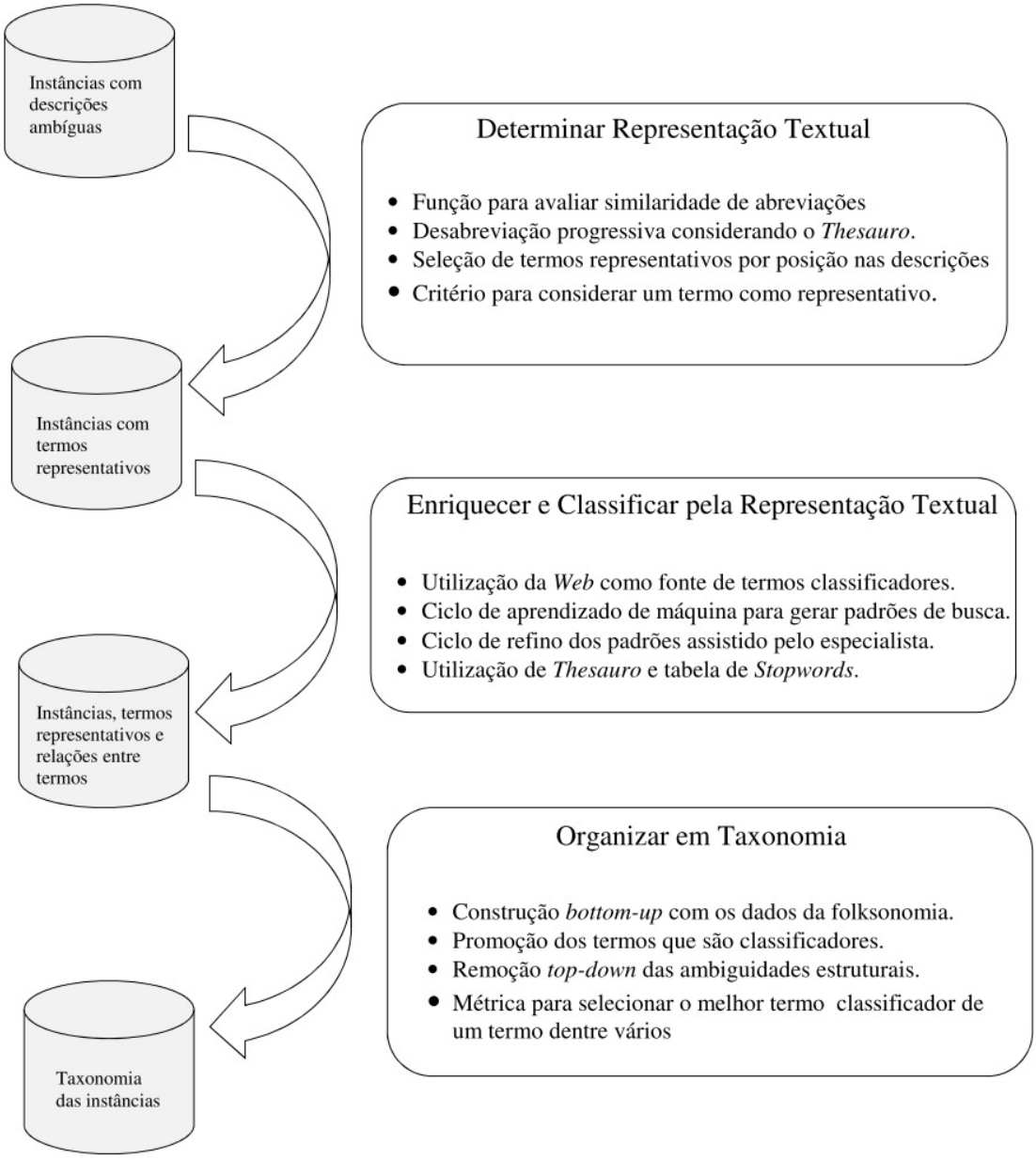


Figura 65 : Processos principais do *framework* e suas contribuições.

6.1 Contribuições

Desenvolvimento de um *framework* que, a partir de dados conflitantes e incompletos, foi capaz de gerar uma taxonomia classificadora destes dados, representa uma contribuição na área de geração de taxonomias com a utilização de informações oriundas da *Web*.

As principais contribuições deste trabalho são:

Uma técnica capaz de recuperar as relações hiponímias de termos na *Web*. O qual que utiliza o aprendizado de máquina para localizar as relações desejadas no texto retornado pelos dicionários *online*. Esta técnica poderá ser utilizada para localizar outras relações no texto recuperado da *Web*, conforme os exemplos fornecidos pelo especialista e também poderá recuperar relações em outras línguas, conforme sejam utilizados dicionários *online* e respectivos exemplos de treinamento escritos em outra língua.

Uma técnica de tratamento de ambiguidades nas relações para a construção de taxonomias. Os Algoritmos 05, 06, 07, 08 e 09 foram capazes de construir uma taxonomia consistente e enxuta a partir de relações ambíguas recuperadas na *Web*. Podendo ser utilizados para a construção de taxonomias classificadores a partir de um conjunto de relações entre os objetos a serem classificados.

Técnicas capazes de definir uma descrição representativa única a partir de um conjunto de descrições ambíguas e abreviadas de objetos. As técnicas utilizadas na etapa 4.4 foram capazes de definir uma descrição única para cada objeto identificado. Estas técnicas podem ser utilizadas no contexto atual, no qual a *Web* fornece diversas informações diferentes sobre o mesmo objeto, permitindo a consolidação das descrições abreviadas, conflitantes e ambíguas em uma única descrição representativa do objeto em questão.

Cada etapa do framework apresenta as seguintes contribuições:

6.1.1 Etapa: Determinar Representação Textual

Nesta etapa, o desenvolvimento de um processo que define uma representação textual única para uma instância a partir de um conjunto conflitante, redundante de descrições, muitas delas

compostas por termos abreviados, representa uma contribuição aos processos atuais de desambiguação e desabreviação. As técnicas apresentadas na etapa 4.4 foram efetivas para a determinação da uma descrição textual única de cada instância, efetuando a desabreviação e desambiguação dos termos das descrições originais e determinando uma sequência de termos representativa para cada instância. A seguir estão as contribuições desta etapa:

Função para avaliar similaridade de abreviações. A função de similaridade abreviada proposta na etapa 4.4.2 permitiu a associação de abreviações nas diversas descrições com suas respectivas palavras originais.

