THIAGO CASTRO FERREIRA

**A variação humana na geração de expressões de
referência**

São Paulo

2014

THIAGO CASTRO FERREIRA

A variação humana na geração de expressões de referência

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 19 de Setembro de 2014. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de Outubro de 2011.

Orientador: Prof. Dr. Ivandré Paraboni

São Paulo

2014

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

Biblioteca

Escola de Artes, Ciências e Humanidades da Universidade de São Paulo

Ferreira, Thiago Castro

A variação humana na geração de expressões de referência / Thiago Castro Ferreira ; orientador, Ivandré Paraboni. – São Paulo, 2014
116 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo
Versão corrigida

1. Linguagem natural. 2. Inteligência artificial. 3. Geração de expressões de referência. I. Paraboni, Ivandré, orient. II. Título.

CDD 22.ed. – 006.35

Dissertação de Mestrado sob o título “*A variação humana na geração de expressões de referência*”, defendida por Thiago Castro Ferreira e aprovada em 19 de Setembro de 2014, em São Paulo, Estado de São Paulo, pela banca julgadora constituída pelos doutores:

Prof. Dr. Ivandré Paraboni
Orientador

Profa. Dra. Sarajane Marques Peres
Universidade de São Paulo

Prof. Dr. Thiago Alexandre Salgueiro
Pardo
Universidade de São Paulo

*Dedico esta dissertação à minha família por todo apoio
prestado.*

Agradeço à toda minha família, em especial minha mãe Angela, meu pai Tomé, minha irmã Tatiane, minha avó Vicentina, minha prima Fernanda, meus padrinhos Luís Paulo e Conceição, e meus tios Mércio e Ivanir, pelo apoio incondicional dado em todos os momentos durante o decorrer deste projeto.

Ao meu orientador Ivandré pela paciência, parceria e total dedicação.

Aos meus amigos do Sion e agregados, em especial para Amanda Penna, Arthur Grava, Bruna Bú, Bruno Arrabal, Caroline Kerestes, Diogo Gouveia, Flávio Aldana, Juliana Biancheze, Kaíssa Nascimento, Mariana Biancheze, Renan Gomes, Renan Kenji, Renan Rodrigues, Vinícius Fernandes e Vitor Oliveira, pelo auxílio e ajuda mútua em todas as fases que um ser humano passa no decorrer dos 11 anos que estamos juntos.

Aos meus amigos do Itaú e agregados pelos diversos passeios em grupo, a admiração mútua e a crença que tudo acaba certo e bem.

Aos meus amigos da USP, entre eles professores e alunos, pelo aprendizado ao longo destes 6 anos que sou parte desta instituição.

Mas as coisas findas,
muito mais que lindas,
essas ficarão.

Carlos Drummond de Andrade

Resumo

FERREIRA, Thiago Castro. **A variação humana na geração de expressões de referência**. 2014. 116 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2014.

Este documento apresenta um estudo em nível de mestrado na área de Geração de Língua Natural (GLN), enfocando a questão da variação humana na tarefa de Geração de Expressões de Referência (GER). O trabalho apresenta um levantamento bibliográfico sobre o tema, a criação de dois algoritmos de GER e a construção de um novo corpus de expressões de referência. Modelos computacionais de GER baseados nos algoritmos criados foram implementados em versões que incorporam e não incorporam a variação humana e empregados em uma série de experimentos de GER em sete corpus de expressões de referência. Resultados comprovam a hipótese inicial de que algoritmos de GER que levam em conta a variação humana podem gerar expressões de referência mais próximas a descrições de seres humanos do que algoritmos que não levam esta questão em conta. Além disso, confirmou-se que algoritmos de GER baseados em técnicas de aprendizado de máquina mostram-se superiores a algoritmos de GER consagrados e amplamente utilizados na literatura, como o algoritmo Incremental.

Palavras-chave: Geração de língua natural. Geração de expressões de referência. Variação humana.

Abstract

FERREIRA, Thiago Castro. **The human variation in the referring expression generation task**. 2014. 116 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2014.

This work concerns a MSc Project in the field of Natural Language Generation (NLG), focusing on the issue of human variation in the Referring Expression Generation task (REG). The study presents a literature review on the topic, the proposal of two REG algorithms and the construction of a new corpus of referring expressions. Based on these algorithms, two REG models are implemented: with and without taking human variation. These models are employed in a series of REG experiments using seven referring expression corpora. Results confirm the initial hypothesis that REG algorithms that take speaker variation into account outperform existing algorithms that generate speaker-independent descriptions. Moreover, the present study confirms that algorithms based on machine learning techniques overperform existing algorithms, as the Dale and Reiter's Incremental algorithm.

Keywords: Natural language generation. Referring expression generation. Human variation.

Lista de Figuras

Figura 1	Contexto com dois cones e um cubo	18
Figura 2	Sentenças com suas expressões de referência destacadas em negrito ...	18
Figura 3	Algoritmo Incremental	26
Figura 4	Contexto com dois cones e um cubo na abordagem de grafos	28
Figura 5	Subgrafos que descrevem o objeto d_1 do contexto na figura 4	28
Figura 6	Algoritmo de GER baseado em grafos <i>Branch and Bound</i>	29
Figura 7	Exemplo de cena mostrada aos participantes no experimento do córpus TUNA no domínio TUNA Furniture	34
Figura 8	Exemplo de cena mostrada aos participantes no experimento do córpus TUNA no domínio TUNA People	34
Figura 9	Contexto de referência do córpus Drawer	35
Figura 10	Exemplo de cena mostrada ao participante no experimento do córpus GRE3D3	36
Figura 11	Exemplo de cena mostrada ao participante no experimento do córpus GRE3D7	39

Figura 12	Exemplo de cena mostrada ao participante no experimento do <i>córpus Stars</i>	41
Figura 13	Exemplos de cenas do <i>córpus Stars2</i> para cada condição investigada ..	42
Figura 14	Exemplos da mesma cena mostrada ao locutor e ao ouvinte no experimento de coleta do <i>córpus Stars2</i>	43
Figura 15	Algoritmo Incremental Estendido	59
Figura 16	Algoritmo para combinação das previsões dos classificadores	70
Figura 17	Contexto com dois cones e um cubo	70
Figura 18	Exemplo de cena mostrada ao participante no experimento do <i>córpus Zoom</i>	78

Lista de Tabelas

Tabela 1	Atributos relacionados a um objeto-alvo TG e seu ponto de referência LM nos corp�us GRE3D3 e GRE3D7	53
Tabela 2	Caracter�sticas de entrada �s �rvores de decis�o, relacionadas a um objeto-alvo TG e seu ponto de refer�ncia LM	53
Tabela 3	Classificadores dos corp�us GRE3D3 e GRE3D7, relacionados a um objeto-alvo TG, um ponto de refer�ncia LM e a rela��o entre eles para predic��o de uma descri��o EXP	64
Tabela 4	Classificadores dos corp�us <i>Stars</i> e <i>Stars2</i> , relacionados a um objeto-alvo TG, um 1� e um 2� ponto de refer�ncia LM e LM2, e a rela��o entre eles para predic��o de uma descri��o EXP	65
Tabela 5	Classificadores do corp�us TUNA Furniture, relacionados a um objeto-alvo TG para predic��o de uma descri��o EXP	66
Tabela 6	Classificadores do corp�us TUNA People, relacionados a um objeto-alvo TG para predic��o de uma descri��o EXP	66
Tabela 7	Classificadores do corp�us <i>Zoom-Pt</i> , relacionados a um objeto-alvo TG, um 1� ponto de refer�ncia LM e a rela��o entre eles para predic��o de uma descri��o EXP	67
Tabela 8	Caracter�sticas de Contexto dos corp�us GRE3D3/7, <i>Stars</i> e <i>Stars2</i> , relacionadas a um objeto-alvo TG e seu ponto de refer�ncia LM	68

Tabela 9	Características de Contexto do corpus TUNA Furniture, relacionadas a um objeto-alvo TG	68
Tabela 10	Características de Contexto do corpus TUNA People, relacionadas a um objeto-alvo TG	68
Tabela 11	Características de Contexto do corpus <i>Zoom-Pt</i> , relacionadas ao objeto-alvo TG	69
Tabela 12	Informações gerais do locutor	73
Tabela 13	Faixas etárias dos locutores	73
Tabela 14	Características que modelam as preferências linguísticas de um locutor na seleção de atributos e no grau de especificação de suas expressões de referência	73
Tabela 15	Variações do algoritmo <i>InteliGER</i>	77
Tabela 16	Número de descrições, participantes e cenas de cada corpus, utilizados como conjunto de dados	81
Tabela 17	Modelos computacionais de GER com e sem variação humana, considerados no teste de H_1	82
Tabela 18	Variações do algoritmo AIE e <i>InteliGER</i> , considerados no teste de H_2	83
Tabela 19	Variações do algoritmo <i>InteliGER</i> nas abordagens de treinamento <i>All</i> , <i>Speaker</i> e <i>Profile</i> , consideradas no teste de H_3	83

Tabela 20	Coeficiente de Dice médio e Desvio Padrão dos modelos computacionais de GER para H_1	85
Tabela 21	Teste estatístico de Wilcoxon sobre os coeficientes de Dice para H_1 ...	85
Tabela 22	Resultados dos experimentos para H_2	86
Tabela 23	Teste estatístico de Wilcoxon sobre os coeficientes de Dice para H_2 ...	86
Tabela 24	Resultados dos experimentos para H_3 e H_4	86
Tabela 25	Teste estatístico de Wilcoxon sobre os coeficientes de Dice para H_3 e H_4	86
Tabela 26	Possíveis valores dos atributos para anotação das expressões de referência da cena do córpis <i>Zoom</i> representada na figura 18	100
Tabela 27	Exemplo de anotação de contexto do córpis GRE3D7	101
Tabela 28	Resultado do experimento para os córpis GRE3D3/7	102
Tabela 29	Resultado do experimento para os córpis <i>Stars</i> e <i>Stars2</i>	102
Tabela 30	Resultado do experimento para os domínios do córpis TUNA	103
Tabela 31	Resultado do experimento para o córpis <i>Zoom-Pt</i>	103
Tabela 32	Resultados das Máquinas de Vetores de Suporte do córpis GRE3D3 ..	104
Tabela 33	Resultados das Árvores de Classificação e Regressão do córpis GRE3D3	105

Tabela 34	Resultados das Máquinas de Vetores de Suporte do córpus GRE3D7	.. 106
Tabela 35	Resultados das Árvores de Classificação e Regressão do córpus GRE3D7	107
Tabela 36	Resultados das Máquinas de Vetores de Suporte do córpus <i>Stars</i>	 108
Tabela 37	Resultados das Árvores de Classificação e Regressão do córpus <i>Stars</i>	.. 109
Tabela 38	Resultados das Máquinas de Vetores de Suporte do córpus <i>Stars2</i>	 110
Tabela 39	Resultados das Árvores de Classificação e Regressão do córpus <i>Stars2</i>	111
Tabela 40	Resultados das Máquinas de Vetores de Suporte do domínio TUNA Furniture	 112
Tabela 41	Resultados das Árvores de Classificação e Regressão do domínio TUNA Furniture	 112
Tabela 42	Resultados das Máquinas de Vetores de Suporte do domínio TUNA People	 113
Tabela 43	Resultados das Árvores de Classificação e Regressão do domínio TUNA People	 114
Tabela 44	Resultados das Máquinas de Vetores de Suporte do córpus <i>Zoom-Pt</i>	.. 115
Tabela 45	Resultados das Árvores de Classificação e Regressão do córpus <i>Zoom-Pt</i>	116

Sumário

1	Introdução	18
2	Objetivo	22
2.1	Objetivo Geral	22
2.2	Objetivos Específicos	22
3	Conceitos Fundamentais	23
3.1	Algoritmos de GER	23
3.1.1	<i>Full Brevity</i>	23
3.1.2	Algoritmo <i>Greedy</i>	24
3.1.3	Algoritmo Incremental	25
3.1.4	Algoritmos Baseado em Grafos	27
3.1.4.1	Algoritmo <i>Branch and Bound</i>	29
3.1.4.2	Algoritmo <i>Longest First</i>	30
3.1.5	Outras Abordagens	30
3.1.5.1	Geração de Expressões de Referência Relacionais	30
3.1.5.2	Geração de Expressões de Referência Superespecificadas	31
3.2	Córpus de Expressões de Referência	32
3.2.1	Córpus de GER de Propósito Geral	32
3.2.2	Córpus de GER Dedicados	33
3.2.2.1	Córpus TUNA	33
3.2.2.2	Córpus Drawer	34

3.2.2.3	Córpus GRE3D3	36
3.2.2.4	Córpus GRE3D7	37
3.2.2.5	Córpus <i>Stars</i>	40
3.2.2.6	Córpus <i>Stars2</i>	42
3.3	Avaliação Intrínseca dos Algoritmos de GER	44
3.4	Considerações	46
4	A Variação Humana em GER	48
4.1	Variação Humana nos corpú <i>Map Task</i> e <i>Coconut</i>	48
4.2	Variação Humana no corpú TUNA	50
4.3	Variação Humana na Seleção de Conteúdo e na Realização Textual de GER ..	51
4.4	Variação Humana nos corpú GRE3D3 e GRE3D7	52
4.5	Grafos e Relações Espaciais em GER	55
4.6	Considerações	57
5	Trabalho Realizado	58
5.1	Algoritmos Básicos	58
5.1.1	Algoritmo Incremental Estendido	58
5.1.1.1	Gerando Expressões de Referência Relacionais	59
5.1.1.2	Listas de Preferência de Atributos	60
5.1.1.3	Funcionamento do Algoritmo AIE	61
5.1.2	<i>InteliGER</i> : um algoritmo de GER baseado em aprendizado de máquina	63
5.1.2.1	Classificadores	64
5.1.2.2	Vetor de Características	66
5.1.2.3	Combinação das Predições	69
5.2	Incorporando a Variação Humana nos Algoritmos Básicos	72
5.3	Variações dos Algoritmos AIE e <i>InteliGER</i> consideradas	75

5.3.1	Variações do Algoritmo Incremental Estendido	75
5.3.2	Variações do Algoritmo <i>InteliGER</i>	75
5.4	Construção do Córpus <i>Zoom</i>	77
5.4.1	Coleta e Anotação dos Dados	77
6	Avaliação	80
6.1	Conjunto de Dados	80
6.2	Modelos Computacionais de GER Avaliados	81
6.3	Hipóteses	81
6.4	Procedimento	83
6.5	Resultados	85
6.6	Discussão	87
6.6.1	Análise de Erros	87
6.6.2	Comparação com trabalho prévio	91
7	Conclusão	93
	Referências	95
	Apêndice A	100
	Apêndice B	101
	Apêndice C	102
	Apêndice D	104

Capítulo 1

Introdução

Sistemas de Geração de Língua Natural (GLN) são modelos computacionais que, tendo tipicamente dados não-linguísticos como entrada, geram uma forma linguística como saída (REITER; DALE, 2000). Como exemplos de aplicações de GLN, encontram-se sistemas geradores de relatórios de previsão do tempo (KITREDGE et al., 1994; BELZ, 2008), diagnósticos médicos textuais para pacientes (BINSTED; CAWSEY; JONES, 1995; PORTET et al., 2007), entre outros.

Um componente essencial de sistemas de GLN é a tarefa de Geração de Expressões de Referência (GER) (KRAHMER; DEEMTER, 2012). Uma expressão de referência é uma forma linguística que descreve uma determinada entidade de modo que a mesma possa ser identificada unicamente em relação às demais entidades do contexto (REITER; DALE, 2000). No contexto representado na figura 1, “O cone pequeno”, “O cone grande” ou “O cubo” são exemplos de expressões de referência que poderiam ser geradas para referenciar as entidades d_1 , d_2 e d_3 , respectivamente. Já na figura 2, são mostradas algumas expressões de referência destacadas em suas sentenças.

Figura 1 – Contexto com dois cones e um cubo

Figura 2 – Sentenças com suas expressões de referência destacadas em negrito

O **cubo verde** foi criado pelo autor.
 Os **cones** estão na **figura 1**.
 Uma **expressão de referência** é uma forma linguística.
 O objetivo geral é propor **um modelo computacional de GER**.
 O primeiro algoritmo de GER foi o algoritmo *Full Brevity*.

Computacionalmente, a tarefa de GER é resolvida em duas fases, conhecidas como seleção de conteúdo e realização textual (PEREIRA; PARABONI, 2007, 2008; LUCENA;

PEREIRA; PARABONI, 2010; NOVAIS; PARABONI, 2013). Na seleção de conteúdo, é feita a seleção das características de uma entidade para distingui-la das demais entidades do contexto; e na realização textual ocorre a representação do conteúdo selecionado em língua natural. O presente trabalho trata em específico da fase de seleção de conteúdo, a qual será referenciada daqui em diante como sinônimo da tarefa de GER.

Apesar da tarefa de GER ser realizada naturalmente por seres humanos, sua resolução não é tão trivial quando realizada computacionalmente (KRAHMER; DEEMTER, 2012). Exemplos de algoritmos que realizam a tarefa de GER incluem os algoritmos *Full Brevity* (DALE, 1989), *Greedy* (DALE, 1989; DALE; HADDOCK, 1991), Incremental e suas extensões (DALE; REITER, 1995; KRAHMER; THEUNE, 2002; KELLEHER; KRUIJFF, 2006), baseado em grafos (KRAHMER; ERK; VERLEG, 2003), baseado em grafos conceituais (CROITORU; DEEMTER, 2007), baseado em fórmulas lógicas (GARDENT; STRIEGNITZ, 2007; ARECES; KOLLER; STRIEGNITZ, 2008), e outros (GARDENT, 2002; BOHNET; DALE, 2005). Neste trabalho, estes algoritmos serão referenciados como algoritmos de GER.

Para representar a tarefa computacional de GER, considera-se D o conjunto que representa as entidades (ou objetos) de um contexto. Cada um destes objetos contidos em D pode ser descrito por um conjunto de pares atributo-valor conhecidos como propriedades. Uma propriedade que diz respeito apenas ao objeto em questão é dita ser atômica, como $\langle \text{tipo}, \text{cone} \rangle$. Quanto a uma propriedade que relaciona dois objetos, essa é dita relacional, como $\langle \text{próximo de}, d_1 \rangle$. Um objeto em especial que se deseja descrever é chamado referente ou objeto-alvo. Os demais objetos, chamados distraidores, formam o contexto de referência C . Para exemplificar, considere o contexto da figura 1 que contém três objetos: d_1 , d_2 e d_3 ($D = \{d_1, d_2, d_3\}$). Dado o conjunto D e os atributos *tipo*, *tamanho* e *cor*, as propriedades de cada objeto de D podem ser representadas pelos seguintes conjuntos de pares atributo-valor:

- $d_1 = \{\langle \text{tipo}, \text{cone} \rangle, \langle \text{tamanho}, \text{pequeno} \rangle, \langle \text{cor}, \text{verde} \rangle\}$
- $d_2 = \{\langle \text{tipo}, \text{cone} \rangle, \langle \text{tamanho}, \text{grande} \rangle, \langle \text{cor}, \text{vermelho} \rangle\}$
- $d_3 = \{\langle \text{tipo}, \text{cubo} \rangle, \langle \text{tamanho}, \text{grande} \rangle, \langle \text{cor}, \text{verde} \rangle\}$

Tomando-se uma base de conhecimento deste tipo como entrada e um objeto-alvo $t \in D$, a tarefa de GER consiste em encontrar um conjunto de pares atributo-valor de t , aqui denominado como conjunto L , tal que L seja verdadeiro para t e falso para todos os distraidores em C . Além da distinção do objeto-alvo, espera-se também que a expressão

de referência produzida seja humanamente plausível, ou seja, seja semelhante a expressões de referência geradas por seres humanos em um mesmo contexto.

Ao avançar dos estudos em GER, cópulas de expressões de referência geradas por seres humanos passaram a ser coletados para avaliar a semelhança destas descrições com expressões de referência produzidas por algoritmos de GER. Como exemplo, encontram-se os cópulas TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), GRE3D3 (VIETHEN; DALE, 2008), GRE3D7 (VIETHEN; DALE, 2011), *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014) e *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014).

A partir da análise destes cópulas, estudos como Viethen e Dale (2008, 2011) mostraram que seres humanos diferentes referenciam uma mesma entidade em um mesmo contexto de maneiras diferentes. Entretanto, grande parte dos algoritmos de GER implementam uma estratégia fixa de referência, ou seja, dada a entrada, tendem a produzir sempre uma mesma descrição de saída, visando apenas distinguir o objeto-alvo e não levando em conta a preferência de cada indivíduo. Um algoritmo de GER como o *Full Brevity* (DALE, 1989), por exemplo, sempre geraria a expressão de referência “O pequeno” para a entidade d_1 apresentada no contexto da figura 1, quando “O cone verde” e o “O cone verde pequeno” são outras expressões de referência que poderiam ser geradas por diferentes seres humanos para referenciar a mesma entidade.

Como apontado em Viethen e Dale (2010), apesar de algumas iniciativas de estudo como Gupta e Stent (2005), Bohnet (2008), Fabbrizio, Stent e Bangalore (2008), Viethen e Dale (2010), Viethen (2011) e Viethen, Mitchell e Krahmer (2013), a variação humana na tarefa de GER é ainda pouco explorada na literatura. Em especial, há pelo menos dois aspectos da variação humana que são de interesse para esta pesquisa que não têm recebido maior atenção na literatura: a seleção de informação para produzir uma descrição, e o grau de especificação dessa.

O primeiro aspecto da variação humana a ser considerado neste trabalho - a seleção de informação - diz respeito a quais propriedades devem ser selecionadas para compor uma expressão de referência baseado no comportamento de cada indivíduo. No experimento de coleta do cópulas GRE3D3 (VIETHEN; DALE, 2008), por exemplo, viu-se que 31% dos participantes nunca utilizaram propriedades relacionais em suas expressões de referência, enquanto 15% utilizaram-nas em todas as suas expressões para as mesmas situações.

O segundo aspecto de variação humana a ser considerado - o grau de especificação da descrição produzida - trata da quantidade de informação que se deve considerar para

compor uma expressão de referência baseado no comportamento de cada indivíduo. No corpus *Stars* (TEIXEIRA et al., 2014), por exemplo, identificou-se que 16% dos participantes geraram somente expressões de referência superespecificadas, ou seja, expressões de referência com mais propriedades do que o necessário para distinguir a entidade referenciada. Com relação às expressões de referência mínimas, ou seja, expressões compostas pela exata quantidade de propriedades necessárias para distinguir o objeto-alvo, viu-se que essas nunca foram produzidas por 61% dos participantes, enquanto que sempre foram geradas por 12%.

