Uso de Informação Linguística e Análise de Conceitos Formais no Aprendizado de Ontologias

Carlos Eduardo Atencio Torres

Dissertação apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Mestre em Ciências

Programa: Ciência da Computação

Orientador: Prof. Dra. Renata Wassermann

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, Agosto de 2012

Resumo

Na atualidade, o interesse no uso de ontologias tem sido incrementado, no entanto, o processo de construção pode ser custoso em termos de tempo. Para uma ontologia ser construída, precisa-se de um especialista com conhecimentos de um editor de ontologias. Com a finalidade de reduzir tal processo de construção pelo especialista, analisamos e propomos um método para realizar um aprendizado de ontologias (OL) de forma supervisionada.

O presente trabalho apresenta uma abordagem combinada de diferentes técnicas no aprendizado de ontologias (AO). Primeiramente usamos uma técnica estatística chamada C/NC-*values* acompanhada da ferramenta Cogroo para extrair os termos, que são as palavras ou conjunto de palavras mais representativas em um texto. Estes termos serão considerados por sua vez como conceitos. Projetamos também uma grámatica de restrições (GR) com base na informação linguística do português, afim de reconhecer e estabeler relações entre conceitos. Para poder enriquecer a informação na ontologia, usamos a análise de conceitos formais (ACF) afim de identificar possíveis superconceitos entre dois conceitos. Finalmente, extraímos ontologias para os textos de três temas, submetendo elas à avaliação dos especialistas na área. Para essa avaliação um site foi feito para mostrar amigavelmente as ontologias aos avaliadores e usamos o questionário de marcos de características proposto pelo método OntoMetrics. Os resultados mostraram que nosso método retornou ontologias aceitavelmente idôneas para representar os respectivos temas.

Palavras-chave: Aprendizado de Ontologias, Extração de Termos, Descoberta de Relações, Análise Sintática, Gramática de Restrições, Análise de Conceitos Formais.

Capítulo 1

Introdução

O uso de ontologias formais no contexto da Web Semântica permite compartilhar conhecimento através de diferentes aplicações indo em direção ao sonho de Berners-Lee e Fischetti (1999). No entanto, a construção destas ontologias é, geralmente, feita por um especialista, proceso este que requer um investimento considerável de tempo e dinheiro. Portanto, é necessário desenvolver um método automático ou semi-automático de construção da ontologia. Este processo é chamado de Aprendizado de Ontologia (AO).

Brank *et al.* (2005); Drumond e Girardi (2008) detalharam diferentes metodologias e aplicações que trataram o problema de AO. A maioria desses trabalhos apresentou uma arquitetura dividida em (i) extração de termos, (ii) extração de conceitos, (iii) construção de taxonomia de conceitos, (iv) descoberta de relações, e (v) extração de axiomas.

É importante notar que a maioria de trabalhos foi feito para o idioma inglês, sendo o trabalho de Junior (2008) o primeiro a realizar AO para textos em português. Ele apresentou um *plugin* para o Protégé chamado OntoLP, que recebe textos, previamente transformados para o formato XCES (formato para representar a estrutura sintática do texto) e utiliza padrões sintáticos para adquirir as relações. Porém, um dos problemas dessa abordagem é que devido à rigidez dos padrões sintáticos muitos casos verdadeiros são omitidos na análise.

Diferente do trabalho de Junior, o projeto ACE [1] (*Attempto Controlled English*) usa uma *gramática de restrições* (GR) para traduzir sentenças escritas em linguagem natural em um formato de propriedades ontológicas.

No presente trabalho também estudamos o uso da Análise de Conceitos Formais (ACF) de Wille (1982) que é uma teoria de análise de dados que identifica estruturas conceituais entre um conjunto de dados. De acordo com Cimiano *et al.* (2003), uma ontologia pode ser adquirida a partir de um reticulado de conceitos traduzido em uma taxonomia de conceitos. Um grave problema nesse enfoque é observado no momento da etiquetação das propriedades. Já para Wang e He (2006) isso não representa problema pois ele utiliza a linguagem SWRL (*Semantic Web Rule Language*) para representar regras em uma ontologia. Ainda assim, o ponto forte da ACF ainda é a taxonomia de conceitos.

O objetivo deste trabalho é a proposta de um método de AO para o português em que:

- Usamos o método NC/*Values* para a extração de termos. Este método utiliza a informação sintática das palavras, junto com a frequência delas e considera também a frequência em que uma palavra candidata a termo é parte de outra palavra candidata a termo. Uma ferramenta muito útil para a tarefa de conseguir a informação sintática é o corretor ortográfico Cogroo, o qual possue uma API para analisar textos.

- Projetamos uma GR para o português, a qual foi testada com uma grande quantidade de construções sintáticas em português.

[1] http://attempto.ifi.uzh.ch

- Usamos a ferramenta proposta por Godin *et al.* (1991) para construir um reticulado de conceitos a partir de uma tabela objeto-atributo. Em seguida, esse reticulado é traduzido a uma forma taxonômica e aqueles conceitos que possuem mais de um conceito inferior são adicionados na ontologia.

- Implementamos uma ferramenta em Java para extrair as ontologias a partir dos textos. Esta ferramenta possue uma opção de filtragem para cada etapa no AO.

- Usamos o marco de características do método OntoMetrics para avaliar a dimensão conteúdo das ontologias retornadas pela nossa ferramenta. Tal dimensão conteúdo possui os fatores de Taxonomia, Conceitos, Relações e Axiomas, porém, o fator de Axiomas não foi desenvolvido.

- Para a avaliação, levamos em consideração textos de três temas: (i) Métodos Ágeis, (ii) Branca de Neve e (iii) Doença do Câncer. Usamos a ferramenta e geramos ontologias para esses textos.

- Fizemos um site para mostrar amigavelmente tais ontologias para que especialistas as avaliem. As avaliações foram muito interessantes e as ontologias provaram ser idoneamente aceitáveis.

Os capítulos 2 e 3 são introdutórios e apresentarão, de forma geral, os conceitos de AO e ACF respectivamente. Já no Capítulo 4 começaremos a descrever um sistema de AO e falaremos sobre diferentes formas de extração de termos, ressaltando a técnica de NC/ *Values*. No Capítulo 5 descreveremos diferentes técnicas de descoberta de relações e discutiremos o problema da GR. No Capítulo 6 faremos um breve resumo da sintaxe da gramática portuguesa e falaremos sobre a gramática gerativa e a análise de constituintes. No Capítulo 7 relataremos nossas alegrias e frustrações nos experimentos. No Capítulo 8 introduziremos o conceito de avaliação de ontologias, detalhando o método de OntoMetrics e apresentaremos os resultados da avaliação pelos especialistas consultados. Finalmente, o Capítulo 9 relataremos as conclusões do trabalho.

Capítulo 2

Aprendizado de Ontologias

Neste capítulo, revisaremos de forma geral os temas de ontologias, lógicas descritivas e aprendizado de ontologias (AO).

2.1 Ontologias

2.1.1 Definição de Ontologias

De acordo com Gruber (1993), *uma ontologia é a especificação de uma conceitualização*. De forma prática, Ivan Kostial (2003) indicou que *o uso de uma ontologia permite definir conceitos e relações representando conhecimento a respeito de um documento, em particular em um domínio específico de termos*.

O desenvolvimento de uma ontologia geralmente está sob a responsabilidade de um especialista com conhecimentos de um editor de ontologias. Tal processo segue um conjunto de boas práticas que vemos por exemplo em Mizoguchi (2003); Noy e Mcguinness (2001).

Uma ontologia possui os seguintes elementos:

1. **Classes**, também chamadas de conceitos.
2. **Propriedades**, também chamadas *papeis* em lógica de descrição; e
3. **Indivíduos** que são instâncias de classes definidas.

2.1.2 Tipos de ontologia

De acordo com Mizoguchi (2003), os tipos de ontologia, segundo a riqueza semântica, podem ser:

1. **Ontologias pesadas** São desenvolvidas prestando muita atenção ao significado de cada conceito, assim como a ordem das relações. Desta maneira, busca-se garantir a consistência e fidelidade do modelo.
2. **Ontologias leves** São hierarquias de conceitos sem definições extensas. Elas melhoram as consultas, porém tornam-se dependentes do contexto.

Em 2004 o W3C recomendou OWL (*Web Ontology Language*) como linguagem para descrição de ontologias. OWL começou com três sub-linguagens:

OWL-Lite atende aqueles usuários que necessitam principalmente de uma classificação hierárquica e restrições simples. Permite um caminho de migração mais rápido de *tesauros* e outras taxonomias.

OWL-DL atende aqueles usuários que desejam a máxima expressividade, enquanto mantém a computabilidade e decibilidade, ou seja, garante-se que todas as conclusões sejam computáveis em tempo finito.

OWL-Full é direcionada àqueles usuários que querem a máxima expressividade e a liberdade sintática do RDF sem nenhuma garantia computacional.

A partir de 2009 o W3C recomendou OWL 2 como linguagem padrão para expressar ontologias. OWL 2 substitui o antigo padrão recomendado em 2004 adicionando mais características. Este novo padrão introduz o conceito de perfil que é um fragmento de OWL 2 para negociar o poder expressivo pela eficiência de raciocínio. Existem três perfis:

OWL 2 EL fornece algoritmos de tempo polinomial para todas as tarefas padrão de raciocínio; é particularmente útil em aplicações empregando ontologias que contém um grande número de propriedades e/ou classes. A sigla *EL* reflete a base do perfil de família de lógicas de descrição $\mathcal{EL}$ (quantificação existencial).

OWL 2 QL destina-se a aplicações que utilizam grandes volumes de instâncias de dados e onde a tarefa mais importante é devolver resultados para as consultas. A sigla QL reflete o fato que pergunta-resposta nesse perfil pode ser implementada reescrevendo perguntas em uma linguagem de consulta relacional padrão.

OWL 2 RL é destinada a aplicações que requerem raciocínio escalável sem sacrificar em excesso o poder expressivo. Sistemas de raciocínio OWL 2 RL podem ser implementados usando mecanismo de raciocínio baseado em regras. A sigla RL reflete o fato que o raciocínio, neste perfil, pode ser implementado usando uma linguagem de regras padrão.

2.1.3 Lógicas de Descrição

Staab e Studer (2004) indicaram que as lógicas de descrição são uma família de linguagens de representação de conhecimento, podendo representar o conhecimento de um domínio de maneira formal e estruturada.

Cada lógica se diferencia entre si pelos construtores que possui, formando diferentes combinações partir do $\mathcal{AL}$ de [1] diversas extensões. A Tabela 2.1 apresenta as definições sintáticas e semânticas de diferentes construtores, respeitando a notação:

- **A,C,D**: são nomes de conceitos.
- **R**: relação.
- $\mathcal{I}$: ou interpretação, é um par $< \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} >$, onde :
  - $\Delta^{\mathcal{I}}$ é o universo.
  - $\cdot^{\mathcal{I}}$ é uma função de mapeamento de:
    - Conceitos para subconjuntos de $\Delta^{\mathcal{I}}$, e
    - Papéis para subconjuntos de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

2.1.4 Raciocínio

Um mecanismo de inferência OWL é um mecanismo capaz de provar a consistência de uma ontologia e/ou inferir consequências lógicas a partir do conjunto de axiomas que a ontologia possui.

O trabalho do mecanismo de inferência consiste em:

- Achar relações de subsunção entre dois conceitos, por exemplo, $C \sqsubseteq D$ quer dizer que as interpretações de C estão contidas em D, ou seja, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Achar equivalência de conceitos ($C \equiv D$).

[1] linguagem de atributos, em inglês *attribute language*

Construtor	**Sintaxe**	**Semântica**
Nome de Conceito	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Top	$\top$	$\Delta^{\mathcal{I}}$
Bottom	$\bot$	0
Conjunção	$C \sqcap D$	$C^{\mathcal{I}}$
Disjunção ($\mathcal{U}$)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{D}}$
Negação ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Universal	$\forall R.C$	$\{x \mid \forall y : R^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y)\}$
Existencial ($\mathcal{E}$)	$\exists R.C$	$\{x \mid \exists y : R^{\mathcal{I}}(x,y) C^{\mathcal{I}}(y)\}$
Restrição numérica $\mathcal{N}$)	$\geq n.R$	$\{x \mid \sharp\{y \mid R^{\mathcal{I}}(x,y) \geq n\}\}$
Restrição numérica qualificada ($\mathcal{Q}$)	$\geq n.R.C$	$\{x \mid \sharp\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\}\}$
Enumeração ($\mathcal{O}$)	$\{a_1, ...a_n\}$	$\{{a_1}^{\mathcal{I}}, ..., {a_n}^{\mathcal{I}}\}$
Seleção $\mathcal{F}$	$f : C$	$\{x \in Dom(f^{\mathcal{I}}) \mid C^{\mathcal{I}}(f^{\mathcal{I}}(x))\}$

Tabela 2.1: *Alguns construtores que formam famílias de lógicas de descrição de $\mathcal{ALC}$*

- Verificar se a ontologia é consistente.

Na Tabela 2.2 foi comparado um conjunto de mecanismos respeitando os seguintes critérios:

Lógicas de Descrição : Afim de evitar nomes extensos para lógicas de descrição, a abreviação $\mathcal{S}$ é usada para se referir a $\mathcal{ALC}_{R^+}$ (estende $\mathcal{ALC}$ com papeis transitivos). Membros de $\mathcal{S}$ são $\mathcal{SHIF}$ (estende $\mathcal{ALC}_{R^+}$ com papeis hierárquicos, papeis inversos e número de restrições da forma $\leq 1R$), $\mathcal{SHIQ}$ (estende $\mathcal{ALC}_{R^+}$ com papeis hierárquicos, papeis inversos e número de restrições avaliado) e outros que podem ser formados com ajuda da Tabela 2.1.

Algoritmo de Raciocínio : Algoritmos como resolução, Tableaux, RETE (descrito em Forgy (1982)) e Datalog.

Verificação de Consistência : Verifica se a ontologia apresenta inconsistências.

Suporte de Regras : SWRL (*Semantic Web Rule Language*) é uma proposta para combinar as sub-linguagens de OWL (OWL DL e Lite) com um conjunto de regras especificadas em RuleML (*Rule Markup Language* - Linguagem de marcação de regras).

Linguagem Em que linguagem de programação foi feito.

O mercado oferece diferentes mecanismos de inferência que podem ser usados junto ao editor Protégé de ontologias, por exemplo o Pellet (http://clarkparsia.com/pellet), que é um mecanismo de código aberto, prático e usado em sistemas que trabalham com *OWL 2 DL*. Ele é completo e aceita OWL-DL, além disso possui uma performance similar aos outros mecanismos de inferência como o FACT++ ou o RacerPRO Sirin *et al.* (2007).

A OWL API (http://owlapi.sourceforge.net/) é uma API escrita em Java para manipular e criar ontologias. A última versão desta API foi projetada para trabalhar com OWL 2 .

O mecanismo de inferência mais atual é o HermiT que usa o algoritmo de *hypertableau*. Segundo Shearer *et al.* (2008), HermiT possui uma maior rapidez com respeito aos seus antagonistas: FACT++ e Pellet.

HermiT é compativel com OWL API a partir da versão 3.1.0.

2.2 Aprendizado de Ontologias

O termo *Aprendizado de Ontologias* (AO) foi originalmente usado por Maedche e Staab (2001) e foi descrito como a aquisição de um modelo de domínio a partir de uma fonte de dados. Segundo Drumond e Girardi (2008), os tipos de fontes de dados podem ser:

	Pellet	**Kaon2**	**RacerPro**	**Jena**	**FaCT++**	**HermiT**
Lógica de Descrição	SROIQ(D)	SHIQ(D)	SHIQ(D-)	Subconjunto de SHIN(D)	SROIQ(D)	SHOIQ+
Algoritmo	Tableau	Resolução e Datalog	Tableau	Basado em Regras	Tableau	Hipertableau
Verificação de Consistência	Sim	-	Sim	Incompleto para OWL DL	Sim	Sim
Suporte de Regras	Sim (SWRL)	Sim (SWRL)	Sim (SWR - não totalmente suportado) e regras próprias	Sim (Regras próprias)	Não	Sim (SWRL-DL)
Linguagem	Java	Java	Java, Lisp	Java	C++	Java

Tabela 2.2: *Comparação de Mecanismos de Inferência (Fonte: Wikipedia)*

Dados estruturados : por exemplo os arquivos `XML`.