Desabreviação progressiva considerando o *Thesauro*. O processo de desabreviação somente foi efetivo com a utilização do *Thesauro* para determinar que um termo não necessita mais ser desabreviado, pois existe no *Thesauro*. A utilização do *Lemma* dos termos removeu a ambiguidade singular/plural dos termos selecionados.

Seleção de termos representativos por posição nas descrições. Diferentes termos posicionados na mesma ordem em descrições diferentes tem grande possibilidade de serem sinônimos ou possuírem o mesmo significado. Neste caso, a prioridade para se selecionar os termos existentes no *Thesauro* (substantivos) levou a seleção de termos identificadores do produto consistentes em detrimento de nomes de marca ou nomes próprios, os quais poderiam prejudicar as próximas etapas.

Critério para considerar um termo como representativo. Os *Thresholds* utilizados neste processo foram efetivos na remoção de termos que poderiam não ser tão representativos da respectiva instância, de forma que os termos selecionados são realmente identificadores de cada instância e podem ser utilizados na construção da taxonomia dos produtos.

6.1.2 Etapa: Enriquecer e Classificar pela Representação Textual

Nesta etapa, o desenvolvimento de técnicas de localização de informações sobre textos, oriundos de diferentes fontes na *Web*, que não estão escritos conforme as regras gramaticais convencionais, representa um avanço na área de folksonomia e recuperação das informações na *Web*. A seguir estão as conclusões desta etapa:

Utilização da *Web* como fonte de termos classificadores. A utilização de dicionários *online* para a recuperação das relações de classificação dos termos apresenta uma possibilidade

promissora no uso da *Web* para a recuperação de informações específicas sobre termos ou palavras.

Ciclo de aprendizado de máquina para gerar padrões de busca. A utilização de aprendizado de máquina na geração dos padrões textuais localizadores de relações permitiu que fossem utilizados como fontes de informação textos que não atendiam aos padrões gramaticais convencionais. Desta forma, o aprendizado de máquina apresenta a possibilidade de ser recuperar informação a partir de *sites* cujo texto não atende aos padrões gramaticais convencionais.

Ciclo de refino dos padrões assistido pelo especialista. O papel desempenhado pelo especialista, avaliando os resultados e ajustando os exemplos permite que os padrões gerados sejam refinados de forma a maximizar a sua capacidade de localização.

Utilização de *Thesauro* e tabela de *Stopwords*. A utilização do *Thesauro* (seleção de substantivos e *lemma*) e da tabela de *Stopwords* foi essencial para filtrar termos sem significado na busca pelos termos classificadores no texto recuperado da *Web*.

6.1.3 Etapa: Organizar em Taxonomia

Nesta etapa, o desenvolvimento de um modelo para construção de taxonomias a partir de informações ambíguas, capaz de tratar as ambiguidades e redundâncias estruturais representa uma contribuição na área, a seguir estão as conclusões desta etapa:

Construção *bottom-up* com os dados da folksonomia. O processo de construção *bottom-up* da taxonomia foi efetivo devido ao fato dos dados originais, organizados em instâncias, corresponderem ao nível básico da taxonomia. Os termos classificadores, recuperados das relações oriundas da *Web*, compõem os níveis superiores da taxonomia.

Promoção dos termos que são classificadores. O processo que promove os termos classificadores, subindo-os de nível conforme estes classificam termos em determinados níveis leva ao posicionamento dos ciclos (do grafo rascunho da taxonomia) nos níveis superiores. A utilização de um nível limite para a promoção de termos previne o processo de *loop* sem fim de promoção de termos que participam de um ciclo, colocando os termos dos ciclos posicionados no último nível.

Remoção *top-down* das ambiguidades estruturais. Como os termos pertencentes aos ciclos (ambiguidades estruturais) foram posicionados no último nível, o processo de remoção das ambiguidades estruturais foi aplicado no sentido de cima para baixo (*top-down*) de forma a priorizar a remoção dos atalhos que compõem os ciclos. Depois das remoções dos atalhos é feita a resolução

das ambiguidades específicas de cada termo, neste caso, múltiplos termos classificadores sobre um único termo (oriunda da utilização de vários dicionários *online* na busca das relações).

Métrica para selecionar o melhor termo classificador de um termo dentre vários termos classificadores. O FC-Fator de Classificação, proposto na etapa 4.6.4.1, mostrou-se efetivo na escolha do melhor termo classificador dentre vários termos classificadores fornecidos pelos dicionários *online*.

6.2 Trabalhos Futuros

Este *framework* é composto de diversas técnicas e processos que criaram uma taxonomia consistente a partir de descrições textuais ambíguas e abreviadas de instâncias. A efetividade do *framework* pode ser comprovada nos resultados apresentados no capítulo anterior, mas não significa que as técnicas e processos utilizados não possam ser evoluídos. A seguir estão diversas propostas de evolução destes processos e técnicas:

Uma evolução deste trabalho será utilizar o processo de busca das relações sobre dicionários *online* de outras línguas, com a utilização do *Thesauro* adequado à língua do texto escrito nestes dicionários. Provavelmente o processo de aprendizado de máquina e as macro-expressões correspondentes funcionarão com sucesso, localizando os padrões das relações apresentadas nos exemplos de treinamento.