Com base nestas constatações, apresenta-se um estudo em nível de mestrado com o objetivo de criar modelos computacionais de GER que levem em conta a variação humana na tarefa de seleção de conteúdo no que diz respeito à seleção de propriedades e ao grau de especificação das expressões de referência produzidas, de forma que essas sejam mais próximas a expressões de referência de seres humanos do que descrições geradas por modelos existentes. Além disso, propõe-se a construção de um corpus de expressões de referência geradas por um grande número de participantes, em um domínio mais realista que os corpus existentes, de modo a investigar a variação humana na tarefa de GER em maior profundidade.

O restante deste documento está organizado da seguinte forma: o objetivo do trabalho é apresentado no Capítulo 2; conceitos fundamentais para o entendimento do problema da variação humana na tarefa de GER são revisados no Capítulo 3; no Capítulo 4, um levantamento bibliográfico dos estudos já feitos a respeito da variação humana na tarefa de GER é apresentado; no Capítulo 5, são apresentados os trabalhos realizados na presente pesquisa; no Capítulo 6, é apresentada a avaliação feita dos modelos computacionais de GER definidos; e no Capítulo 7, é apresentada a conclusão feita sobre a presente pesquisa.

Capítulo 2

Objetivo

2.1 Objetivo Geral

O objetivo geral do trabalho é propor modelos computacionais de GER que levem em conta a variação humana para geração de expressões de referência semelhantes a expressões de referência geradas por cada indivíduo ou perfil desses.

2.2 Objetivos Específicos

1. Coletar e anotar um *cópus* de expressões de referência geradas por um grande número de participantes em um domínio mais realista que *cópus* anteriores apresentados na literatura.
2. Implementar modelos computacionais de GER que levem em conta a variação humana na fase de seleção de conteúdo no que diz respeito à seleção de propriedades e ao grau de especificação das expressões de referência geradas.

Capítulo 3

Conceitos Fundamentais

Neste capítulo, será apresentado um levantamento bibliográfico de algoritmos de GER (Seção 3.1), corpus de expressões de referência (Seção 3.2) e métricas de avaliação intrínseca de algoritmos de GER (Seção 3.3).

3.1 Algoritmos de GER

Nesta seção, será apresentada uma visão geral de alguns algoritmos de GER tradicionais, como *Full Brevity* (DALE, 1989), *Greedy* (DALE, 1989; DALE; HADDOCK, 1991), Incremental (DALE; REITER, 1995), da abordagem de grafos (KRAHMER; ERK; VERLEG, 2003), dentre outros.

3.1.1 *Full Brevity*

Segundo Viethen e Dale (2006), o primeiro algoritmo formalmente definido para a tarefa de GER foi o algoritmo *Full Brevity* (DALE, 1989). Este algoritmo baseia-se no princípio de brevidade apresentado em Grice (1975), o qual propõe que a descrição que distingue um objeto-alvo t dos demais objetos com o menor número de propriedades é a ideal. Pode-se dizer que o objetivo do algoritmo *Full Brevity* é encontrar um conjunto L com o menor número possível de propriedades de t , de forma que não haja distraidores com este conjunto como descrição. Basicamente, este algoritmo segue os seguintes passos:

1. Procurar por uma descrição L do objeto-alvo de forma que o distinga usando apenas uma propriedade. Se encontrá-la, retorne-a. Caso contrário, vá ao passo 2.
2. Procurar por uma descrição L do objeto-alvo de forma que o distinga usando apenas duas propriedades. Se encontrá-la, retorne-a. Caso contrário, vá ao passo 3.
3. E assim por diante...

Como exemplo de funcionamento do algoritmo *Full Brevity*, considera-se como objeto-alvo o objeto d_1 do contexto apresentado na figura 1 do Capítulo 1. Neste exemplo,

o algoritmo *Full Brevity* receberia como entrada o conjunto D e o elemento d_1 como objeto-alvo ($t = d_1$).

Inicialmente, a descrição do objeto-alvo seria inicializada vazia ($L = \{\}$) e os distraidores seriam definidos como todos os objetos do contexto exceto o objeto-alvo ($C = \{d_2, d_3\}$). As possíveis descrições de t com uma propriedade computadas pelo algoritmo seriam: $\{\langle tipo, cone \rangle\}$, $\{\langle tamanho, pequeno \rangle\}$ e $\{\langle cor, verde \rangle\}$. Destas possíveis descrições, apenas a que leva em conta o atributo *tamanho* eliminaria todos os distraidores, pois os objetos d_2 e d_3 possuem o valor *grande* relacionado ao atributo *tamanho*. Logo, a descrição do objeto-alvo a ser retornada pelo algoritmo *Full Brevity* seria $L = \{\langle tamanho, pequeno \rangle\}$, que poderia ser realizada como “O Pequeno”.

A abordagem adotada por este algoritmo possui algumas desvantagens. Em primeiro lugar, a ineficiência: o problema de encontrar a menor descrição é NP-completo. Em segundo lugar, dado que as expressões de referência geradas devem se assemelhar às geradas por seres humanos, sabe-se que esses nem sempre geram expressões de referência breves (DALE; REITER, 1995).

3.1.2 Algoritmo *Greedy*

O algoritmo *Greedy* propõe que o problema de GER seja resolvido testando-se os atributos do objeto-alvo por ordem de seu poder discriminatório em relação aos distraidores (DALE, 1989; DALE; HADDOCK, 1991).

O poder discriminatório de um atributo é definido como o número de distraidores que possuem valor diferente do objeto-alvo para este atributo. Quanto ao término do algoritmo, esse se dá quando é encontrado um conjunto de propriedades L que distinga o objeto-alvo dos distraidores do contexto, ou quando todas as propriedades do objeto-alvo foram inseridas no conjunto L e, mesmo assim, esse não distingue o objeto-alvo dos demais objetos do contexto de forma única. Neste último caso, o algoritmo pode retornar o conjunto L com todas propriedades do objeto-alvo ou um conjunto vazio.

Para exemplificar o funcionamento do algoritmo *Greedy*, considera-se como objeto-alvo o objeto d_1 do contexto representado na figura 1 do Capítulo 1. Neste caso, o algoritmo receberia como entrada o conjunto D e o elemento d_1 como objeto-alvo ($t = d_1$).

Inicialmente, a descrição do objeto-alvo seria inicializada como vazia ($L = \{\}$) e os distraidores seriam definidos como todos os objetos do contexto exceto o objeto-alvo ($C = \{d_2, d_3\}$). O atributo *tamanho* teria poder discriminatório de valor 2, pois o valor do

objeto-alvo relacionado a este atributo ($\langle tamanho, pequeno \rangle$) eliminaria os distraidores d_2 e d_3 , ambos de *tamanho* grande ($\langle tamanho, grande \rangle$). Quanto aos atributos *tipo* e *cor*, ambos teriam poder discriminatório de valor 1, pois seus respectivos valores para o objeto-alvo d_1 ($\langle tipo, cone \rangle$ e $\langle cor, verde \rangle$) eliminariam apenas um distraidor cada: d_2 para a *cor* e d_3 para o *tipo*.

Pelo fato do atributo *tamanho* ter o maior poder discriminatório, a propriedade do objeto-alvo d_1 relacionada a este atributo é a primeira a ser inserida no conjunto L . Como esta propriedade elimina todos os distraidores do objeto-alvo, o critério de parada do algoritmo é alcançado. Logo, a descrição do objeto-alvo d_1 feita pelo algoritmo *Greedy* seria $L = \{\langle tamanho, pequeno \rangle\}$, que poderia ser realizada como “O Pequeno”.

O algoritmo *Greedy* possui custo computacional de execução polinomial, o que é menor do que o algoritmo *Full Brevity*. Apesar disso, esta abordagem também não se assemelha ao comportamento humano para geração de expressões de referência, pois não há evidências de que humanos ordenem os atributos de um objeto-alvo de acordo com seu poder discriminatório, podendo inclusive utilizar mais propriedades em uma descrição do que o necessário para distinguir um determinado objeto-alvo (DALE; REITER, 1995).

3.1.3 Algoritmo Incremental

Em Dale e Reiter (1995), são feitas duas observações com base em estudos da área de psicologia sobre a produção de expressões de referência:

1. Participantes do discurso incluem propriedades não necessárias para desambiguação em suas expressões de referência.
2. Participantes do discurso iniciam o processo de Realização Textual antes de terminar a Seleção de Conteúdo, ou seja, antes de ter eliminado todos os distraidores.

Influenciado por estas observações, em Dale e Reiter (1995) é proposto o algoritmo Incremental, cujo pseudo-código é apresentado na figura 3. A ideia central deste algoritmo é a seleção dos atributos do objeto-alvo de acordo com uma determinada ordem de preferência. Os atributos são assim iterados seguindo a ordem desta lista de preferência e são incluídos na descrição de um objeto-alvo desde que eliminem pelo menos um dos distraidores. O critério de parada do algoritmo se dá quando é encontrado um conjunto de propriedades L do objeto-alvo que consiga distingui-lo dos distraidores do contexto,

ou quando toda a lista de preferência foi iterada e não foi possível distinguir o objeto-alvo dos distraidores.

Figura 3 – *Algoritmo Incremental*

```

1: function getDescricao( $D, Pref, t$ )
2:    $L[t] \leftarrow \{\}$ 
3:    $C \leftarrow D - \{t\}$ 
4:   for  $A_i \in Pref$  do
5:      $V \leftarrow valor(t, A_i)$ 
6:     if  $|eliminaDistraidor(C, \langle A_i, V \rangle)| > 0$  then
7:        $L[t] \leftarrow L[t] \cup \langle A_i, V \rangle$ 
8:        $C \leftarrow C - eliminaDistraidor(C, \langle A_i, V \rangle)$ 
9:     end if
10:    if  $|C| == 0$  then
11:      return  $L$ 
12:    end if
13:  end for
14:  return  $L$ 
15: end function
16:
17: function eliminaDistraidor( $C, \langle A, V \rangle$ )
18:   if  $V = null$  then
19:     return  $\emptyset$ 
20:   else
21:     return  $\{x \in C : valor(x, A) \neq V\}$ 
22:   end if
23: end function

```

Para exemplificar, considere novamente o contexto apresentado na figura 1 do Capítulo 1. Define-se a lista de preferência de atributos como $Pref = \{tipo, cor, tamanho\}$ e d_1 como o objeto-alvo t a ser referenciado. O algoritmo Incremental receberia como entrada os conjuntos D , $Pref$ e o elemento d_1 como objeto-alvo ($t = d_1$).

Inicialmente, a descrição L do objeto-alvo seria inicializada como vazia ($L = \{\}$) e os distraidores seriam todos os objetos do contexto exceto o objeto-alvo ($C = \{d_2, d_3\}$). Na primeira iteração, seguindo a ordem da lista de preferência $Pref$, o *tipo* do objeto-alvo é comparado com o *tipo* dos distraidores. Nesta primeira iteração, d_3 é eliminado do conjunto de distraidores devido ao fato do valor de seu atributo *tipo* ser diferente do objeto-alvo. Em outras palavras, o objeto-alvo é um cone e d_3 um cubo. Consequentemente, a propriedade $\langle tipo, cone \rangle$ do objeto-alvo é incluída no conjunto de descrição ($L = \{\langle tipo, cone \rangle\}$) e o conjunto de distraidores permanece somente com o

elemento d_2 ($C = \{d_2\}$).

Como o conjunto de distraidores não se encontra vazio e ainda há atributos a serem examinados, o algoritmo segue para a segunda iteração, onde o atributo *cor* é testado. Nesta segunda iteração, d_2 é eliminado do conjunto de distraidores devido ao fato de sua *cor* ser diferente da *cor* do objeto-alvo e a propriedade $\langle cor, verde \rangle$ do objeto-alvo t é incluída no conjunto L . Como todos os distraidores foram eliminados, respeitando o critério de parada, o algoritmo Incremental se encerra, devolvendo como resultado o conjunto $L = \{\langle tipo, cone \rangle, \langle cor, verde \rangle\}$, que poderia ser realizado como “O Cone Verde”.

Em Dale e Reiter (1995), argumenta-se que o algoritmo Incremental tenha uma abordagem mais próxima de uma estratégia humana. Além disso, ele também apresenta um custo computacional polinomial mais eficiente em comparação com os algoritmos *Full Brevity* e *Greedy* (DALE; REITER, 1995).

Um ponto a ser ressaltado com relação ao algoritmo Incremental, entretanto, é o fato de não haver uma regra para a ordenação da lista de preferência de atributos dos objetos do contexto. Este fato realça a capacidade de adaptação do algoritmo para geração de expressões de referência mais adequadas a determinados contextos, situações ou preferências e capacidades de cada ser humano. Por outro lado, pode ser visto como uma solução incompleta, pois dependendo da ordem de escolha, o resultado pode variar muito. Considerando o exemplo anterior, caso a lista de preferência fosse definida como $Pref = \{tamanho, cor, tipo\}$, o algoritmo produziria a expressão de referência “O Pequeno” para o objeto d_1 .

3.1.4 Algoritmos Baseado em Grafos

A abordagem da tarefa de GER baseada em grafos é apresentada em Krahmer, Erk e Verleg (2003). Nesta abordagem, os objetos de um contexto D são representados na forma de um grafo direcionado $G = \langle V_G, E_G \rangle$, onde V_G é o conjunto de vértices do grafo que representará o domínio do contexto ($V_G = D$), e E_G é o conjunto de arestas valoradas do grafo, as quais representarão as propriedades de cada objeto do contexto ($E_G \subseteq V_G \times atributo \times V_G$). No grafo, atributos atômicos são representados como laços, enquanto atributos relacionais são representados como arestas na conexão de dois vértices. Na figura 4, pode ser visto a representação do contexto apresentado na figura 1 no formato de um grafo.

O problema de GER na abordagem de grafos é análogo ao problema de Isomor-

Figura 4 – Contexto com dois cones e um cubo na abordagem de grafos

fismo de Subgrafos (KRAHMER; ERK; VERLEG, 2003). Em outras palavras, seja t um objeto-alvo, representado por um vértice $v \in V_G$, o problema de seleção de conteúdo é achar um subgrafo H de G , tal que contenha v , de forma com que este subgrafo seja único no grafo G . Na figura 5, exemplos de subgrafos que descrevem o objeto d_1 do contexto na figura 4 são apresentados.

Figura 5 – Subgrafos que descrevem o objeto d_1 do contexto na figura 4

Assim como o algoritmo Incremental possui uma lista de preferência de atributos que permite ao algoritmo adequar a geração de expressões de referência a contextos diferentes, situações ou às preferências de cada falante, pode-se atribuir custos às arestas que representam as propriedades na abordagem de grafos, conforme feito no contexto representado na figura 4. Mais especificamente, é atribuído às arestas menos preferíveis um maior custo. Desta forma, além de buscar um subgrafo único H que contenha o objeto-alvo v em G , pode-se também buscar o subgrafo de menor custo entre os subgrafos que

representam o objeto-alvo.

3.1.4.1 Algoritmo *Branch and Bound*

Um dos algoritmos propostos para a tarefa de GER na abordagem de grafos é o *Branch and Bound* (KRAHMER; ERK; VERLEG, 2003), representado em partes na figura 6. Este algoritmo recebe como entrada um grafo G e um elemento t a ser referenciado. Inicialmente, o algoritmo inicializa um subgrafo H com apenas o nó correspondente ao objeto-alvo t (linha 3). Após a inicialização, as arestas adjacentes ao(s) nó(s) de H são incluídas uma a uma neste subgrafo para que se faça um teste de exclusividade no grafo G (linhas 15-20). O critério de parada do algoritmo é alcançado quando se encontra uma versão única do subgrafo H com menor custo em G (linha 4).

Figura 6 – Algoritmo de GER baseado em grafos *Branch and Bound*

```

1: function getDescricao( $G, t$ )
2:    $bestGraph \leftarrow \emptyset$ 
3:    $H \leftarrow \langle \{t\}, \emptyset \rangle$ 
4:   return getSubgrafo( $t, bestGraph, H$ )
5: end function
6:
7: function getSubgrafo( $t, bestGraph, H$ )
8:   if  $bestGraph \neq \emptyset$  and  $custo(bestGraph) \leq custo(H)$  then
9:     return  $bestGraph$ 
10:  end if
11:   $C \leftarrow \{n | V_G \wedge isIsomorfos(v, H, n, G) \wedge n \neq v\}$ 
12:  if  $C == \emptyset$  then
13:    return  $H$ 
14:  end if
15:  for  $e \in G.adjacentes(H)$  do
16:     $I \leftarrow getSubgrafo(v, bestGraph, H + e)$ 
17:    if  $bestGraph == \emptyset$  and  $custo(I) \leq custo(bestGraph)$  then
18:       $bestGraph \leftarrow I$ 
19:    end if
20:  end for
21:  return  $bestGraph$ 
22: end function

```

Para exemplificar, considere o elemento d_1 do contexto demonstrado na figura 4 como o objeto-alvo t a ser referenciado. Neste caso, o subgrafo H único e de menor custo retornado pelo algoritmo *Branch and Bound* seria $\{\{d_1\}, \langle \langle tamanho, pequeno \rangle, 2 \rangle\}$, representado na figura 5. Com isso, a descrição do objeto-alvo d_1 resultante poderia ser

realizada como “O Pequeno”.

3.1.4.2 Algoritmo *Longest First*

Um algoritmo recente de GER baseado na abordagem de grafos, chamado *Longest First*, é apresentado em Viethen, Mitchell e Krahmer (2013). Este algoritmo procura gerar a descrição de menor custo com o maior número possível de propriedades.

A motivação para esta abordagem veio da observação feita em Viethen, Mitchell e Krahmer (2013) de que o algoritmo *Branch and Bound* garante a criação do primeiro subgrafo menos custoso que distinga o objeto-alvo. Em uma situação em que haja dois subgrafos representantes de duas expressões de referência do objeto-alvo com o mesmo custo mínimo, será assim retornado a descrição mais breve. Este comportamento torna o algoritmo *Branch and Bound* limitado do ponto de vista da superespecificação, principalmente ao se tratar das propriedades relacionais (VIETHEN; MITCHELL; KRAHMER, 2013).

Ainda em comparação com o algoritmo *Branch and Bound*, o algoritmo *Longest First* tem um número de operações maior a ser executado. Por este motivo, este algoritmo também precisa receber como um de seus parâmetros de entrada o número máximo de propriedades que a descrição gerada deve ter.

3.1.5 Outras Abordagens

Diversas outras abordagens de interesse para a presente pesquisa são apresentadas na literatura. Entre elas está o estudo feito em Bohnet e Dale (2005), onde a tarefa de GER é apresentada como um problema de busca; em Gardent e Striegnitz (2007) e Areces, Koller e Striegnitz (2008), onde o conteúdo semântico das expressões de referência é apresentado como uma fórmula lógica; e em Croitoru e Deemter (2007), onde a tarefa de GER recebe uma abordagem baseada em grafos conceituais. Além disso, novos algoritmos de GER ou extensões dos existentes abordam a resolução de problemas pontuais em GER, como a geração de expressões de referência relacionais ou superespecificadas. Algumas destas abordagens são discutidas brevemente a seguir.

3.1.5.1 Geração de Expressões de Referência Relacionais

Defini-se uma expressão de referência relacional como a descrição de um objeto-alvo a partir de outros objetos, conhecidos como pontos de referência. A descrição “O

cubo à direita do cone” é um exemplo de expressão de referência relacional de um dos objetos da figura 1 no Capítulo 1. Nela, reconhece-se o *cubo* como o objeto-alvo e o *cone* como um ponto de referência.

Em Dale e Haddock (1991), é proposta uma extensão do algoritmo *Greedy* capaz de gerar expressões de referência relacionais. Extensões do algoritmo Incremental com o mesmo fim são apresentadas em Krahmer e Theune (2002), Siddharthan e Copestake (2004), Kelleher e Kruijff (2006) e Silva e Paraboni (2013). Quanto à abordagem de grafos (KRAHMER; ERK; VERLEG, 2003), essa lida naturalmente com a geração de expressões de referência relacionais. Nela, uma relação é representada a partir de uma aresta conectada a dois vértices, representantes de dois objetos distintos, como apresentado na Seção 3.1.4.

3.1.5.2 Geração de Expressões de Referência Superespecificadas

Define-se por expressões de referência superespecificadas as descrições que possuem mais propriedades do que o estritamente necessário para distinguir um determinado objeto-alvo de seus distraidores. No contexto representado na figura 1, “O cubo verde” é um exemplo de expressão de referência superespecificada, pois a expressão “O cubo” com a menção apenas do *tipo* do objeto seria suficiente para distingui-lo dos demais objetos na cena.

A proposta do algoritmo Incremental (DALE; REITER, 1995) teve como uma de suas motivações o fato de seres humanos gerarem expressões de referência superespecificadas. Além disso, o estudo em Paraboni, Deemter e Masthoff (2007) e Paraboni e Deemter (2013) demonstra que, em certos contextos, expressões de referência superespecificadas são necessárias para identificação do objeto-alvo. No contexto representado na figura 1, por exemplo, “O cubo” ou “O cubo verde” não seriam expressões de referência ideais se o objeto-alvo em questão estivesse oculto dentro de uma caixa e o receptor da expressão não soubesse deste fato. O mais adequado neste caso seriam expressões de referência superespecificadas como “O cubo dentro da caixa” ou “O cubo verde dentro da caixa”, mesmo havendo um único objeto do tipo *cubo* no contexto.

Por causa da tendência humana em produzir descrições relacionais e superespecificadas, estes tipos de descrição estão presentes em muitos corpúsculos de expressões de referência. Descrições relacionais, por exemplo, são encontradas nos corpúsculos Drawer (VIETHEN; DALE, 2006, 2007), GRE3D3 (VIETHEN; DALE, 2008), GRE3D7 (VIETHEN; DALE, 2011), *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014), *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014) e *Zoom-Pt* (Seção 5.4); e expressões de referência superespecificadas,

além dos cópus citados, também são encontradas no cópus TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009). Com exceção do cópus Drawer, todos os outros são utilizados na presente pesquisa. Com base nestas constatações, faz-se necessário que os modelos computacionais de GER aqui utilizados contemplem a geração destes tipos de expressão de referência.

3.2 **Cópus de Expressões de Referência**

Dentre os muitos objetivos que um algoritmo de GER precisa atingir, como brevidade, unicidade etc. há a questão de gerar descrições humanamente plausíveis (BELZ; GATT, 2007). Para entender como algoritmos de GER podem gerar expressões de referência semelhantes às expressões de referência produzidas por seres humanos, é comum coletar-se cópus destas descrições. Segundo Krahmer e Deemter (2012), estes cópus podem ser separados em cópus de propósito geral e cópus dedicados.

3.2.1 **Cópus de GER de Propósito Geral**

Cópus de GER de propósito geral são aqueles nos quais se passa uma tarefa qualquer ao participante na qual ele necessariamente terá de referir-se a objetos como uma tarefa secundária. Como exemplos, estão os cópus *Map Task* e *Coconut*.