Dados semi-estruturados : como alguns arquivos `HTML` que possuem conteúdo semântico (tags no cabeçalho).

Dados não-estruturados : geralmente a maior parte da informação vem deste tipo de dados, e é o que chamamos de texto sem formatação.

Esta categorização se aplica de forma similar aos tipos de AO, existindo AO para dados estruturados, semi-estruturados e não-estruturados.

De acordo com Yang e Callan (2008), o AO é definido também como um problema de otimização em que tenta-se minimizar as mudanças que se fazem a uma ontologia parcial no processo de adicionar novos conceitos.

Os métodos atuais apostam no aprendizado semi-automático, mesmo no trabalho de Blomqvist (2005) que, embora proponha um método completamente automático no título, usa padrões (que o usuário insere previamente) para identificar construções léxico-sintáticas, e finalmente padrões morfológicos para construir a ontologia.

2.2.1 Modelo de Camadas de Phillip Cimiano

A metodologia de Cimiano (2006) é usada como padrão no AO. Consiste em um conjunto de camadas (ver Figura 2.1) onde primeiramente extraímos os termos de um texto, depois realizamos o aprendizado de conceitos, de relações e finalmente na parte superior da pilha, o aprendizado de axiomas.

2.2.2 Text2Onto

Originalmente tinha o nome de *TextToOnto* e foi uma ferramenta parte do projeto KAON [2], que é um projeto de código aberto para o desenvolvimento de ontologias com fins empresariais.

O Text2Onto [3] adquiriu o seu atual nome a partir do trabalho de Cimiano e Volköer (2005), onde os autores apresentam três novas melhorías: (i) Um modelo ontológico probabilístico (MOP), (ii) uma interface de interação com o usuário, e (iii) uma estratégia de atualização das probabilidades no MOP.

[2] http://kaon.semanticweb.org/
[3] http://ontoware.org/projects/text2onto/

Figura 2.1: *Conjunto de Camadas para AO de acordo com a metodologia de Philipp Cimiano*

2.2.3 Gramática Restritiva (GR)

É uma técnica que consiste em aplicar padrões léxico-sintáticos com informação contextual sobre textos afim de adquirir apenas certa informação desejada.

Os sistemas baseados em GR são o foco de pesquisa do VISL [4], que é um projeto importante de desenvolvimento de ferramentas baseadas em gramáticas online para fins acadêmicos.

Outro exemplo do uso deste paradigma encontra-se no projeto ACE [5] (Attempto Controlled English) cujo objetivo é a construção de ontologias usando uma entrada em linguagem natural com certas restrições.

Na Tabela 2.3 vemos diferentes construções em linguagem natural que o projeto ACE traduz para uma ontologia. Vale a pena resaltar que: na construção 1, vemos que as classes `man` e `human` foram reconhecidas; na construção 3, os indivíduos `John` e `Mary` foram reconhecidos; na construção 7, vemos a definição da propriedade `mary`; e, na construção 8, vemos uma regra envolvendo duas propiedades `like` e `love`.

	ACE	**Linguagem Formal**
1	Every man is a human.	man $\subseteq$ human
2	Every human is a male or is a female.	human $\subseteq$ male $\bigcup$ female
3	John is a student and Mary is a student.	{John,Mary} $\subseteq$ student.
4	No dog is a cat.	dog $\subseteq$ *not* cat
5	Every driver owns a car.	driver $\subseteq \exists$ own car
6	Everything that a goat eats is some grass.	goat $\subseteq \forall$ eat grass
7	John likes Mary.	<John,Mary > $\in$ like
8	Everybody who loves somebody likes him/her.	love(X,Y) $\Rightarrow$ like(X,Y)

Tabela 2.3: *Exemplo do ACE, extraído do site de Attempto.*

[4] http://beta.visl.sdu.dk/
[5] http://attempto.ifi.uzh.ch

2.2.4 OntoLP

É um plugin feito para o Protégé como parte do trabalho de mestrado de Junior (2008). Ele usa técnicas híbridas entre estatística e padrões léxico-sintáticos para achar a lista de termos, além de aplicar padrões para identificar uma taxonomia de conceitos na ontologia.

Dentre as características mais importantes deste plugin estão:

- A entrada são arquivos em formatação XCES[6], que é uma representação em XML da anotação linguística de certos textos.
- A partir deste formato XCES extrai-se facilmente os núcleos dos sintagmas nominais que serão considerados termos.
- Opcionalmente, realiza-se uma filtragem por grupos semânticos de acordo com a informação do analisador sintático PALAVRAS que foi desenvolvido por Bick (2000).
- Para calcular a relevância dos termos, foram usados três métodos: Frequência Relativa, *tf-idf* e *NC-Value*.
- Os termos simples são considerados subconceitos dos termos complexos, por exemplo: *arquitetura* é superconceito de *arquitetura colonial*, *arquitetura eclesiastica* e *arquitetura moderna*.
- Para a extração de taxonomias usa-se os padrões léxico-sintáticos de Baségio.

Neste capítulo apresentamos brevemente uma introdução ao tema de ontologias e vimos também conceitos genéricos de um sistema de AO. Falamos um pouco de GR e analisamos também o OntoLP a maneira de exemplo de um sistema de AO. No próximo capítulo falaremos de outro tópico importantes: *a análise de conceitos formais*.

[6]XCES: XML-based Encoding Standard for Linguistic Corpora

Capítulo 3

Análise de Conceitos Formais

A Análise de Conceitos Formais (ACF) é definida por Ganter e Wille (1999) como uma teoria de análise de dados que identifica estruturas de conceitos entre conjunto de dados.

Uma vez que definimos um mapeamento entre objetos e atributos, esta ferramenta conceitual se torna a forma mais simples de construir uma taxonomia usando o reticulado de conceitos.

Contexto Formal Consiste em uma tripla $< O, A, I >$, onde O é um conjunto de objetos, A é um conjunto de atributos e $I \subseteq O \times A$ é uma relação binária.

Conceito Formal

- Consiste em uma tupla $< O', A' >$ onde:
  - $O' \subseteq O$ e é também chamado *extent*
  - $A' \subseteq A$ e é também chamado *intent*
- Podem seguir uma ordem parcial respeitando a inclusão de elementos, assim:
  $< O', A' > \quad \leq \quad < O'', A'' > \Leftrightarrow O' \subseteq O''$
  e de forma equivante:
  $< O', A' > \quad \leq \quad < O'', A'' > \Leftrightarrow A'' \subseteq A'$

3.1 Exemplo do domínio de Turismo

Este é um exemplo clássico apresentado por Cimiano (2006), e trata de um domínio de viagens turísticas onde a informação contextual é apresentada em uma lista de objetos asociados a um determinado verbo, tal como pode ser visto na Tabela 3.1.

verbo	**objetos**
reservar	hotel, apartamento, carro, bicicleta, excursão, viagem
alugar	apartamento, carro, bicicleta
dirigir	carro, bicicleta
montar	bicicleta
acompanhar	excursão, viagem

Tabela 3.1: *Exemplo de verbos e objetos extraídos de um corpus textual sobre o domínio de Viagens Turísticas*

Tais verbos são adjetivizados para ser tratados como atributos. Dessa forma, é construída uma tabela de asociação *objeto-atributo* como pode ser visto na Tabela 3.2.

A seguir, aplica-se o algoritmo de Ganter (1984) resultando no reticulado de conceitos da Figura 3.1.

	reservável	alugável	dirigível	montável	acompanhável
hotel	X				
apartamento	X	X			
carro	X	X	X		
bicicleta	X	X	X	X	
excursão	X				X
viagem	X				X

Tabela 3.2: *Domínio de Viagens Turísticas como um contexto formal*

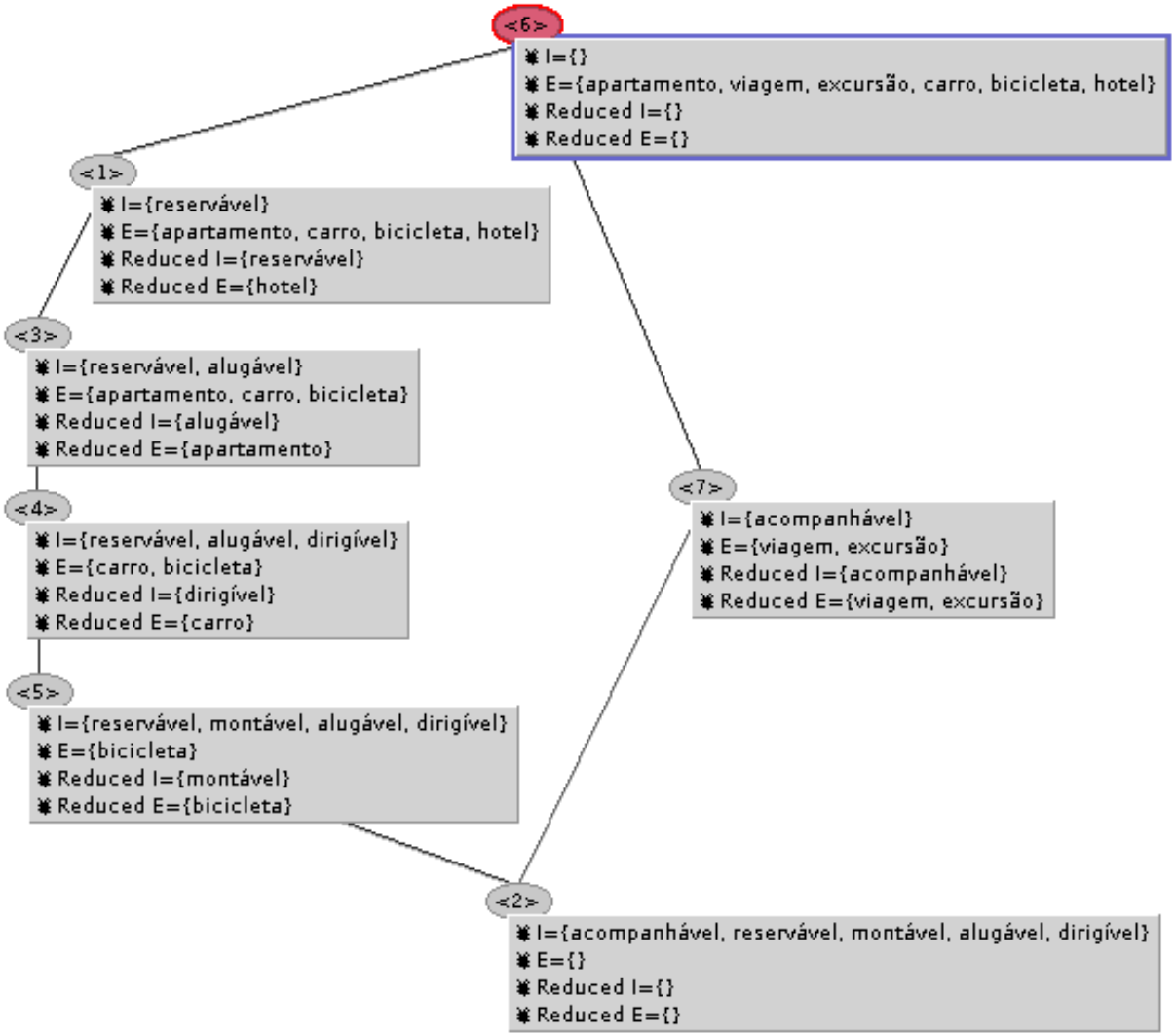


Figura 3.1: *Reticulado de conceitos para o domínio de Viagens Turísticas*

O reticulado é uma estrutura que descreve uma ordem entre conceitos baseada no conjunto de atributos que cada conceito possui. Esta estrutura coloca no topo os conceitos mais gerais, ou seja, aqueles que possuem a menor quantidade de atributos, e em consequência, os conceitos com maior especificação ficarão na parte baixa do reticulado. Essa relação pode descrever uma taxonomia.

3.2 Forma Reduzida do Reticulado

No reticulado da Figura 3.1 encontramos informação redundante, por exemplo *apartamento* é repetido nos nós $< 6 >$,$< 1 >$ e $< 3 >$. Godin *et al.* (1991) apresentam uma forma de reduzir o reticulado com a ferramenta Galicia [1].

A Figura 3.2 mostra a taxonomia sem a informação redundante.

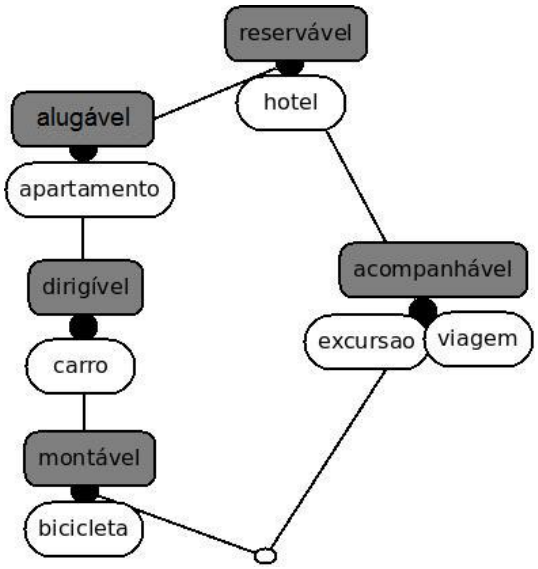


Figura 3.2: *Taxonomia de conceitos para o domínio de Turismo*

3.3 Aplicação nas Ontologias

Como foi bem apontado por Hitzler (2011), os usos de ACF na área de ontologias são:

- Construção de taxonomias.
- Melhoras na ontologia, por exemplo, a exploração de atributos.
- Enriquecimento semântico interativo.
- Alinhamento e *Merging* de Ontologias.

Alguns trabalhos descrevem o uso da ACF para auxiliar o especialista na construção de ontologias. Por exemplo, o trabaho de Peng e Zhao (2007) descreve uma metodologia incremental, auxiliada pela ACF, para a construção de uma ontologia. Essa ontologia foi usada para recuperação de componentes de *software*.

Outros trabalhos aproveitam a ordem parcial entre conceitos no reticulado para construir uma taxonomia. Tomamos como exemplo o trabalho de Cimiano *et al.* (2003), que descreve os passos para realizar o AO a partir de textos usando ACF da seguinte forma:

[1] http://www.iro.umontreal.ca/ galicia/

- Realiza-se uma análise sintática do texto para extrair os pares *verbo-objeto*. Os objetos são tratados como os elementos do *extent* e para os verbos adiciona-se o sufixo *ABLE* afim de ser tratados como elementos do *intent*.
- Constrói-se o reticulado de forma reduzida, ou seja, de tal forma que cada objeto ou atributo é nomeado apenas uma vez no diagrama.
- Para cada conceito formal, cria-se um conceito na ontologia com a etiqueta formada pelos elementos do *intent*. E para cada elemento do *extent* criamos um subconceito de tal conceito novo criado.

Outro trabalho interessante é de Wang e He (2006), que propõe o uso de SWRL (*Semantic Web Rule Language*) para traduzir o reticulado em forma de um conjunto de cláusulas de Horn. Tal trabalho usa uma lógica modal específica para ontologias proposto no trabalho por Haav (2004).

3.4 Conceito Superior Comum

A ACF também pode ajudar na identificação de um conceito superior comum LCA [2] entre dois conceitos c_1 e c_2. Tal processo é descrito no algoritmo 1, onde a função $ancestorOf(c_1)$ devolve o conjunto de elementos superiores a c_1 no reticulado, e a função $distance(c_{1,2})$ devolve a quantidade de arcos entre os dois nós.

Algoritmo 1 LCA(c_1,c_2) - Lowest Common Ancestor

Entrada: c_1, c_2 ▷ Conceitos Formais
  $Ld_1 \leftarrow$ EmptyList, $Ld_2 \leftarrow$ EmptyList
  para cada $c \in ancestorOf(c_1)$ **fazer**
    $t \leftarrow < c, distance(c, c_1) >$
    Ld_1.Add(t)
  fim para
  para cada $c \in ancestorOf(c_2)$ **fazer**
    $t \leftarrow < c, distance(c, c_2) >$
    $d_1 \leftarrow d_2 + t$
  fim para
  $minimal\_height \leftarrow \infty$
  $better\_node \leftarrow$ *'Thing'*
  para cada $t \in d_1$ **fazer**
    se $\exists s \in d_2$ where $s.node = t.node$ **então**
      $tmp\_max \leftarrow max(s.distance, t.distance)$
      se $tmp\_max < minimal\_height$ **então**
        $minimal\_height \leftarrow tmp\_max$
        $better\_node = s.node$
      fim se
    fim se
  fim para
  return $better\_node$

A estratégia do algoritmo é escolher aquele conceito superior comum (LCA) cuja maior distância dos nós inferiores for mínima. Por exemplo, na Figura 3.3, o LCA entre os nós $< 5 >$ e $< 9 >$ é o nó $< 4 >$ pois a maior distância dele aos nós filhos é 2. E não pode ser $< 2 >$ por exemplo, pois a maior distancia deste último aos filhos é 3.