Outras evoluções deste trabalho consistiriam em aperfeiçoamentos dos processos dentro do ciclo de aprendizado de máquina assistido por especialista em domínio, apresentado na etapa 4.5.2. Após a geração das macro-expressões a partir dos exemplos (*Training Positive* e *Training Negative*) e ciclo de aprendizado de máquina, as macro-expressões geradas são testadas sobre os exemplos de avaliação (*Evaluation Positive* e *Evaluation Negative*), sendo avaliadas pelo especialista em domínio. O aperfeiçoamento consistiria em três atividades processadas e avaliadas de forma cíclica após os testes sobre os exemplos de avaliação:

1. A identificação dos casos de localizações erradas, as respectivas macro-expressões que falharam e os respectivos exemplos de treinamento (*Training Positive* e *Training Negative*) que geraram estas macro-expressões. O status destes exemplos de treinamento seria ajustado de *Trainng* para *Evaluation* caso apresentem similaridade textual com os casos de localizações erradas, de forma a minimizar a ocorrência destas localizações erradas.
2. A ordem de utilização das macro-expressões sobre o texto retornado pelos dicionários *online* também poderia ter um processo de aprendizado de máquina, que testaria a melhor ordem de aplicação destas para se localizar a informação desejada no texto com base no resultado da localização dos dois conjuntos (*Evaluation Positive* e *Evaluation Negative*).
3. A identificação dos casos de não localização. Neste caso, seriam processados e avaliados os exemplos de avaliação (*Evaluation Positive* e *Evaluation Negative*) que apresentarem resultados negativos (coluna *Err* da Figura 42) e determinados

exemplos deste conjunto teriam seu respectivo status ajustado de *Evaluation* para *Training* caso tenham padrão textual comum. Outra mudança seria sobre os exemplos de avaliação que apresentarem localizações erradas, neste caso seriam verificados o conjunto de exemplos (*Training Positive* e *Training Negative*).

É importante ressaltar que os pontos de configuração gerados pelos processos automáticos propostos acima seriam armazenados em memória, de forma que o algoritmo "aprenda" que determinadas mudanças degradam a capacidade de localização e outras mudanças melhoram esta capacidade. Desta forma o algoritmo poderia direcionar as mudanças para otimizar a capacidade de localização do processo como um todo e identificar o ponto em que se atingiu a melhor capacidade de localização com os exemplos fornecidos pelo especialista em domínio.

Outra evolução deste trabalho seria a construção de uma taxonomia-base a partir dos termos que definem as alíquotas de imposto (ICMS), apresentada no Apêndice A. Esta taxonomia-base seria utilizada para a construção da taxonomia de produtos. Nesta proposta, será necessária uma métrica de similaridade semântica de termos do produto com os termos da taxonomia base (poderia ser a quantidade de nós do grafo presente no caminho entre os termos ou o nível (grau) do termo classificador comum aos dois termos). Esta métrica seria utilizada para se definir qual a provável alíquota de um determinado produto em um grafo de similaridade entre os termos. Seriam necessárias mudanças nos exemplos das macro-expressões, de forma que busquem termos sinônimos ou similares do termo pesquisado nos textos retornados pelos dicionários *online* para que seja construído este grafo de similaridade.

A utilização de dicionários especializados também poderia resultar em evoluções neste trabalho. Por exemplo: a utilização de um dicionário especializado em serviços ou vinhos. Seriam construídas taxonomias especializadas e identificados os produtos associados a estas taxonomias.

Outra evolução deste trabalho seria a identificação do ramo de atividade de um prefixo de EAN (código do fabricante do produto). A frequência dos termos representativos de cada produto pertencente a esta faixa de códigos de produtos (prefixo de EAN) seria verificada e seriam selecionados termos candidatos a identificarem o fabricante do produto. Seria verificada na taxonomia construída neste trabalho a estrutura de classificação dos produtos pertencentes a este prefixo de EAN, podendo-se associar o conjunto de termos identificadores do fabricante com o conjunto de termos classificadores dos respectivos produtos, de forma que seja possível identificar o ramo de atividade e o nome do fabricante.

7 REFERÊNCIAS

Aliquotas Internas do ICMS. Disponível em http://jlassessoriaempresarial.com.br/_indicadores/JL%20Assessoria%20Empresarial%20-%20Al%C3%ADquotas%20Internas%20do%20ICMS.pdf Acesso em 2014-08-14.

ANDRADE, S.F.R.; NETO, M.G.M. **Uma abordagem de modelagem multidimensional para data mart de compras públicas, usando taxonomia.** Third Workshop on Ontologies and Metamodeling in Software and Data Engineering - WOMSDE 2008 http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=7605 Acesso em 12/11/2013.

ANGELETOU, Sonia. SABOU, Marta. MOTTA, Enrico. **Semantically Enriching Folksonomies with FLOR**. 1° International Workshop on Collective Semantics: Collective Intelligence & the Semantic Web (CISWeb 2008).

ARANHA, Christian, Nunes. **Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência Computacional**. Tese D.Sc. Puc-Rio 2007.

ITTOO, Ashwin. BOUMA, Gosse. **Minimally-supervised learning of domain-specific causal relations using an open-domain corpus as knowledge base**. Data & Knowledge Engineering 88, 142-163 http://www.sciencedirect.com/science/article/pii/S0169023X12000638
Acessado em 30/09/2013

BACCAR, Feten Amar. GARGOURI, Bilel. HAMADOU, Abdelmajid Ben. **Towards Generation of Domain Ontology from LMF Standardized Dictionaries**. The International Conference on Intelligent Semantic Web Services and Applications (ISWSA 2010)

BALAKRISHNA, Mithun. MOLDOVAN Dan. TATU, Marta. OLTEANU, Marian. **Semi-Automatic Domain Ontology Creation from Text Resources**. LREC 2010, 7th Language Resources and Evaluation Conference, Malta, Maio 2010.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Editora Campus, 2a edição, 2007. ISBN 85-352-169602.

BILENKO, Mikhail; MOONEY Raymond; COHEN, William; RAVIKUMAR Pradeep; and FIENBERG Stephen. **Adaptive Name Matching in Information Integration**. Journal IEEE Intelligent Systems archive Volume 18 Issue 5, September 2003 Page 16-23 http://dl.acm.org/citation.cfm?id=1137369 Acesso em 16/08/2012.

BISHOP, Christopher. **Pattern Recognition and Machine Learning**. Information Science and Statistics 2006. ISBN 978-0-387-31073-2 http://www.springer.com/us/book/9780387310732 Acesso em 25/10/2013.