O cópus *Map Task* (ANDERSON et al., 1991) consiste de um cópus de propósito geral contendo diálogos entre pares de participantes munidos de mapas com pontos de referência indicados. Na tarefa por trás do experimento, um dos participantes da dupla fazia o papel de instrutor e tinha no seu mapa uma rota traçada. Sua função consistia em instruir o outro participante a percorrer esta rota, a qual não estava explícita em seu próprio mapa. O experimento foi conduzido com 64 participantes, 32 do sexo masculino e 32 do sexo feminino. Cada participante atuou em 4 diálogos diferentes, o que soma 128 diálogos.

Quanto ao cópus *Coconut* (EUGENIO et al., 2000), esse consiste de uma série de diálogos em um contexto onde participantes negociam itens de mobília. Estes itens deveriam ser negociados e pagos com a quantia de dinheiro fictício que recebiam no experimento. “A cadeira” ou “A cadeira de 300 dólares” são exemplos de expressões de referência deste cópus.

3.2.2 **Cópus de GER Dedicados**

Os cópus de GER dedicados são aqueles onde a tarefa principal passada aos participantes é referenciar objetos. Nesta pesquisa, apenas cópus deste tipo foram utilizados. Como exemplos desses, encontram-se os cópus TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), Drawer (VIETHEN; DALE, 2006, 2007), GRE3D3 (VIETHEN; DALE, 2008), GRE3D7 (VIETHEN; DALE, 2011), *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014) e *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014). Estes cópus são discutidos individualmente a seguir.

3.2.2.1 **Cópus TUNA**

O cópus TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009) é um conjunto de expressões de referência sobre itens de mobília e fotografias em preto e branco de famosos matemáticos. Sua coleta foi feita a partir de um experimento *online* na qual foi requisitado aos participantes descrever um ou dois objetos-alvo. No caso das imagens onde era requisitada a referência a dois objetos-alvo, esses podiam ser do mesmo tipo ou não. Em cada imagem, os objetos-alvo eram acompanhados de 6 distraidores, posicionados de maneira aleatória em uma tela 3x5, como pode ser visto nas figuras 7 e 8.

O experimento TUNA contém dois domínios. O primeiro, chamado de TUNA Furniture, trata de imagens de itens de mobília, anotadas semanticamente pelos atributos *tipo*, *cor*, *localização*, *tamanho*, *dimensão-x* e *dimensão-y*. O segundo domínio, chamado de TUNA People, consiste de fotografias em preto e branco de matemáticos famosos, anotadas semanticamente pelos atributos *tipo*, *idade*, *localização*, *cor do cabelo*, *tem barba?*, *tem cabelo?*, *usa óculos?*, *usa camisa?*, *usa gravata?*, *usa terno?*, *dimensão-x* e *dimensão-y*. No total, o cópus TUNA contou com 60 participantes e tem 2280 expressões de referência, sendo 1200 para o domínio TUNA Furniture e 1080 para o domínio TUNA People.

O cópus TUNA tem como limitação o fato de não possuir expressões de referência relacionais. Contudo, este cópus foi empregado em uma série de desafios (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), assim como em diversos outros estudos de GER (BOHNET, 2008; FABBRIZIO; STENT; BANGALORE, 2008). Por estes motivos, julga-se o cópus TUNA como importante para a presente pesquisa. Nela, os dois domínios deste cópus foram utilizados em experimentos para explorar a variação humana na tarefa de GER, conforme apresentado no Capítulo 6.

Figura 7 – *Exemplo de cena mostrada aos participantes no experimento do cópus TUNA no domínio TUNA Furniture*

Fonte: (FABBRIZIO; STENT; BANGALORE, 2008)

Figura 8 – *Exemplo de cena mostrada aos participantes no experimento do cópus TUNA no domínio TUNA People*

Fonte: (FABBRIZIO; STENT; BANGALORE, 2008)

3.2.2.2 **Cópus Drawer**

O cópus Drawer (VIETHEN; DALE, 2006, 2007) é um pequeno conjunto de expressões de referência produzido para um experimento que visava comparar como as expressões de referência produzidas por certos algoritmos de GER divergiam das expressões de referência produzidas por humanos (VIETHEN; DALE, 2006).

Este cópus foi coletado em um contexto de referência onde havia 4 armários, cada um com 4 gavetas. Cada armário foi disposto um ao lado do outro, conforme a figura 9. Cada gaveta foi caracterizada com um número, entre 1 e 16, e por sua cor, que podia ser azul, rosa, amarelo ou laranja. Neste contexto, conjuntos de 4 gavetas tinham a mesma cor, as quais estavam distribuídas de maneira aleatória ao longo da estrutura. O objetivo dos participantes do experimento era descrever as gavetas que eram

apontadas como objeto-alvo. O estudo teve 20 participantes e contabilizou 140 expressões de referência. Dessas, 22 expressões de referência foram eliminadas por não distinguirem o objeto-alvo, restando um total de 118 descrições.

Figura 9 – Contexto de referência do córpus Drawer

1 (blue)	2 (orange)	3 (pink)	4 (yellow)
8 (blue)	7 (blue)	6 (yellow)	5 (pink)
9 (orange)	10 (blue)	11 (yellow)	12 (orange)
16 (yellow)	15 (pink)	14 (orange)	13 (pink)

Fonte: (VIETHEN, 2011)

A anotação das expressões do córpus Drawer deu-se a partir dos atributos que foram identificados no experimento. Dentre os atributos não-relacionais, encontram-se *cor*, *linha*, *coluna* e *canto da estrutura*; e dentre os atributos relacionais estão *em cima*, *abaixo*, *próximo*, *à direita de*, *à esquerda de* e *entre*.

Uma representação semântica do contexto do experimento também foi feita para que algoritmos de GER pudessem ser executados e, de fato, tivessem suas expressões contrastadas com os dados do córpus. Para tal, as gavetas do experimento foram rotuladas com os símbolos $d_1, d_2, d_3, \dots, d_{16}$ e tiveram suas propriedades representadas conforme o exemplo abaixo:

$$d_2 = \{\langle cor, laranja \rangle, \langle linha, 1 \rangle, \langle coluna, 2 \rangle, \langle a direita de, d_1 \rangle, \langle a esquerda de, d_3 \rangle, \langle proximo, d_1 \rangle, \langle abaixo, d_7 \rangle\}$$

Dentre as conclusões iniciais feitas em Viethen e Dale (2006) e Viethen e Dale (2007), viu-se que aproximadamente 83% das expressões geradas não utilizavam atributos relacionais. Também detectou-se que apenas 25% das expressões de referência possuíam superespecificação.

Do ponto de vista da presente pesquisa, a coleta do córpus Drawer apresentou dados relevantes a respeito do uso de superespecificação e relações espaciais em expressões de referência geradas por seres humanos. Além disso, mostrou um formato de notação semântica para contextos e expressões de referência útil para a pesquisa aqui desenvolvida. Porém, decidiu-se pelo não uso deste córpus devido ao seu baixo número de expressões

de referência e participantes.

3.2.2.3 **Cópus GRE3D3**

O cópus GRE3D3 (VIETHEN; DALE, 2008) foi coletado com o intuito de investigar o uso de relações espaciais na geração de expressões de referência. Para este fim, foram tomados como princípio trabalhos anteriores como Krahmer e Theune (2002) e Kelleher e Kruijff (2006), que defendem a utilização de atributos relacionais em expressões de referência como último recurso, ou seja, apenas quando não consegue-se descrever um objeto somente com suas propriedades atômicas.

O cópus GRE3D3 originou-se de um experimento conduzido com 74 participantes. Neste experimento, os participantes tinham de descrever objetos individuais de 10 cenas simples em um ambiente tridimensional. Os contextos apresentados aos participantes constituíam-se de até três objetos em uma cena, de tipo cubo ou esfera, de cor vermelha, amarela, azul ou verde e de tamanho pequeno ou grande. Dos três objetos no contexto, aquele apontado por uma seta era o objeto a ser referenciado, e o restante eram os objetos que poderiam ser utilizados para descrições relacionais (i.e., pontos de referência). Todos os contextos foram planejados de forma que não fosse obrigatória a utilização de atributos relacionais para distinguir o objeto-alvo.

Figura 10 – *Exemplo de cena mostrada ao participante no experimento do cópus GRE3D3*

Fonte: (VIETHEN; DALE, 2008)

No total, foram coletadas 740 descrições, das quais 10 foram descartadas a pedido do participante, e outras 10 pelo fato de um participante ser daltônico. Ao final, mais 90 descrições de 9 participantes foram eliminadas por não convergirem com o objetivo do

experimento. Logo, restaram 630 descrições como cópus do experimento GRE3D3.

Na análise dos resultados em Viethen e Dale (2008), viu-se que um terço das 630 expressões coletadas (231 ou 36,6%), fizeram uso de relações espaciais, mesmo essas não sendo necessárias para a desambiguação do objeto-alvo. Estas 231 expressões de referência foram produzidas por 40 (63,5%) dos 63 participantes considerados no experimento, enquanto 23 (36,5%) dos participantes nunca utilizaram relações espaciais. Segundo Viethen e Dale (2008), isso sugere que o uso de relações espaciais está muito atrelado à preferência de cada participante, confirmada pelo fato de que 11 (25%) dos participantes que utilizaram relações espaciais fizeram uso dessas nas 10 cenas que lhes foram apresentadas no experimento.

Da parcela de expressões de referência que fizeram uso de relações espaciais, notou-se que 63,6% delas foram utilizadas em cenas cujo objeto-alvo encontrava-se no topo de um dos pontos de referência, enquanto 36,4% foram utilizadas em cenas cujo objeto-alvo encontrava-se a frente do ponto de referência. Segundo Viethen e Dale (2008), isso demonstra a possibilidade de haver uma preferência pelo atributo *no topo de*, o que comprova estudos anteriores como Arts et al. (2011).

Na conclusão do estudo em Viethen e Dale (2008), ressaltou-se que relações espaciais são utilizadas mesmo quando não necessárias, e que o número de relações espaciais por participante diminuiu no decorrer do experimento. Este último fato demonstra que os participantes puderam perceber que não havia necessidade da utilização de relações espaciais para a descrição do objeto-alvo, o que caracteriza um defeito do projeto do experimento. Concluiu-se também que a saliência dos pontos de referência encoraja o uso de atributos relacionais; a preferência do atributo *no topo de* ao invés de *em frente de*; e o fato do uso de relações espaciais estar relacionado à preferência de cada participante.

3.2.2.4 **Cópus GRE3D7**

O cópus de expressões de referência humana GRE3D7 é uma extensão do GRE3D3, contendo 4480 expressões de referência (VIETHEN; DALE, 2011). Na data de sua divulgação, era o maior cópus de expressões de referência humanas até então publicamente disponível. Sua criação foi motivada pelo fato de que os dados de cópus existentes até então faziam uso de contextos simples e frequentemente abstratos, ao invés de contextos mais realistas. Além disso, baseado na natureza não-determinística do processo de seleção de conteúdo para geração de uma expressão de referência, os cópus devem ter uma grande massa de dados e conter expressões de referência de um número variado de participantes

para cada cenário (VIETHEN; DALE, 2011).

Assim como o cópus GRE3D3, as cenas mostradas aos participantes são cenas tridimensionais, contendo figuras geométricas simples. Porém, diferentemente de seu antecessor, as cenas do cópus GRE3D7 são compostas de sete objetos, sendo quatro de uma cor, e os outros três de outra. A relação entre o objeto-alvo apontado por uma seta na cena e o ponto de referência nunca é uma relação única, ou seja, se o objeto-alvo possuir uma relação espacial horizontal com algum ponto de referência, outras duas entidades da cena possuem esta mesma relação entre si.

O projeto do experimento GRE3D7 foi feito com base em cinco hipóteses:

1. Um ponto de referência de tamanho maior é mais saliente que um ponto de referência menor, e tem maior chance de ser selecionado como relação espacial ao objeto-alvo.
2. Um ponto de referência que possui o mesmo tamanho que outros pontos de referência é menos saliente que um ponto de referência com tamanho único, o qual tem maior chance de ser selecionado como relação espacial ao objeto-alvo.
3. Um ponto de referência que é visualmente diferente do objeto-alvo tem maior probabilidade de ser selecionado para compor uma referência ao objeto-alvo do que um ponto de referência semelhante ao objeto-alvo.
4. Quanto mais semelhante um ponto de referência for do objeto-alvo, mais os dois parecerão ser um único objeto. Quanto mais parecerem um único objeto, maior é a probabilidade de serem mencionados juntos.
5. O atributo relacional *no topo de* tem uma maior probabilidade de ser utilizado do que atributos relacionais laterais, como *à direita de* e *à esquerda de*.

A cada participante, foram apresentadas 16 cenas de um dos dois conjuntos utilizados no experimento. Nestas cenas, o uso de atributos relacionais não era necessário para distinguir os objetos-alvo. Com o intuito de evitar que os participantes descobrissem este padrão, cada uma das 16 cenas foram intercaladas com cenas que não representavam condições do experimento, conhecidas como *fillers*. Nos *fillers*, fazia-se necessário o uso de atributos relacionais para distinção do objeto-alvo dos demais objetos do contexto. Desta forma, foram apresentadas 32 cenas aos participantes, embora apenas 16 representem condições do experimento.

O experimento GRE3D7 teve participação de 318 nativos da língua inglesa. Desse, 294 completaram as 32 cenas. Dos participantes, 54% eram do sexo feminino. Dos

Figura 11 – Exemplo de cena mostrada ao participante no experimento do cópus GRE3D7

Fonte: (VIETHEN; DALE, 2011)

294 participantes do experimento, cinco foram descartados por gerarem expressões de referência ambíguas. Além desses, outros dois participantes foram também descartados por suas expressões de referência não convergirem com o propósito real do experimento. Com isso, restaram 287 participantes, contabilizando 4480 expressões de referência.

Na análise dos resultados, viu-se que apenas 81 das 4480 expressões (1,8%) fizeram menção à localização do objeto-alvo na cena, como na descrição “A bola azul *à direita*”; e das 600 expressões de referência que fizeram uso de atributos relacionais, apenas dez (1,7%) fizeram menção à localização do ponto de referência, como na descrição “A pequena bola próxima ao cubo grande *à esquerda*”. Ainda tratando dos atributos não-relacionais, foi observado ampla utilização do atributo *cor*: 98,7% de todas as expressões fizeram menção à cor do objeto-alvo e 86,8% das expressões de referência relacionais fizeram menção à cor do ponto de referência. Menções ao tamanho do objeto-alvo foram feitas em 57,8% das descrições, e 54,8% das descrições relacionais fazem menção ao tamanho dos pontos de referência.

Das 4480 expressões de referência, 600 (13,4%) fazem menção a relações espaciais. Em média, mais da metade dos participantes do experimento (50,3%) utilizaram relações espaciais em 22,7% de suas descrições, ou seja, estes participantes assumiram suas próprias estratégias no uso de relações espaciais de acordo com o contexto. Por outro lado, viu-se que 135 participantes não fizeram nenhuma menção a relações espaciais, enquanto apenas seis participantes utilizaram relações espaciais em todas as cenas.

Na análise em Viethen e Dale (2011), as hipóteses 1 e 2 mostram o efeito contrário

ao esperado: relações espaciais com pontos de referência que possuem o mesmo tamanho que outros objetos do contexto são significativamente mais prováveis de serem incluídas em expressões de referência do que relações espaciais com pontos de referência de tamanho único. Para as hipóteses 3 e 4, viu-se que pontos de referência que compartilham a mesma cor que o objeto-alvo têm maior probabilidade de serem selecionados. Quanto à hipótese 5, pode-se confirmar que o atributo *no topo de* é preferível a atributos relacionais laterais, novamente confirmando estudos como Arts et al. (2011).

Os cópus GRE3D3 e GRE3D7 mostram-se úteis ao presente trabalho por apresentarem dados relevantes sobre a variação do comportamento humano na seleção de atributos e propriedades para compor uma expressão de referência. Além disso, o cópus GRE3D7 possui uma grande variedade de expressões de referência de diferentes participantes, disponíveis anotadas semanticamente. Estes fatos contribuíram para que estes cópus tenham sido escolhidos para os experimentos feitos neste trabalho, conforme apresentado no Capítulo 6.

3.2.2.5 *Cópus Stars*

O cópus *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014) é um conjunto com 704 expressões de referência produzidas por 64 participantes de um experimento *online*. O experimento *Stars* consistiu em mostrar a cada um dos 64 participantes 11 cenas, representando 6 condições do experimento, intercaladas com 5 cenas *fillers*, as quais tiveram como intuito não deixar que os participantes estabelecessem padrões de referência. As cenas eram simples, compostas de figuras no formato de estrelas, círculos e cubos. O objetivo dos participantes era descrever a estrela apontada por uma seta e próxima a um cubo, conforme mostrado na figura 12.

O cópus *Stars* surgiu a partir de um experimento piloto de investigação do fenômeno de subespecificação em expressões de referência relacionais. Entende-se por subespecificação o fenômeno de considerar um número menor de propriedades para um objeto do que o necessário para distingui-lo de todos distraidores possíveis. Para o contexto da figura 12, a expressão de referência ao objeto *e2* “A estrela ao lado do quadrado”, embora não seja ambígua, possui uma descrição subespecificada do ponto de referência *q6*. Já a descrição “A estrela ao lado do quadrado branco” fornece informação para a completa desambiguação de *q6* em relação aos demais quadrados da cena.

As seis cenas que representam as condições do experimento foram denominadas de *abs1*, *abs3*, *abs5*, *proj1*, *proj3* e *proj5*. Nas cenas *abs1*, *abs3* e *abs5*, o cubo próximo

Figura 12 – Exemplo de cena mostrada ao participante no experimento do cópus Stars

Fonte: (TEIXEIRA et al., 2014)

ao objeto-alvo tinha sua cor diferente dos demais cubos, enquanto nas cenas proj1, proj3 e proj5 todos os cubos tinham a mesma cor. Em outras palavras, nas cenas “abs”, a subespecificação do cubo poderia ser evitada simplesmente com uso de um atributo absoluto (i.e., cor), enquanto nas cenas “proj”, exigir-se-ia um atributo projetivo (i.e., *à esquerda de*) (KELLEHER; KRUIJFF, 2006).

O objetivo dessa comparação era ganhar entendimento sobre as situações em que a subespecificação de um ponto de referência é mais comum. Em Teixeira et al. (2014), duas hipóteses foram levantadas:

1. A subespecificação de pontos de referência é mais frequente quando não é possível distinguir o ponto de referência dos demais objetos da cena usando um atributo absoluto.
2. Descrições mínimas são geralmente menos frequentes que descrições não mínimas.

Um total de 73 estudantes de Bacharelado em Sistemas de Informação, nativos da língua portuguesa e com média de idade de 20,9 anos, participaram do experimento. Coletou-se 803 expressões de referência destes participantes, os quais levaram em média 5,5 minutos cada um para terminarem a tarefa. Deste total, as expressões de referência de 9 participantes foram descartadas, pois divergiam do objetivo do experimento, restando 704 expressões de referência de 64 participantes. Essas foram anotadas semanticamente na linguagem XML, assim como o domínio das cenas. Das 704 descrições, 320 foram desconsideradas no estudo sobre subespecificação em Paraboni et al. (2014) por terem

sido geradas para cenas *fillers*. No teste de hipóteses, foram assim consideradas 384 expressões de referência de 64 participantes.

Na análise dos resultados, detectou-se que a maioria dos participantes usaram tanto descrições superespecificadas quanto subespecificadas. Apenas 10 participantes (15,6%) produziram todas expressões de referência de forma superespecificadas. Também pôde-se detectar que expressões de referência subespecificadas são menos frequentes que as superespecificadas quando é possível distinguir o ponto de referência pela sua cor (cenas *abs1*, *abs3* e *abs5*), assim como apontado pela primeira hipótese. Confirmando a segunda hipótese, expressões de referência mínimas foram raras entre as expressões de referência coletadas: 39 participantes (60,9%) não produziram nenhuma expressão de referência mínima, enquanto apenas 8 participantes (6,3%) sempre as gerou.

3.2.2.6 *Cópus Stars2*

O cópus *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014) é um conjunto criado com o objetivo de investigar a superespecificação de expressões de referência em condições envolvendo diferentes intensidades de esforço cognitivo. Estas condições, denominadas *over_colour*, *over_size*, *over_rel*, *full_colour*, *full_size* e *full_rel*, foram representadas a partir de diferentes tipos de cenas mostradas aos participantes no experimento de coleta do cópus. Exemplos de cenas para cada uma destas condições investigadas estão representados na figura 13.

Figura 13 – Exemplos de cenas do cópus *Stars2* para cada condição investigada

O experimento de coleta do cópus deu-se a partir da organização dos participantes em duplas, as quais interagiam através de um sistema *online* chamado *Lausanne* (IACOVELLI; GALINDO; PARABONI, 2014). Em cada dupla, um participante assumia o papel de locutor, e outro de ouvinte. De 60 cenas de contexto como as mostradas na figura 13, 16 eram escolhidas aleatoriamente e apresentadas, uma de cada vez, aos membros da dupla em telas diferentes. Na tela onde a cena era mostrada ao locutor, o objeto a ser referenciado era apontado por uma seta. O objetivo do locutor no experimento era descrever o objeto-alvo da cena em uma caixa de texto e enviá-la ao ouvinte através de uma mensagem, enquanto o objetivo do ouvinte era assinalar o objeto-alvo da cena com base nesta descrição, conforme mostra a figura 14.

Figura 14 – Exemplos da mesma cena mostrada ao locutor e ao ouvinte no experimento de coleta do cópus *Stars2*

Fonte: (IACOVELLI; GALINDO; PARABONI, 2014)

No total, o experimento de coleta dos dados contou com 76 participantes, sendo 51 do gênero masculino e 25 do gênero feminino. A média de idade dos mesmos foi de 20,4 anos. O número de expressões de referência geradas foi de 1216. Dessas, 12 foram excluídas por ambiguidade, restando assim 1204 expressões de referência válidas. Tanto expressões de referência, quanto o domínios das cenas foram anotados semanticamente na linguagem XML.

Em relação a semelhanças com o cópus *Stars*, o cópus *Stars2* também apresenta contextos onde se pode exigir a referência de até 3 objetos em uma mesma descrição. Como diferenças entre eles, há o fato do cópus *Stars2* possuir um maior número de contextos, tipos de objetos e oportunidades para produção de expressões de referência mais complexas; uma melhor distribuição e variedade nas condições do experimento ao apresentar diferentes tipos de agrupamentos de objetos, orientação de cena, objetos com e

sem sobreposição, etc.; e o balanceamento entre situações de referência, onde há provável uso de relações com outras em que o uso de relação não é necessário, de forma que não há uma vantagem evidente para uma estratégia única de GER. Além disso, o corpus *Stars2* foi construído em modo colaborativo, com falante e ouvinte em uma situação de comunicação. Esta característica fez com que o nível de ruído do experimento fosse praticamente nulo, pois descrições incorretas (i.e., ambíguas) não podiam ser interpretadas pelo ouvinte, exigindo assim uma nova formulação por parte do locutor.