[2] LCA - Lowest Common Ancestor

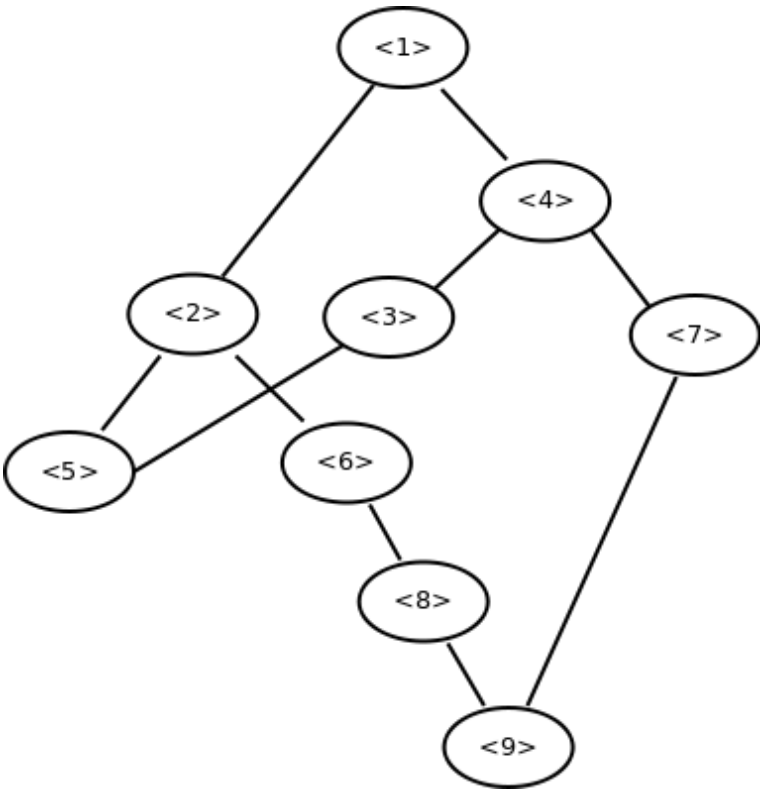


Figura 3.3: *Exemplo de reticulado*

Neste capítulo falamos sobre ACF e como poderíamos usar esta técnica no AO. A partir do próximo capítulo começamos a descrever um sistema de AO e começamos com a etapa inicial de todo AO: *a extração de termos*.

Capítulo 4

Extração de Termos

Segundo Cabré (1999); Maria da Graça Krieger (2004), um termo é uma unidade léxica que possui um significado concreto dentro de um certo domínio. Ele pode ser considerado também como uma expressão multipalavra (EMP), definida por Sag *et al.* (2001) como um conjunto de palavras que possuem um significado proprio e cuja frequência em certo texto tem caráter idiossincrático. . Um sinônimo estabelecido por Wray (2002) a estas expressões multipalavras é o de *sequências formulaic*, que são definidas como: *uma sequência, continua ou descontinua, de palavras ou outros elementos, os quais são, ou parecem, pre-fabricados: isto é, armazenados e recuperados totalmente da memória no momento do uso, ao invés de serem sujeitos a generalização ou análise pela gramática da linguagem.*

De acordo com Sag *et al.* (2001) existem dois tipos de EMP:

Frases léxicas : as quais possuem sintaxe ou semântica idiossincrática, ou contém palavras que não ocorrem em isolamento. As expressões multipalavras deste tipo podem ser:

- Expressões idiomáticas, também chamadas expressões populares: Caracterizam-se por não ter um significado literal como por exemplo gírias: "show de bola".
- Compostos nominais: "centro estadual de educação tecnológica".
- Nomes próprios: "Instituto de Matemática e Estatística"
- Construções verbais: "dar é melhor que receber".
- Verbos de suporte: "Tomar banho" ou "Por em risco".

Frases institucionalizadas : são construções semânticas e sintáticas, mas estatísticamente idiossincráticas. Por exemplo em "banco de dados", as palavras "banco" e "dados" possuem significado próprio, mas juntas possuem outro significado próprio.

O trabalho de Duan *et al.* (2009) propôs um método híbrido para identificar EMPs a partir de documentos em inglês. Tal método usa processamento de linguagem natural para extrair a informação gramatical de cada palavra, e aplica uma técnica de alinhamento de sequencias para obter aqueles segmentos mais similares entre todas elas.

Quando testamos esta técnica em um documento sobre métodos ágeis de desenvolvimento de software, obtivemos vários segmentos muito compridos, como o seguinte:

```
Então se você tem um método que prevê , que se baseia em achar, dá
para prever o futuro, então você já tem um problema em desenvolvimento
de software ...
```

Segmentos como o anterior foram encontrados em todo o texto, e isso deu-se ao fato de que muitas palavras, com pontuação elevada, estavam dentro de tais segmentos, por exemplo *método*, *desenvolvimento* e *software*.

Reduzir tais EMPs a simples termos demandaria um grande esforço, e sobretudo tempo, o que não é justificável já que existem outros métodos mais diretos.

4.1 Métodos para extração de termos

A maioria dos trabalhos usam técnicas estatísticas para extrair os termos, por exemplo o trabalho de Pantel e Lin (2001) apresenta uma métrica híbrida baseada no valor de informação mútua junto à verossimilhança logarítmica.

O uso desses dois métodos está fundamentado, já que a verossimilhança devolve valores altos para termos que acontecem como erros, como por exemplo "*the the*" no idioma inglês. Porém, a informação mútua diminui a chance deste tipo de expressão ser escolhida como termo.

Para melhorar os resultados de Pantel e Lin, o trabalho Tomokiyo e Hurst (2003) propôs o uso de uma fonte externa maior (*background*) para calcular as probabilidades com maior precisão que usando apenas o texto a ser analisado (*foreground*). Para cada fonte foi criado um modelo de linguagem, que é uma estrutura de dados que armazena probabilidades de ocorrências de uma palavra (*unigram*) ou várias palavras (*N-gram*) aparecerem no texto.

Tomokiyo e Hurst usaram também o ponto de divergência *KL* (Kullback-Leibler) para medir a ineficácia de assumir uma certa distribuição de probabilidade p quando a verdadeira é q. A divergência KL é conhecida também como entropia relativa.

Finalmente, o trabalho de Deane (2005) compara diferentes métricas, a *MutualRank* entre elas, e explica que os métodos estatísticos não tem bons resultados devido à incorreta suposição da distribuição de probabilidade (como por exemplo distribuição normal ou de Poisson), mas que a distribuição Zipf é a mais apropriada porque ela se amolda melhor em distribuições assimétricas (skewed distribution em inglês), as quais são mais comuns em ocorrências de palavras.

A conclusão de Deane com respeito à incorreta suposição da distribuição de probabilidade também é defendida pelo trabalho de Duan et al. Duan *et al.* (2009), que usou o alinhamento de sequencias na extração de termos.

4.2 Ferramentas para a extração de termos

A seguir apresentamos uma lista de ferramentas disponíveis na Internet, dedicadas à extração de termos.

- ExatoLP é uma ferramenta desenvolvida na Universidade de Rio Grande do Sul para extrair termos a partir de textos escritos em português. Seu lançamento para baixar e usar livremente está anunciado desde 2009.
- KEA é um extrator de palavras chave para documentos, portanto, devolve poucas palavras chaves para um texto. O trabalho de Lacerda Dias (2004) adaptou este sistema para o português.
- TOPIA é parte da biblioteca ZOPE, que é uma biblioteca de aplicações escrita em Python. De código simples, compete contra a ferramenta online de Yahoo. Usa um léxico em inglês com muitas anotações sintáticas. Ele usa a métrica N/NC-Values.
- Text-NSP (*Ngram Statistic Package*) é uma biblioteca Perl de utilidades para a análise estatística em textos. É uma ferramenta muito usada para trabalhos em linguística que analizam palavras.
- AlchemyAPI é uma biblioteca com outras utilidades interessantes para aplicação em sites, tais como extração de entidades nomeadas, marcações de conceitos, análise de sentimentos em textos, entre outras. Existe a versão da biblioteca para Android OS (Mobile SDK), Java, Perl, Ruby, Python, PHP, C++ e C#. Não é livre e precisa se registrar para obter um *API Key*.
- MAUI é uma ferramenta cujo objetivo é etiquetar um texto com os termos que definem tal texto. Ele tem suporte a espanhol, francês e alemão. Esta ferramenta consulta a Web para adquirir informação contextual.

Existem também sistemas online que realizam a tarefa de extração de termos. Vemos eles a continuação:

- *Five Filter Term Extraction* suporta inglês e usa a ferramenta TOPIA.
- ZEMANTA é um serviço web. Uma versão cliente pode ser baixada para realizar consultas em informações contextuais com respeito dos textos que o usuario ingressa.
- *Yahoo's Term Extraction* é um dos melhores. A quantidade de consultas são restritas por dia.
- *Terme Extractor* que trabalha apenas com textos no inglês. É lento no processamento.
- *Translated.net, Terminology Extraction* suporta os idiomas inglês, italiano e francês.
- TermMine suporta apenas o idioma inglês. possui uma versão para baixar e usar com o Protégé. Usa a métrica C/NC-value. Como lematizador usa o TreeTagger (pago) e o Genia Tagger.
- OpenCalais [1] é uma ferramenta similar a MAUI, porém, mais completa pois ela faz marcações semânticas para um determinado texto. É usado em Drupal e outros CMS (*Content Management System*).

4.3 Reconhecimento de sintagmas nominais e preposicionais

O Cogroo, apresentado em Kinoshita *et al.* (2007), é um projeto de código livre para correção gramatical de textos em língua portuguesa. Ele possui uma API que pode ser usada para realizar a lematização [2] do corpus.

O projeto Cogroo usou o corpus da Floresta Sintática [3] para treinar e gerar modelos gramaticais. Tal corpus foi gerado pelo PALAVRAS, que é uma ferramenta paga para a análise gramatical.

Uma vantagem de usar o Cogroo é a desambiguação de frases, por exemplo na sentença *quem casa quer casa*, ele identifica positivamente que a primeira ocorrência de *casa* refere-se a um verbo e a segunda ocorrência refere-se a um substantivo.

O Cogroo usa um *shallow parser* para construir uma árvore sintática simples. A partir dessa árvore, pode-se obter uma lista de tokens, cada um contendo a informação sintática (NP,SUB,etc) e/ou morfológica (Noun, third person, etc) das unidades gramaticais da oração.

Por exemplo, na frase *Os métodos ágeis são importantes no desenvolvimento de software*, o cogroo realizou uma análise sintática, obtendo uma lista de tokens e uma árvore sintática que apreciamos na Tabela 4.1 e na Figura 4.1 respectivamente.

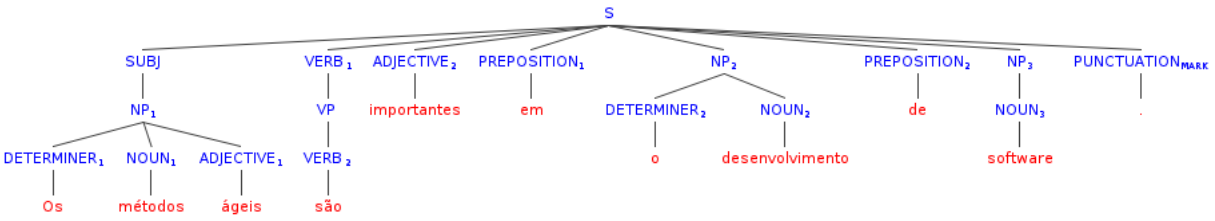


Figura 4.1: *Árvore sintática retornada pelo Cogroo*

[1] http://www.opencalais.com/

[2] A lematização é o processo de extrair as palavras sem infleções de tempo, género e número, chamando elas de *lemas*

[3] Floresta Sintática http://www.linguateca.pt/floresta/corpus.html

Token	Lema	Tag Morfológico
Os	o	determiner,male,plural
métodos	método	noun,male,plural
ágeis	ágil	adjective,male,plural
são	ser	verb,plural,third,present,indicative,finite
importantes	importante	adjective,male,plural
no	em	preposition
no	o	determiner,male,singular
desenvolvimento	desenvolvimento	noun,male,singular
de	de	preposition
software	software	noun,male,singular

Tabela 4.1: *Lematização e informação morfológica*

4.4 Extração de candidatos a termos com a métrica C-*value*/NC-*value*

A métrica C-*value*/NC-*value* é prática pois precisa das frequências das palavras, e também confiável, já que vários sistemas a usam, como o Copia, o Terme Extractor e OntoLP (plugin desenvolvido no trabalho de Junior (2008)).

Esta métrica foi introduzada no trabalho Frantzi *et al.* (1998) e combina informação linguística e estatística. O processo divide-se em duas etapas: (i) Calcular os C-*value*s e os NC-*value*s.

4.4.1 Calculando os C-*values*

Aqui calculamos um valor de ranking para os candidados de termos usando apenas a frequência das palavras no texto. Consiste em duas partes:

1. Parte Linguística:
   (a) Realiza-se um processo de lematização em todo o texto.
   (b) Aplica-se um filtro linguístico para conseguir os sintagmas nominais e preposicionais.
   (c) Remove-se os *stop-words*.
2. Na parte estatística, retorna-se um valor para uma cadeia candidata a termo usando:
   (a) A frequência total de ocorrências da cadeia candidata no corpus.
   (b) A frequência da cadeia candidata como parte de outros candidatos a termos maiores.
   (c) O número desses candidatos maiores.
   (d) O tamanho dos candidados (quantidade de palavras).

Finalmente a métrica *C-value* fica explícita da seguinte forma:

$$C\text{–}value(a) = \begin{cases} log_2|a| \times f(a), \text{ se } a \text{ estiver isolado} \\ log_2|a| \times (f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)), \text{ caso contrário} \end{cases} \quad (4.1)$$

onde:

a é o candidato a termo,

$f(.)$ é a frequencia de ocorrência no corpus,

T_a é o conjunto de candidato a termos extraído que contém a,

$P(T_a)$ é o número desses candidatos a termos.

4.4.2 Calculando os NC-*values*

Uma vez calculado os C-*values*, a informação contextual é usada para ajudar na identificação dos termos corretos. O fundamento de usar tal recurso se deve a que normalmente podemos identificar o significado de uma palavra analisando o contexto onde ela aparece.

Os tipos de palavras contextuais podem ser adjetivos, substantivos ou verbos que estão próximos a um candidato a termo. A relevancia de uma palavra contextual $weight(.)$:

$$weight(w) = \frac{t(w)}{n} \tag{4.2}$$

onde :

w é a palavra contextual (substantivo, verbo ou adjetivo)

$t(w)$ é o número de termos onde a palavra w aparece junta.

n número total de termos considerados

Finalmente, temos:

$$NC\text{–}value(a) = 0.8 \times C\text{–}value(a) + 0.2 \times \sum_{b \in C_a} f_a(b) weight(b) \tag{4.3}$$

onde:

a é o termo candidato,

C_a é o conjunto de diferentes palavras contextuais de a

b é uma palavra de C_a,

$f_a(b)$ é a frequencia de b como uma palavra de contexto de a

Neste capítulo descrevemos alguns métodos e ferramentas para a extração de termos. Falamos sobre a ferramenta Cogroo e descrevemos a métrica NC/C-*Value*. No próximo capítulo continuamos descrevendo um sistema de AO falando de: *a descoberta de relações*.

Capítulo 5

Descoberta de Relações

De acordo com Shamsfard e Abdollahzadeh Barforoush (2003), existem dois tipos de relações conceituais:

1. **Taxonômicas :** representam relações de generalização ou especialização. Pertencem a este tipo as relações de hiponímia e/ou hipernímia. De acordo com Sánchez (2009) existem dois métodos para a descoberta de relações deste tipo: (i) usando padrões léxico-sintáticos como os padrões de Hearst; e (ii) achando padrões do tipo [[ADJETIVO*]SUBSTANTIVO]; esse último enfoque foi abordado por Grefenstette (1997).