BORTH 2011. Marcelo Rafael. **Uma abordagem de recomendação de tags semânticas para sistemas baseados em tagging.** Dissertação M.Sc. UEM, Maringá 2011

BREITMAN Karin Koogan. HOWARD Carolina. CASANOVA, Marco A. **CATO - A Lightweight Ontology Alignment Tool.** CAiSE Short Paper Proceedings 2005.

BREUING, Alexa; WALTINGER, Ulli; WACHSMUTH, Ipke; **Harvesting Wikipedia Knowledge to Identify Topics in Ongoing Natural Language Dialogs**. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on Volume: 1, Agosto de 2011. Páginas 445 - 450. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6040710 Acesso em 30/10/2012

CANTADOR, Iván; KONSTASB, Ioannis; JOSE, Joemon M.; **Categorising social tags to improve folksonomy-based recommendations.** Web Semantics: Science, Services and Agents on the World Wide Web Volume 9, Issue 1, March 2011, Pages 1-15. http://www.sciencedirect.com/science/article/pii/S1570826810000685 Acesso em 17/10/2013.

CARPINETO, Claudio. ROMANO, Giovanni. **GALOIS: An order-theoretic approach to conceptual clustering.** Proceedings of 10th International Conference on Machine Learning, Amherst. June 1993, pp. 33-40.

CHANDRASEKARAN, B.; JOSEPHSON, R.; BENJAMINS, V. R. **What are Ontologies, and why do we need then?** IEEE Inteligent Systems, v. 14, n. 1, p. 20-25, jan.1999.

CHAPMAN, Sam. **String Similarity Metrics for Information Integration**. Sheffield, UK, 2005. http://www.dcs.shef.ac.uk/~sam/stringmetrics.html Acesso em 16/08/2012

CHEN, David M. QIN, Jian. **Deriving ontology from folksonomy and controlled vocabulary.** Conference 2008, University of California Los Angeles, February 28-29, 2008.

CONFAZ. A**RRECADAÇÃO DO ICMS - VALORES CORRENTES – 2010.** <http://www.fazenda.gov.br/confaz/boletim/Valores.asp> Acesso em 14/08/2012.

COMPUTERWORLD, 17 de Junho de 2013. **Efficiency will hold down storage growth, IDC says.** http://www.computerworld.com/article/2497852/data-center/efficiency-will-hold-down-storage-growth--idc-says.html. Acesso em 31/03/2015.

DOMINGUES, Marco Aurélio. REZENDE, Solange Oliveira. **Using Taxonomies to Facilitate the Analysis of the Association Rules**. ECML/PKDD'05 The Second International Workshop on Knowledge Discovery and Ontologies (KDO'05), 2011. http://arxiv.org/abs/1112.1734. Acesso em 06/05/2015.

DOTSIKA, Felfie. **Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies**. International Journal of Information Management 29, 2009. Pags 407–415. http://www.sciencedirect.com/science/article/pii/S0268401209000127 Acesso em 30/09/2013.

DOWLING, Geoff R; HALL, Patrick A. V. **Approximate String Matching**. England, Computing Surveys, Vol. 12, No. 4, December 1980. http://dl.acm.org/citation.cfm?id=356830 Acesso em 16/08/2012

ESCHENBACH, Carola. GRÜNINGER, Michael. **Formal Ontology in Information Systems**; Proceedings of the 15th International Conference - FOIS 2008.

FISCHER, Bern. **Specification-Based Browsing of Software Component.** IEEE/ACM International Conference on Automated Software Engineering, 1998.

FREEMAN, Linton C. WHITE, Douglas R. **Using Galois lattices to represent network data.** Sociological Methodolog, Pags 127-146. 1993.

GONDIM, Flavio Melo. **Algoritmo de Comparação de Strings para Integração de Esquemas de Dados.** Trabalho de Graduação em Ciencia da Computação. UFPE, 2006. http://www.cin.ufpe.br/~tg/2005-2/fmg.pdf . Acesso em 16/08/2012

GODIN, Robert. MILI, Hafedh. **Building and maintaining analysis-level class hierarchies using Galois lattices**. OOPSLA '93. ACM Sigplan Notices, 28, 10,394-410.

GOLDER, Scott A.; HUBERMAN, Bernardo A. **Usage patterns of collaborative tagging systems.** Journal of Information Science April 2006 vol. 32 no. 2 198-208
http://dl.acm.org/citation.cfm?id=1119747 Acesso em 22/05/2014

GOMES da Silva, Fenanda. SANT´ANNA, Simone. **A Semântica Lexical e as Relações de Sentido: Sinomínia, Antomínia, Hipomínia e hiperonímia**. Livro dos Minicursos. Cadernos do CNLF, Vol. XIII, Nº 03 34 Circulo Fluminense de Estudos Filosóficos e Linguísticos. Instituto de Letras da UERJ, 24 a 28 de agosto de 2009.
http://www.filologia.org.br/xiiicnlf/indice_textos_completos.htm Acesso em 02/09/2014.

GONZALEZ, Marco. LIMA, Vera L. S. **Recuperação de Informação e Processamento da Linguagem Natural.** PUC/RS 2003.

GRUBER. Thomas R. **A translation approach to portable ontology specifications**. Knowledge Systems Laboratory - Technical Report KSL 92-71 1993.

GUARINO. Nicola. **Formal Ontologies and Information Systems.** Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

HAN, Jiawei. KAMBER, Micheline. **Data Mining: Concepts and Techniques,** 2nd ed. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, March 2006. ISBN 1-55860-901-6

HOVY, Eduard; NAVIGLI, Roberto; PONZETTO, Simone Paolo. **Collaboratively built semi-structured content and Artificial Intelligence: The story so far.** Journal Artificial Intelligence archive, Volume 194, January, 2013, Páginas 2-27 http://dl.acm.org/citation.cfm?id=2405907 Acesso em 06/05/2014.

ITTOO, Ashwin; BOUMA, Gosse; **Minimally-supervised extraction of domain-specific part-whole relations using Wikipedia as knowledge-base.** Journal Data & Knowledge Engineering archive Volume 85, Maio de 2013 Páginas 57-79 http://dl.acm.org/citation.cfm?id=2452042 Acesso em 30/09/2013.