Os corpus *Stars* e *Stars2* mostram-se relevantes para a pesquisa aqui desenvolvida, pois ilustram aspectos importantes a respeito da variação humana na tarefa de GER, principalmente do ponto de vista do grau de especificação que cada participante adotou para suas expressões de referência. Além disso, ambos corpus são compostos por uma variedade de expressões de referência de um número significativo de participantes. Tanto as expressões de referência, quanto os contextos encontram-se disponíveis anotados semanticamente. Estes fatos contribuíram para que estes corpus tenham sido utilizados na avaliação dos algoritmos de GER que incorporam a variação humana, conforme apresentado no Capítulo 6.

3.3 Avaliação Intrínseca dos Algoritmos de GER

Uma questão recorrente em GLN é como avaliar o resultado de algoritmos de GER, ou seja, suas expressões de referência. Dentre as diversas maneiras possíveis, esta avaliação pode ser feita de forma extrínseca, avaliando o desempenho de um sistema completo de GLN que inclua o módulo de GER em questão. Neste tipo de avaliação, entretanto, o impacto do algoritmo de GER pode ser parcial ou totalmente ocultado por outros componentes do sistema. Além disso, este tipo de avaliação tende a apresentar um custo elevado, razão pela qual a avaliação de forma intrínseca tende a ser mais comum.

A avaliação de forma intrínseca consiste na comparação das saídas de um algoritmo de GER com um conjunto de expressões de referência geradas por seres humanos em condições idênticas, ou seja, expressões de referência geradas para um mesmo objeto-alvo em um mesmo contexto. A semelhança de expressões de referência geradas por algoritmos de GER com expressões de referência geradas por seres humanos pode ser medida com o auxílio de corpus de expressões de referência. As descrições geradas por algoritmos de GER podem ser comparadas a expressões de referência dos corpus a partir das métricas de Dice (DICE, 1945), MASI (PASSONNEAU, 2006), de (GUPTA; STENT, 2005) e Acurácia.

O coeficiente de Dice (DICE, 1945) é uma métrica para comparação entre dois conjuntos representantes de expressões de referência a partir da divisão entre o número de elementos que os dois conjuntos têm em comum pela soma do tamanho de ambos, multiplicada por dois. Essa é a métrica mais comum para avaliação de algoritmos de GER baseada em corpus (KRAHMER; DEEMTER, 2012; BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009).

$$Dice(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (3.1)$$

O valor obtido pelo coeficiente de Dice varia entre 0 e 1, sendo 0 o valor indicativo de uma total divergência entre as duas expressões de referência comparadas e o valor 1 a total convergência.

Para a tarefa computacional de GER, entende-se os elementos dos conjuntos A e B como atributos ou propriedades que compõem uma expressão de referência, ou a combinação de ambos. Por exemplo, se o conjunto A representasse a expressão de referência “O cone verde”, este conjunto poderia ser representado das seguintes maneiras:

- $A_1 = \{tipo, cor\}$ representaria os atributos que compõem a expressão de referência.
- $A_2 = \{tipo.cone, cor.verde\}$ representaria as propriedades que compõem a expressão de referência.
- $A_3 = \{tipo, cor, tipo.cone, cor.verde\}$ representaria a combinação dos atributos e propriedades que compõem a expressão de referência.

Neste trabalho, a representação das expressões de referência em conjuntos seguiu a abordagem A_3 acima, onde é feita a combinação de atributos e propriedades.

Uma alternativa à métrica de Dice é o coeficiente de MASI (PASSONNEAU, 2006):

$$MASI(A, B) = \delta \times \frac{|A \cap B|}{|A \cup B|} \quad (3.2)$$

Como é possível visualizar pela Fórmula 3.2, o coeficiente de MASI consiste na divisão do tamanho do conjunto de intersecção entre dois conjuntos pelo tamanho do conjunto de união destes dois conjuntos, multiplicada por um peso δ . Este peso δ favorece a similaridade quando um conjunto é um subconjunto ou superconjunto do outro:

$$\delta = \begin{cases} 1, & \text{se } A = B, \\ \frac{2}{3}, & \text{se } A \subset B \text{ ou } B \subset A, \\ 0, & \text{se } A \cap B = \emptyset, \\ \frac{1}{3}, & \text{caso contrário.} \end{cases} \quad (3.3)$$

Assim como o coeficiente de Dice, o valor obtido pelo coeficiente de MASI varia entre 0 e 1, sendo 0 o valor indicativo de uma total divergência entre as duas expressões de referência comparadas e o valor 1 a total convergência.

Além dos coeficientes de Dice e MASI, uma forma alternativa de comparar dois conjuntos representantes de expressões de referência é apresentado em Gupta e Stent (2005). Nesta forma, operações de inserção, remoção e troca de ordem de elementos são feitas sobre o conjunto representante da expressão de referência gerada pelo algoritmo de GER para igualá-lo ao conjunto representante da expressão de referência do *córpus* em questão. A partir destas operações, a seguinte fórmula é calculada (GUPTA; STENT, 2005):

$$S = \frac{C}{C + I + D + M} \quad (3.4)$$

Nesta fórmula, C representa o número de atributos coincidentes entre os conjuntos, I o número de elementos inseridos, D o número de elementos removidos e M o número de elementos movidos. O valor S varia entre 0 e 1, onde um maior valor representa expressões de referência mais próximas.

Finalmente, a medida de Acurácia consiste em atribuir valor 1 caso as duas expressões de referência comparadas sejam idênticas, e 0 caso contrário. Em outras palavras, pode-se dizer que quando o valor dos coeficientes de Dice, MASI e de (GUPTA; STENT, 2005) for 1, a Acurácia também terá valor 1, e quando os valores dos coeficientes de Dice, MASI e (GUPTA; STENT, 2005) forem diferentes de 1, a Acurácia terá valor 0. Pela sua baixa granularidade, a Acurácia é normalmente apresentada em trabalhos da área para fins ilustrativos, mas tende a ser menos útil do que as métricas de Dice e MASI para fins de avaliação mais detalhada do comportamento de algoritmos de GER.

3.4 Considerações

Este capítulo apresentou uma revisão de conceitos fundamentais de GER, como algoritmos de GER, *córpus* de expressões de referência e métricas de avaliação intrínseca

de algoritmos de GER.

Na Seção 3.1, algoritmos de GER tradicionais, como o algoritmo *Full Brevity* (DALE, 1989), *Greedy* (DALE, 1989; DALE; HADDOCK, 1991), Incremental (DALE; REITER, 1995) e da abordagem de grafos (KRAHMER; ERK; VERLEG, 2003; VIETHEN; MITCHELL; KRAHMER, 2013) foram apresentados. Os algoritmos Incremental e da abordagem de grafos mostram-se importantes para a presente pesquisa, pois são modelos que apresentam medidas paramétricas, como a lista de preferência de atributos no caso do algoritmo Incremental, e o peso das arestas no caso dos algoritmos da abordagem de grafos. Estas medidas são configuráveis e permitem que o algoritmo possa aproximar suas expressões de referência das expressões de referência geradas por seres humanos.

Os corpúscos de expressões de referência foram revisados na Seção 3.2. Estes corpúscos apresentam uma forma de entender o comportamento humano na tarefa de GER para aplicá-lo na criação de novos algoritmos ou para configuração das medidas paramétricas dos já existentes. Dentre os corpúscos revisados, os corpúscos TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), GRE3D3 (VIETHEN; DALE, 2008), GRE3D7 (VIETHEN; DALE, 2011), *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014) e *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014) apresentam um grande número de expressões de referência geradas por diferentes seres humanos, havendo assim a oportunidade de explorar a variação humana na tarefa de GER com base nestes dados.

As métricas de avaliação intrínseca de algoritmos de GER foram apresentadas na Seção 3.3. Na avaliação do presente trabalho, a ser discutida no Capítulo 6, serão consideradas as métricas de Dice (DICE, 1945), MASI (PASSONNEAU, 2006) e Acurácia para avaliação das expressões de referência geradas pelos algoritmos de GER utilizados.

Capítulo 4

A Variação Humana em GER

Entre as constatações feitas com a análise de corpus de expressões de referência está a de que seres humanos podem referenciar uma determinada entidade em um determinado contexto de maneiras diferentes (VIETHEN; DALE, 2008, 2011). Neste capítulo, discute-se alguns dos estudos que apresentam algoritmos de GER que incorporam esta variação do comportamento humano na geração de expressões de referência.

4.1 Variação Humana nos corpus *Map Task* e *Cocunut*

Em Gupta e Stent (2005), foram realizados experimentos de GER utilizando o algoritmo Incremental (DALE; REITER, 1995) e algumas de suas extensões nos corpus de diálogos entre participantes *Map Task* (ANDERSON et al., 1991) e *Coconut* (EUGENIO et al., 2000) com o objetivo de avaliar a tarefa de GER em diálogos.

Cada um destes algoritmos recebeu como entrada o objeto-alvo a ser referenciado, uma lista de preferência dos atributos e um conjunto de distraidores. Para a formação deste conjunto de distraidores foram considerados dois contextos diferentes: o contexto do diálogo e a base de conhecimento do locutor em questão. Se o objeto-alvo fosse achado no contexto do diálogo, o conjunto de distraidores seria o conjunto dos objetos nele contidos. Caso contrário, o conjunto de distraidores seria o conjunto dos objetos contidos no contexto do diálogo mais os objetos contidos na base de conhecimento do locutor em questão.

As estratégias consideradas em Gupta e Stent (2005) para a ordenação da lista de preferência de atributos do algoritmo Incremental e suas extensões foram:

- *Baseline*: algoritmo semelhante ao Incremental com a diferença de que o atributo *tipo* é sempre escolhido em primeiro lugar, seguido de uma seleção dos outros atributos de maneira aleatória.
- Algoritmo Incremental original (DR): para o corpus *Map Task*, a lista de preferência

de atributos do algoritmo é definida como o conjunto ordenado $\langle \textit{tipo}, \textit{tamanho}, \textit{conteúdo}, \textit{estado}, \textit{localização}, \textit{cor} \rangle$ e para o *córpus Coconut*, como o conjunto ordenado $\langle \textit{tipo}, \textit{cor}, \textit{custo}, \textit{quantidade}, \textit{estado} \rangle$.

- Extensão do algoritmo Incremental de Siddharthan e Copestake (2004) (SC): a lista de preferência de atributos passa a ser ordenada de forma que os atributos que possuam um coeficiente de discriminação maior sejam os primeiros. Este coeficiente de discriminação é calculado através da soma de um coeficiente de similaridade com um coeficiente de contraste (SIDDHARTHAN; COPESTAKE, 2004).
- *Partner-Specific Adaptation* (DRP e SCP): a lista de preferência de atributos é reordenada segundo os atributos recentemente utilizados por cada participante. Os atributos não utilizados ficam no final da lista.
- *Partner-Specific Adaptation Variant* (DRPV e SCPV): nesta variante da abordagem anterior, a lista de preferência de atributos não é somente reordenada com base nos últimos atributos recentemente utilizados por cada participante, mas também exige-se que os atributos utilizados na última menção feita pelo algoritmo sejam reutilizados, mesmo que não eliminem nenhum distrator.

As abordagens *Partner-Specific Adaptation* e *Partner-Specific Adaptation Variant* foram utilizadas em conjunto com o algoritmo Incremental original (DRP e DRPV) e com sua extensão proposta em Siddharthan e Copestake (2004) (SCP e SCPV).

Para comparação dos resultados, foi utilizada a métrica de Gupta e Stent (2005), discutida na Seção 3.3. Ao contrário do esperado pela hipótese inicial, observou-se que a abordagem *Baseline* não teve um desempenho significativamente inferior, comparado ao algoritmo Incremental (DR), à extensão proposta por Siddharthan e Copestake (2004) (SC) e à abordagem *Partner-Specific Adaptation* (DRP e SCP) nos dois *córpus*. Por outro lado, os algoritmos com a abordagem *Partner-Specific Adaptation Variant* (DRPV e SCPV) obtiveram os melhores desempenhos. O estudo conclui que levar em conta a variação humana faz com que os algoritmos de GER gerem expressões de referência mais próximas das descrições produzidas por seres humanos.

O estudo apresentado em Gupta e Stent (2005) é relevante para o presente trabalho por apresentar um método de tratar a variação humana nos algoritmos de GER existentes. Este método supera a abordagem que não leva a questão da variação humana em conta, o que é consistente com a hipótese central do presente trabalho.

4.2 Variação Humana no *córpus* TUNA

Em Bohnet (2008), é discutido um estudo de desempenho de algoritmos de GER no *córpus* TUNA (BELZ; GATT, 2007). Um dos algoritmos utilizados para este fim foi o IS-FBN (BOHNET, 2007, 2008). Este algoritmo é uma implementação do algoritmo *Full Brevity* (DALE, 1989; DALE; HADDOCK, 1991) na qual a tarefa de GER é vista como um problema de busca (BOHNET; DALE, 2005). O funcionamento do algoritmo IS-FBN se dá pela computação de todas as descrições possíveis de um objeto-alvo, que posteriormente são comparadas com um conjunto de treinamento de expressões de referência para o mesmo objeto-alvo utilizando o algoritmo dos Vizinhos Mais Próximos (KNUTH, 1998) e a métrica Dice (DICE, 1945). A expressão de referência com maior coeficiente de Dice, ou seja, a expressão de referência mais semelhante às expressões do conjunto de treinamento, é retornada como resultado.

Além do algoritmo IS-FBN, uma extensão desse também é apresentada: o algoritmo IS-FP. Assim como o algoritmo IS-FBN, o algoritmo IS-FP também computa todas as expressões de referência possíveis para um objeto-alvo. Porém, diferentemente do IS-FBN, a partir do algoritmo dos Vizinhos Mais Próximos, as expressões de referência computadas são comparadas com as expressões de referência do participante analisado e do participante cujas expressões de referência são mais semelhantes às do participante em questão. A similaridade dos participantes é computada a partir da métrica Dice. Do resultado obtido pelo algoritmo dos Vizinhos Mais Próximos é selecionada a descrição mais breve. Caso exista mais de uma expressão de referência com a mesma quantidade de atributos, o algoritmo dos Vizinhos Mais Próximos é executado novamente sobre elas. Se ainda não houver um resultado único, é escolhida a expressão de referência cujos atributos tenham maior frequência.

O experimento conduzido em Bohnet (2008) consistiu em comparar quatro algoritmos: o algoritmo Incremental usando uma lista de preferência ordenada pela frequência dos atributos no conjunto de dados; o algoritmo Incremental usando uma lista de preferência ordenada pela frequência dos atributos por participante; e os algoritmos IS-FBN e IS-FP. Nos resultados obtidos para o *córpus* TUNA, observou-se que o algoritmo Incremental que levava em conta a variação dos participantes superou o algoritmo Incremental sem a variação, assim como o algoritmo IS-FP superou o algoritmo IS-FBN.

O trabalho apresentado em Bohnet (2008) é importante para o presente trabalho, porque explora a variação humana na tarefa de GER, foco principal da pesquisa aqui

apresentada. Além disso, o estudo apresenta uma heurística para ordenação da lista de preferência de atributos do algoritmo Incremental baseada na frequência desses.

4.3 Variação Humana na Seleção de Conteúdo e na Realização Textual de GER

Em Fabrizio, Stent e Bangalore (2008), a partir do uso do corpus TUNA (BELZ; GATT, 2007), as tarefas de Seleção de Conteúdo e Realização Textual de GER foram desenvolvidas levando-se em conta a variação na preferência dos participantes.

Para a tarefa de seleção de conteúdo, dois algoritmos foram utilizados: *Full Brevity* (DALE, 1989) e o algoritmo Incremental (DALE; REITER, 1995). Em suas execuções sobre o corpus TUNA, o algoritmo *Full Brevity* usou quatro estratégias de referência, enquanto o Incremental usou duas. Para o algoritmo *Full Brevity*, todos os conjuntos de atributo-valores que distinguem o objeto-alvo são computados e uma das 4 abordagens a seguir é aplicada (FABBRIZIO; STENT; BANGALORE, 2008):

1. *Critério da Brevidade*: é escolhida a expressão de referência mais breve. Em caso de haver mais de uma, a escolha é feita de forma aleatória.
2. *Critério da Frequência*: seleciona a expressão de referência cujo conjunto de atributos é o mais frequente no conjunto de treinamento.
3. *Critério da Frequência por Participante*: seleciona a expressão de referência cujo conjunto de atributos é o mais frequente no conjunto de treinamento para o participante em questão.
4. *Critério de escolha de Atributos Recém-utilizados por Participante*: seleciona a expressão de referência cujo conjunto de atributos foi recém-utilizado pelo participante, voltando ao terceiro critério caso necessário.

Para o algoritmo Incremental, duas estratégias de ordenação com base em sua lista de preferência de atributos são consideradas (FABBRIZIO; STENT; BANGALORE, 2008):

1. Ordenação baseada em suas frequências no conjunto de treinamento.
2. Ordenação baseada em suas frequências no conjunto de treinamento para cada participante, havendo a possibilidade de consultar o primeiro critério caso não haja frequência para um determinado atributo.

O resultado do experimento foi mensurado a partir das métricas MASI, Dice, Acurácia, unicidade e brevidade (BELZ; GATT, 2007). Como esperado, o algoritmo *Full Brevity* gerou altos índices de unicidade. Nos dois corpúsculos do conjunto TUNA houve um considerável aumento de desempenho do algoritmo *Full Brevity* com o cálculo das frequências por participante, com exceção da métrica de brevidade. Porém, ao utilizar o *Critério de escolha de Atributos Recém-utilizados por Participante*, os resultados foram mais baixos. Segundo Fabbrizio, Stent e Bangalore (2008) esta baixa é devida a dois fatores: os participantes não estavam em uma conversa, e nem sempre se tem conhecimento dos últimos atributos utilizados por um participante no conjunto de treinamento. Para o algoritmo Incremental, também houve um aumento de desempenho ao incorporar a variação do participante, embora não tenha superado o algoritmo *Full Brevity*. De modo geral, pôde-se concluir que levar em conta as preferências de cada participante, de fato, melhora o desempenho dos algoritmos utilizados.

Assim como em Bohnet (2008), o estudo apresentado em Fabbrizio, Stent e Bangalore (2008) apresenta a heurística de ordenação da lista de preferência de atributos do algoritmo Incremental baseada na frequência dos atributos por participante. Além disso, também é apresentado resultados os quais também sugerem que algoritmos de GER que levam em conta a variação humana superam aqueles que não levam esta questão em conta.

4.4 Variação Humana nos corpúsculos GRE3D3 e GRE3D7

Em Viethen e Dale (2010) e Viethen (2011), são apresentados experimentos de aprendizado de máquina com o intuito de realizar a tarefa de GER nos corpúsculos GRE3D3 (VIETHEN; DALE, 2008) e GRE3D7 (VIETHEN; DALE, 2011). Os dois principais objetivos do estudo eram entender quais são os fatores relevantes para inclusão de uma propriedade em particular em uma expressão de referência, e como estes fatores interagem entre si. Especificamente, o estudo inclui o uso de informações de identidade de cada locutor e procura assim gerar expressões de referência de acordo com as preferências de cada locutor.

A técnica de aprendizado de máquina escolhida para o experimento foi as Árvores de Decisão C4.5 (QUINLAN, 1993). Segundo Viethen e Dale (2010), a escolha deveu-se ao fato desta técnica permitir que se visualize o processo de decisão sobre os dados dos corpúsculos. As árvores de decisão utilizadas foram treinadas com e sem poda, em duas formas de treinamento: uma utilizando as mesmas expressões de referência do corpúsculo como conjunto de treinamento e teste, e outra utilizando o método de validação cruzada.

Para modelagem do experimento, identificou-se quais atributos foram utilizados nas expressões de referência dos *córpus GRE3D3 e GRE3D7*, chegando-se a uma lista de 9 atributos (VIETHEN; DALE, 2010), representados na tabela 1. Como entrada para as árvores de decisão, um vetor com as características de aprendizado representadas na tabela 2 foi utilizado.

Tabela 1 – Atributos relacionados a um objeto-alvo TG e seu ponto de referência LM nos *córpus GRE3D3 e GRE3D7*

Id	Descrição
TG_Type	atributo <i>tipo</i> de TG
TG_Colour	atributo <i>cor</i> de TG
TG_Size	atributo <i>tamanho</i> de TG
TG_Location	atributo <i>localização</i> de TG
Relation	atributo indicativo da relação entre TG e LM
LM_Type	atributo <i>tipo</i> de LM
LM_Colour	atributo <i>cor</i> de LM
LM_Size	atributo <i>tamanho</i> de LM
LM_Location	atributo <i>localização</i> de LM

Fonte: (VIETHEN; DALE, 2010)

Tabela 2 – Características de entrada às árvores de decisão, relacionadas a um objeto-alvo TG e seu ponto de referência LM

Id	Descrição
TG_Size	atributo <i>tamanho</i> de TG
LM_Size	atributo <i>tamanho</i> de LM
Relation_Type	tipo de relação entre TG e LM
Num_TG_Size	número de objetos com o mesmo tamanho que TG
Num_LM_Size	número de objetos com o mesmo tamanho que LM
TG_LM_Same_Size	valor que indica se TG e LM compartilham o mesmo tamanho
Num_TG_Col	número de objetos com a mesma cor que TG
Num_LM_Col	número de objetos com a mesma cor que LM
TG_LM_Same_Col	valor que indica se TG e LM compartilham a mesma cor
Num_TG_Type	número de objetos com o mesmo tipo que TG
Num_LM_Type	número de objetos com o mesmo tipo que LM
TG_LM_Same_Type	valor que indica se TG e LM compartilham o mesmo tipo
Participant_ID	número de identificação do participante

Fonte: (VIETHEN; DALE, 2010)

Na análise das expressões de referência dos *córpus*, reconheceram-se 31 combinações de atributos no *córpus GRE3D3* e 27 no *córpus GRE3D7*. Estas combinações foram utilizadas como classes das respectivas árvores de decisão para uma das abordagens seguidas. Em Viethen e Dale (2010) e Viethen (2011), considerou-se duas abordagens:

1. Árvores de decisão com objetivo de classificar a combinação adequada de atributos do objeto-alvo e ponto de referência para compor a expressão de referência, treinadas

com e sem poda, com e sem a identificação do participante nas duas formas de treinamento citadas.

2. Árvores de decisão com o objetivo de classificar se um determinado atributo deve ser ou não incluído numa expressão de referência, treinadas com e sem poda nas duas formas de treinamento citadas. Nesta abordagem, a identificação do participante não foi utilizada como um componente do vetor de características e árvores de decisão específicas foram treinadas para cada atributo e para cada atributo e participante. Estas últimas foram definidas em Viethen e Dale (2010) e Viethen (2011) como um perfil do participante, e eram somente árvores de decisão com poda.

Na segunda abordagem também foi realizado o cruzamento das árvores de decisão entre os *córpus*, ou seja, as árvores de decisão treinadas com as expressões de referência do *córpus* GRE3D3 foram testadas para prever as expressões de referência do *córpus* GRE3D7 e vice-versa.