2. **Não-Taxonômicas :** são o resto de relações, por exemplo sinonímia (mesmo significado), meronímia (parte de), antonímia, possessão, e outras relações que são obtidas de forma automática por algum sistema de aprendizado, por exemplo `gostar_de`, `andar_com`, etc.

5.1 Descoberta de Relações Taxonômicas

De acordo com Cimiano (2006), existem três métodos para adquirir uma hierarquia de conceitos.

1. Baseados em similaridades,

2. Teoria de Conjuntos, e

3. Clustering

5.1.1 Baseado em Similaridade

O trabalho de Hearst (1992) propõe seis padrões léxico-sintáticos para serem procurados nos textos de língua inglesa. Mais tarde, Baségio (2006) fez a adaptação deles para a língua portuguesa (ver Tabela 5.1).

Por exemplo na sentença *"Agar is a substance prepared from a mixture of red algae,* ***such as*** *Gelidium, for laboratory or industrial use"*, depois de aplicar o padrão $h1$ extraímos a relação: hiponímia("Gelidium", "red algae") que indica que *"red algae"* tem como subconceito *"Gelidium"*.

Para a sentença *"... works by* ***such*** *authors* ***as*** *Herrick, Goldsmith and Shakespare"*, depois de aplicar o padrão $h2$, extraímos as relações:

- hiponímia("author","Herrick");
- hiponímia("author","Goldsmith"); e
- hiponímia("author","Shakespeare").

que indica que *"Herrick"*, *"Goldsmith"* e *"Shakespeare"* são subconceitos de *"author"*.

	Padrão Original	**Adaptação**
$h1$	NP such as {NP,}* {(and|or)} NP	SUB tal(is) como {(SUB)*(ou-e)} SUB
$h2$	such NP as {NP,}* {(and|or)} NP	SUB tal(is) como {(SUB)*(ou-e)} SUB
$h3$	NP ,NP* {,} or other NP	tal(is) SUB como {(SUB)*(ou-e)} SUB
$h4$	NP ,NP* {,} and other NP	SUB {,SUB}*{,} ou outro(s) SUB
$h5$	NP including {NP,}* NP {(and|or)} NP	SUB {,SUB}* {,} e outro(s) SUB
$h6$	NP especially {NP,}* {(and|or)} NP	• SUB , especialmente SUB,*ou-e SUB • SUB , principalmente SUB,*ou-e SUB • SUB , particularmente SUB,*ou-e SUB • SUB , em especial SUB,*ou-e SUB • SUB , em particular SUB,*ou-e SUB • SUB , de maneira especial SUB,*ou-e SUB • SUB , sobretudo SUB,*ou-e SUB

Tabela 5.1: *Padrões de Hearst adaptados por Baségio. Em que* NP: Noun Phrase (Sintagma Nominal) *e* SUB: Substantivo

5.1.2 Teoria de Conjuntos

Neste tópico, Cimiano propõe a Análise de Conceitos Formais como a técnica mais simples para obter uma taxonomia de conceitos.

Esta metodologia indica que uma taxonomia pode ser construida estabelecendo relações de similaridade entre conceitos contando com a quantidade de especificações em comum que eles possuam.

5.1.3 Clustering

A suposição básica tomada por esses métodos está fundada na hipótese da distribuição de Harris (1968), que simplesmente indica que palavras similares aparecem em contextos similares.

Traduzindo a distribuição de Harris a um enfoque linguístico-computacional, o objetivo é estabelecer relações entre termos que possuem elementos sintáticos contextuais comuns.

5.2 Descoberta de Relações Não-Taxonômicas

Sánchez (2009) indica que a descoberta de relações deve se ocupar de duas tarefas: (i) encontrar os conceitos relacioados entre si e estabelecer uma relação entre eles, e (ii) etiquetar tal relação.

Sánchez também indica alguns trabalhos que usaram um conjunto fechado de relações genéricas, por exemplo: *parte-de*, **qualia**, **causation**.

O trabalho de Botero e Ricarte (2008) propôs um algoritmo de clustering para achar relações semânticas. Um problema desse trabalho é o estabelecimento de conceitos compostos como *algorithm_genetic* pois o trabalho agrupava dois grupos diferentes de conceitos: (i) aqueles relaciados a *algorithm* e (ii) conceitos relacioados a *genetic*, mas que não necessáriamente representava fielmente o conceito de *genetic_algorithm*. Entre as conclusões, Botero deixa aberto como trabalho futuro a otimização do algoritmo com ajuste de certas variáveis.

O trabalho de Schutz e Buitelaar (2005) estabelece verbos como relações entre conceitos. Ele analiza um determinado texto e fornece uma lista com as triplas relevantes (pares de conceitos, conetados por um verbo) sobre outra ontologia de conceitos existente.

5.3 Gramática de Restrições

De acordo com Karlsson (1995), a Gramática de Restrições é um paradigma no Processamento de Linguagem Natural. Ela consiste na compilação de um conjunto de regras em uma gramática. Cada regra mantém um contexto e determina um comportamento diferente para um componente analisado na sentença.

Nesta seção, discutiremos como usar o conceito de GR (visto na seção 2.2.3) na língua portuguesa.

Problemas de GR no Português

A informação em qualquer texto não aparece em uma forma ideal onde podemos aplicar simples padrões. Por exemplo:

- *A Branca de Neve, bonita como nenhuma, cantava alegremente no bosque.*

Se quisermos encontrar a estrutura SVO (sujeito-verbo-objeto) na sentença anterior, uma simples análise não ajudaria, pois temos a seguinte ordem:

$$\underbrace{\textit{A Branca de Neve}}_{\text{sujeito}}\ ,\ \underbrace{\textit{bonita como nenhuma}}_{\text{aposto}}\ ,\ \underbrace{\textit{cantava}}_{\text{verbo}}\ \underbrace{\textit{alegremente}}_{\text{C. Circunstancial}}\ \underbrace{\textit{no bosque}}_{\text{C. Indireto}}\ .$$

Notamos que o aposto é um elemento que dificulta a análise, mas que é importante pois faz parte da semântica da oração. Ele indica que *Branca de Neve é bonita como nenhuma.*

Também notamos que a frase não possui apenas um mas dois objetos ou complementos: *alegremente* (circunstancial) *no bosque* (indireto).

As complicações se acrescentam quando analisamos frases de período composto, por exemplo:

- *Branca de Neve crescia e ficava bonita.*

Podemos obter duas expressões nucleares, ou orações de período simples:

- *Branca de Neve crescia*
- *Branca de Neve ficava bonita*

Ainda parece trivial, mas a complexidade cresce conforme vamos analisando orações mais extensas, por exemplo:

- *Branca de Neve despediu-se dos sete anões e partiu junto com o príncipe para um castelo distante onde se casaram e foram felizes para sempre.*

Uma análise ótima poderia obter as seguintes frases de período simples:

- *Branca de Neve despediu-se dos sete anões*
- *Branca de Neve partiu junto com o príncipe*
- *Branca de Neve partiu para um castelo distante*
- *Branca de Neve e o príncipe casaram*
- *Branca de Neve e o príncipe foram felizes para sempre*

Existe também uma frase que não está de forma explícita:

- *O príncipe partiu junto com a Branca de Neve para um castelo distante*

Aparentemente, uma pessoa pode extrair tal informação de uma forma natural. Porém, o trabalho para um programa de computador é desafiador.

Neste capítulo, falamos sobre métodos de descoberta de relações e estudamos também o problema de aplicar uma simples GR para textos escritos em português. Para poder projetar uma GR, precisaremos de alguns conceitos básicos de gramática e linguística que veremos no próximo capítulo.

Capítulo 6

Análise Sintática

Neste capítulo faremos uma revisão da sintaxe da língua portuguesa onde, além de conceitos básicos, falaremos também da análise de constituintes. Para isto, tomamos como livro base Bechara (2009) e também o livro de sintaxe e cognição de Lagunilla e Rebollo (2004).

6.1 Sintaxe

A sintaxe é uma parte da gramática que estuda a ordem dos constituintes em uma oração.

A oração encerra a menor unidade de sentido do discurso com propósitos definidos Bechara (2009). Ela pode estar constituída por um ou mais vocábulos, por exemplo:

- *Atenção!*
- *Vamos embora!*
- *Claro que sim!*
- *Margaret viaja com Bruno na terça a noite*

Quando uma mensagem é completa de acordo com o contexto, e delimitada por um silêncio precedente e uma pausa final (sinais de pontuação), chama-se de *período*, existindo oração simples e oração composta quando a sentença possui um ou mais de um período respectivamente.

6.1.1 Oração

Existem diferentes tipos de orações:

- **Declarativas**, por exemplo:
  - *A biblioteca abriu ontem.*
  - *Miguel está falando pelo telefone.*
  - *Ainda não sei.*
- **Interrogativas**, por exemplo:
  - *Quantos anos você tem?*
  - *Tudo bem?*
- **Imperativas**, por exemplo:
  - *Vá comer!*
  - *Seja forte!*
- **Exclamativa**, por exemplo:

- *Oh, meu Deus!*
- *Que interessante!*

Os termos essencias da oração são o sujeito e o predicado. A omissão de um deles é chamada de *elipse*:

- $\underbrace{\textit{Marcos}}_{\text{Sujeito}}$ $\underbrace{\textit{estuda muito}}_{\text{Predicado}}$.
- $\underbrace{\textit{Maia!}}_{\text{Sujeito}}$ $\underbrace{\quad}_{\text{Predicado}}$, *eclipse* por ausência do predicado.
- $\underbrace{\quad}_{\text{Sujeito}}$ $\underbrace{\textit{Corre!}}_{\text{Predicado}}$, *eclipse* por ausência do sujeito.

6.1.2 Sujeito

De acordo com Bechara, o sujeito indica a pessoa ou coisa de que afirmamos ou negamos uma ação ou qualidade, enquanto que o predicado é tudo o que se declara na oração, ordinariamente em referência ao sujeito.

O núcleo do sujeito (NS) é o substantivo, que geralmente está rodeado de determinantes. Em geral, tais determinantes podem ser artigos, adjetivos, pronomes e quantificadores.

- $\underbrace{\textit{O}}_{\text{artigo}}$ $\underbrace{\textit{livro}}_{\text{NS}}$
- $\underbrace{\textit{Os}}_{\text{artigo}}$ $\underbrace{\textit{livros}}_{\text{NS}}$ $\underbrace{\textit{bons}}_{\text{adjetivo}}$
- $\underbrace{\textit{Aqueles}}_{\substack{\text{Pronome}\\\text{demonstrativo}}}$ $\underbrace{\textit{dois}}_{\text{quantificador}}$ $\underbrace{\textit{livros}}_{\text{NS}}$ $\underbrace{\textit{seus}}_{\substack{\text{pronome}\\\text{possessivo}}}$

Existe também, a um nível sintático-semântico, os chamados *termos nucleares* que estão íntimamente relacioados à relação predicativa. Por exemplo:

- *Tim Berner's Lee propôs a Web Semântica como meio de comunicação entre máquinas.*

Além de [*Tim Berner's Lee*] e [*propôs*], existem as frases [*Web Semântica*] e [*meio de comunicação entre máquinas*] que são chamadas *nucleares* porque:

- A oração fala da [*Web Semântica*], e
- que foi proposta como [*meio de comunicação entre máquinas*].

Quando existe um elemento que não está referido nem ao sujeito nem ao predicado, mas a toda a oração, então pode ser deslocado a qualquer posição, por exemplo:

- ***Certamente***, *Tim Berner's Lee propôs a Web Semântica como meio de comunicação entre máquinas.*
- *Tim Berner's Lee*, ***certamente***, *propôs a Web Semântica como meio de comunicação entre máquinas.*
- *Tim Berner's Lee propôs a Web Semântica como meio de comunicação entre máquinas*, ***certamente***.

6.1.3 Predicado

O núcleo do predicado é o verbo. Quando o verbo é intransitivo, o predicado é chamado *simples*, pois possui um significado concreto.

Quando o predicado é extenso, possui delimitações chamadas *argumentos* ou *complementos verbais*. Tais complementos podem ser:

Complemento direto ou objeto direto

São aqueles que completam o significado dos verbos transitivos diretos, por exemplo:

- *O estudante leu um livro.*
- *Eu estudei vários temas.*

Complemento indireto ou objeto indireto

Aqueles que estão do lado dos verbos transitivos indiretos. Estes complementos caraterizam-se por estar junto a uma preposição, por exemplo:

- Os professores gostam ***da*** *pesquisa.*
- O aluno enviou seu trabalho ***ao*** *professor.*
- Os dois enviaram artigos ***para*** *o congresso.*

Complemento adverbiais

Os adjuntos adverbiais podem ser

- de lugar (respondem a perguntas como *onde?*, *por onde?*, etc.):

  Adriana trabalha na Embratel. (*onde*)

  Passamos por várias cidades do interior. (*por onde?*)

  Thiago viaja para Recife. (*aonde?*)

- temporais (responde a perguntas como *quando?*, *desde quando?*, etc.):

  Limpamos a casa no domingo. (*quando?*)

  Não paramos desde as quatro até as oito. (*desde quando?*, *até quando?*)

- modais (responde a perguntas como *como?*, *de que maneira?*):

  Bruno está ajudando bem. (*como?*)

  Pâmela ficou muito cansada. (*de que maneira?*)

- de fim, de causa, de instrumento e de companhia:

  Timóteo estudou matemática para seu trabalho. (*fim*)

  Choravam de emoção. (*causa*)

  Chutou a bola com a perna esquerda. (*instrumento*)

  Saiu com Ellen. (*companhia*)

- de quantidade:

  Talita trabalha muito *mais* nos domingos.

  Peru acabou o jogo com *dois* gols contra.

- de distribuição:

  Os biscoitos foram entregues *para cada vizinho*.

- de inclinação e oposição:

  Sempre esteve *a favor dos mais necessitados*.

  Ela sempre ficou *contra o aborto*.

- de substituição, troca ou equivalência:

  Estou bricando *no lugar* de estudar.

  Sirley trocou a casa *por mais* carros.

- de campo ou aspecto:

  O novo estagiário é um especialista *em base de dados*.

  Ele conseguirá nos ajudar com seus *conhecimentos atualizados*.

- de assunto ou matéria tratada

  O programa de hoje *tratou* { das doenças do corpo / sobre a segurança nas ruas / a respeito da crise financieira }

6.2 Orações Complexas

Também chamada de orações de período composto. Na gramática tradicional identificavam-se dois tipos de orações de acordo com a função sintática: independente e dependente, onde a oração principal era definida como *aquela que pede uma dependente*, ou seja, aquela que a que as dependentes são aderidas, por exemplo:

$$\underbrace{\textit{A Branca de Neve dormiu}}_{\text{Oração Principal}} \quad \underbrace{\textit{porque comeu a maçã envenenada}}_{\text{Oração Dependente}} .$$

Ora, a atual gramática fala de dois tipos de orações: subordinada e oração coordenada, as quais veremos na continuação.

6.2.1 Orações subordinadas

A partir do exemplo anterior, observamos que a oração primitiva *comeu a maçã* é independente de *Branca de Neve dormiu*, mas passsou a exercer uma função de adjunto adverbial de causa já que *porque comeu da maçã envenenada* é a oração subordinada.