JOUSSELME, Anne-Laure. MAUPIN, Patrick. BOSSÉ, Éloi. **Uncertainty in a Situation Analysis Perspective**. Proceedings of the Sixth International Conference of Information Fusion, 2003.

JURAFSKY, Daniel. MARTIN, H, James. **Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition**. Prentice Hall, Englewood Cliffs, New Jersey, ISBN-13: 978-0131873216. 2nd Edition Hardcover 2008,

KIMBALL, Ralph. **The data warehouse lifecycle toolkit: expert methods for designing, developing data Warehouses**. New York: Wiley Computer Publishing, 1998.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de Metodologia Científica**. São Paulo: Atlas, 1993.

LANGIE, Leonardo Cavalheiro. LIMA, Vera Lúcia Strube. **Classificação hierárquica de documentos textuais digitais usando o algoritmo kNN**. In TIL'2003 - 1º Workshop em

Tecnologia da Informação e Linguagem Humana, evento associado ao XVI Brazilian Symposium on Computer Graphics and Image Processing - (SIBGRAPI) (São Paulo, SP, Brasil, 12 de Outubro de 2003).

LAU, Raymond Y.K.; CHUNG, Albert Y.K.; SONG, Dawei; HUANG, Qiang; **Toward a fuzzy domain ontology extraction method for adaptive e-learning.** Knowledge and Data Engineering, IEEE Transactions (Volume: 21, Issue: 6) Junho de 2009, Paginas: 800 – 813 http://cs.tju.edu.cn/faculty/dsong/papers/7x60l48368438371.pdf Acesso em 17/10/2013.

LMF. **Lexical Markup Framework** <http://www.lexicalmarkupframework.org/> LMF is the ISO standard for Natural Language Processing (NLP) lexicons and Machine Readable Dictionaries (MRD). The ISO code number for LMF is ISO-24613:2008

KONDRAK, Grzegorz. **Phonetic alignment and similarity**. Edmonton, Canada, 2003. Disponível em: http://www.cs.ualberta.ca/~kondrak/papers/chum.pdf Acesso em 16/08/2012

KOZAREVA, Zornitsa; HOVY, Eduard. **A Semi-Supervised Method to Learn and Construct Taxonomies using the Web.** Proceeding EMNLP '10 Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Páginas 1110-1118. http://dl.acm.org/citation.cfm?id=1870766 Acesso em 18/10/2013.

KRAUSE, Paul. CLARK, Dominic. **Representing Uncertain Knowledge: An Artificial Intelligence Approach**. Springer; 1° edition (1° de Dezembro de 1993)

MACQUEEN. James B. **Some methods for classification and analysis of multivariate observations**. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics (University of California. Press, 1967), 281-297.

MARTA, Tatu. MOLDOVAN. Dan. **Inducing Ontologies from from Folksonomies using Natural Language Understanding.** LREC 2010, Seventh International Conference on Language Resources and Evaluation. Mediterranean Conference Centre, Valletta, Malta, 2010. http://www.lrec-conf.org/proceedings/lrec2010/summaries/203.html, Acesso em 7/8/2012.

MARTINS, Camila Delefrate. **Construção semi-automática de taxonomías para generalização de regras de associação. Tese de Mestrado. USP.** WTDIA 2006 - Workshop de Teses e Dissertações em Inteligência Artificial. http://www.teses.usp.br/teses/disponiveis/55/55134/tde-06102006-104314/pt-br.php. Acesso em 29/09/2010.

METZ, Jean. MONARD, Maria Carolina. **Estudo e Análise das Diversas Representações e Estruturas de Dados Utilizadas nos Algoritmos de Clustering Hierárquico**, Relatórios Técnicos do ICMC - USP - ICMC, São Carlos, Brasil, 2006.

MINEAU, Guy. STUMME, Gerd. WILLE, Rudolf. **Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis**. 7th International Conference on Conceptual Structures, ICCS'99 Blacksburg, VA, USA, July 12–15, 1999 Proceedings.

MOLOSSI. Sinara. **Inserção da biblioteca digital de teses contexto web semântica construção e uso da ontologia**. Dissertação M.Sc. UFSC, 2008

MORESI, Eduardo TARAPANOFF, Kira. **O contexto Organizacional. Inteligência Organizacional e Competitiva**. Cap. Gestão da Informação e do Conhecimento. Brasília, Brasil, UnB.

NAVARRO, Gonzalo. **A guided tour to approximate string matching**. Journal ACM Computing Surveys (CSUR) Volume 33 Issue 1, March 2001. Páginas 31-88. http://dl.acm.org/citation.cfm?id=375365 Acesso em 16/08/2012.

NAVIGLI, Roberto; VELARDI, Paola; FARALLI, Stefano; **A graph-based algorithm for inducing lexical taxonomies from scratch.** Proceeding IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, 2004 - Volume 3 Páginas 1872-1877 http://www.aaai.org/ocs/index.php/IJCAI/IJCAI11/paper/viewFile/3345/3751 Acesso em 06/05/2014.

NAVIGLI, Roberto. **Word Sense Disambiguation: A Survey**. ACM Computing Surveys, Vol. 41, No. 2, Article 10, Publication date: February 2009.

NOY, Natalya F. MCGUINESS, Deborah L. **Ontology Developement 101: A Guide to Creating Your First Ontology**. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

OLIVEIRA. Taís da Veiga. **Adaptação do Algoritmo Levenshtein Distance para o Cálculo de Similaridade entre Frases.** Universidade Católica de Pelotas PPGINF-2009

OSIEK, B.A.; **Estração de acrônimos e seus significados com modelos ocultos de Markov**; Dissertação MSc. UFRJ/COPPE 2008.

OTHERO, Gabriel de Ávila. **A Gramática da frase em Português, algumas reflexões para a formalização da estrutura frasal em Português**. ISBN 978-85-7430-854-8, Editora EDIPUCRS, 2009.

PLANGPRASOPCHOK, Anon; LERMAN, Kristina; GETOOR, Lise. **Growing a Tree in the Forest: Constructing Folksonomies by Integrating Structured Metadata**. Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2008, http://arxiv.org/abs/1005.5114 . Acesso em 17/10/2013.