Por fim, as conclusões feitas em Viethen e Dale (2010) e Viethen (2011) foram:

- Expressões de referência completas são difíceis de prever a partir das árvores de decisão da primeira abordagem. Nela, o uso da identificação do participante proporciona uma boa caracterização dos dados, mas não necessariamente uma boa predição de novas expressões de referência.
- Com o cruzamento das árvores de decisão entre os *córpus*, viu-se que o *córpus* GRE3D7 é menos variável que o *córpus* GRE3D3.
- A definição de árvores de decisão específicas por atributo e treinadas para cada participante na segunda abordagem apresenta um novo e eficiente modelo de GER. A acurácia obtida para estas árvores de decisão foi mais alta que a acurácia das árvores de decisão da primeira abordagem, mesmo para predição de novas expressões de referência.
- A grande maioria dos participantes compartilham das mesmas estratégias no momento de escolher um atributo específico, e apenas uma minoria de participantes tem suas próprias estratégias no momento de fazer esta escolha.

Do ponto de vista da presente pesquisa, o trabalho apresentado em Viethen e Dale (2010) e Viethen (2011) é útil por apresentar um método para resolução da tarefa

de GER baseado em aprendizado de máquina. Além disso, o estudo aborda o problema da variação humana em GER a partir da inclusão da identidade de cada locutor como entrada a este método.

4.5 Grafos e Relações Espaciais em GER

O estudo em Viethen, Mitchell e Krahmer (2013) parte da premissa de que a tarefa de GER na abordagem de grafos (KRAHMER; ERK; VERLEG, 2003) lida naturalmente com propriedades relacionais, e do fato desta abordagem nunca ter sido testada empiricamente. Assim, dois experimentos são conduzidos com esta abordagem no *cópus* de expressões de referência GRE3D3 (VIETHEN; DALE, 2008).

No primeiro experimento em Viethen, Mitchell e Krahmer (2013), os pesos das arestas do grafo foram treinadas de seis formas diferentes:

- 0-COST-PROP: é atribuído o valor 0 para todas as arestas do grafo. Mesmo que não elimine nenhum distraidor, cada propriedade é incluída na descrição até que essa distinga o objeto-alvo de todos distraidores.
- 0-COST-ATT: semelhante à 0-COST-PROP, com a diferença que os valores das arestas são baseados na frequência de seus respectivos atributos.
- FREE-NAIVE-PROP: propriedades que ocorrerem em 75% das descrições têm o peso 0 atribuído às arestas, pares atributo-valor com frequência abaixo de 20% têm peso 2 atribuído, e para o restante das arestas o peso 1 é atribuído.
- FREE-NAIVE-ATT: semelhante ao FREE-NAIVE-PROP, com a diferença de que os pesos são atribuídos baseados na frequência dos atributos, e não das propriedades.
- K-PROP: os pesos das arestas são atribuídos a partir do uso do algoritmo K-médias (JAIN, 2010) sobre a frequência das propriedades com $k = 2$.
- K-ATT: semelhante ao K-PROP, mas com a diferença de que os pesos são atribuídos baseados na frequência dos atributos, e não das propriedades.

Nos experimentos conduzidos em Viethen, Mitchell e Krahmer (2013), quatro algoritmos de GER foram utilizados, sendo dois desses da abordagem de grafos:

- Abordagem dos 50%: algoritmo onde propriedades do objeto-alvo são incluídas na descrição caso possuam frequência maior que 50% no conjunto de treinamento.
- Algoritmo Incremental (DALE; REITER, 1995), discutido na Seção 3.1.3. Este algoritmo teve uma versão com a lista de preferência de atributos ordenada pela frequência de atributos, e outra pela frequência das propriedades.
- Algoritmo *Branch and Bound* (KRAHMER; ERK; VERLEG, 2003) da abordagem de grafos, discutido na Seção 3.1.4.
- Algoritmo *Longest First* (VIETHEN; MITCHELL; KRAHMER, 2013) da abordagem de grafos, discutido na Seção 3.1.4.

O treinamento da ordem de preferência dos atributos ou pares atributo-valor foi feito a partir da metodologia de validação cruzada com 10 *folds*. A efetividade dos algoritmos para cada forma de atribuição de pesos foi calculada usando-se as medidas de Acurácia e Dice, conforme discutido na Seção 3.3.

Nos resultados do primeiro experimento, o algoritmo *Branch and Bound* apresentou o mesmo resultado para todas as abordagens (Acurácia de 39,21% e Dice de 73,4%). O algoritmo *Longest First* com as abordagens 0-COST e K-PROP foram os únicos que incluíram atributos relacionais em suas descrições. A abordagem dos 50% superou a abordagem do algoritmo *Longest First* com 0-COST. O melhor resultado foi obtido para os algoritmos baseados na abordagem FREE-NAIVE, embora essas não tenham gerado nenhuma descrição com atributos relacionais. A dificuldade em utilizar atributos relacionais comprova o baixo desempenho das abordagens deste primeiro experimento, já que um terço das expressões de referência do corpus GRE3D3 possuem atributos relacionais (VIETHEN; DALE, 2008).

No segundo experimento, todos os pesos foram treinados a partir de perfis dos participantes do corpus. O algoritmo de clusterização K-médias (JAIN, 2010) foi utilizado para agrupar os participantes do corpus GRE3D3 de acordo com o número de vezes com que cada participante utilizou cada atributo e a média de tamanho de cada uma de suas descrições. Viu-se que o algoritmo K-médias executado com dois agrupamentos obteve o melhor resultado. Destes dois agrupamentos criados, um tratava dos participantes que tinham preferência por descrições longas com relações espaciais (16 participantes e 160 descrições), enquanto o segundo agrupamento consistia dos participantes que preferiam descrições mais curtas não relacionais (47 participante e 470 descrições). Os dois agrupamentos foram treinados separadamente utilizando a função de custo FREE-NAIVE, que

tinha produzido melhores resultados no primeiro experimento. Novamente, a metodologia de validação cruzada com 10 *folds* foi utilizada. Além disso, o algoritmo *Longest First* teve uma execução com o número máximo de 4 atributos por descrição e uma segunda execução onde o número máximo de atributos por descrição é a média do tamanho das descrições de cada agrupamento.

Na análise dos resultados do segundo experimento, o algoritmo *Longest First*, com a abordagem FREE-NAIVE-PROP e com o parâmetro de máxima descrição atribuído pela média do tamanho das descrições do conjunto de treinamento conseguiu gerar expressões de referência relacionais mais próximas das expressões de referência do corpus. Ao treinar as funções de custo e ordens de preferência separadamente para os diferentes agrupamentos, o algoritmo *Longest First* conseguiu reproduzir corretamente 100% das relações espaciais usadas pelos participantes. Por fim, concluiu-se que a variação humana encontrada na tarefa de GER deve ser levada em conta pelos algoritmos para que o uso de relações espaciais nas expressões de referência seja feito de maneira parecida com a preferência individual de cada locutor.

Do ponto de vista da presente pesquisa, considera-se o trabalho em Viethen e Dale (2008) potencialmente útil porque apresenta um método para agrupar locutores em perfis com base em suas expressões de referência. Posteriormente, estes perfis podem ser utilizados no treinamento de modelos computacionais de GER.

4.6 Considerações

Neste capítulo, estudos que levam em conta a variação humana na tarefa de GER foram apresentados. Nestes trabalhos, diversos métodos para incorporar a variação humana em algoritmos de GER foram discutidos, como a seleção de propriedades a partir das frequências de uso dos atributos por participante (BOHNET, 2008; FABBRIZIO; STENT; BANGALORE, 2008; VIETHEN; MITCHELL; KRAHMER, 2013) e a partir dos atributos recentemente utilizados por participante (GUPTA; STENT, 2005; FABBRIZIO; STENT; BANGALORE, 2008). Além desses, o trabalho apresentado em Viethen e Dale (2010) e Viethen (2011) mostra-se relevante para a pesquisa aqui desenvolvida, pois apresenta um método de resolução da tarefa de GER, levando em conta a variação humana, através de técnicas de aprendizado de máquina.

Capítulo 5

Trabalho Realizado

O capítulo anterior revisou uma série de estudos sobre a variação humana em GER. Entretanto, grande parte destas iniciativas, como os trabalhos de Gupta e Stent (2005), Bohnet (2008), Fabbrizio, Stent e Bangalore (2008) e Viethen, Mitchell e Krahmer (2013), limitam-se a agrupar locutores e/ou manipular a ordem de seleção de atributos para obter resultados mais próximos das descrições existentes em *cópus* de GER. Apesar de estudos introdutórios como Viethen e Dale (2010) e Viethen (2011), não há uma investigação mais detalhada do efeito da variação humana sobre a seleção de atributos individuais ou sobre o grau de especificação adotado por diferentes locutores. Parte deste capítulo é destinado para um estudo desta natureza.

Com relação à organização do capítulo, esse está dividido em quatro seções: a proposta de algoritmos de GER é detalhada na Seção 5.1; as abordagens para incorporar a variação humana em cada algoritmo de GER utilizado são apresentadas na Seção 5.2; as variações dos algoritmos de GER, utilizadas como modelos computacionais na presente pesquisa, são apresentadas na Seção 5.3; e a construção do *cópus Zoom* é apresentado na Seção 5.4.

5.1 Algoritmos Básicos

Nesta seção, serão apresentados os algoritmos de GER desenvolvidos e implementados neste trabalho. Na Seção 5.1.1, é apresentado uma extensão do Algoritmo Incremental (DALE; REITER, 1995) capaz de gerar descrições relacionais; e na Seção 5.1.2, é proposto um algoritmo de GER baseado em técnicas de aprendizado de máquina. Ambos algoritmos serão utilizados como sistemas de *baseline* para a proposta de algoritmos de GER considerando a preferência humana, a ser discutida na Seção 5.2.

5.1.1 Algoritmo Incremental Estendido

O algoritmo Incremental, como apresentado em Dale e Reiter (1995), tem como limitação o fato de não lidar com expressões de referência relacionais. Além disso, não há

uma heurística definida de como ordenar sua lista de preferência de atributos. Ao invés disso, apenas se estipula que esta ordem seria dependente do domínio. Visando preencher estas lacunas, nesta seção, será apresentada uma extensão do algoritmo Incremental denominada “Algoritmo Incremental Estendido” (AIE). Este algoritmo é ilustrado na figura 15 e será discutido a seguir.

Figura 15 – Algoritmo Incremental Estendido

```

1: function getDescricao( $t, L, D, Preferencia, objetosAtivos$ )
2:    $L[t] \leftarrow \{\}$ 
3:    $C \leftarrow D - \{t\}$  //Distraidores
4:    $objetosAtivos \leftarrow objetosAtivos \cup t$ 
5:
6:    $v \leftarrow valor(t, tipo)$ 
7:    $Pref \leftarrow Preferencia[v]$ 
8:
9:   for  $A_i \in Pref$  do
10:      $V \leftarrow valor(t, A_i)$ 
11:     for  $v_i \in V$  do
12:       if  $|eliminaDistraidor(C, \langle A_i, v_i \rangle)| > 0$  then
13:         if  $atributoRelacional(A_i) == true$  then
14:           if  $v_i \notin objetosAtivos$  then
15:              $L[t] \leftarrow L[t] \cup \langle A_i, v_i \rangle$ 
16:              $C \leftarrow C - eliminaDistraidor(C, \langle A_i, v_i \rangle)$ 
17:              $L \leftarrow getDescricao(v_i, L, D, Preferencia, objetosAtivos)$ 
18:           end if
19:         else
20:            $L[t] \leftarrow L[t] \cup \langle A_i, v_i \rangle$ 
21:            $C \leftarrow C - eliminaDistraidor(C, \langle A_i, v_i \rangle)$ 
22:         end if
23:       end if
24:       if  $|C| == 0$  then
25:         return  $L$ 
26:       end if
27:     end for
28:   end for
29:   return  $L$ 
30: end function

```

5.1.1.1 Gerando Expressões de Referência Relacionais

Em sua versão original, apresentada em Dale e Reiter (1995), o algoritmo Incremental não lida com expressões de referência relacionais. Em Krahmer e Theune (2002)

e Kelleher e Kruijff (2006), extensões deste algoritmo são propostas para a geração deste tipo de expressão de referência. Ambas extensões adotam uma abordagem que prioriza os atributos não-relacionais ao invés dos atributos relacionais. Em outras palavras, estes algoritmos buscam inicialmente uma descrição de um determinado objeto-alvo a partir de seus atributos atômicos, assim como o algoritmo Incremental original, e só consideram os atributos relacionais caso não consigam uma descrição do objeto-alvo apenas com os atributos atômicos.

Diferentemente das extensões do algoritmo Incremental propostas em Krahmer e Theune (2002) e Kelleher e Kruijff (2006), no presente algoritmo AIE, os atributos relacionais podem estar intercalados com os atributos não-relacionais na lista de preferência de atributos do algoritmo. Esta decisão é embasada pela análise dos *corpus* GRE3D3/7 em Viethen e Dale (2008) e Viethen e Dale (2011), onde viu-se que atributos relacionais são selecionados em detrimento de atributos atômicos mesmo quando não são necessários para fins de desambiguação.

Para lidar com expressões de referência relacionais, o algoritmo AIE passa a produzir um conjunto L de propriedades do objeto-alvo e dos pontos de referência a ele associados. Por exemplo, para a expressão de referência “O cone à esquerda do cone vermelho” gerada para o objeto-alvo d_1 via o ponto de referência d_2 do contexto representado na figura 1, o conjunto L seria:

$$\{ d_1 = \{ \langle \text{tipo}, \text{cone} \rangle, \langle \text{à esquerda de}, d_2 \rangle \}, d_2 = \{ \langle \text{tipo}, \text{cone} \rangle, \langle \text{cor}, \text{vermelho} \rangle \} \}$$

Para evitar o problema de expressões de referência recursivas (DALE; HADDOCK, 1991) como “O cubo abaixo do círculo acima do cubo abaixo...”, uma lista de objetos ativos é criada, representada na figura 15 pelo parâmetro de entrada *objetosAtivos*. Ao longo da execução do algoritmo, a lista de objetos ativos tem como função armazenar os objetos os quais estão presentes na descrição L . Na iteração de um atributo relacional, o objeto tido como seu valor será descrito e inserido na lista de objetos ativos somente se ainda não estiver nela presente.

5.1.1.2 Listas de Preferência de Atributos

Com relação à lista de preferência de atributos, o algoritmo AIE passa a considerar uma para cada tipo de objeto existente no contexto em questão. Por exemplo, no contexto representado na figura 1 do Capítulo 1, existem objetos do tipo cone ($\langle \text{tipo}, \text{cone} \rangle$) e do tipo cubo ($\langle \text{tipo}, \text{cubo} \rangle$). Neste caso, uma lista de preferência de atributos seria considerada

quando o objeto a ser descrito fosse do tipo *cone*, e outra para quando o objeto fosse do tipo *cubo*. Este feito permite adaptar melhor o algoritmo às possíveis variantes na forma de referenciar os objetos que compõem uma expressão de referência. Na figura 15, o conjunto com as listas de preferência é representado pelo parâmetro de entrada *Preferencia*.

A ordenação das listas de preferência de atributos do algoritmo AIE é feita a partir da contabilização da frequência dos atributos nas expressões de referência dos corpos. Esta estratégia foi baseada nos trabalhos apresentados em Fabbrizio, Stent e Bangalore (2008), Bohnet (2008) e Viethen, Mitchell e Krahmer (2013), e discutidos nas Seções 4.2, 4.3 e 4.5, respectivamente.

Para ordenação de cada uma das listas de preferência de atributos, definidas de acordo com os tipos de objeto presentes no contexto, são criadas primeiramente uma lista de atributos relacionais e outra lista de atributos não-relacionais, ambas ordenadas decrescentemente pelas frequências de seus atributos. Após a determinação das duas listas ordenadas, é computada a frequência total do uso de atributos relacionais. Posteriormente, a lista de atributos relacionais é inserida na lista de atributos não-relacionais entre os atributos que possuírem frequência menor/igual e maior que a frequência total do uso de atributos relacionais, formando assim a lista de preferência para o tipo de objeto em questão. Esta abordagem objetiva incentivar o uso de atributos relacionais, colocando-os nas primeiras posições das listas de preferência de atributos. Quanto aos atributos que possuírem frequência zero, esses são inseridos no final da respectiva lista de preferência.

5.1.1.3 Funcionamento do Algoritmo AIE

Como entrada, o algoritmo AIE recebe quatro parâmetros. Esses são:

- $t \in D$: objeto a ser referenciado, pertencente ao domínio D . Na primeira chamada do algoritmo, este parâmetro é o objeto-alvo.
- D : conjunto representante do domínio do contexto.
- *Preferencia*: conjunto que representa as listas de preferência de atributos para cada tipo de objeto presente no contexto.
- *objetosAtivos*: lista de objetos já inseridas na descrição. Na primeira chamada do algoritmo, recebe o conjunto vazio ($objetosAtivos = \{\}$).

Após a ordenação das listas de preferência, o algoritmo AIE tem seu início (linhas 2-4) a partir da inserção de um conjunto vazio em L como descrição do objeto t , da definição do conjunto C com os distraidores deste objeto e a inserção de t à lista de objetos ativos. Posteriormente (linhas 6-7), é definida qual a lista de preferência que deve ser iterada segundo o valor do atributo *tipo* do objeto a ser descrito.

O processo de seleção de conteúdo se dá a partir da iteração dos atributos da lista de preferência definida (linha 9). Na linha 10, atribui-se na variável V , a partir da função auxiliar *valor*, os valores das propriedades de t relacionadas ao atributo iterado. Em seguida, para cada valor em V (linha 11), é testado se o tamanho do conjunto resultante da função auxiliar *eliminaDistraidor* é maior que 0 (linha 12). Esta função auxiliar retorna os objetos do conjunto C cujos valores da propriedade relacionada ao atributo iterado são diferentes da propriedade de t para o mesmo atributo.

Dado que o tamanho do conjunto resultante da função auxiliar *eliminaDistraidor* seja maior que 0, testa-se se o atributo iterado é um atributo relacional a partir da função auxiliar *atributoRelacional* (linha 13). Caso a condição seja verdadeira e o valor da propriedade de t relacionada ao atributo iterado não esteja na lista de objetos ativos (i.e., *objetosAtivos*), a propriedade de t relacionada a este atributo é inserida no conjunto L , o resultado da função auxiliar *eliminaDistraidor* é eliminado do conjunto de distraidores C e é feita a chamada recursiva ao algoritmo para descrição do objeto v_i (linhas 15-17). Caso o atributo iterado não seja um atributo relacional, a propriedade de t relacionada a este atributo é inserida na descrição L e o resultado da função auxiliar *eliminaDistraidor* é eliminado do conjunto de distraidores C (linhas 20 e 21).

O encerramento do algoritmo se dá com o retorno do conjunto de propriedades L em duas ocasiões: quando o conjunto de distraidores C encontra-se vazio (linhas 24-26) ou quando toda a lista de preferência de atributos foi iterada (linha 29).

Tratando da complexidade, o algoritmo Incremental original possui custo computacional polinomial $O(n * d)$, sendo n o número de atributos presentes na lista de preferência de atributos e d o número de distraidores do objeto-alvo. Quanto ao algoritmo AIE, esse possui custo computacional polinomial $O(k * n * v * d)$, sendo k o número de objetos descritos na execução do algoritmo e v o número de propriedades analisadas por atributo. Apesar de ser mais custoso, não há mudança na ordem de grandeza entre o algoritmo Incremental original e o algoritmo AIE, sendo esse também de custo computacional polinomial.

5.1.2 *InteliGER*: um algoritmo de GER baseado em aprendizado de máquina

Apesar dos resultados e da longevidade do algoritmo Incremental (DALE; REITER, 1995), atualmente, a disponibilidade de corpus de expressões de referência permite que técnicas da área de Inteligência Artificial, como as de aprendizado de máquina, sejam utilizadas para o desenvolvimento de modelos computacionais de GER mais generalizáveis, como já feito em Viethen e Dale (2010), Viethen (2011) e Pereira et al. (2012). Em Viethen e Dale (2010) e Viethen (2011), por exemplo, modelos de aprendizado de máquina são treinados com expressões de referência geradas por seres humanos para que as descrições geradas sejam próximas a essas, como explicado na Seção 4.4.

A proposta de um algoritmo de GER aqui apresentada é baseada em um dos modelos apresentados em Viethen e Dale (2010) e Viethen (2011), onde classificadores são treinados para cada atributo com o intuito de prever se esses devem compor uma determinada expressão de referência. Para definir este modelo, considera-se D o conjunto que representa as entidades de um domínio e A o conjunto de atributos que o descreve. Dada a tarefa de gerar uma expressão de referência para um determinado objeto-alvo $t \in D$, define-se o conjunto de classificadores $C = \{c^{(1)}, c^{(2)}, \dots, c^{(|A|)}\}$, onde $c^{(i)} \in C$ é treinado para prever se o atributo $a^{(i)} \in A$ deve compor a expressão de referência de t .

O treinamento é realizado com expressões de referência feitas a objetos de D , representadas pelo conjunto de treinamento $E = \{e^{(1)}, e^{(2)}, \dots, e^{(|E|)}\}$. Seja uma expressão de referência de treinamento $e \in E$, sua representação é dada por $e = \{X, Y\}$, onde X representa seu vetor n -dimensional de características e Y , seu conjunto de classes $\{y^{(1)}, y^{(2)}, \dots, y^{(|A|)}\}$, atribuindo-se $y^{(k)} = 0$, caso o atributo $a^{(k)} \in A$ não componha e , e $y^{(k)} = 1$ caso contrário.

No modelo aqui proposto, apelidado de *InteliGER*, estende-se o trabalho de Viethen e Dale (2010) e Viethen (2011), combinando as predições individuais dos atributos, feitas pelos classificadores responsáveis, para produzir a expressão de referência do determinado objeto-alvo. A definição dos classificadores do algoritmo *InteliGER* está fortemente relacionada aos atributos que contextualizam cada domínio. Neste estudo, os corpus de expressões de referência TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), GRE3D3 (VIETHEN; DALE, 2008), GRE3D7 (VIETHEN; DALE, 2011), *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014), *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014) e *Zoom-Pt* foram utilizados como conjunto de dados.

Como as descrições de cada um destes *córpus* são geradas em domínios diferentes, os atributos referenciais de cada um também são distintos. O domínio TUNA Furniture do *córpus* TUNA, por exemplo, é anotado apenas com os atributos atômicos *tipo*, *cor*, *localização*, *tamanho*, *dimensão-x* e *dimensão-y*, enquanto o *córpus* GRE3D3 é anotado com os atributos atômicos *tipo*, *cor*, *tamanho* e *localização*, além dos atributos relacionais *próximo de*, *à direita de*, *à esquerda de*, *no topo de* e *em frente de*. Logo, projetou-se modelos específicos para cada *córpus*, absorvendo suas peculiaridades. Os classificadores, vetores de características e o algoritmo de combinação das predições para cada um dos *córpus* utilizados são explicados com maiores detalhes nas seções a seguir.