Oração Subordinada Substantiva

A Oração Subordinada Substantiva (O.S.S.) exerce a função de substantivo e está precedida da conjunção *que*. Podem ser:

- Subjetiva (O.S.S.S): *Melhor dar do que receber.*
- Objeto Direta (O.S.S.O.D): *O cliente disse que fizemos um bom trabalho*
- Objeto Indireta (O.S.S.O.I): *Precisamos de que os proveedores agilizem os processos.*
- Predicativa (O.S.S.P): *O motivo foi que o outro time não conseguiu*
- Completiva Nominal (O.S.S.C.N): *Tenho esperança que o artigo seja aceito*
- Apositivas (O.S.S.A.): *Algo é fato, que a gente trabalhou muito para isto*

Oração Subordinada Adjetiva

São orações que modificam um substantivo. A ligação é feita com ajuda de um pronome relativo *que*, por exemplo:

- *O projeto que era mais dificil acabou nesta semana*
- *O teste que tinha menos erros falhou no ultimo momento*

Também pode ter uma preposição do lado do pronome relativo, por exemplo:

- *O lugar a que vamos ainda está longe*
- *O celular de que gostas chegou no pais*

Podem estar no meio de pausas também, por exemplo:

- *O cavalo, que ganhou a ultima corrida, saiu nas capas das revistas*

Oração Subordinada Adverbial

Podem exercer uma função de advérbio mesmo ou de locução adverbial:

- *O prazo foi adiado para ter mais tempo*

Podem ser também:

- Causais: *Acabei o mestrado porque me esforcei*
- Comparativas: *Os comentários de Marcelo foram tão bons como os de Flávio*

6.2.2 Orações coordenadas

São orações independentes que forman uma sequência. Por exemplo:

- *Branca de Neve crescia e ficava mais bonita.*

Aqui temos duas orações:

- *Branca de Neve crescia*
- *ficava mais bonita*

Quando as orações coordenadas possuem um *'-'* como conector, chamam-se *intercaladas*. Por exemplo:

- *Uma mordida nesta maçã fará Branca de Neve dormir para sempre - disse a bruxa.*

Caso contrário, as orações coordenadas com conetivos chamam-se *conectivas*. Estas últimas podem ser de diferentes tipos:

- aditivas:

  Faremos nossos trabalhos neste fim de semana *e também* seremos rápidos.

  Não sou sábio *nem* pretendo sê-lo.

- adversativas:

  Jogaram bem, *mas* perderam como sempre.

  Ora você sai com os amigos, *ora* você fica em casa.

- alternativas:

  Ou é verdadeira *ou* é falsa.

  Pode ir na esquerda *ou* na direita.

- conclusivas:

  Eu não estudei pra prova, *logo*, tirei baixa nota.

  Me esforcei muito, *portanto* espero uma grande recompensa.

- explicativas:

  Os humanos não voam *porque* não são aves.

  A comida está gostosa, *pois* eu que fiz.

6.3 Gramática Gerativa

Na linguística existem três níves de representação, os quais foram bem introduzidos por Costa-Campos e Xavier (1991):

- O nível 1, que é inacessível pelo linguísta, possui os mecanismos e as representações abstratas das atividades da linguagem. Este nível é adquirido desde a infância com o mundo que nos rodeia. Aqui são geradas sequências linguísticas, que podem ser observadas no nível 2.

- A partir do nível 2, o linguista tem acesso e formular hipóteses sobre o nível 1. Aqui podemos falar de formas linguísticas, por exemplo *casa* dá a idéia de um imóvel onde podem morar pessoas, uma frase com *casa* daria a idéia da interação dela com outros objetos.

- No nível 3, é construído um sistema de representação metalinguístico para responder pela relação entre as sequências do nível 2 e os mecanismos ou representação do nível 1. Por exemplo, quando realizamos uma análise gramatical de uma oração: *casa* - substantivo.

De acordo com Ruwet (1975), uma gramática gerativa é uma gramática explícita, que enumera explicitamente todas e não mais que as frases gramaticais de uma língua, assim como suas descrições estruturais.

Para Lagunilla e Rebollo (2004), uma gramática *explícita* consiste em um modelo de descrição sistemática da competência dos falantes de uma certa língua. O termo *competência* de acordo com Perini (1976), designa o conjunto de normas ou regras que nos permite emitir, receber e julgar enunciados em uma certa língua.

Chomsky (1957) indicou que o verdadeiro objetivo da linguística deveria de ser *a formulação de uma gramática que, por meio de um número finito de regras, fosse capaz de gerar todas as frases de um idioma, do mesmo modo que um falante pode formar um número infinito de frases em sua língua, mesmo quando nunca as tenha ouvido ou pronunciado.*

Nesse trabalho Chomsky define dois níveis: o superficial e o profundo, e determina o conjunto de transformações para ir de um nível a outro. A seguir estudaremos uma ferramenta para a análise de frases.

6.3.1 Análise de Constituintes

De acordo com Silva e Koch (2001), o sintagma é um conjunto de elementos, com valor significativo dentro da oração e que mantém entre si uma relação de dependência e de ordem. Existem os diferentes tipos:

- Sintagma Nominal (SN), cujo núcleo é o substantivo.

- Sintagma Verbal (SV), cujo núcleo é o verbo.

- Sintagma Adjetival (SA), cujo núcleo é um adjetivo.
- Sintagma Preposicional (SP), que é um SN acompanhado de uma preposição.

A estrutura sintática é uma configuração formada por unidades sintáticas, chamadas *categorias*, de diferentes classes (nome, verbo, adjetivo, preposição, flexão, sintagma nominal, sintagma verbal, etc.), entre as quais, estabelecem-se dois tipos de relações fundamentais: domínio e precedência.

Quando uma frase é analisada, o linguista constrói um indicador sintagmático, que não é mais que um gráfico em forma de árvore que representa a estrutura sintática da frase. Podemos ver na Figura 6.1 o indicador sintagmático da frase *A formiga comeu a folha*.

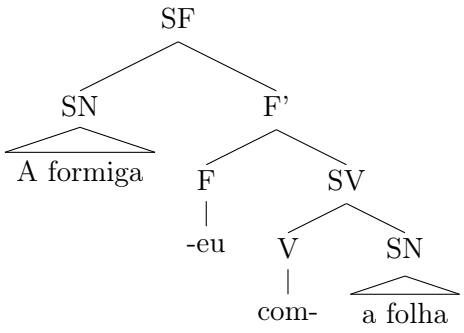
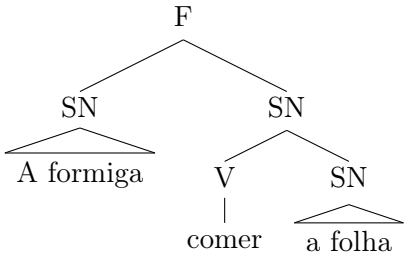


Figura 6.1: *Exemplo de indicador sintagmático*

Notemos que a inflexão verbal *eu*, afixo do pretérito de *comer*, foi deslocada à esquerda do verbo, aliás, é anterior ao sintagma verbal. Tal inflexão possui um nível a mais para indicar o tempo em que acontece o SV.

O conjunto de regras para obter uma estrutura superficial, que seja mais simples, se chama *transformações*. Aplicando tais transformações, podemos obter:



A transformação mais simples é a *supressão de sujeito idêntico* (SSI), relatada por Perin, que consiste em suprimir o sujeito de uma oração subordinada quando este for idêntico a qualquer outro SN presente na oração. Na frase:

- [Antônio querer [Antônio sambar com a Portela]]

Com a SSI, a frase ficaria simplificada em:

- [Antônio quer sambar com a Portela]

Neste capítulo revisamos conceitos da gramática portuguesa. Vimos que as orações podem ser de período simples e composto. Também estudamos brevemente a Gramática Gerativa e vimos como a Análise de Constituintes nos permitiu apreciar a riqueza sintática contida em uma simples oração. No próximo capítulo descrevemos os experimentos e finalmente concretizamos dentro de uma GR o estudado até aqui sobre análise sintática .

Capítulo 7

Experimentos e Resultados

Para poder avaliar nosso trabalho, escolhemos textos sobre três temas nos domínios de engenharia de software, conto de fadas e biologia. As ontologias foram aprendidas e submetidas à avaliação usando o modelo de características de conteúdo proposta no método *OntoMetrics*.

7.1 Descrição dos textos

Engenharia de Software: Métodos Ágeis de Desenvolvimento de Software

Consiste na transcrição de um conjunto de palestras sobre métodos ágeis, ministradas e registradas pelo grupo AgilCoop [1] na Universidade de São Paulo desde 2007. O texto final com a união de todas as transcrições possuia 11098 palavras.

Conto de Fadas: Branca de Neve e os Sete Anões

A versão usada é a adaptação no português do conto dos irmãos Grimm no http://www.educacional.com.br/projetos/ef1a4/contosdefadas/brancadeneve.html. O texto possuia apenas 837 palavras.

Biologia: Doença do Câncer

Usamos dois textos:

- O primeiro foi uma publicação pelo Instituto Nacional do Câncer - INCA, que fala do carcinoma de células escamosas da cabeça e pescoço (HNSCC), que é o quinto tipo de câncer mais comum mundialmente. Tal artigo foi publicado na Revista Brasileira de Cancerologia em 2009 Colombo e Rahal (2009). Este texto possui 6263 palavras.
- O segundo texto é um artigo do wikipedia [2], onde o documento descreve de forma geral os aspectos básicos do cancro (português europeu) ou câncer (português brasileiro). O texto possuia 7374 palavras.

7.2 Procedimentos

A seguir, descreveremos os passos realizados e as experiências em cada um deles.

[1] http://ccsl.ime.usp.br/agilcoop/agilvideo
[2] http://pt.wikipedia.org/wiki/Cancro

7.2.1 Extração de Termos

Usamos o Cogroo para lidar com a informação lingústica dos textos e experimentamos o seguinte:

- Antes da versão 3, a forma de usar o Cogroo era através de uma interface UIMA. Tal interface padronizava o Cogroo para poder ser utilizado por diferentes ferramentas. Isso possibilitava que, com o mesmo código, pudessemos trocar de Português a Inglês ou outras línguas.
- O Cogroo UIMA era um projeto ambicioso, porém, percebemos que apareciam alguns erros na etiquetação, ou seja, alguns elementos gramaticais não eram corretamente estabelecidos. Portanto decidimos re-projetar nossa arquitetura para trabalhar com o Cogroo diretamente sem usar o UIMA.
- Como qualquer sistema, o Cogroo as vezes não é preciso no reconhecimento de elementos gramaticais. Por exemplo, em certas ocasiões, o advérbio *aí* é marcado como um substantivo. A palavra *que* também é interpretada de diferentes formas, as vezes como um especificador, as vezes como um substantivo e as vezes como um pronome.
- O Cogroo 3 possui um *shallow parser* que retorna uma árvore sintática com sintagmas nominais, por exemplo, vemos na Figura 7.1 a árvore sintática da frase *A Branca de Neve vagou pela floresta.*

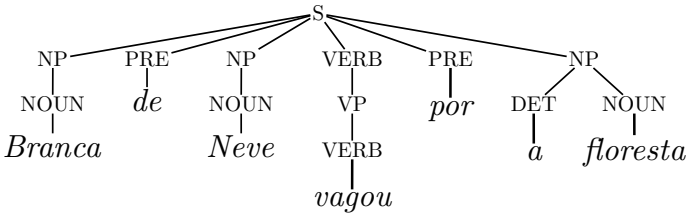
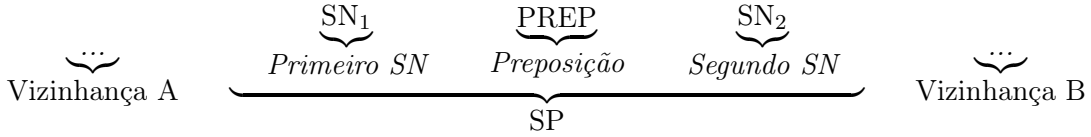


Figura 7.1: *Árvore sintática da frase* A Branca de Neve vagou pela floresta

Nós usamos a métrica NC/Value, detalhada na Seção 4.4, com as seguintes características:

- Usamos o Cogroo para realizar a lematização das palavras. Também o usamos como filtro linguístico para obter os sintagmas nominais e consequentemente os sintagmas preposicionais:
  - Como vimos na Figura 7.1, o Cogroo reconhece positivamente quais são os sintagmas nominais, mas para reconhecer os sintagmas preposicionais localizamos dois sintagmas nominais que possuam uma preposição entre eles, tais como:
  - 

  - Para cada SP, removemos os *stop-words* que estiverem no começo e no final, por exemplo a frase *cada sintoma na lista acima* ficaria reduzida a *sintoma na lista.*
  - As palavras contextuais do SN_1 são a última palavra da vizinhança A e a primeira palavra do SN_2. Similarmente, as palavras contextuais do SN_2 são a ultima palavra do SN_1 e a primeira palavra da vizinhança B. Finalmente, as palavras das vizinhanças são as palavras contextuais do SP.
- Enquanto aos *Stop Words*, nós consideramos uma lista de 150 palavras, entre símbolos de pontuação, adjetivos de posição, interjeições, conjunções, artigos, conectivos, etc. Vemos tal lista no Apéndice A.

Finalmente, a Tabela 7.1 mostra os primeiros quinze elementos escolhidos como termos para cada texto.

7.2.2 Descoberta de Relações

No desenvolvimento do presente trabalho, tentamos aplicar a estratégia do trabalho de Junior (2008), ou seja, aplicamos os padrões de Baségio (adaptação de Hearst) nos textos mas não encontramos resultados positivos, apenas uma ou até três relações reconhecidas em cada um. Esse resultado frustrante foi devido ao fato de que os padrões eram muito rígidos, por exemplo no seguinte texto:

```
A gente se depara com software que fez algo errado, que não funcionou
direto. Pode ser o seu Mac OS ou o seu Windows que tem que dar reboot dez
vezes, pode ser o seu Firefox que morre ou seu Internet Explorer, pode
ser o seu banco cujo site de Hong Bank não funciona ou não faz alguma
coisa que deveria fazer. Isso muitas vezes acontece por culpa de
software mal feito.
```

Nenhum padrão é reconhecido, porém, os termos *Mac OS* e *Windows* compartilham a mesma natureza, e deveriam ser reconhecidos como equivalentes, de forma similar, *Firefox* e *Internet Explorer*.

Com o uso da GR, conseguimos identificar as seguintes relações:

- Mac **dar** reboot _dez_vezes
- Windows **dar** reboot _dez_vezes
- Gente **deparar** software
- Software **faz** errado
- Software **not_funcionar** direto

A estratégia adotada é parecida a RelText (mencionado na Seção 5.2), já que estabelecemos verbos como relações enquanto que o domínio e a imagem são o sujeito e predicado da oração onde o verbo aparece. Na continuação descreveremos as regras usadas para analisar os textos.

Regras para uma GR na Língua Portuguesa

Visitamos cada elemento de uma sentença e tentamos identificar a estrutura *sujeito-verbo-objeto*, onde o *sujeito* e o *objeto* serão de tipo `Lista`, e o verbo será apenas de tipo `String`.

1. Quando encontramos um sintagma nominal (`SN`), consideramos a alternativa deste ser uma sequência destes e/ou Sintagmas Preposicionais (`SPs`), como por exemplo:

   - *O SCRUM, o XP e o Desenvolvimento Guiado por Funcionalidades são metodologias ágeis.*

   Neste exemplo, os sintagmas nominais SCRUM e XP são reconhecidos; e de igual maneira o `SP` *Desenvolvimento Guiado por Funcionalidades*.

   Finalmente, dependendo se o verbo já foi encontrado, será colocado na lista sujeitos (antes do verbo) ou na lista de objetos (depois do verbo).

2. Se encontrarmos um verbo, primeiro verificamos se o verbo principal já foi inicializado, e se ainda não foi, simplesmente estabelecemos o verbo encontrado como o principal da atual sentença; caso contrário, suponhamos que é uma oração de período composto e chamamos ao processo de análise recursiva, passando como parâmetro o primeiro elemento não vazio entre:

Métodos Ágeis de Desenvolvimento de Software	**Branca de Neve e os Sete Anões**	**Diagnóstico de câncer de pescoço e cabeça**	**Carcoma**
desenvolvimento de software	branca de neve	HNSCC	cancro
método ágil	coração em uma caixa	recidiva locorregional	câncer
software	neve	câncer de cabeça	célula normal
seis mês	belo de todo o mulher	superexpressão de EGFR	célula
engenharia de software	dia seguinte	ciclo celular	órgãos linfoides
cliente	príncipe para um castelo distante	paciente com HNSCC	câncer de pulmão
manifesto ágil	princesa para seu castelo	expressão de EGFR	câncer colorretal
gente	sete anão	carcinogênese de cabeça	paciente com câncer
três mil linha	lindo caixão de cristal	cabeça	cancro de o estómago
com	princesa branco de neve	metástase em linfonodos cervical	colo de o útero
dia a dia	animal de a floresta	enzima ciclo-oxigenases	artigo principal
desenvolvimento	passeio em a floresta	gene supressores	célula canceroso
método tradicional	branca de neve cair	célula escamoso de cabeça	tumor benigno
colaboração com o cliente	história para o anão	câncer	gene
cara de o negócio	dono de cabana	ativação de oncogeneses	dna

Tabela 7.1: *Lista de termos para os nossos textos testes*

(a) o último objeto do predicado,

(b) o sujeito da sentença atual, ou

(c) o sujeito recebido de uma sentença anterior analisada.