POPPER, Karl. **A logica da Pesquisa Cientifica**. Editora Cultrix São Paulo 1993.

PRISS. Uta Priss. **Formal Concept Analysis in Information Science.** Journal Annual Review of Information Science and Technology Volume 40 Issue 1, January 2007

PRISS. Uta. **Linguistic Applications of Formal Concept Analysis. In: Formal Concept Analysis - State of the Art.** Proceedings of the First international Conference on Formal Concept Analysis 2005. Berlin: Springer.

QUONIAM, Luc. **Los x- en biblioméFrie et dossier & travaux. Marseille: Université de Droit d'konomie et des Sciences**. d'Aix-Marseille 111, 1996.

RISTAD, Eric Sven; YANILOS, Peter N.. **Learning String Edit Distance**. Research Report CS-TR-532-96, October 1996. http://grfia.dlsi.ua.es/ml/algorithms/references/lsed_ristad.pdf Acesso em 16/08/2012.

RISTAD, Eric Sven. YIANILOSY,. Peter N. **Finite Growth Models and the Learning of Edit Distance Costs**. IEEE Transactions on Pattern Recognition and Machine Intelligence 1996.

RUSSEL, Stuart. NORVIG, Peter. **Artificial Intelligence A Modern Approach.** 3rd edition. ISBN-13: 978-0136042594. Pearson (December 11, 2009)

SCIENCE DAILY. Science Daily Magazine. **How Much Information Is There in the World?** http://www.sciencedaily.com/releases/2011/02/110210141219.htm Acesso em 14/09/2012.

SHAFFER, Clifford A. **Data Structures and Algorithm Analysis**. Department of Computer Science Virginia Tech. Edition 3.2 update 3.2.0.3. January 2, 2012. http://people.cs.vt.edu/shaffer/Book/Java3e20120102.pdf. Acesso em 24 de Abril de 2015.

SMITH, T.F. WATERMAN, M.S. **Identification of common molecular subsequences.** Journal of Molecular Biology Volume 147, Issue 1, 25 March 1981, Pages 195-197 http://www.sciencedirect.com/science/article/pii/0022283681900875 Acesso em 16/08/2012

SOFIA, Helena. MARTINS. João P. **Ontologies: How can They be Built?** Knowledge and Information Systems *(2004) 6: 441–464*

SOUZA, Renato Rocha. **Uma proposta de metodologia para escolha automática de descritores utilizando sintagmas nominais**. Tese DSc. UFMG. 2005 http://www.gercinalima.com/mhtx/pages/prototipo-btdeci/teses/souza-rr

STUMME, Gerd. **Formal concept analysis on its way from mathematics to computer science**. In ICCS, Lecture Notes in Computer Science, pages 2–19, London, UK, 2002. Springer-Verlag.

SUN, Aixin.LIM. Ee-Peng. **Hierarchical Text Classification and Evaluation**. Proceedings of

the 2001 IEEE International Conference on Data Mining.

TANG, Jie; LEUNG, Ho-fung; LUO, Qiong; CHEN, Dewei; GONG, Jibin. **Towards Ontology Learning from Folksonomies**. Proceedings of the 21st international jont conference on Artifical intelligence (2009). Páginas 2089-2094 http://dl.acm.org/citation.cfm?id=1661779 Acesso em 19/02/2014

LABEL-LEX . *Thesauro* com contendo aproximadamente um milhão de palavras inflexionadas, com os devidos *lemmas*, classificações gramaticais e atributos morfológicos Extraído em http://label.ist.utl.pt

TOMURO, Noriko. SHEPITSEN, Andriy. **Construction of Disambiguated Folksonomy Ontologies Using Wikipedia.** Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009, pages 42–50, Suntec, Singapore, 7 August 2009.

VANDER. Thomas Vander Wal. **Folksonomy**
<http://vanderwal.net/folksonomy.html> Acesso em 21/08/2012

WILLE, Rudolf. **Restructuring lattice theory: an approach based on hierarchies of concepts**. 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009

YULE, George**. The Study of Language (4th edition)**. Cambridge: University Press, 1998. 294 p.

APENDICE A

A1. Tabela de Alíquotas do ICMS por produto

Alíquota	Descrição do serviço/Produto
37%	a) arma e munição, suas partes e acessórios; b) perfume e cosmético; c) bebida alcoólica, exceto cerveja, chope e aguardente de cana e de melaço; d) peleteria e suas obras e peleteria artificial; e) embarcações de esporte e de recreio.
35%	cigarro, charuto, cigarrilha, fumo e artigo correlato.
30%	gasolina e álcool carburante.
25%	energia elétrica consumo acima de 300 quilowatts/hora mensais
18%	energia elétrica até o consumo de 300 quilowatts/hora mensais
17%	cerveja e chope. Aguardente
16%	refrigerante.
15%	Operação de importação querosene de aviação (QAV), Fund. Legal: Art. 14, XXV, § 4º da Lei nº 2.657/96
13%	Operação de importação quando a operação de importação for realizada através do Aeroporto Internacional Tom Jobim
12%	a) com arroz, feijão , pão e sal; b) com gado, ave e coelho, bem como os produtos comestíveis resultantes de sua matança, em estado natural, resfriado ou congelado; c) restaurantes, lanchonetes, bar, café e similares; d) óleo diesel; e) de fornecimento de energia elétrica para cooperativas de eletrificação, rural e sua distribuição para produtor rural, assim entendido aquele que mantenha exploração agrícola ou pastoril e esteja inscrito no Caderj; f) com máquinas, aparelhos, equipamentos e veículos destinados a implantação, ampliação e modernização de unidades industriais ou agroindustriais, e visem a incorporação de novas tecnologias, desconcentração industrial, defesa do meio ambiente, segurança e saúde do trabalhador e redução das disparidades regionais.
7%	material ou equipamento especializado para pessoas portadoras de deficiência física e medicamentos para os doentes renais crônicos e transplantados e produtos de informática e automação.
6%	Operação com energia elétrica quando utilizada no transporte público eletrificado de passageiros. Operações com óleo diesel, quando consumido no transporte de passageiros por ônibus urbano, hidroviário. Operações com Gás Natural Veicular – GNV quando consumido por empresa concessionária ou permissionária de transporte coletivo de passageiros por ônibus ou por veículo hidroviário

Fonte: http://www.idealsoftwares.com.br/tabelas/tabela.php?id=136

APENDICE B

B1. EAN (European Article Number)

O padrão EAN (*European Article Numbering*) e, na falta deste, admite-se a utilização de código próprio do estabelecimento usuário.