5.1.2.1 Classificadores

Para predição de atributos atômicos, classificadores binários foram treinados para definir se o respectivo atributo deve ou não deve compor uma determinada expressão de referência. Quanto aos atributos relacionais, com exceção do *córpus* *Zoom-Pt* como será explicado, classificadores multi-classe foram treinados para prever se existe relação entre dois objetos, e qual tipo de relação existe entre eles.

Nos *córpus* GRE3D3/7 (VIETHEN; DALE, 2008, 2011), oito classificadores binários são treinados, correspondentes aos atributos do objeto-alvo e do ponto de referência que se deseja prever; e um classificador multi-classe é treinado para prever a relação entre o objeto-alvo e um possível ponto de referência. Os possíveis valores que podem ser preditos por este classificador são *sem relação* (indica que a descrição não é relacional), *à direita de*, *à esquerda de*, *próximo de*, *no topo de* e, no caso do *córpus* GRE3D3, também *em frente de*. Todos os classificadores dos *córpus* GRE3D3/7, extraídos de Viethen e Dale (2010), estão representados na tabela 3.

Tabela 3 – *Classificadores dos *córpus* GRE3D3 e GRE3D7, relacionados a um objeto-alvo TG, um ponto de referência LM e a relação entre eles para predição de uma descrição EXP*

Classificador	Descrição
TG_Type	1 para inclusão do tipo de TG em EXP; 0 caso contrário
TG_Colour	1 para inclusão da cor de TG em EXP; 0 caso contrário
TG_Size	1 para inclusão do tamanho de TG em EXP; 0 caso contrário
TG_Location	1 para inclusão da localização de TG em EXP; 0 caso contrário
LM_Type	1 para inclusão do tipo de LM em EXP; 0 caso contrário
LM_Colour	1 para inclusão da cor de LM em EXP; 0 caso contrário
LM_Size	1 para inclusão do tamanho de LM em EXP; 0 caso contrário
LM_Location	1 para inclusão da localização de LM em EXP; 0 caso contrário
Relation	prediz a relação entre TG e LM em EXP

Nos corpúscos *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014) e *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014), existem descrições com até dois pontos de referência, como “A estrela vazada ao lado do círculo vermelho e do quadrado vazado” e “O cone localizado na frente de um cubo, que está atrás de uma esfera”. Logo, classificadores foram treinados para os objetos-alvo, para um possível 1º e para outro possível 2º ponto de referência. No total, são treinados 14 classificadores binários para o corpúscos *Stars* e 17 para o corpúscos *Stars2*. Além desses, mais dois classificadores multi-classe são treinados em cada corpúscos para predição da relação entre o objeto-alvo e o 1º ponto de referência e a relação entre o 1º e o 2º ponto de referência. Os possíveis valores que podem ser preditos pelos classificadores multi-classe são *sem relação* (indica que não há relação entre os dois objetos), *à direita de*, *à esquerda de*, *próximo de*, *acima de*, *abaixo de* e, no caso do corpúscos *Stars2*, também *em frente de*. Os classificadores para ambos os corpúscos estão representados na tabela 4.

Tabela 4 – *Classificadores dos corpúscos Stars e Stars2, relacionados a um objeto-alvo TG, um 1º e um 2º ponto de referência LM e LM2, e a relação entre eles para predição de uma descrição EXP*

Classificador	Descrição	Stars	Stars2
TG_Type	1 para inclusão do tipo de TG em EXP; 0 caso contrário	X	X
TG_Colour	1 para inclusão da cor de TG em EXP; 0 caso contrário	X	X
TG_Size	1 para inclusão do tamanho de TG em EXP; 0 caso contrário		X
TG_VPos	1 para inclusão da posição vertical de TG em EXP; 0 caso contrário	X	X
TG_HPos	1 para inclusão da posição horizontal de TG em EXP; 0 caso contrário	X	X
LM_Type	1 para inclusão do tipo de LM em EXP; 0 caso contrário	X	X
LM_Colour	1 para inclusão da cor de LM em EXP; 0 caso contrário	X	X
LM_Size	1 para inclusão do tamanho de LM em EXP; 0 caso contrário		X
LM_VPos	1 para inclusão da posição vertical de LM em EXP; 0 caso contrário	X	X
LM_HPos	1 para inclusão da posição horizontal de LM em EXP; 0 caso contrário	X	X
LM2_Type	1 para inclusão do tipo de LM2 em EXP; 0 caso contrário	X	X
LM2_Colour	1 para inclusão da cor de LM2 em EXP; 0 caso contrário	X	X
LM2_Size	1 para inclusão do tamanho de LM2 em EXP; 0 caso contrário		X
LM2_VPos	1 para inclusão da posição vertical de LM em EXP; 0 caso contrário	X	X
LM2_HPos	1 para inclusão da posição horizontal de LM2 em EXP; 0 caso contrário	X	X
Relation	prediz a relação entre TG e LM em EXP	X	X
LM_Relation	prediz a relação entre LM e LM2 em EXP	X	X

O corpúscos TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009) não possui expressões de referência relacionais. Logo, somente classificadores binários foram treinados para prever se atributos atômicos deveriam compor a expressão de referência de um objeto-alvo. No total, 6 classificadores binários são treinados para o domínio TUNA Furniture e 12 para o domínio TUNA People. Estes classificadores encontram-se nas tabelas 5 e 6.

No corpúscos *Zoom-Pt*, existem expressões de referência que possuem muitos pontos de referência, sendo mais de dois pontos de referência para um mesmo objeto em algumas

Tabela 5 – *Classificadores do corpus TUNA Furniture, relacionados a um objeto-alvo TG para predição de uma descrição EXP*

Classificador	Descrição
TG_Type	1 para inclusão do tipo de TG em EXP; 0 caso contrário
TG_Colour	1 para inclusão da cor de TG em EXP; 0 caso contrário
TG_Size	1 para inclusão do tamanho de TG em EXP; 0 caso contrário
TG_Orientation	1 para inclusão da localização de TG em EXP; 0 caso contrário
TG_XDimension	1 para inclusão da dimensão x onde se encontra TG em EXP; 0 caso contrário
TG_YDimension	1 para inclusão da dimensão y onde se encontra TG em EXP; 0 caso contrário

Tabela 6 – *Classificadores do corpus TUNA People, relacionados a um objeto-alvo TG para predição de uma descrição EXP*

Classificador	Descrição
TG_Type	1 para inclusão do tipo de TG em EXP; 0 caso contrário
TG_Orientation	1 para inclusão da localização de TG em EXP; 0 caso contrário
TG_Age	1 para inclusão da idade de TG em EXP; 0 caso contrário
TG_HairColour	1 para inclusão da cor do cabelo de TG em EXP; 0 caso contrário
TG_HasBeard	1 para inclusão de informação do uso de barba de TG em EXP; 0 caso contrário
TG_HasHair	1 para inclusão de informação do cabelo de TG em EXP; 0 caso contrário
TG_HasGlasses	1 para inclusão de informação do uso de óculos de TG em EXP; 0 caso contrário
TG_HasShirt	1 para inclusão de informação do uso de camisa de TG em EXP; 0 caso contrário
TG_HasTie	1 para inclusão de informação do uso de gravata de TG em EXP; 0 caso contrário
TG_HasSuit	1 para inclusão de informação do uso de terno de TG em EXP; 0 caso contrário
TG_XDimension	1 para inclusão da dimensão x onde se encontra TG em EXP; 0 caso contrário
TG_YDimension	1 para inclusão da dimensão y onde se encontra TG em EXP; 0 caso contrário

situações, como na descrição “O restaurante na esquina da *Rua Anchieta* com a *Rua Capelo*”. Portanto, optou-se por não utilizar classificadores multi-classe para fazer a predição de atributos relacionais. Ao invés disso, como feito para os atributos atômicos, classificadores binários foram treinados para cada atributo relacional com o objetivo de prever se este deve ou não compor uma determinada descrição. No total, 12 classificadores binários foram treinados para o corpus *Zoom-Pt*, como representado na tabela 7.

5.1.2.2 Vetor de Características

No algoritmo *InteliGER*, um vetor de características é a entrada fornecida a um classificador para caracterizar uma determinada expressão de referência. Cada dimensão do vetor é uma característica de aprendizagem da expressão de referência descrita. Em uma visão tradicional de GER (i.e., sem considerar a variação humana), estas características são extraídas do contexto de referência e serão aqui denominadas de características de contexto.

As características de contexto tratam de caracterizar o objeto-alvo da expressão,

Tabela 7 – Classificadores do *cópus Zoom-Pt*, relacionados a um objeto-alvo TG, um 1º ponto de referência LM e a relação entre eles para predição de uma descrição EXP

Classificador	Descrição
TG_Type	1 para inclusão do tipo de TG em EXP; 0 caso contrário
TG_Name	1 para inclusão do nome de TG em EXP; 0 caso contrário
TG_Other	1 para inclusão de outras informações de TG em EXP; 0 caso contrário
LM_Type	1 para inclusão do tipo de LM em EXP; 0 caso contrário
LM_Name	1 para inclusão do nome de LM em EXP; 0 caso contrário
LM_Other	1 para inclusão de outras informações de LM em EXP; 0 caso contrário
Next-To	1 para inclusão da relação <i>próximo de</i> entre TG e LM; 0 caso contrário
Right-Of	1 para inclusão da relação <i>à direita de</i> entre TG e LM; 0 caso contrário
Left-Of	1 para inclusão da relação <i>à esquerda de</i> entre TG e LM; 0 caso contrário
In-Front-Of	1 para inclusão da relação <i>em frente de</i> entre TG e LM; 0 caso contrário
Behind-Of	1 para inclusão da relação <i>atrás de</i> entre TG e LM; 0 caso contrário
In	1 para inclusão da relação <i>em</i> entre TG e LM; 0 caso contrário

os pontos de referência utilizados e suas semelhanças com os outros objetos do contexto. De modo geral, as características de contexto visam descrever o ambiente onde o objeto-alvo da expressão de referência se encontra.

Nos *cópus* GRE3D3/7 (VIETHEN; DALE, 2008, 2011), os objetos são caracterizados pelos atributos atômicos *tipo*, *cor*, *tamanho* e *localização*; e pelos atributos relacionais *próximo de*, *à direita de*, *à esquerda de*, *no topo de* e, no caso do *cópus* GRE3D3, também *em frente de*. Quanto ao *cópus* *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014) e *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014), os objetos são caracterizados pelos atributos atômicos *tipo*, *cor*, *orientação vertical*, *orientação horizontal* e, no caso do *cópus* *Stars2*, também *tamanho*. Com relação aos atributos relacionais, os objetos destes dois *cópus* são caracterizados pelos atributos *próximo de*, *à esquerda de*, *à direita de*, *acima de*, *abaixo de* e, no caso do *cópus* *Stars2*, também *em frente de*. Baseadas nestes atributos, as características de contexto modeladas para os *cópus* GRE3D3/7, *Stars* e *Stars2* são apresentadas na tabela 8. Vale observar que características relacionadas ao atributo *tamanho* não foram modeladas para o vetor de características do *cópus* *Stars*, porque seus objetos não são caracterizados por este atributo.

A representação dos objetos no domínio TUNA Furniture (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009) é feita pelos atributos atômicos *tipo*, *cor*, *localização*, *tamanho*, *dimensão-x* e *dimensão-y*. No domínio TUNA People (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), caracteriza-se os objetos pelos atributos atômicos *tipo*, *idade*, *localização*, *cor do cabelo*, *tem barba?*, *tem cabelo?*, *usa óculos?*, *usa camisa?*, *usa gravata?*, *usa terno?*, *dimensão-x* e *dimensão-y*. As características de contexto modeladas para

Tabela 8 – Características de Contexto dos corpúscos GRE3D3/7, Stars e Stars2, relacionadas a um objeto-alvo TG e seu ponto de referência LM

Característica	Descrição
TG_Size	valor do atributo <i>tamanho</i> de TG
LM_Size	valor do atributo <i>tamanho</i> de LM
Relation_Type	o tipo de relação entre o TG e LM (Vertical ou Horizontal)
Num_TG_Size	número de objetos com o mesmo tamanho que TG
Num_LM_Size	número de objetos com o mesmo tamanho que LM
TG_LM_Same_Size	valor que indica se TG e LM compartilham o mesmo tamanho
Num_TG_Col	número de objetos com a mesma cor que TG
Num_LM_Col	número de objetos com a mesma cor que LM
TG_LM_Same_Col	valor que indica se TG e LM compartilham a mesma cor
Num_TG_Type	número de objetos com o mesmo tipo que TG
Num_LM_Type	número de objetos com o mesmo tipo que LM
TG_LM_Same_Type	valor que indica se TG e LM compartilham o mesmo tipo

Fonte: (VIETHEN; DALE, 2010; VIETHEN, 2011)

ambos domínios do corpúscos TUNA, baseadas em seus atributos, estão representadas nas tabelas 9 e 10. Vale observar que, como não há descrições relacionais em ambos domínios, características baseadas em atributos relacionais não foram modeladas.

Tabela 9 – Características de Contexto do corpúscos TUNA Furniture, relacionadas a um objeto-alvo TG

Característica	Descrição
LOC	recomendação dada aos participantes
Num_TG_Size	número de objetos com o mesmo tamanho que TG
Num_TG_Col	número de objetos com a mesma cor que TG
Num_TG_Type	número de objetos com o mesmo tipo que TG
Num_TG_Orientation	número de objetos com a mesma orientação que TG

Tabela 10 – Características de Contexto do corpúscos TUNA People, relacionadas a um objeto-alvo TG

Id	Descrição
LOC	recomendação dada aos participantes
Num_TG_Age	número de objetos com a mesma idade que TG
Num_TG_Type	número de objetos com o mesmo tipo que TG
Num_TG_Orientation	número de objetos com a mesma localização que TG
Num_TG_HairColour	número de objetos com a mesma cor de cabelo que TG
Num_TG_HasSuit	número de objetos com o mesmo valor para o atributo <i>usa terno?</i> que TG
Num_TG_HasShirt	número de objetos com o mesmo valor para o atributo <i>usa camisa?</i> que TG
Num_TG_HasTie	número de objetos com o mesmo valor para o atributo <i>usa gravata?</i> que TG
Num_TG_HasBeard	número de objetos com o mesmo valor para o atributo <i>tem barba?</i> que TG
Num_TG_HasGlasses	número de objetos com o mesmo valor para o atributo <i>usa óculos?</i> que TG
Num_TG_HasHair	número de objetos com o mesmo valor para o atributo <i>tem cabelo?</i> que TG

Parte das características de contexto definidas para os corpúscos TUNA foi adaptada

do trabalho em Pereira et al. (2012). Com relação à característica “LOC”, apresentada nas tabelas 9 e 10, essa representa uma recomendação dada aos participantes no experimento de coleta de ambos corpúscos TUNA. Esta recomendação consistiu em sugerir para alguns participantes do experimento que utilizassem os atributos que representam as coordenadas dos objetos na tela para referenciá-los, enquanto que, para outros participantes, o uso dessas foi desencorajado.

Por último, o corpúscos *Zoom-Pt* teve seu contexto modelado pelos atributos atômicos *tipo*, *nome* e *outros*; e pelos atributos relacionais *em*, *próximo de*, *à direita de*, *à esquerda de*, *em frente de*, *atrás de* e *entre*. Baseado nestes atributos, duas características de contexto foram modeladas para o corpúscos *Zoom-Pt*, apresentadas na tabela 11.

Tabela 11 – Características de Contexto do corpúscos *Zoom-Pt*, relacionadas ao objeto-alvo *TG*

Id	Descrição
Num_TG_Next	número de objetos que possuem o mesmo número de pontos de referência que TG
Num_TG_Type	número de objetos com o mesmo tipo que TG

5.1.2.3 Combinação das Predições

Após os classificadores destinados para cada atributo terem sido treinados, as predições de quais atributos deverão compor a expressão de referência são feitas. Com base nestas predições, o algoritmo apresentado na figura 16 é executado com o intuito de combiná-las e, por fim, construir a expressão de referência para o determinado objeto-alvo.

O algoritmo é representado pela função principal *getDescricao* (linha 1) e pela função secundária *getDescricaoObjeto* (linha 13). A função principal recebe como parâmetros de entrada um objeto-alvo *t*, o conjunto *P* de predições feitas para o objeto-alvo e para possíveis pontos de referência, e o conjunto de atributos *A* que caracteriza o domínio em questão. Quanto à função secundária, essa recebe como entrada um objeto *obj*, o conjunto de propriedades *L*, o conjunto *P* de predições feitas para *obj* e o conjunto de atributos *A*.

O início do algoritmo se dá pela definição da descrição não-relacional do objeto-alvo *t* a partir da função *getDescricaoObjeto* (linha 2). Esta função tem como objetivo complementar um conjunto de propriedades *L* com as propriedades de um determinado objeto *obj* baseado num conjunto de predições *P* (linhas 13-22). As propriedades de *obj* incluídas em *L* são as propriedades cujos atributos relacionados foram preditos como positivo ($P[A_i] = 1$) pelos classificadores. Vale observar que as propriedades são incluídas

Figura 16 – Algoritmo para combinação das predições dos classificadores

```

1: function getDescricao( $t, P, A$ )
2:    $L \leftarrow \text{getDescricaoObjeto}(t, \emptyset, P[tg], A)$ 
3:   if  $P[\text{relacao}] \neq \text{sem\_relacao}$  then
4:      $V \leftarrow \text{random}(\text{valor}(t, P[\text{relacao}])))$ 
5:     if  $V \neq \text{null}$  then
6:        $L[t] \leftarrow L[t] \cup \langle P[\text{relacao}], V \rangle$ 
7:        $L \leftarrow \text{getDescricaoObjeto}(V, L, P[lm], A)$ 
8:     end if
9:   end if
10:  return  $L$ 
11: end function
12:
13: function getDescricaoObjeto( $obj, L, P, A$ )
14:   $L[obj] \leftarrow \{\}$ 
15:  for  $A_i \in A$  do
16:     $V \leftarrow \text{random}(\text{valor}(obj, A_i))$ 
17:    if  $P[A_i] == 1$  then
18:       $L[obj] \leftarrow L[obj] \cup \langle A_i, V \rangle$ 
19:    end if
20:  end for
21:  return  $L$ 
22: end function

```

na descrição independente de seu poder discriminatório ou da quantidade de distraidores do objeto-alvo.

Posteriormente, analisa-se o resultado do classificador responsável pela predição da relação entre o objeto-alvo e o ponto de referência (linha 3). Caso a predição deste classificador tenha sido *sem relação*, a função retorna a descrição não-relacional L de t gerada até então. Caso contrário, se houver um ponto de referência do objeto-alvo de acordo com o atributo relacional predito pelo classificador, a propriedade atrelada a este atributo é incluída na descrição de t e a função *getDescricaoObjeto* é chamada novamente para produzir a descrição do ponto de referência (linhas 4-7).

Figura 17 – Contexto com dois cones e um cubo

Para exemplificar, considere o contexto da figura 17. Dado os conjuntos $D = \{d_1, d_2, d_3\}$ e $A = \{tipo, cor, tamanho, \grave{a} esquerda de, \grave{a} direita de\}$, as propriedades de cada objeto de D podem ser representadas pelos seguintes conjuntos de propriedades:

- $d_1 = \{\langle tipo, cone \rangle, \langle tamanho, pequeno \rangle, \langle cor, verde \rangle, \langle \grave{a} esquerda de, null \rangle, \langle \grave{a} direita de, d_2 \rangle\}$
- $d_2 = \{\langle tipo, cone \rangle, \langle tamanho, grande \rangle, \langle cor, vermelho \rangle, \langle \grave{a} esquerda de, d_1 \rangle, \langle \grave{a} direita de, d_3 \rangle\}$
- $d_3 = \{\langle tipo, cubo \rangle, \langle tamanho, grande \rangle, \langle cor, verde \rangle, \langle \grave{a} esquerda de, d_2 \rangle, \langle \grave{a} direita de, null \rangle\}$

Neste exemplo, considera-se $d_1 \in D$ o objeto-alvo e P o conjunto de predições feitas pelos classificadores. Este conjunto de predições P é definido por:

$$P = \{tg = \{\langle tipo, 1 \rangle, \langle tamanho, 0 \rangle, \langle cor, 1 \rangle, \langle relação, \grave{a} direita de \rangle\}, \\ lm = \{\langle tipo, 1 \rangle, \langle tamanho, 1 \rangle, \langle cor, 0 \rangle\}\}$$

Para este exemplo, a chamada à função *getDescricaoObjeto* para descrição do objeto-alvo d_1 (linha 2) retornaria a expressão de referência preliminar abaixo:

$$L = \{d_1 = \{\langle tipo, cone \rangle, \langle cor, verde \rangle\}\}$$

Em seguida, o atributo relacional *à direita de* seria adicionado a descrição de acordo com o classificador responsável pela predição da relação entre o objeto-alvo e o ponto de referência (linhas 3-6):

$$L = \{d_1 = \{\langle tipo, cone \rangle, \langle cor, verde \rangle, \langle \grave{a} direita de, d_2 \rangle\}\}$$

Por último, a função *getDescricaoObjeto* é chamada novamente para descrição do ponto de referência d_2 (linha 7) e o algoritmo é finalizado, retornando a expressão de referência “O cone verde à direita do cone grande” (linha 10):

$$L = \{d_1 = \{\langle tipo, cone \rangle, \langle cor, verde \rangle, \langle \grave{a} direita de, d_2 \rangle\}, \\ \{d_2 = \{\langle tipo, cone \rangle, \langle tamanho, grande \rangle\}\}\}$$

Vale observar que o código apresentado na figura 16 é genérico, pois certas peculiaridades tiveram de ser consideradas de acordo com cada um dos domínios em questão. No corpus TUNA, por exemplo, a condição definida entre as linhas 3 e 9 não é utilizada, pois este corpus não possui descrições relacionais. Por outro lado, nos corpus *Stars* e *Stars2*, além do teste entre a relação do objeto-alvo e um 1º ponto de referência, também testou-se a relação entre um 1º e 2º ponto de referência.

5.2 Incorporando a Variação Humana nos Algoritmos Básicos

O foco principal deste trabalho é explorar a variação humana na tarefa de GER. Para isso, nesta seção serão mostradas as adaptações feitas para incorporá-la nos algoritmos AIE e *InteliGER*, ambos apresentados na Seção 5.1.