3. No caso de um advérbio, analisamos se ele é de negação como *não, nunca, jamais, nem, tampouco*, e se ele está do lado do verbo. Exemplo:

   - *O analista não pode falhar*

   Se não for de negação, consideramos a chance da frase ser de período composto verificando se tal advérbio é de causa (*porque, porquê, conseguintemente, consequentemente*); ou de exclusão (*apenas, exclusivamente, salvo, senão, simplesmente, só, somente, unicamente*); ou de inclusão (*ainda, até, inclusivamente, mesmo, nomeadamente, também), de lugar (aí, aonde, onde*); ou de modo (*aliás*). Exemplo:

   - *Deve-se entregar no prazo porque o cliente está exigindo.*

   Se não for algum dos tipos de advérbios anteriores, simplesmente o ignoramos.

4. Para o caso de uma preposição, verificamos se estamos tratando o caso de um verbo composto como por exemplo: Eu "gosto de viajar" no estrangeiro.

5. Se encontramos um adjetivo, então é considerado como um SN, e o tratamos como na primeira regra.

6. Se encontramos um especificador, como por exemplo "que", então estamos frente a uma frase subordinada, e chamamos recursivamente o algoritmo para tal frase.

7. Se encontramos uma conjunção, estamos frente a um caso de oração coordenada, porém, o tratamento será o mesmo que na regra anterior.

8. Se encontramos uma vírgula, então, pode ser o caso de outra frase e tratamos similarmente aos casos anteriores.

9. Qualquer outro caso é simplesmente ignorado.

7.2.3 Análise de Conceitos Formais

Testamos três diferentes formas de introduzir a ACF dentro do processo de construção da ontologia:

1. Construção de uma ontologia considerando os termos como *objetos* e os verbos como *atributos*

2. Com a mesma estratégia anterior, porém, usando etiquetas reduzidas.

3. Estabelecendo superconceitos em nossa ontologia já aprendida com nosso método de análise sintática.

7.2.4 Experimentando o uso da ACF

Além de usar a ACF para obter um superconceito a partir do reticulado, implementamos também uma forma de traduzir um reticulado a uma ontologia e experimentamos usar uma técnica de redução de etiquetas para conceitos formais.

Básicamente, nosso roteiro foi o seguinte:

1. Identificamos o termos principais e os adicionamos dentro do conjunto de *objetos*.

2. Reconhecemos os verbos que estavam junto a estes termos e construímos um Contexto Formal. Por exemplo:

   - $\underbrace{\textit{Branca de Neve}}_{\text{primeiro termo}}$ $\underbrace{\textit{dormiu}}_{\text{verbo}}$ na $\underbrace{\textit{floresta}}_{\text{segundo termo}}$.

   O atributo *dormir* é mapeado ao termo *Branca de Neve*, e o atributo *está_ dormido* é relacionado com o termo *floresta*. A regra para trocar a terminação AR e ER/IR do verbo por um sufixo ANDO ou INDO nem sempre é correta, por exemplo para o verbo *ir*, nossa regra geraria o atributo *esta_indo*, no entanto, o correto seria usar outro verbo mais adequado, de acordo com o contexto em que se fala. Por exemplo, se o sujeito está indo a outra cidade, então essa cidade recebe os objetos que possuem o atributo: *está_ chegando*.

3. Usamos o Galicia para construir o reticulado de conceitos e o traduzimos para um foramto de ontologia. Todos os conceitos formais foram traduzidas em classes da ontologia, tendo como etiquetas os números dos nós. Visualmente resultou difícil de compreender, por exemplo apreciamos na Figura 7.2 o reticulado de conceitos para o tema de Métodos Ágeis. Esse reticulado teve 155 conceitos sem redução de etiquetas.

4. Estabelecemos as relações tomando como domínio as extensões, e como imagem as intensões. A etiqueta recebeu uma palavra genérica (*active*) junto a um número, por exemplo a *active12* possue como domínio `Desenvolvimento` e como imagem *está_ dividido*, *está_ melhorado* e *terminar*.

O principal problema com o estabelecimento de relações é que alguns conceitos possuíam uma enorme quantidade de objetos ou de atributos, por exemplo o *active11* possuia apenas 1 domínio, mas 27 elementos na imagem.

A expressividade indicava que a ontologia estava escrita na família de lógica de descrição $\mathcal{ALU}$ e $\mathcal{ALC}$.

Devido à inviabilidade da avaliação com outros métodos, bem como ao fato de ter um menor poder de expressividade, descartamos esta técnica foi descartada e nos concentramos na análise sintática e no uso da ACF como ferramenta para estabelecer um conceito superior comum para dois ou mais conceitos.

Neste capítulo detalhamos os passos seguidos na realização das experiências. Descrevemos primeiro os textos analisados e depois falamos como foi realizada a extração de termos, depois a aplicação da GR e finalmente o uso do análise de conceitos. No próximo capítulo descrevemos o método usado para a avaliação das ontologias aprendidas.

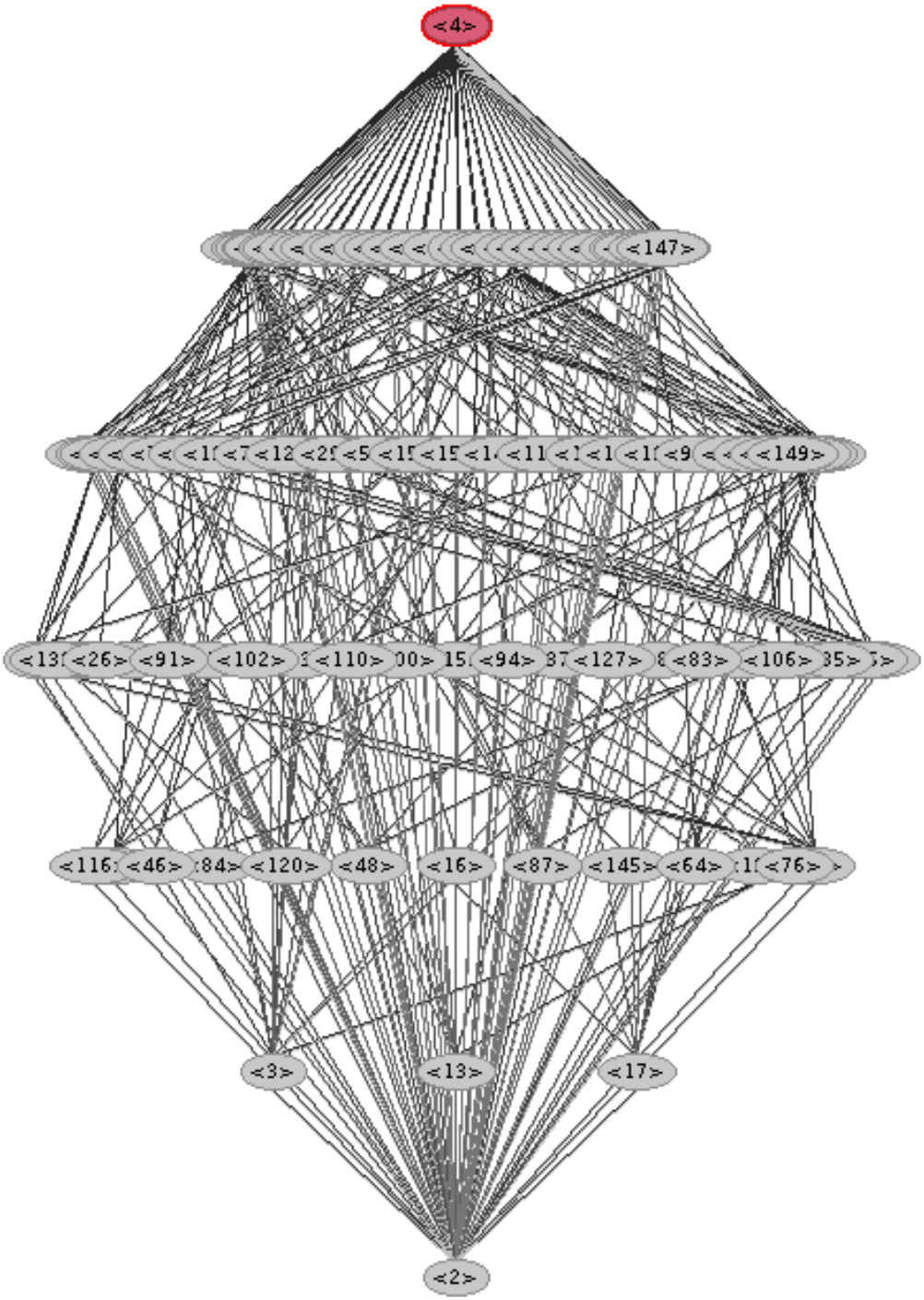


Figura 7.2: *Reticulado de conceitos para o tema de Métodos Ágeis*

Capítulo 8

Avaliação de Ontologias

Este capítulo foi dividido em três partes: (i) uma introdução nos métodos de avaliação de ontologias, (ii) detalhamos o Ontometrics e (iii) mostramos os resultados retornados pelo método OntoMetrics.

8.1 Introdução

É difícil definir o que é uma "boa ontologia", mas para alguns autores, como Freitas (2007), uma boa ontologia é apenas aquela que cumpre com os requisitos do sistema onde queremos aplicá-la.

Outra definição interessante também é a de Vrandečić (2010) que indica que a uma *boa ontologia* é aquela que tem a intensão de ser amplamente reutilizável e ser capaz de colaborar com outras aplicações.

Ante a preocupação por obter uma ontologia que represente da maneira mais fiel um certo domínio de conhecimento, os investigadores propuseram durante os últimos anos diferentes formas de avaliar tais ontologias. De acordo com o trabalho de Brank *et al.* (2005), existem as seguintes formas de avaliação de ontologias:

(i) *baseando-se em uma referência dourada*, onde a referência dourada é uma ontologia fruto da síntese de outras ontologias de referência, e da discussão com um especialista do domínio;

(ii) *usando a ontologia em uma aplicação*, onde respondemos as perguntas: satisfazemos as restrições do sistema?. Resolve as consultas que os usuários precisam?;

(iii) *comparar com uma fonte de dados*, isto é, se a ontologia possui termos para conceitos e relações congruentes com o domínio estudado.

(iv) *feita por um humano*, que na verdade é feita por um especialista que determina se a ontologia representa realmente o domínio de estudo.

Devemos levar em consideração também que as ontologias desenvolvidas manualmente são diferentes daquelas que foram obtidas através de um método de aprendizado semi ou completamente automático e requerem outra forma de avaliação. O trabalho de Dellschaft e Staab (2008) analiza justamente diferentes enfoques para a avaliação de ontologias fruto de algum método de aprendizado. Podemos ver tal análise na Tabela 8.1.

8.2 OntoMetric

Lozano-Tello Lozano-Tello (2002), propôs na tese de doutorado o método *OntoMetric* para reconhecer que ontologia entre outras deve ser usada para um certo domínio.

Esse método usou a metodologia *METHONTOLOGY* para construir as ontologias de referência e descreveu também, um marco multinível de características onde 160 ítens foram agrupados em 5 dimensões para avaliar a idoneidade de uma ontologia:

Enfoques	Características
Baseado em tarefas	• Diferentes métricas de recuperação de informação são usadas. • A tendência deles é indicar o que falta, o que sobra e o que deve ser mudado na ontologia, no entanto, não existe uma só forma certa de avaliar tais coisas.
Baseado em corpus	• Úteis para avaliar se uma ontologia cobre efetivamente um certo domínio. • Parcialmente úteis pois as ontologias retornadas por estes métodos podem ser usadas como benchmark para avaliar as ontologias fruto de outras técnicas de AO.
Baseado em criterios	• Uma grande variedade de medicões podem ser estabelecidas. • Pluralidade de avaliações. Cada uma pode dar uma diferente ontologia ganhadora.

Tabela 8.1: *Enfoques usados na avaliacão de métodos de AO*

- conteúdo,
- linguagem de implementação,
- metodologia de desenvolvimento,
- ambientes de desenvolvimento, e
- custos de uso

As características da dimensão *conteúdo*, avaliam os elementos internos da ontologia. Possui os seguintes fatores:

- Conceitos.
- Relações.
- Taxonomia.
- Axiomas.

As características da dimensão *linguagem* englobam aspectos mais formais sobre lógica, atributos, instâncias, inferência e outras particularidades.

As características relacionadas com a dimensão de *metodologia de desenvolvimento* possuem fatores como:

- Precisão da Metodologia, onde se fala sobre partes técnicas no planejamento do projeto de desenvolvimento da ontologia.
- Usabilidade da Metodologia, onde verifica-se a claridade das instruções do projeto.
- Maturidade da Metodologia, para medir a magnitude do projeto.

As características relacionadas com a dimensão *ambiente de desenvolvimento* possuem os seguintes fatores:

- Prestações do Ambiente.
- Aspectos de Visualização.
- Aspectos de Edição.
- Aspectos de Interação.
- Aspectos Metodológicos.
- Aspectos Cooperativos.
- Aspectos de Tradução.
- Aspectos de Integração de Ontologias.

Finalmente, as características relacionadas com a dimensão *custos de uso* são destinadas a fins comerciais.

No presente trabalho estamos interessados apenas na idoneidade do conteúdo da ontologia, porém, a parte de axiomas não será considerada pois nosso método não considerou tais elementos. Vemos na Tabela 8.2 as características da dimensão *conteúdo*.

Código	**Característica**
	Fatores de Conceitos
C11	Os conceitos essenciais para o projeto encontram-se na ontologia
C12	Os conceitos essenciais para o projeto estão em níveis superiores da ontologia
C13	Os conceitos estão descritos adequadamente em linguagem natural
C14	A especificação formal dos conceitos coincide com a descrição em linguagem natural
C15	Os atributos descrevem de forma precisa os conceitos
C16	O número de conceitos é adequado para representar a ontologia
	Fatores de Relações
C21	As relações essenciais para o projeto estão na ontologia
C22	Os conceitos encontram-se relacionados corretamente
C23	As relações estão descritas adequadamente em linguagem natural
C24	As aridades das relações são adequadas para o projeto
C25	As relações tem especificadas suas propriedades formais
C26	O número de relações é adequado para representar a ontologia
	Fatores de Taxonomia
C31	Classificação dos conceitos desde várias perspectivas
C32	As relações Not-Subclass-Of são apropriadas
C33	As relações de partição disjunta são usadas apropriadamente
C34	As relações de partição exaustiva são usadas apropriadamente
C35	Profundidade máxima adequada
C36	A média de filhos por conceito é adequada

Tabela 8.2: *Características na dimensão conteúdo sem a avaliação do fator de axiomas.*

Fator	Métodos Ágeis	Branca de Neve	Doença do Câncer
Fatores de Conceito			
C11	4,3	4,7	4,4
C12	4,3	4	4
C13	4	4,7	3,7
C14	4,3	4,7	3,4
C16	4	4,7	4
Fatores de Relações			
C21	4	5	3,4
C22	4	4	3,2
C23	4,3	5	3,4
C24	3,7	4,3	3,4
C25	4,3	5	3,5
C26	4,3	4,7	4,1
Fatores de Taxonomia			
C31	4,3	5	3,7
C33	4,7	4,7	4,4
C35	4	3,7	3,4
C36	3	3,3	3

Tabela 8.3: *Média dos resultados dos questionários sobre os fatores de conceito, relações e taxonomia. O valor mínimo era* 1 *e o máximo* 5.

8.3 Avaliação com Ontometrics

As ontologias geradas com nosso método de análise sintático e ACF foram colocadas em formato web e um pequeno site foi feito para que outras pessoas as avaliem. Podemos ver algumas imagens no Apêndice B. Neste site foi incorporado um questionário para medir os Fatores de Conceitos, Relações e Taxonomia. Vemos um exemplo do questionário no Apêndice C.

Ressaltamos que a notação usou os lemas das palavras. Por tanto, os nomes das relações são verbos em infinitivo. Aqueles SN com adjetivos estarão em singular e gênero masculino. Quando uma negação for encontrada, um prefixo "not_" é adicionado e quando uma forma passiva é reconhecida, o prefixo "can_be" é adicionado no verbo.

- As pessoas que avaliaram a ontologia de Métodos Ágeis foram os membros do laboratório de métodos ágeis da USP, os mesmos que formaram AgilCoop.
- A ontologia da Doença de Câncer foi avaliada por estudantes de medicina cursando o terceiro ano e também por membros do grupo de pesquisa no Hospital do Câncer A.C. Camargo.
- Finalmente, a ontologia do conto de fada Branca de Neve foi avaliada pelo público em geral.