O código deve estar indicado em Tabela de Mercadorias e Serviços do PAF-ECF. No caso de utilização de código próprio, é vedada a reutilização de códigos. A codificação deve ser única para todos os estabelecimentos da empresa, conforme apresentado na Figura 66.

Figura 66 : Formatação do código EAN-13.

O padrão EAN (*European Article Numbering*) foi adotado a partir de 1976 em muitos países, exceto nos Estados Unidos e Canadá. Baseada na UPC12, porém capaz de identificar também o país de origem do produto.

O sistema EAN é um conjunto de padrões, que possibilita a identificação dos produtos, unidades logísticas e localizações. Ele facilita os processos de comércio eletrônico. Foi desenvolvido para fornecer soluções que garantam identificação exclusiva e sem ambiguidades. É um padrão internacional rígido onde cada produto terá seu código exclusivo, aplicável no mundo inteiro, sem repetição, o que possibilita a integração e a troca de informações entre os vários elos da cadeia produtiva, desde o fabricante até o consumidor final. Exceto Estados Unidos e Canadá, onde o responsável é o UCC (*Uniform Code Council*), o órgão controlador dos códigos é o EAN. Particularmente o EAN13 é formado por 13 dígitos: NNNEEEE PPPPP D, onde:

- NNN identifica o país (789 = Brasil);
- EEEE representa o prefixo EAN de empresa;

- PPPPP identifica o produto;
- D é dígito verificador.

APENDICE C

C.1 O Cupom Fiscal

A Secretaria de Fazenda do Estado do Rio de Janeiro (SEFAZ/RJ) é o órgão estadual destinado a gerir os recursos e executar o controle fiscal do governo estadual. O controle fiscal é executado pelos fiscais, que possuem a autoridade de execução das ações de fiscalização. Os contribuintes localizados no Estado do Rio de Janeiro interagem com a SEFAZ na maioria das vezes de forma eletrônica, seja para a emissão de certidões pelo portal da SEFAZ, seja emitindo declarações eletrônicas anuais (DECLANs) ou declarações mensais (GIAs). Diversos impostos são controlados pela SEFAZ/RJ, dentre eles o ICMS, IPVA e ITD. Todos os registros eletrônicos relativos ao pagamento destes impostos nos bancos são transmitidos a SEFAZ/RJ para serem consolidados para o acompanhamento das metas de arrecadação.

Os contribuintes que possuem equipamento de emissão de cupom fiscal (ECF) devem mensalmente transmitir eletronicamente para a SEFAZ os arquivos referentes aos cupons fiscal emitidos no mês anterior (ECF-MFD), referentes a vendas ao consumidor.

Como cada mercadoria vendida por meio de cupom fiscal dentro do estado do Rio de Janeiro obriga a geração de um registro de cupom fiscal, a base de dados de ECF-MFD da SEFAZ possui um crescimento de aproximadamente dois bilhões de registros por ano. Nessa base estão informações sobre as data de venda, valores de produtos vendidos, código, descrição e também a respectiva alíquota de imposto do produto vendido. O cupom fiscal ideal deveria possuir informações padronizadas sobre o produto vendido, com a utilização do código de produto correto (EAN correto) e uma única descrição sem abreviações do produto por todos os contribuintes, além da falta de padronização da descrição, existe a limitação do número de caracteres a serem impressos no cupom fiscal, o que leva ao mesmo produto (de mesmo EAN) a possuir diversas descrições diferentes em função de que cada contribuinte possuir seu sistema próprio de emissão de cupom fiscal com dados próprios.

Na Figura 67 está um exemplo de cupom fiscal:

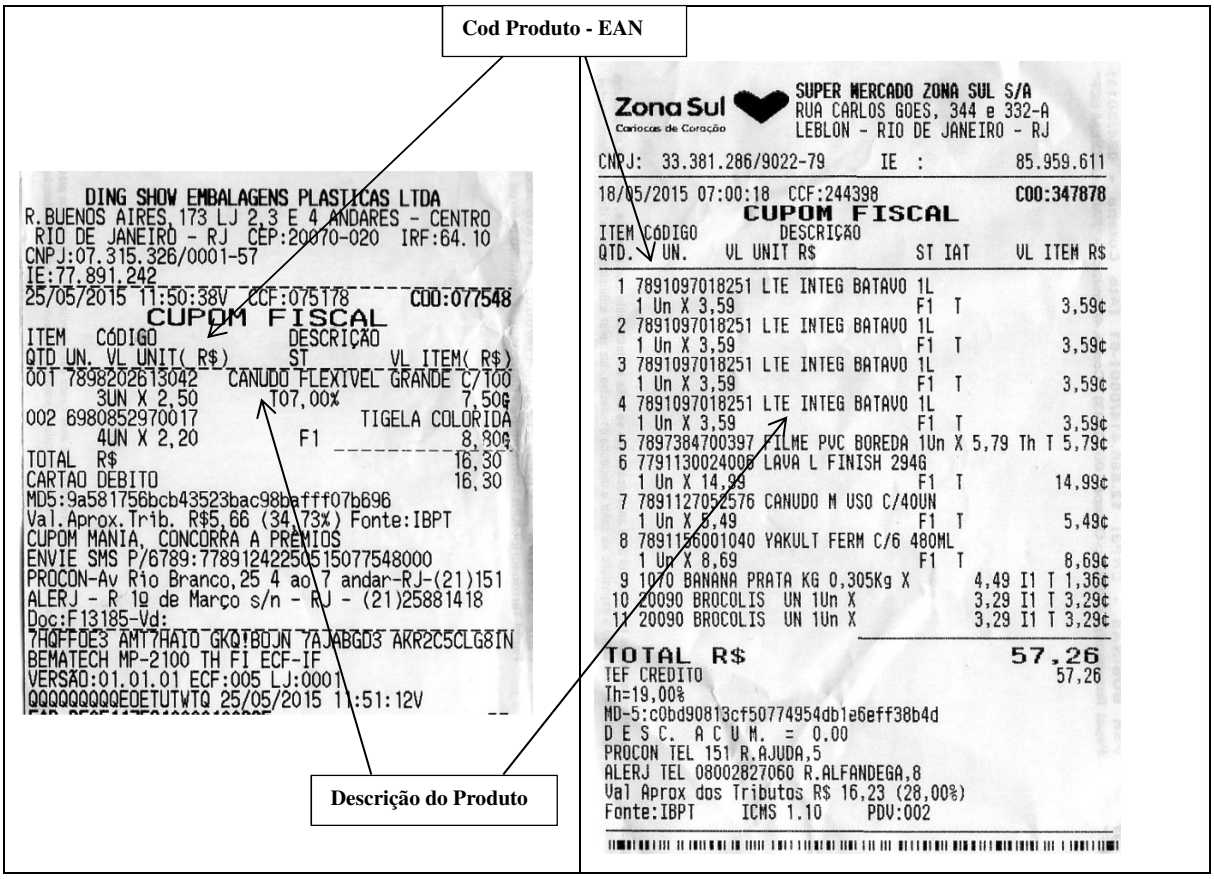


Figura 67 : Amostras de cupons fiscais.

O contribuinte que fizer operações de venda de mercadorias ao consumidor final e possuir faturamento (de todas as filiais) superior a cento e vinte mil Reais, deverá ser obrigado a possuir um equipamento denominado de ECF (emissor de cupom fiscal).

Cada contribuinte que possuir o ECF tem por obrigação:

- Enviar mensalmente até o 10º dia do mês, em mídia ótica não regravável, a Memória de Fita-detalhe (MFD), arquivo em formato texto (TXT), contendo informações relativas aos documentos emitidos pelo ECF no mês imediatamente anterior, no formato e conforme especificações contidas no Ato COTEPE/ICMS 17/04 contendo o registro de assinatura digital **OU**
- Transmitir à SEFAZ até o 15º dia do mês arquivo MFD ou registro60 I, conforme o caso, referente às operações e prestações efetuadas no mês anterior;

O código utilizado para identificar as mercadorias ou prestações registradas em ECF deve ser o Número Global de Item Comercial - GTIN (*Global Trade Item Number*) do Sistema EAN. Na impossibilidade de se adotar o GTIN, deverá ser utilizado o padrão EAN (*European Article*

Numbering) e, na falta deste, admite-se a utilização de código próprio do estabelecimento usuário.

O código deve estar indicado em Tabela de Mercadorias e Serviços do PAF-ECF. No caso de utilização de código próprio, é vedada a reutilização de códigos. A codificação deve ser única para todos os estabelecimentos da empresa.

Quando não identificadas conforme os critérios estabelecidos na legislação, as mercadorias serão tributadas pela maior alíquota prevista para as operações ou prestações internas promovidas pelo estabelecimento.

Os principais problemas dos dados de cupom fiscal são:

- Falta da qualidade dos campos identificadores do produto vendido no registro do cupom fiscal.
- A descrição do produto não possui padronização, é limitada em número de caracteres do *layout* do cupom (raramente superior a 20 caracteres), possuindo abreviações variadas e apresentando descrições conflitantes para o mesmo código de produto.
- As informações enviadas a SEFAZ são geradas por pelo menos 100.000 contribuintes diferentes, levando a descrição do produto a um alto grau de heterogeneidade.
- Existência de registros inválidos referentes a outros produtos dentro do mesmo EAN (identificador) do produto.
- Massiva quantidade de dados (3.5 bilhões de registros – cupons fiscais emtidos no ano de 2010).

APENDICE D

D.1 Autorização da SEFAZ/RJ para utilização dos dados e compromisso de sigilo de dados dos contribuintes do Estado do Rio de Janeiro:

Rio de Janeiro, 14 de Março de 2013

Prezado Senhores

Sou Edival Ponciano de Carvalho Filho, funcionário da Secretaria de Fazenda do Estado do Rio de Janeiro (SEFAZ/RJ), identidade funcional 4331112-1, lotado na ATI.

Estou no final do curso de doutorado em engenharia de sistemas pela COPPE/UFRJ/RJ e desejo apresentar a tese de nome "*Enriquecimento e limpeza de dados oriundos de fontes heterogêneas por meio de extração de informações da WEB*". Esta tese utilizará dados do sistema de cupom fiscal (ECF/MFD), consolidando as diversas descrições de produtos vendidos nos cupons fiscais e construindo um cadastro hierárquico de produto, o qual poderá futuramente viabilizar um processo de determinação/validação da alíquota de ICMS do produto vendido.

Consciente das limitações definidas pelo Artigo 5° da Constituição de 1988, informo que não haverá exposição de informações individualizadas de contribuintes, sendo utilizados apenas os seguintes campos:

- EAN – Código do produto
- Quantidade total vendida (com este código EAN).
- Descrições utilizadas com o respectivo EAN.
- Quantidade de produtos vendidos com este EAN e a respectiva descrição.

Com o objetivo de preservar o sigilo fiscal dos contribuintes, comprometo-me a não expor nenhuma referência ao emissor do cupom fiscal nem valores totais do cupom fiscal ou do produto vendido.

Estou colocando em anexo uma cópia da proposta de qualificação, aprovada em vinte e cinco de Setembro de 2012 pelos professores D.Sc. Geraldo Bonorino Xexéo, D.Sc. Luis Alfredo Vidal de Carvalho e professora Ph.D. Priscila Machado Vieira Lima.

Atenciosamente

Edival Ponciano de Carvalho Filho ID: 4331112-1

Edival Ponciano de Carvalho Filho

Assessor de Informática / ATI/ SEFAZ

De acordo

José Correa da Silva
Superintendente-SUACIEF
Mat. 6.199727-6