No algoritmo AIE, explora-se o aspecto da variação humana a partir da forma com que é feita a contabilização das frequências de atributos nas expressões de referência de treinamento para ordenação das listas de preferência do algoritmo. Na versão original do algoritmo AIE, os atributos são ordenados na lista de preferência de acordo com suas frequências em todo o conjunto de treinamento, enquanto que na versão que leva em conta a variação humana, as listas de preferência de atributos são ordenadas pela frequência dos atributos no conjunto de treinamento *por participante*. Esta estratégia foi baseada nos trabalhos Fabbriozzi, Stent e Bangalore (2008), Bohnet (2008) e Viethen, Mitchell e Krahmer (2013), discutidos nas Seções 4.2, 4.3 e 4.5, respectivamente.

No algoritmo *InteliGER*, a variação humana é explorada a partir da inclusão de novas características nos vetores de entrada dos modelos de aprendizado de máquina. Na versão do algoritmo apresentada em 5.1.2, os vetores de características são compostos apenas por características de contexto, enquanto que na versão em que se leva em conta a variação humana, estes vetores são complementados com características que modelam informações gerais e de preferência linguística de cada locutor, denominadas neste estudo de características de perfil.

Dentre as características de perfil, as características de informações gerais referem-se exclusivamente ao locutor que gerou a expressão de referência, e têm o intuito de modelar aspectos mais básicos da variação humana na tarefa de GER, semelhante aos estudos em Viethen e Dale (2010) e Viethen (2011). Como características deste tipo, foram modeladas informações do locutor como seu gênero, sua faixa etária (vide tabela

13) e seu código de identificação no corpus. Estas características, mostradas na tabela 12, foram obtidas através das informações adicionais do corpus onde o locutor foi participante.

Tabela 12 – Informações gerais do locutor

Característica	Descrição
Speaker_ID	identificação do locutor no corpus onde foi participante
Speaker_Gender	gênero do locutor (<i>{masculino, feminino}</i>)
Speaker_Age-Group	faixa etária do Locutor (Tabela 13)

Tabela 13 – Faixas etárias dos locutores

Faixa Etária	Descrição
20 $\geq$ Idade	Locutores de até 20 anos
20 < Idade $\leq$ 25	Locutores entre 21 e 25 anos
25 < Idade $\leq$ 30	Locutores entre 26 e 30 anos
30 < Idade $\leq$ 40	Locutores entre 31 e 40 anos
40 < Idade $\leq$ 50	Locutores entre 41 e 50 anos
50 < Idade $\leq$ 60	Locutores entre 51 e 60 anos
60 < Idade	Locutores maiores de 61 anos

Além das informações gerais do locutor, também foram definidas como características de perfil as preferências linguísticas de cada locutor, visando modelar a variação humana na seleção de atributos e no grau de especificação das expressões de referência. Estas características, obtidas com base em expressões de referência de treinamento de cada locutor, são apresentadas na tabela 14.

Tabela 14 – Características que modelam as preferências linguísticas de um locutor na seleção de atributos e no grau de especificação de suas expressões de referência

Característica	Descrição
TG_Frequency	frequência dos atributos na descrição dos objetos-alvo
LM_Frequency	frequência dos atributos na descrição dos pontos de referência
TG_Description_Size	tamanho médio das descrições dos objetos-alvo
LM_Description_Size	tamanho médio das descrições dos pontos de referência
Overspecification_Mean	porcentagem de descrições superespecificadas
Underspecification_Mean	porcentagem de descrições subespecificadas
Minimal_Mean	porcentagem de descrições mínimas
Relation_Mean	média de relações/pontos de referência utilizados por descrição

As características que modelam as preferências linguísticas do locutor estão atreladas à frequência de atributos (*TG_Frequency* e *LM_Frequency*), ao tamanho (*TG_Description_Size* e *LM_Description_Size*) e especificação das expressões de referência (*Overspecification_Mean*, *Underspecification_Mean*, *Minimal_Mean* e *Relation_Mean*). Espera-se que as características relacionadas à frequência de atributos ajudem a modelar o problema de variação humana

na seleção de atributos para compor uma expressão de referência, enquanto as características relacionadas ao tamanho e especificação da expressão de referência contribuam para determinar o exato grau de especificação da descrição a ser predita.

Para classificar se uma determinada expressão de referência é subespecificada, atividade essa feita como parte da formulação da característica *Underspecification_Mean*, executa-se o algoritmo Incremental (DALE; REITER, 1995) com sua lista de preferência composta apenas com os atributos que compõem esta descrição com o intuito de gerá-la novamente. Caso ainda haja distraidores para o objeto-alvo ao fim do algoritmo, a expressão de referência em questão é classificada como subespecificada.

Quanto à questão de classificar uma determinada expressão de referência como superespecificada ou não, atividade essa feita como parte da formulação da característica *Overspecification_Mean*, executa-se o algoritmo Incremental mais de uma vez com a lista de preferência composta apenas dos atributos que compõem esta descrição. Em cada execução, utiliza-se uma ordem diferente dos atributos na lista de preferência. Caso em alguma das execuções o algoritmo gere uma expressão de referência que distinga o objeto-alvo antes de terminar a iteração de todos os atributos da lista de preferência, a descrição é classificada como superespecificada. Por último, caso a descrição não seja classificada como subespecificada ou superespecificada, essa é classificada como uma expressão de referência mínima.

No caso especial do corpus TUNA, nenhuma das características de perfil relacionadas a pontos de referência (*LM_Frequency*, *LM_Description_Size* e *Relation_Mean*) são utilizadas, pois este corpus não possui expressões de referência relacionais. Além disso, as características de informações gerais *Speaker_Gender* e *Speaker_Age-Group* também não são utilizadas neste caso, pois não são fornecidas pelos provedores do corpus (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009).

Espera-se que a construção de modelos de aprendizado de máquina, combinando características de contexto com as características de perfil, possibilite detectar os padrões de referência correlacionados ao tipo de contexto e preferência do locutor. Com o sucesso dos modelos em aprender estes padrões, expressões de referência serão então geradas de acordo com a preferência de cada locutor para cada tipo de contexto.

5.3 Variações dos Algoritmos AIE e *InteliGER* consideradas

Nos experimentos desenvolvidos na presente pesquisa, detalhados no Capítulo 6, serão utilizadas como modelos computacionais de GER diversas variações dos algoritmos Incremental Estendido (AIE) e *InteliGER*. Nesta seção, estas variações serão detalhadas tanto na versão que não leva em conta a variação humana, caracterizadas pelo símbolo $-$, quanto na versão que leva esta questão em conta, reconhecidas pelo símbolo $+$. Especificamente, as variações do algoritmo AIE são tratadas na Seção 5.3.1, e as variações do algoritmo *InteliGER* na Seção 5.3.2.

5.3.1 Variações do Algoritmo Incremental Estendido

Duas variações do algoritmo AIE foram consideradas no experimento:

1. *Algoritmo AIE com contabilização das frequências (AIE-)*: variação do algoritmo AIE onde computa-se uma lista de frequência de atributos única para cada domínio, ou seja, sem distinção entre locutores.
2. *Algoritmo AIE com contabilização das frequências por participante (AIE+)*: semelhante à variação acima, com a diferença que a ordenação das listas de preferência de atributos é feita de forma individualizada para cada locutor.

5.3.2 Variações do Algoritmo *InteliGER*

Para utilização do algoritmo *InteliGER*, os classificadores foram implementados em duas versões. Na primeira, os classificadores são Máquinas de Vetores de Suporte (CORTES; VAPNIK, 1995) com *kernel Gaussiano (SVM)*, enquanto na segunda versão, os mesmos são Árvores de Classificação e Regressão (BREIMAN et al., 1984) com o Coeficiente de Gini (GINI, 1912) como medida de impureza (*CART*). Para treinamento e teste dos classificadores multi-classe, foi utilizada a estratégia “um-contra-um” (KNERR; PERSONNAZ; DREYFUS, 1990). Para as duas versões de classificadores, o treinamento foi feito seguindo três abordagens:

1. *All*: treinamento de classificadores por atributo.
2. *Speaker*: treinamento de classificadores por atributo e por locutor.

3. *Profile*: treinamento de classificadores por atributo e por perfil de locutor. Definiu-se três perfis de locutores com base em suas expressões de referência de treinamento: locutores que sempre superespecificam suas descrições, locutores que sempre geram descrições mínimas e locutores que não seguem uma estratégia fixa de descrição.

Na abordagem *All*, classificadores são treinados para cada atributo com expressões de referência de diferentes locutores. Na abordagem *Speaker*, os classificadores são segmentados por locutores, ou seja, um grupo de classificadores, correspondentes a cada atributo, é treinado para cada locutor unicamente com suas expressões de referência. Por fim, na abordagem *Profile*, um grupo de classificadores, correspondentes a cada atributo, é treinado para cada um dos três perfis definidos.

Para exemplificar, considere um contexto caracterizado por 4 atributos. Suponha que, neste contexto, 8 locutores produziram descrições para referenciar os objetos nele existentes. Destes locutores, 4 sempre superespecificaram suas descrições, enquanto 1 sempre produziu descrições mínimas. Para a abordagem *All* neste caso, um total de 4 classificadores, um para cada atributo, seriam treinados com todas as expressões de referência de todos locutores. Na abordagem *Speaker*, um grupo de 4 classificadores, correspondentes a cada atributo, seria treinado para cada um dos 8 locutores, totalizando 32 classificadores. Quanto à abordagem *Profile*, um grupo de 4 classificadores, correspondentes a cada atributo, seria treinado para cada um dos 3 perfis definidos, totalizando 12 classificadores.

Para cada uma das versões dos classificadores em cada abordagem de treinamento, treinou-se modelos com dois tipos de vetores de características diferentes:

1. *VAR-*: vetores de entrada apenas com características de contexto.
2. *VAR+*: vetores de entrada com características de contexto e de perfil.

Como resultado, foram definidas 12 variações do algoritmo *InteliGER*, sumarizadas na tabela 15.

Em todas as variações do algoritmo *InteliGER*, o treinamento dos classificadores relacionados aos objetos-alvo foi feito com todos tipos de expressões de referência, enquanto que os classificadores relacionados aos pontos de referência foram treinados apenas com descrições relacionais.

Tabela 15 – *Variações do algoritmo InteliGER*

Classificador	Treinamento	Entrada	Variação
SVM	<i>All</i>	VAR- VAR+	SVM_ <i>All</i> _VAR- SVM_ <i>All</i> _VAR+
	<i>Speaker</i>	VAR- VAR+	SVM_ <i>Speaker</i> _VAR- SVM_ <i>Speaker</i> _VAR+
	<i>Profile</i>	VAR- VAR+	SVM_ <i>Profile</i> _VAR- SVM_ <i>Profile</i> _VAR+
CART	<i>All</i>	VAR- VAR+	CART_ <i>All</i> _VAR- CART_ <i>All</i> _VAR+
	<i>Speaker</i>	VAR- VAR+	CART_ <i>Speaker</i> _VAR- CART_ <i>Speaker</i> _VAR+
	<i>Profile</i>	VAR- VAR+	CART_ <i>Profile</i> _VAR- CART_ <i>Profile</i> _VAR+

5.4 Construção do *Córpus Zoom*

Com o intuito de investigar a variação humana em GER, construiu-se um novo *córpus* de expressões de referência denominado *Zoom*. Este *córpus* foi elaborado em parceria com o Departamento de Ciências da Computação da Universidade de Córdoba¹ para coletar descrições a objetos em mapas geográficos de participantes que tenham como língua nativa o português e o espanhol.

Como motivação para sua criação, acredita-se que em um *córpus* composto por expressões de referência a um ou dois objetos-alvo, geradas por um grande número de participantes de línguas nativas diferentes, em um domínio mais realista que *córpus* anteriores realizados em ambientes controlados, permitirá um estudo mais aprofundado sobre a variação humana na tarefa de GER. Partindo desta premissa, embora a construção deste recurso não seja o objetivo principal do trabalho, coletou-se o *córpus Zoom* de maneira que houvesse descrições produzidas por indivíduos distintos nos mesmos contextos, ao contrário de *córpus* já existentes em que contextos de referência podem variar de um locutor para outro. A coleta e anotação dos dados deste *córpus* é explicada a seguir.

5.4.1 Coleta e Anotação dos Dados

Para coleta dos dados, foi elaborado um experimento *online* na forma de dois questionários abertos, um para cada categoria de participante, levando em conta sua língua nativa. Nestes questionários, são mostradas 22 imagens de mapas retirados do sítio *OpenStreetMap*². Nestas imagens editadas, os objetos-alvo são apontados por uma seta,

¹<http://www.cs.famaf.unc.edu.ar>

²<http://www.openstreetmap.org>

conforme a figura 18. Para cada imagem, pede-se ao participante completar a frase “Seria interessante conhecer” em uma caixa de texto, onde se deve referenciar os objetos-alvo. Para o questionário destinado aos nativos da língua portuguesa, os mapas apresentados mostram regiões da cidade de Lisboa em Portugal; e para os nativos da língua espanhola, os mapas apresentados mostram regiões da cidade de Madrid na Espanha. Das 22 imagens, as duas primeiras servem apenas para ambientação dos participantes ao experimento e não são incluídas nos dados coletados.

Os questionários foram disponibilizados através de um *website*³ criado pelos colaboradores da Universidade de Córdoba. Pelo fato de ser um experimento *online*, a tarefa é conduzida de maneira não-supervisionada e em regime voluntário.

Figura 18 – Exemplo de cena mostrada ao participante no experimento do *cópus Zoom*

Estabeleceu-se como meta a coleta de expressões de referência de 100 participantes para cada uma das línguas que o experimento abrange. A coleta das descrições dos participantes nativos da língua portuguesa foi finalizada em Setembro de 2013. Posteriormente, as 2000 expressões de referência resultantes foram anotadas por dois colaboradores independentes segundo um esquema previamente estabelecido pelos idealizadores do *cópus*.

³<http://www.cs.famaf.unc.edu.ar/romina/pagina/>

Neste esquema, cada expressão de referência é composta de uma descrição para cada objeto-alvo e para cada um dos quatro pontos de referência que podem ser descritos para cada objeto-alvo. Os objetos-alvo podem ser representados por 11 atributos, sendo 4 não-relacionais e 9 relacionais. Quanto aos pontos de referência, esses somente podem ser representados pelos 4 atributos atômicos. Assim, a anotação de uma expressão completa pode envolver até 54 atributos. Os atributos e seus valores possíveis para a cena do experimento representada na figura 18 são ilustrados no Apêndice A.

Após o fim da anotação, o responsável por este projeto assumiu o papel de juiz e realizou a análise de concordância entre os trabalhos dos dois anotadores para a produção da versão final do *córpus* em português. Segundo o índice Kappa (COHEN, 1960), a concordância entre os anotadores foi de 84% em nível de atributo. Nesta análise de concordância entre os anotadores, todas as descrições de 4 participantes, juntamente com mais 106 expressões de referência de diversos outros, foram descartadas por não convergi-rem com o objetivo do experimento. Sendo assim, o *córpus Zoom* na versão em português conta com 1814 expressões de referência de 96 participantes. Além da anotação das expressões de referência, as 22 imagens do sítio *OpenStreetMap*, utilizadas no experimento, também foram anotadas para uso como entrada a algoritmos de GER. A porção em espanhol do *córpus* ainda encontra-se em fase de anotação pela equipe de colaboradores no exterior e não foi aproveitada no presente trabalho. A porção em português aqui considerada é denominada *Zoom-Pt*.

Capítulo 6

Avaliação

Neste capítulo é apresentada a avaliação intrínseca dos modelos computacionais de GER implementados na presente pesquisa. Na Seção 6.1, os corpú de expressões de referência utilizados como conjunto de dados são apresentados; na Seção 6.2, apresenta-se os modelos computacionais de GER avaliados; na Seção 6.3, estão as hipóteses do estudo; na Seção 6.4, o procedimento adotado para condução dos diferentes experimentos feitos está detalhado; na Seção 6.5, os resultados obtidos são apresentados; e na Seção 6.6, apresenta-se a discussão sobre esses.

6.1 Conjunto de Dados

A avaliação dos modelos propostos foi realizada com base em sete corpú de GER: ambos domínios do corpú TUNA (BELZ; GATT, 2007; GATT; BELZ; KOW, 2008, 2009), GRE3D3 (VIETHEN; DALE, 2008), GRE3D7 (VIETHEN; DALE, 2011), *Stars* (TEIXEIRA et al., 2014; PARABONI et al., 2014), *Stars2* (IACOVELLI; GALINDO; PARABONI, 2014) e *Zoom-Pt*. Na tabela 16, é mostrado o número de expressões de referência, participantes e contextos de cada corpú utilizado nos experimentos aqui apresentados.

A escolha destes corpú como conjunto de dados deu-se pela grande disponibilidade de expressões de referência, geradas por diferentes participantes nos mesmos contextos, que esses possuem. Esta variedade permite que a questão da variação humana na tarefa de GER se torne mais evidente. Além disso, os corpú GRE3D3/7, *Stars* e TUNA já foram utilizados para avaliação de algoritmos de GER (GATT; BELZ; KOW, 2008, 2009; BOHNET, 2008; FABBRIZIO; STENT; BANGALORE, 2008; VIETHEN; DALE, 2010; VIETHEN, 2011; VIETHEN; MITCHELL; KRAHMER, 2013; TEIXEIRA et al., 2014). Logo, a avaliação destes algoritmos nos trabalhos citados puderam servir de termo de comparação para os modelos computacionais de GER propostos e utilizados no presente estudo.

No caso dos corpú TUNA e *Zoom-Pt*, apenas suas expressões de referência singulares foram utilizadas. Quanto aos corpú GRE3D3/7, as cenas que compunham os respectivos contextos tiveram de ser anotadas semanticamente, pois apenas as expressões

Tabela 16 – Número de descrições, participantes e cenas de cada *córpus*, utilizados como conjunto de dados

Córpus	Descrições	Participantes	Cenas
TUNA Furniture	420	60	7
TUNA People	360	60	6
GRE3D3	630	63	10
GRE3D7	4480	287	16
<i>Stars</i>	384	64	6
<i>Stars2</i>	1204	76	64
<i>Zoom-Pt</i>	821	96	9

de referência geradas pelos participantes são fornecidas anotadas. No Apêndice B consta um exemplo da anotação de um dos domínios do *córpus* GRE3D7. Com relação ao *córpus* *Stars*, foram utilizados apenas as expressões de referência produzidas para as cenas das condições do experimento, conforme descrito em Teixeira et al. (2014). As descrições referentes as cenas de *filler* deste *córpus* foram desconsideradas, pois não haviam sido completamente anotadas na ocasião deste estudo. Além dessas, nenhuma outra alteração foi realizada nestes conjuntos de dados.

6.2 Modelos Computacionais de GER Avaliados

Nos experimentos feitos para explorar a variação humana na tarefa de GER, variações dos algoritmos Incremental Extendido (AIE) e *InteliGER* foram utilizados como modelos computacionais de GER. Ambos algoritmos foram utilizados em versões que não incorporam a variação humana, como mostrado na Seção 5.1; e outras que a incorporam, conforme detalhado na Seção 5.2. As diferentes variações de ambos algoritmos são apresentadas na Seção 5.3 tanto nas versões que levam em conta a variação humana, quanto nas versões que não levam esta questão em conta.

6.3 Hipóteses

As hipóteses avaliadas a partir dos experimentos foram:

- H_1 : modelos computacionais de GER que levam em conta a variação humana geram expressões de referência mais próximas às de seres humanos do que modelos que não levam esta questão em conta.

Para confirmação desta hipótese, espera-se que o coeficiente de Dice médio de todos os modelos computacionais de GER que levam a variação humana em conta supere o coeficiente de Dice médio de todos os modelos que não levam esta questão em conta. Os modelos com e sem variação humana utilizados para obtenção dos coeficientes de Dice médio estão sumarizados na tabela 17.

Tabela 17 – Modelos computacionais de GER com e sem variação humana, considerados no teste de H_1

Sem Variação Humana	Com Variação Humana
AIE-	AIE+
SVM_All_VAR-	SVM_All_VAR+
SVM_Speaker_VAR-	SVM_Speaker_VAR+
SVM_Profile_VAR-	SVM_Profile_VAR+
CART_All_VAR-	CART_All_VAR+
CART_Speaker_VAR-	CART_Speaker_VAR+
CART_Profile_VAR-	CART_Profile_VAR+

- H_2 : modelos computacionais de GER baseados em técnicas de aprendizado de máquina produzem expressões de referência mais próximas a descrições de seres humanos do que modelos existentes, como o algoritmo Incremental.

Para confirmação desta hipótese, espera-se que os coeficientes de Dice médio das variações do algoritmo *InteliGER*, tanto nas versões cujos classificadores são Máquinas de Vetores de Suporte, quanto nas versões onde os mesmos são Árvores de Classificação e Regressão, superem o coeficiente de Dice médio das variações do algoritmo AIE. Na tabela 18, estão sumarizadas as variações dos algoritmos AIE e *InteliGER* utilizadas para obtenção dos coeficientes de Dice médio.

- H_3 : variações do algoritmo *InteliGER* treinadas a partir da abordagem *Speaker* produzem expressões de referência mais próximas a descrições de seres humanos do que as variações treinadas com as abordagens *All* e *Profile*.

Para confirmação desta hipótese, espera-se que o coeficiente de Dice médio das variações treinadas com a abordagem *Speaker* supere o coeficiente de Dice médio das variações treinadas com as abordagens *All* e *Profile*. As variações do algoritmo *InteliGER* utilizadas para obtenção dos coeficientes de Dice médio estão sumarizadas na tabela 19.

- H_4 : variações do algoritmo *InteliGER* treinadas a partir da abordagem *Profile* geram expressões de referência mais próximas a descrições de seres humanos do que as variações treinadas com a abordagem *All*.

Tabela 18 – Variações do algoritmo AIE e InteliGER, considerados no teste de H_2

Algoritmo	Variação
AIE	AIE-
	AIE+
InteliGER	SVM_All_VAR-
	SVM_All_VAR+
	SVM_Speaker_VAR-
	SVM_Speaker_VAR+
	SVM_Profile_VAR-
	SVM_Profile_VAR+
	CART_All_VAR-
	CART_All_VAR+
	CART_Speaker_VAR-
	CART_Speaker_VAR+
	CART_Profile_VAR-
	CART_Profile_VAR+

Tabela 19 – Variações do algoritmo InteliGER nas abordagens de treinamento *All*, *Speaker* e *Profile*, consideradas no teste de H_3

<i>All</i>	<i>Speaker</i>	<i>Profile</i>
SVM_All_VAR-	SVM_Speaker_VAR-	SVM_Profile_VAR-
SVM_All_VAR+	SVM_Speaker_VAR+	SVM_Profile_VAR+
CART_All_VAR-	CART_Speaker_VAR-	CART_Profile_VAR-
CART_All_VAR+	CART_Speaker_VAR+	CART_Profile_VAR+

Para confirmação desta hipótese, espera-se que o coeficiente de Dice médio das variações treinadas com a abordagem *Profile* supere o coeficiente de Dice médio das variações treinadas com a abordagem *All*.