As respostas dos questionários estavam no intervalo de 1 a 5, e encontram-se na Tabela 8.3.

Observações

- O fator C15 (os atributos descrevem de forma precisa os conceitos) não foi considerado pois não existiu um aprendizado de atributos para os conceitos.
- Os fatores C32 e C34 (relação *Not-SubClasse* e partição exaustiva respectivamente), não foram considerados.

Capítulo 9

Conclusões

9.1 Discussões

Nesta seção, discutimos as ontologias que obtimos através do nosso método.

Sobre Métodos Ágeis

Dois exemplos ótimos encontrado foram de **Cliente** e **Programador**

$$\texttt{Cliente} \equiv \texttt{Figura\_Criativo} \sqcap \texttt{Fundamental} \sqcap \texttt{Escopo} \sqcap \texttt{Razão} \sqcap \texttt{Versão}$$

$$\texttt{Cliente} \sqsubseteq \textit{colaborar}\ \texttt{Processo} \sqcup \textit{conseguir\_mudar}\ \texttt{Tecnologia} \sqcup \textit{estar}\ \texttt{Feliz\_Com\_O\_Software} \sqcup \textit{estar}\ \texttt{Rio\_De\_Janeiro} \sqcup \textit{estar\_aprender}\ \texttt{Desenvolvedor} \sqcup \textit{estar\_aprender}\ \texttt{Sistema} \sqcup \textit{mudar}\ \texttt{Ideia} \sqcup \textit{pedir}\ \texttt{Sexto\_Feira} \sqcup \textit{pedir}\ \texttt{Sistema\_De\_Web} \sqcup \textit{solicitar}\ \texttt{Funcionalidade} \sqcup \textit{ter}\ \texttt{Mente} \sqcup \textit{ter}\ \texttt{Razão}$$

$$\texttt{Programador} \sqsubseteq \text{acreditar}\ \texttt{Método\_Ágil}$$

$$\texttt{Programador} \neq \textit{entender}\ \texttt{Pessoal} \sqcap \textit{entender}\ \texttt{Regra\_De\_Negócio}$$

- Resaltamos estes dois exemplos pois na verdade, o foco das palestras era explicar o papel das pessoas (clientes e programadores) no processo de desenvolvimento.
- Das relações de equivalência encontradas, as primeiras parecem corretas, mas `Razão` não parece. No entanto, é possível que uma empressa suponha corretamente que "a *razão* da empresa é o *cliente*"e isso validaria a equivalência. Finalmente `Escopo` e `Versão` não parecem ser equivalentes ao conceito de `Cliente`.
- Algumas relações taxonômicas só são verdade para casos específicos, por exemplo, *estar* `Feliz_Com_O_Software` e *estar* `Rio_De_Janeiro`. Outras relações só fazem sentido com um contexto, por exemplo, *ter* `Mente` só faz sentido na sentença "o *cliente* tem em mente uma ideia do que quer", mas mesmo assim, tal frase não justifica a necesidade de estabelecer essa relação.

Sobre Branca de Neve

Um conceito interessante para analizar é a `Pele`, a qual possui as seguintes equivalências:

$$\texttt{Pele} \equiv \texttt{Cabelo\_Preto} \sqcap \texttt{Lábio\_Vermelho} \sqcap \texttt{Neve}$$

Tais equivalências foram encontradas a partir da frase:

`Sua pele era branca como a neve, os lábios vermelhos como o sangue e os cabelos pretos como o ébano.`

Na frase, `Pele` foi reconhecido como sujeito, **era** como verbo, e o predicado foi considerado como uma sequência de conceitos, estabelecendo uma relação de equivalência com o sujeito.

Sobre Doença do Câncer

Encontramos comentários muito interessantes de um especialista, que comentou algumas relações:

`Aquisição_De_Um_Fenótipo_Angiogênico` $\equiv$ `Ocorrer_Tumor`

`"fenótipo angiogênico é a característica de alguns tumores em que há criação de novos vasos sanguíneos. Os dois termos não são equivalentes, mas definitivamente estão relacionados."`

`Fumo` $\sqsubseteq$ *constituir* `Câncer_de_Pescoço`

`"e fato, um dois fatores mais importantes para câncer de pescoço (e todos os de cabeça) é o fumo."`

`Tumor` $\sqsubseteq$ *adquirir* `Propriedade_Invasiva` $\sqcup$ *apresentar* `Estadiamento_Clínico`

`"bem interessante; Estadiamento clínico é uma variável que é determinada pelo médico para avaliar a gravidade de um tumor; e o tumor (benigno) que adquire propriedades invasivas torna-se um câncer (maligno)."`

9.2 Considerações Finais

- A extração de termos no final não representou um elemento principal no trabalho, pois se quisermos ser mais precisos perderemos muitos termos importantes. Por exemplo, é impossível criar uma ontologia de métodos ágeis com apenas 10 termos, são necessários mais.
- A tese foi tratada de forma multidisciplinar, abrangido temas sobre estatística, linguística, grafos e ontologias.
- De forma similar ao trabalho de Jean-Pierre *et al.* (1997), a imprecisão dos resultados deve-se por causa das diferentes fontes de erros:
  - Erros por causa do Cogroo, por exemplo:
    - Um erro muito comum era a indentificação do adverbo *aí* como substantivo. Porém, tal erro era correto em frases como por exemplo: *Deixarei a nota **aí**.*
    - Outro erro muito comúm era a identificação da palavra *pode* como conjugação do verbo **podar**, enquanto o verbo correto deveria ser **poder**.
  - Erros por causa da fonte:

  - ∗ O texto de Métodos Ágeis possui muitas estruturas informais por serem parte de uma palestra, o que dificulta o processo de aprendizado. Como por exemplo: "*A, isso que eu chamo colaborar*". "*Vamos tocar o barco*".

- Reforzamos o problema de não ter tido ferramentas prontas para a extração de termos, reconhecimentos de relações, etc.

- Os resultados que apreciamos na Tabela 8.3 mostram que as ontologias retornadas por nosso método foram avaliadas positivamente pelos especialistas. As duas ontologias para a doença do Câncer obteve os valores mais baixos, porém os comentários dos especialistas indicaram que houveram coisas pontuais que relatavam fatos verdadeiros sobre o tema e que davam valor a nossos resultados.

- O desafio inicial foi atingido, que era propor uma ferramenta que consiga extrair uma ontologia de um texto mas não de forma automática e soberana, mas sim de forma supervisionada e dependente de um especialista. O resultado foi uma ontologia inicial sem instâncias que, de forma simples, descreve certas características do domínio analisado.

9.3 Sugestões para Pesquisas Futuras

Acreditamos que a descoberta de relações requer uma maior atenção. Os recursos linguísticos são muito extensos, existem diferentes gramáticas e diferentes formalismos. Um melhor estudo desta informação de um ponto de vista prático seria necessário. Assim como o tema de ontologias foi revivido em 2004 a partir de conceitos filosóficos muito anteriores à época, acreditamos que a representação de conhecimento pode ser muito mais enriquecida se aplicássemos mais a teoria linguística.

Também pensamos que uma fonte de informação semântica seria muito útil, mas não apenas um dicionário como o Amazonas, se não um lexicón. Um lexicón poderia abrir muito mais os caminhos das pesquisas relacionadas com a representação de conhecimento.

Finalmente, a tradução do reticulado a uma forma de ontologia não foi estudada apropriadamente, ela requer ainda um estudo mais profundo. Vimos que com diferentes critérios, podemos criar diferentes ontologias.

Apêndice A

Lista de Stop-Words

Total de 150 palavras:

.	bem	esses	não	sob
,	boa	em	nome	sobre
:	bom	embaixo	nosso	somente
/	cada	enquanto	novo	tem
-	cima	então	o	tal
*	com	este	onde	também
+	como	estes	os	têm
’	comprido	eu	ou	tempo
a	conhecido	fim	outra	tenho
ai	corrente	horas	outras	teu
as	da	isso	outro	tipo
à	dai	ista	outros	todo
às	das	iste	para	todos
acima	de	isto	parte	tu
acerca	debaixo	ligado	pela	tudo
agora	dentro	maioria	pelas	um
algum	desde	maiorias	pelo	uma
alguma	desligado	mais	pelos	umas
algumas	deve	mas	por	uns
alguns	devem	mesmo	porque	valor
ali	deverá	meu	que	veja
ambos	direita	minha	qual	verdade
antes	dois	minhas	qualquer	verdadeiro
aquela	dos	muita	quando	vós
aquelas	e	muitas	quem	você
aquele	ela	muito	quito	vocês
aqueles	ele	muitos	se	
aqui	eles	no	sem	
assim	essa	nosso	seu	
atrás	essas	nossos	seus	
até	esse	nós	sim	

Apêndice B

AOntoWEB

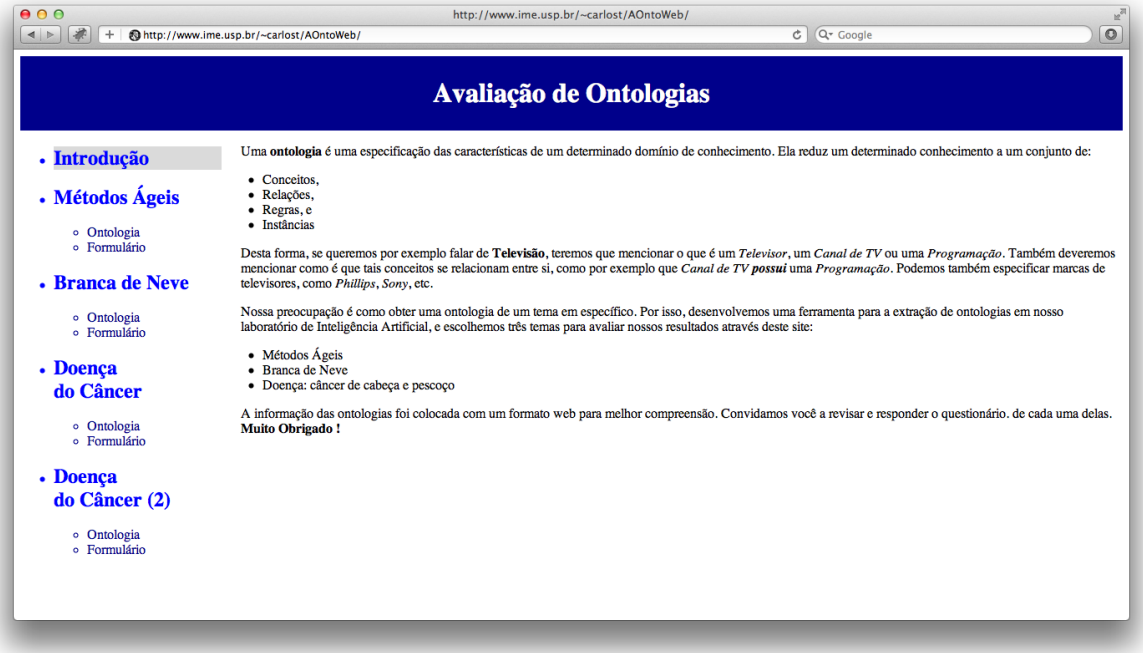


Figura B.1: *AOntoWeb - Página Inicial*

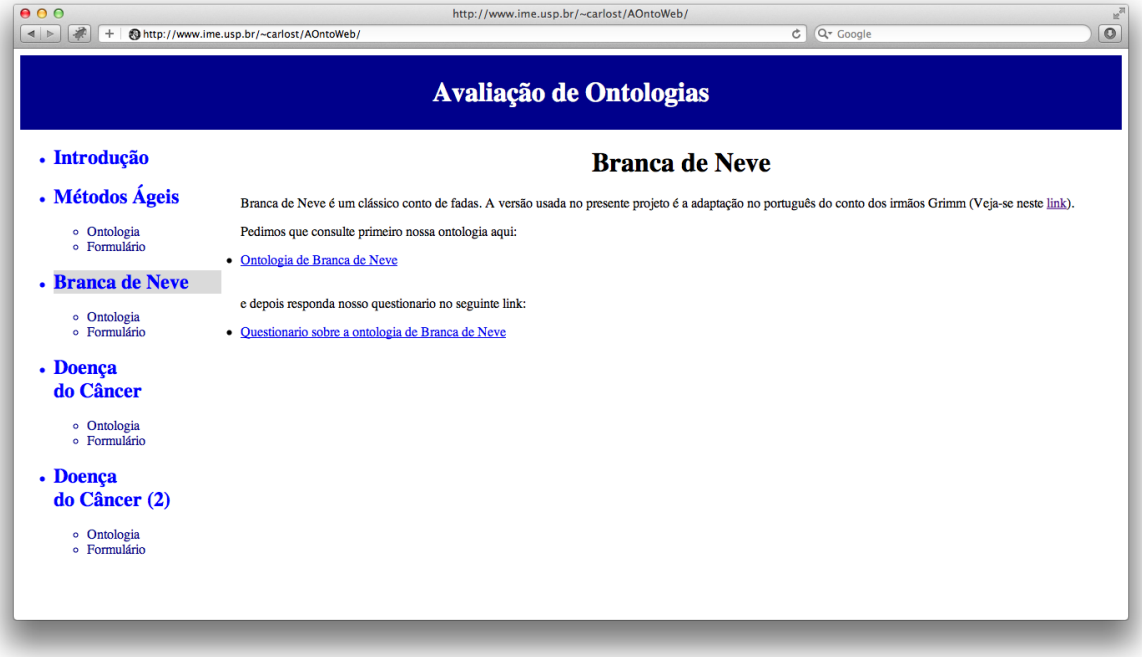


Figura B.2: *AOntoWeb - Introdução à avaliação da ontologia de Branca de Neve*

Figura B.3: *AOntoWeb - Branca de Neve - Termos*

Figura B.4: *AOntoWeb - Branca de Neve - Relações de Equivalência*

Figura B.5: *AOntoWeb - Branca de Neve - Relações Taxonómicas*

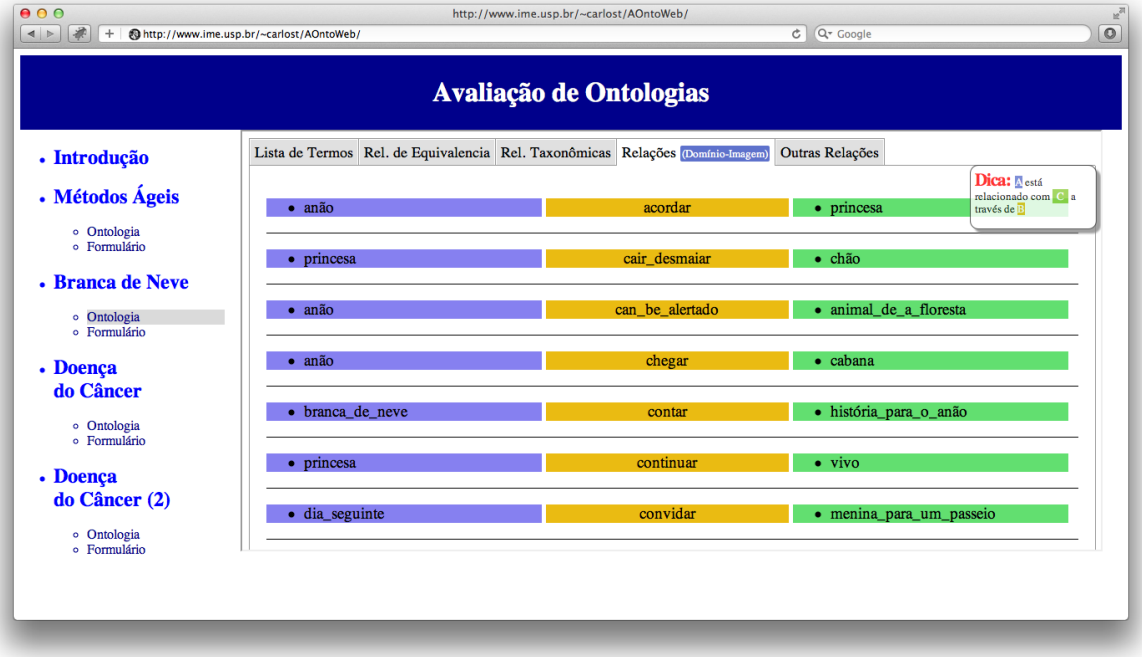


Figura B.6: *AOntoWeb - Branca de Neve - Dominio é Imagem*

Figura B.7: *AOntoWeb - Branca de Neve - Outras*

Apêndice C

Questionário sobre o Marco de Características da Dimensão Conteúdo da Ontologia

Exemplo de Branca de Neve

1. Lembre que este é um questionário para avaliar a idoneidade da ontologia gerada pelo nosso sistema.
2. Valores baixos indicam um retorno pobre
3. Valores altos indicam uma grande riqueza de informação

C11 Os conceitos essenciais se encontram presentes na ontologia?

(quando falamos de conceitos, nos referimos a lista de termos)

	1	2	3	4	5	
Muito baixo	○	○	○	○	○	Muito alto

C12 Os conceitos essenciais estão nos níveis superiores da ontologia?

(os conceitos mais relevantes apareceram entre os primeiros elementos da lista de termos)

	1	2	3	4	5	
Muito baixo	○	○	○	○	○	Muito alto

C13 Os conceitos encontram-se descritos em linguagem natural?

(os nomes dos termos ficaram claros?)

	1	2	3	4	5	
Muito baixo	○	○	○	○	○	Muito alto

C14 A especificação formal dos conceitos casa com a descrição em linguagem natural?

(a forma em que os conceitos apareceram nas relações foi correto?)

	1	2	3	4	5	
Muito baixo	○	○	○	○	○	Muito alto

C16 O número de conceitos é adequado?

(a quantidade de termos foi boa?)

	1	2	3	4	5	
Muito baixo	○	○	○	○	○	Muito alto

C21 As relações essenciais estão na ontologia?

(as relações entre conceitos foram suficientes para falar de Branca de Neve?)

C22 Os conceitos estão correctamente relacionados?

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C23 As relações estão escritas convenientemente em linguagem natural?

(conseguiu entender os nomes das relações?)

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C24 ariedade das relações são apropriadas para o projecto?

(Padre => n-Tem => Hijos) - Opcional

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C25 As relações possuem suas propriedades formais especificadas?

(O domínio e a imagem está correctamente definida?)

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C26 O número de relações é adequado para o projecto?

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C31 A classificação dos conceitos desde várias perspectivas é a correta?

(A taxonomia está correta?)

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C33 A relação de disjunção é usada convenientemente para as necessidades do sistema?

(As relações de "não é"estavam corretas?)

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C35 A profundidade máxima na hierarquia de conceitos é a adequada?

Opcional

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

C36 A média de filhos por conceitos é a adequada?

(algumas relações taxonômicas possuem mais de um elemento, a média deles é adequada?)

	1	2	3	4	5	
Muito baixo	◯	◯	◯	◯	◯	Muito alto

Referências Bibliográficas

Baségio(2006) Túlio Lima Baségio. Uma Abordagem Semi-automática para Identificação de Estruturas Ontológicas a partir de Textos na Língua Portuguesa do Brasil. Dissertação de Mestrado, Pontificia Universidade Católica do Rio Grande do Sul, Rio Grande do Sul. Citado na pág. 21

Bechara(2009) Evanildo Bechara. *Moderna Gramática Portuguesa, atualizada pelo novo Acordo Ortográfico*. 37 edição. Citado na pág. 25

Berners-Lee e Fischetti(1999) Tim Berners-Lee e Mark Fischetti. *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco. ISBN 0062515861. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0062515861. Citado na pág. 1

Bick(2000) E. Bick. *The Parsing System"Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press. Citado na pág. 8

Blomqvist(2005) Eva Blomqvist. Fully automatic construction of enterprise ontologies using design patterns: Initial method and first experiences. Em *OTM Conferences (2)*, páginas 1314–1329. Citado na pág. 6

Botero e Ricarte(2008) Sergio William Botero e Ivan L. M. Ricarte. Extração de relações semânticas via análise de correlação de termos em documentos. Em *Brazilian Symposium in Information and Human Language Technology - STIL*. Citado na pág. 22

Brank *et al.***(2005)** Janez Brank, Marko Grobelnik e Dunja Mladenic. A survey of ontology evaluation techniques. Em *In In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005*. Citado na pág. 1, 41

Cabré(1999) M. T. Cabré. *La terminologia: representación y comunicación*, volume 1a ed. Citado na pág. 15

Chomsky(1957) N. Chomsky. *Syntactic structures*. Janua linguarum: Series minor. Mouton. URL http://books.google.com.br/books?id=55YaAAAAIAAJ. Citado na pág. 30

Cimiano(2006) Philipp Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 0387306323. Citado na pág. 6, 9, 21

Cimiano e Volköer(2005) Philipp Cimiano e Johanna Volköer. Text2onto - a framework for ontology learning and data-driven change discovery, 2005. Citado na pág. 6

Cimiano *et al.***(2003)** Philipp Cimiano, Steffen Staab e Julien Tane. Automatic acquisition of taxonomies from text: Fca meets nlp. Em *Proceedings of the ECML / PKDD Workshop on Adaptive Text Extraction and Mining*, páginas 10–17, Cavtat-Dubrovnik, Croatia. URL http://www.dcs.shef.ac.uk/~fabio/ATEM03/cimiano-ecml03-atem.pdf. Citado na pág. 1, 11

Colombo e Rahal(2009) Jucimara Colombo e Paula Rahal. Alterações genéticas em câncer de cabeça e pescoço. *Revista Brasileira de Cancerologia*, páginas 165–174. Citado na pág. 33

Costa-Campos e Xavier(1991) Maria Henriqueta Costa-Campos e Maria Francisca Xavier. *Sintaxe e Semântica do Português*. Universidade Aberta. Citado na pág. 30

Deane(2005) Paul Deane. A nonparametric method for extraction of candidate phrasal terms. Em *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, páginas 605–613, Morristown, NJ, USA. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1219840.1219915. Citado na pág. 16

Dellschaft e Staab(2008) Klaas Dellschaft e Steffen Staab. Strategies for the evaluation of ontology learning. Em *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, páginas 253–272, Amsterdam, The Netherlands, The Netherlands. IOS Press. ISBN 978-1-58603-818-2. URL http://dl.acm.org/citation.cfm?id=1563823.1563842. Citado na pág. 41

Dias(2004) Maria Abadia Lacerda Dias. Extração automática de palavras-chave na língua portuguesa aplicada a dissertações e teses da área das engenharias. Dissertação de Mestrado, Universidade Estadual de Campinas - UNICAMP. Citado na pág. 16

Drumond e Girardi(2008) Lucas Drumond e Rosario Girardi. A survey of ontology learning procedures. Em *WONTO*. Citado na pág. 1, 5

Duan *et al.***(2009)** Jianyong Duan, Ru Li e Yi Hu. A bio-inspired application of natural language processing: A case study in extracting multiword expression. *Expert Syst. Appl.*, 36(3):4876–4883. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2008.05.046. Citado na pág. 15, 16

Forgy(1982) Charles Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligences*, 19(1):17–37. ISSN 0004-3702. URL http://dx.doi.org/10.1016/0004-3702(82)90020-0. Citado na pág. 5

Frantzi *et al.***(1998)** Katerina Frantzi, Sophia Ananiadou e Junichi Tsujii. The c-value/nc-value: Method of automatic recognition for multi-word terms. Em *Research and Advanced Technology for Digital Libraries*, volume 1513 of *Lecture Notes in Computer Science*, páginas 520–520. Springer Berlin / Heidelberg. Citado na pág. 18

Freitas(2007) Maria Cláudia De Freitas. *Elaboração automática de ontologias de domínio. Discussão e resultados*. Tese de Doutorado, Pontificia Universidade Católica do Rio de Janeiro. Citado na pág. 41

Ganter(1984) Bernhard Ganter. Two basic algorithms in concept analysis. FB4–Preprint 831, TH Darmstadt. Citado na pág. 9

Ganter e Wille(1999) Bernhard Ganter e Rudolph Wille. *Formal concept analysis: Mathematical foundations*. Springer, Berlin-Heidelberg. Citado na pág. 9

Godin *et al.***(1991)** R. Godin, R. Missaoui e H. Alaoui. Learning algorithms using a galois lattice structure. Em *Tools for Artificial Intelligence, 1991. TAI '91., Third International Conference on*, páginas 22 –29. doi: 10.1109/TAI.1991.167072. Citado na pág. 2, 11

Grefenstette(1997) Gregory Grefenstette. Sqlet: Short query linguistic expansion techniques, palliating one-word queries by providing intermediate structure to text. Em *In Proceedings of the RIAO'97*, páginas 500–509. Springer Verlag. Citado na pág. 21

Gruber(1993) Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220. ISSN 1042-8143. doi: 10.1006/knac.1993.1008. URL http://dx.doi.org/10.1006/knac.1993.1008. Citado na pág. 3

Haav(2004) Hele-mai Haav. A semi-automatic method to ontology design by using fca. Em *University of Ostrava, Department of Computer Science*. Citado na pág. 12

Harris(1968) Z.S. Harris. *Mathematical structures of language*. Wiley. Citado na pág. 22

Hearst(1992) Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. Em *In Proceedings of the 14th International Conference on Computational Linguistics*, páginas 539–545. Citado na pág. 21

Hitzler(2011) Pascal Hitzler. What's happening in semantic web: and what fca could have to do with it. Em *Proceedings of the 9th international conference on Formal concept analysis*, ICFCA'11, páginas 18–23, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-642-20513-2. URL http://dl.acm.org/citation.cfm?id=2008557.2008560. Citado na pág. 11

Ivan Kostial(2003) Jan Paralic Ivan Kostial. I.: Ontology-based information retrieval. *Information and Intelligent Systems, Croatia*, páginas 23–28. Citado na pág. 3

Jean-Pierre *et al.***(1997)** Salah Ayt-Mokhtar Jean-Pierre, Salah Ait-mokhtar e Jean pierre Chanod. Subject and object dependency extraction using finite-state transducers. Em *In Proceedings of the ACL/EACL'97 Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources*, páginas 71–77. Citado na pág. 46

Junior(2008) Luiz Carlos Ribeiro Junior. OntoLP: Construção Semi-Automática de Ontologias a partir de Textos da Língua Portuguesa. Dissertação de Mestrado, Universidade do Vale do Rio dos Sinos, São Leopoldo. Citado na pág. 1, 8, 18, 35

Karlsson(1995) F. Karlsson. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Natural Language Processing. Mouton De Gruyter. ISBN 9783110141795. URL http://books.google.com.br/books?id=70IvVPlH63cC. Citado na pág. 23

Kinoshita *et al.***(2007)** Jorge Kinoshita, Lais N. Salvador e Carlos E. D. Menezes. Cogroo - an openoffice grammar checker. Em *Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*, páginas 525–530, Washington, DC, USA. IEEE Computer Society. ISBN 0-7695-2976-3. URL http://portal.acm.org/citation.cfm?id=1317534.1318289. Citado na pág. 17

Lagunilla e Rebollo(2004) Mariana Fernández Lagunilla e Alberto Anula Rebollo. *Sintaxis y Cognición: Introducción a la gramática generativa*. Citado na pág. 25, 30

Lozano-Tello(2002) Adolfo Lozano-Tello. *Métrica de Idoneidad de Ontologías*. Tese de Doutorado, Universidad de Extremadura, Escuela Politécnica de Cáceres, Departamento de Informática. Citado na pág. 41

Maedche e Staab(2001) Alexander Maedche e Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79. ISSN 1541-1672. doi: http://dx.doi.org/10.1109/5254.920602. Citado na pág. 5

Maria da Graça Krieger(2004) Maria José Bocorny Finatto Maria da Graça Krieger. *Introdução a Terminologia: teoria & prática*. Citado na pág. 15

Mizoguchi(2003) Riichiro Mizoguchi. Part 1: introduction to ontological engineering. *New Gen. Comput.*, 21(4):365–384. ISSN 0288-3635. doi: http://dx.doi.org/10.1007/BF03037311. Citado na pág. 3

Noy e Mcguinness(2001) Natalya F. Noy e Deborah L. Mcguinness. Ontology development 101: A guide to creating your first ontology. Online, 2001. URL http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html. Citado na pág. 3

Pantel e Lin(2001) Patrick Pantel e Dekang Lin. A statistical corpus-based term extractor. Em *AI '01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, páginas 36–46, London, UK. Springer-Verlag. ISBN 3-540-42144-0. Citado na pág. 16

Peng e Zhao(2007) Xin Peng e Wenyun Zhao. An incremental and fca-based ontology construction method for semantics-based component retrieval. Em *QSIC, Seventh Quality Software International Conference*, páginas 309–315. Citado na pág. 11

Perini(1976) Mário A. Perini. *A Gramática Gerativa; Introdução ao Estudo da Sintaxe Portuguesa*. Belo Horizonte. Citado na pág. 30

Ruwet(1975) Nicolas Ruwet. *Introdução à Gramática Gerativa*. Universidade de São Paulo. Citado na pág. 30

Sag *et al.***(2001)** Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake e Dan Flickinger. Multiword expressions: A pain in the neck for nlp. Em *In Proc. of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002*, páginas 1–15. Citado na pág. 15

Sánchez(2009) David Sánchez. Domain ontology learning from the web. *The Knowledge Engineering Review*, 24(04):413. URL http://www.journals.cambridge.org/abstract_S0269888909990300. Citado na pág. 21, 22

Schutz e Buitelaar(2005) Er Schutz e Paul Buitelaar. Relext: A tool for relation extraction from text in ontology extension. Em *In: Proceedings of the 4th International Semantic Web Conference (ISWC). (2005*, página 05. Citado na pág. 22

Shamsfard e Abdollahzadeh Barforoush(2003) Mehrnoush Shamsfard e Ahmad Abdollahzadeh Barforoush. The state of the art in ontology learning: a framework for comparison. *Knowl. Eng. Rev.*, 18(4):293–316. ISSN 0269-8889. doi: http://dx.doi.org/10.1017/S0269888903000687. Citado na pág. 21

Shearer *et al.***(2008)** Rob Shearer, Boris Motik e Ian Horrocks. Hermit: A highly-efficient owl reasoner. Em Alan Ruttenberg, Ulrile Sattler e Cathy Dolbear, editors, *Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 EU)*, Karlsruhe, Germany. Citado na pág. 5

Silva e Koch(2001) Maria Cecilia Perez De Souza E Silva e Ingedore Grunfeld Villaça Koch. *Lingüìstica aplicada ao português: sintaxe*. São Paulo. ISBN 85-249-0204-3. Citado na pág. 30

Sirin *et al.***(2007)** Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur e Yarden Katz. Pellet: A practical owl-dl reasoner. *J. Web Sem.*, 5(2):51–53. Citado na pág. 5

Staab e Studer(2004) Steffen Staab e Rudi Studer. *Handbook on Ontologies*, chapter 1, páginas 1–23. International Handbooks on Information Systems. Springer. ISBN 3-540-40834-7. Citado na pág. 4

Tomokiyo e Hurst(2003) Takashi Tomokiyo e Matthew Hurst. A language model approach to keyphrase extraction. Em *Proceedings of the ACL 2003 workshop on Multiword expressions*, páginas 33–40, Morristown, NJ, USA. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1119282.1119287. Citado na pág. 16

Vrandečić(2010) Denny Vrandečić. *Ontology Evaluation*. Tese de Doutorado, KIT, Fakultät für Wirtschaftswissenschaften, Karlsruhe. Citado na pág. 41

Wang e He(2006) Jian Wang e Keqing He. Towards representing fca-based ontologies in semantic web rule language. Em *Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*, CIT '06, páginas 41–, Washington, DC, USA. IEEE Computer Society. ISBN 0-7695-2687-X. doi: http://dx.doi.org/10.1109/CIT.2006.186. URL http://dx.doi.org/10.1109/CIT.2006.186. Citado na pág. 1, 12

Wille(1982) Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. Em Ivan Rival, editor, *Ordered sets*, páginas 445–470, Dordrecht–Boston. Reidel. Citado na pág. 1

Wray(2002) Alison Wray. *Formulaic Language and the Lexicon*. Cambridge University Press. Citado na pág. 15

Yang e Callan(2008) Hui Yang e Jamie Callan. Metric-based ontology learning. Em *ONISW '08: Proceeding of the 2nd international workshop on Ontologies and nformation systems for the semantic web*, páginas 1–8, New York, NY, USA. ACM. ISBN 978-1-60558-255-9. doi: http://doi.acm.org/10.1145/1458484.1458486. Citado na pág. 6