Conforme detalhado na Seção 6.1, os modelos computacionais de GER implementados na presente pesquisa, definidos na tabela 17, foram testados em sete corpú. Logo, para realização dos testes das hipóteses acima, foi utilizado o coeficiente de Dice médio de cada modelo calculado com base nos sete corpú sob consideração.

6.4 Procedimento

Além de medir o grau de semelhança das expressões de referência geradas pelos modelos computacionais de GER com as descrições originais dos corpú pelo coeficiente de Dice, foi também medido o coeficiente de MASI e Acurácia para fins ilustrativos. Quanto às variações do algoritmo *InteliGER*, essas tiveram ainda seus classificadores avaliados pelas métricas de Precisão, Revocação e *F-Score*. Nas variações deste algoritmo em que os classificadores são Máquinas de Vetores de Suporte, a medida de área abaixo da curva

ROC (*Area Under the Curve* - AUC) também foi calculada.

Vale observar que para obtenção das medidas de Precisão, Revocação, *F-Score* e AUC dos classificadores multi-classe que previam a relação entre dois objetos, esses foram analisados como classificadores binários, indicando 1 se na determinada descrição houvesse relação, e 0 caso contrário.

Para validação dos algoritmos utilizados no experimento, a metodologia de validação cruzada foi utilizada com 10 *folds* para os corpúscos GRE3D3, GRE3D7 e *Stars2*; e 6 *folds* para os corpúscos *Stars*, TUNA e *Zoom-Pt*. A escolha do número de *folds* para ambos os corpúscos foi feita de modo que o número de expressões de referência de cada locutor por *fold* fosse mantido equilibrado.

No caso especial das variações do algoritmo *InteliGER* em que os classificadores são Máquinas de Vetores de Suporte com *kernel Gaussiano*, o método de *Grid Search* foi utilizado com o objetivo de buscar os melhores valores para os parâmetros de custo da SVM (C) e γ (γ) do *kernel* em questão. Este processo é feito da seguinte maneira: para cada uma das fases do método de validação cruzada, 1 *fold* é separado para teste e os $k - 1$ *folds* restantes são utilizados num segundo processo de validação cruzada, onde classificadores são treinados com diferentes combinações dos parâmetros C (1, 10, 100 e 1000) e γ (1, 0.1, 0.001 e 0.0001). Para todas SVMs treinadas com diferentes combinações de parâmetros para predição de um determinado atributo, é escolhida a combinação cujo classificador obtiver o melhor resultado de acordo com a Acurácia nas predições. Posteriormente, a combinação escolhida para cada atributo será utilizada como parâmetro para o classificador treinado nos $k - 1$ *folds* e testado no *fold* de teste. Esta abordagem se repete em todas as iterações do primeiro processo de validação cruzada.

O teste das hipóteses deu-se através da comparação dos coeficientes de Dice médio dos diferentes modelos computacionais de GER de acordo com suas execuções nos sete corpúscos definidos. Para fins estatísticos, o teste de Wilcoxon foi utilizado sobre estes coeficientes. Além disso, o teste estatístico de Chi-Quadrado foi realizado sobre a Acurácia dos modelos para fins de complementação desta avaliação.

Os experimentos mostrados neste trabalho, incluindo os algoritmos básicos, suas variações propostas como modelos computacionais de GER, os leitores de corpúscos e as métricas de avaliação foram implementados na linguagem *Python*. Para implementação das técnicas de aprendizado de máquina do algoritmo *InteliGER*, utilizou-se a biblioteca *Scikit-learn* (PEDREGOSA et al., 2011). Toda implementação está disponível no repositório

Github do autor da pesquisa¹.

6.5 Resultados

Na tabela 20 é apresentado o coeficiente de Dice médio para cada um dos modelos computacionais de GER segundo seu respectivo desempenho nos sete corpúscos utilizados na presente pesquisa. Para H_1 , as médias dos coeficientes de Dice dos modelos que levam em conta a variação humana e dos modelos que não levam esta questão em conta também são apresentadas.

Tabela 20 – Coeficiente de Dice médio e Desvio Padrão dos modelos computacionais de GER para H_1

Sem Variação Humana			Com Variação Humana		
Modelo	Dice	δ	Modelo	Dice	δ
AIE-	0,77	0,06	AIE+	0,78	0,06
SVM_All_VAR-	0,78	0,07	SVM_All_VAR+	0,85	0,05
SVM_Speaker_VAR-	0,80	0,06	SVM_Speaker_VAR+	0,83	0,06
SVM_Profile_VAR-	0,79	0,06	SVM_Profile_VAR+	0,85	0,05
CART_All_VAR-	0,78	0,06	CART_All_VAR+	0,84	0,05
CART_Speaker_VAR-	0,75	0,09	CART_Speaker_VAR+	0,80	0,08
CART_Profile_VAR-	0,78	0,06	CART_Profile_VAR+	0,85	0,05
Total	0,78	0,07	Total	0,83	0,06

Com base nas médias apresentadas na tabela 20, visualiza-se que modelos computacionais de GER que incorporam a variação humana são superiores aos que não levam esta questão em conta. Este resultado é comprovado estatisticamente conforme teste de Wilcoxon apresentado na tabela 21. Logo, comprova-se H_1 : modelos computacionais de GER que levam em conta a variação humana geram expressões de referência mais próximas a descrições de seres humanos do que modelos que não levam esta questão em conta.

Tabela 21 – Teste estatístico de Wilcoxon sobre os coeficientes de Dice para H_1

Pares	W	p
AIE- X AIE+	507075,0	< 0,0001
SVM_All_VAR- X SVM_All_VAR+	1320714,0	< 0,0001
SVM_Speaker_VAR- X SVM_Speaker_VAR+	342043,0	< 0,0001
SVM_Profile_VAR- X SVM_Profile_VAR+	1348513,0	< 0,0001
CART_All_VAR- X CART_All_VAR+	2293514,5	< 0,0001
CART_Speaker_VAR- X CART_Speaker_VAR+	1791273,5	< 0,0001
CART_Profile_VAR- X CART_Profile_VAR+	2172332,0	< 0,0001

¹<https://github.com/ThiagoCF05/EACH-GRE>

Para H_2 , os coeficientes de Dice das variações do algoritmo AIE, do algoritmo *InteliGER* cujos classificadores são Máquinas de Vetores de Suporte, e das variações deste mesmo algoritmo cujos classificadores são Árvores de Classificação e Regressão são apresentadas na tabela 22. A partir destes resultados, confirmados estatisticamente conforme o teste de Wilcoxon apresentado na tabela 23, pode-se visualizar que os coeficientes de Dice das variações do algoritmo *InteliGER*, tanto nas versões cujos classificadores são Máquinas de Vetores de Suporte, quanto nas versões cujos classificadores são Árvores de Classificação e Regressão, são superiores ao coeficiente de Dice das variações do algoritmo AIE. Logo, H_2 pode ser comprovada.

Tabela 22 – Resultados dos experimentos para H_2

Algoritmo	Dice	δ
AIE	0,77	0,06
<i>InteliGER SVM</i>	0,82	0,06
<i>InteliGER CART</i>	0,80	0,07

Tabela 23 – Teste estatístico de Wilcoxon sobre os coeficientes de Dice para H_2

Pares	W	p
<i>AIE X SVM</i>	215169626,5	< 0,0001
<i>AIE X CART</i>	228820422,0	< 0,0001
<i>SVM X CART</i>	201082455,0	< 0,0001

Com relação à H_3 e H_4 , os coeficientes de Dice das variações do algoritmo *InteliGER* treinados nas abordagens *All*, *Speaker* e *Profile* são apresentadas na tabela 24. Os resultados do teste de Wilcoxon sobre os coeficientes de Dice destes três grupos de variações é apresentado para fins estatísticos na tabela 25.

Tabela 24 – Resultados dos experimentos para H_3 e H_4

Abordagem	Dice	δ
<i>All</i>	0,81	0,06
<i>Speaker</i>	0,80	0,07
<i>Profile</i>	0,82	0,06

Tabela 25 – Teste estatístico de Wilcoxon sobre os coeficientes de Dice para H_3 e H_4

Pares	W	p
<i>All X Speaker</i>	115119329,0	< 0,0001
<i>All X Profile</i>	100334854,0	< 0,0001
<i>Speaker X Profile</i>	109299469,0	< 0,0001

A partir dos resultados nas tabelas 24 e 25, pode-se visualizar que as variações do algoritmo *InteliGER* treinadas de acordo com a abordagem *Profile* apresentam um coeficiente de Dice médio superior aos das variações treinadas com as abordagens *All* e *Speaker*. Quanto às duas abordagens inferiores, visualiza-se que as variações treinadas com a abordagem *All* apresentam coeficiente de Dice médio superior ao mesmo coeficiente das variações treinadas com a abordagem *Speaker*. Ambos resultados contrariam H_3 , na qual acredita-se que as variações do algoritmo *InteliGER* treinadas de acordo com a abordagem *Speaker* geram expressões de referência mais próximas a expressões geradas por seres humanos do que as variações treinadas com as abordagens *All* e *Profile*. Entretanto, H_4 pode ser comprovada a partir da superioridade das variações do algoritmo *InteliGER* treinadas com a abordagem *Profile* frente às variações treinadas com a abordagem *All*.

Por fim, os coeficientes de Dice, MASI e Acurácia de cada modelo computacional de GER em cada corpus são apresentados no Apêndice C; e os resultados de Precisão, Revocação, *F-Score* e AUC dos classificadores do algoritmo *InteliGER* para cada atributo de cada corpus são apresentados no Apêndice D.

6.6 Discussão

6.6.1 Análise de Erros

A partir dos resultados apresentados na Seção 6.5, as hipóteses H_1 , H_2 e H_4 puderam ser comprovadas, enquanto a hipótese H_3 foi refutada. Em outras palavras, pode-se constatar que modelos computacionais de GER que levam em conta a variação humana e modelos baseados em técnicas de aprendizado de máquina geram respectivamente expressões de referência mais próximas a descrições de seres humanos do que modelos que não levam a variação humana em conta e modelos baseados em algoritmos consagrados de GER. Quanto aos modelos baseados em técnicas de aprendizado de máquina, viu-se que, quando treinados com a abordagem *Profile*, têm melhor desempenho que quando treinados com as abordagens *All* e *Speaker*.

Sobre a hipótese H_1 , essa foi comprovada a partir da superioridade da média de Dice de todos os modelos computacionais de GER que incorporam a variação humana frente à média de todos os modelos que não a incorporam, como pode ser visualizado na tabela 20 e comprovado estatisticamente na tabela 21. Do ponto de vista do desempenho nos sete corpus utilizados na presente pesquisa, nas tabelas 20 e 21 pode-se também constatar que todos os modelos em suas versões que levam a variação humana em conta

possuem coeficiente de Dice médio significativamente superior às versões que não levam esta questão em conta. Contudo, alguns modelos computacionais de GER não apresentaram diferença significativa entre seus coeficientes de Dice nas versões que incorporam e não incorporam a variação humana nos corpúscos GRE3D3, GRE3D7, *Stars* e *Zoom-Pt*, conforme resultados apresentados no Apêndice C.

Nos corpúscos GRE3D3 e GRE3D7, as variações do algoritmo AIE nas versões que incorporam e não incorporam a variação humana (*AIE-/AIE+*) não apresentam diferença significativa entre seus coeficientes de Dice segundo teste de Wilcoxon (GRE3D3: $W = 663,0 / p < 0,8126$; GRE3D7: $W = 1871,5 / p < 0,2970$). Estes corpúscos são compostos por cenas de contextos simples, onde seus objetos podem ser desambiguados uns dos outros com apenas o uso, em diferentes combinações, dos atributos atômicos *tipo*, *cor* e *tamanho*. Este fato faz com que estes três atributos sejam os mais frequentes em todas as expressões de referência de ambos os corpúscos, assim como nas expressões de referência de 49 (78%) participantes do corpúscos GRE3D3 e de 264 (94%) do corpúscos GRE3D7. Logo, como a incorporação da variação humana nas variações do algoritmo AIE se dá pelo cálculo da frequência de atributos por participante, acredita-se que as listas de preferência de atributos do algoritmo AIE ordenadas de acordo a frequência do uso de atributos em todo conjunto de treinamento ou ordenadas segundo a frequência do uso de atributos por participantes são muito semelhantes para que expressões de referência significativamente diferentes umas das outras sejam geradas.

No corpúscos *Stars*, os pares de variações *SVM_All_VAR-/SVM_All_VAR+* e *SVM_Profile_VAR-/SVM_Profile_VAR+* não possuem diferença significativa entre seus coeficientes de Dice: $W = 17493,5 / p < 0,9177$; e $W = 18358,5 / p < 0,4416$, respectivamente. Entretanto, as versões destes dois pares que incorporam a variação humana apresentam graus de acurácia significativamente superiores às versões que não a incorporam, conforme teste de Chi-Quadrado: $\chi^2 = 24,28 / p < 0,0001$ e $\chi^2 = 58,74 / p < 0,0001$, respectivamente. Em outras palavras, constata-se que as versões que incorporam a variação humana destes pares geram um maior número de expressões de referência exatamente iguais a descrições dos participantes do corpúscos *Stars* do que as versões que não levam esta questão em conta.

Ainda tratando da hipótese H_1 no corpúscos *Stars*, também constata-se que o par de variações *SVM_Speaker_VAR-/SVM_Speaker_VAR+* não possui diferença significativa entre seus coeficientes de Dice e graus de acurácia neste corpúscos: $W = 498,5 / p < 0,6462$ e $\chi^2 = 0,71 / p < 0,3984$. Neste caso, ao analisar as medidas de *F-Score* e AUC de

cada classificador de cada atributo do corpus *Stars* destas duas variações no Apêndice D, visualiza-se que os resultados dos classificadores que apenas têm características de contexto como entrada são semelhantes e em alguns casos, como os classificadores *TG_VPos*, *LM_Type* e *LM_Relation*, até superiores aos resultados dos classificadores que também possuem as características de perfil como entrada. Logo, percebe-se que não há relevância das características de perfil neste caso.

Com relação ao corpus *Zoom-Pt*, esse, no geral, foi o corpus onde os modelos implementados obtiveram os piores desempenhos. Especificamente, o coeficiente de Dice médio dos modelos computacionais de GER executados no corpus *Zoom-Pt* foi muito abaixo do coeficiente de Dice médio dos modelos computacionais de GER executados em todos os corpus utilizados. Com relação ao teste de H_1 neste corpus, visualiza-se que os pares *AIE-/AIE+* ($W = 71308,5 / p < 0,1066$), *SVM_Speaker_VAR-/SVM_Speaker_VAR+* ($W = 15453,5 / p < 0,6751$), *SVM_Profile_VAR-/SVM_Profile_VAR+* ($W = 28911,5 / p < 0,0007$), *CART_All_VAR-/CART_All_VAR+* ($W = 63768,0 / p < 0,0319$) e *CART_Profile_VAR-/CART_Profile_VAR+* ($W = 63540,0 / p < 0,0271$) não mostraram diferença significativa entre seus coeficientes de Dice médio.

Acredita-se que o mau resultado no corpus *Zoom-Pt* se deva a dois motivos: sua anotação semântica e suas expressões de referência relacionais. Considera-se a anotação semântica do corpus *Zoom-Pt* um gargalo, pois essa só considera três atributos atômicos: *tipo* do objeto, *nome* do objeto e um atributo *outros*, onde características diversas do objeto, que não sejam seu tipo ou seu nome, são armazenadas. Este fato faz com que o corpus *Zoom-Pt*, dentre os sete corpus utilizados, seja o que possui o menor número de tipos de atributos atômicos disponíveis para serem utilizados na desambiguação dos objetos-alvo, contribuindo para a ineficiência de explorar a variação humana na tarefa de GER. Espera-se em um trabalho futuro segmentar o atributo *outros* em diferentes atributos semânticos no intuito de solucionar este problema.

O segundo gargalo do corpus *Zoom-Pt* tange suas expressões de referência relacionais. Conforme citado na Seção 5.1.2.1, existem expressões de referência neste corpus as quais possuem dois pontos de referência para um mesmo objeto, como em “O restaurante na esquina da *Rua Anchieta* com a *Rua Capelo*”. Estas expressões de referência são difíceis de serem reproduzidas pelos algoritmos de GER utilizados na presente pesquisa, principalmente pelo algoritmo *InteliGER*. Ainda assim, considerando-se que esse é um primeiro estudo de GER fazendo uso deste corpus, espera-se que estes resultados sirvam de referência para melhorias futuras.

Com relação à hipótese H_2 , essa foi confirmada com base na superioridade dos coeficientes de Dice médio das variações do algoritmo *InteliGER* frente ao coeficiente de Dice médio das variações do algoritmo AIE. Esta confirmação reforça trabalhos como Viethen e Dale (2010), Viethen (2011) e Pereira et al. (2012), os quais utilizaram técnicas de aprendizado de máquina para executar a tarefa de GER previamente.

Apesar de na presente pesquisa não ter sido definida nenhuma hipótese de comparação entre as variações do algoritmo *InteliGER* cujos classificadores são Máquinas de Vetores de Suporte e as variações cujos classificadores são Árvores de Classificação e Regressão, nas tabelas 22 e 23 pode-se observar a superioridade das variações do algoritmo *InteliGER* na qual os classificadores são Máquinas de Vetores de Suporte. Contudo, espera-se em um trabalho futuro criar variações do algoritmo *InteliGER* híbridas, combinando Máquinas de Vetores de Suporte e Árvores de Classificação e Regressão. Defende-se esta ideia pelo desempenho dos classificadores apresentados nos Apêndices C e D: segundo coeficientes de Dice, MASI e Acurácia apresentados no Apêndice C, nos corpúscos GRE3D3 e GRE3D7, por exemplo, o melhor desempenho foi obtido pela variação *SVM_Profile_VAR+*, cujos classificadores são Máquinas de Vetores de Suporte. Entretanto, segundo a medida de *F-Score* apresentada no Apêndice D, pode-se ver que para os atributos *TG_Location*, *LM_Location* e, no corpúscos GRE3D3, também para *TG_Colour*, *LM_Colour* e *LM_Size*, as Árvores de Classificação e Regressão da variação *CART_Profile_VAR+* obtiveram melhores resultados que as Máquinas de Vetores de Suporte da variação *SVM_Profile_VAR+*.

Por fim, refutou-se a hipótese H_3 e confirmou-se a hipótese H_4 , pois as variações do algoritmo *InteliGER* treinadas de acordo com a abordagem *Profile* apresentam um coeficiente de Dice médio superior aos das variações treinadas com as abordagens *All* e *Speaker*. Além disso, as variações treinadas com a abordagem *All* apresentam coeficiente de Dice médio superior ao das variações treinadas com a abordagem *Speaker*.

Na abordagem *All*, classificadores são treinados com todas as descrições de treinamento, enquanto que nas abordagens *Speaker* e *Profile* é feita a divisão destas descrições em grupos distintos de acordo com algum fator de semelhança para posteriormente treinar classificadores diferentes para cada um destes grupos. Na abordagem *Speaker*, a divisão das descrições é feita por participante, enquanto que na abordagem *Profile* a mesma é feita por perfil de participante, como já explicado na Seção 5.3.2. Acredita-se que esta metodologia de divisão das descrições em grupos distintos e treinamento de classificadores para cada um destes grupos seja uma abordagem melhor para desempenhar a tarefa de

GER quando feito o uso de técnicas de aprendizado de máquina do que a abordagem *All*. Entre as diferentes metodologias de divisão das expressões de referência, assume-se que, quanto mais específica, melhor. Logo, esperava-se que as variações treinadas com a abordagem *Speaker* superassem as variações treinadas com a abordagem *Profile*, assim como as variações desta última abordagem superassem as variações da abordagem *All*. De fato, as variações da abordagem *Profile* obtiveram melhores resultados que as variações da abordagem *All*. Porém, as variações da abordagem *Speaker* não superaram as variações das abordagens *All* e *Profile*.

Explica-se o menor desempenho das variações treinadas pela abordagem *Speaker*, comparado com o desempenho das variações treinadas com as outras duas abordagens, pelo baixo número de expressões de referência por participante nos corpúis utilizados. Dentre os sete corpúis utilizados, o que possui o maior número de expressões de referência por participante é o corpúis GRE3D7 com 16 descrições, número muito limitado para treinar classificadores para desempenhar a tarefa de GER.

6.6.2 Comparação com trabalho prévio

Sobre os sete corpúis utilizados nos experimentos desta pesquisa, seis desses são corpúis já conhecidos da literatura, e um, conhecido como corpúis *Zoom-Pt*, é contribuição da presente pesquisa. Nos corpúis GRE3D3, GRE3D7, *Stars* e TUNA, variações do algoritmo *InteliGER* apresentaram os melhores resultados já obtidos e disponibilizados na literatura, como analisado no Apêndice C e sumariado abaixo:

- No corpúis GRE3D3, o maior resultado obtido na presente pesquisa foi da variação *SVM_Profile_VAR+*, que obteve um coeficiente de Dice de 0,93 e Acurácia de 0,75. Este resultado é superior às Árvores de Decisão treinadas de acordo com a metodologia de validação cruzada, apresentadas em Viethen e Dale (2010), as quais apresentaram Acurácia de 0,58; e superior ao algoritmo *Longest First*, apresentado em (VIETHEN; MITCHELL; KRAHMER, 2013), o qual apresentou coeficiente de Dice de 0,85 e Acurácia de 0,6.
- No corpúis GRE3D7, o maior resultado obtido na presente pesquisa foi da variação *SVM_Profile_VAR+*, que obteve um coeficiente de Dice de 0,94 e Acurácia de 0,77. Este resultado é superior às Árvores de Decisão treinadas de acordo com a metodologia de validação cruzada, apresentadas em Viethen e Dale (2010), as quais apresentaram Acurácia de 0,67 neste corpúis.

- No corpus *Stars*, o maior resultado obtido na presente pesquisa foi da variação *CART_Profile_VAR+*, que obteve um coeficiente de Dice de 0,88 e Acurácia de 0,58. Este resultado é maior do que o apresentado em Teixeira et al. (2014), o qual registrou coeficiente de Dice de 0,61 e Acurácia de 0,11.
- No domínio TUNA Furniture do corpus TUNA, o maior resultado obtido na presente pesquisa foi da variação *CART_Speaker_VAR+*, que obteve um coeficiente de Dice de 0,89 e Acurácia de 0,58. Este resultado é maior do que o apresentado pelos algoritmos vencedores dos desafios Belz e Gatt (2007) e TUNA REG 2008 (GATT; BELZ; KOW, 2008). Em Belz e Gatt (2007), o algoritmo vencedor registrou coeficiente de Dice de 0,8. Já no desafio TUNA 2008, o algoritmo vencedor registrou coeficiente de Dice de 0,86 e Acurácia de 0,53. Os resultados apresentados na presente pesquisa são também maiores do que as Árvores de Decisão apresentadas em Pereira et al. (2012), as quais registraram coeficiente de Dice de 0,78.
- No domínio TUNA People do corpus TUNA, o maior resultado obtido na presente pesquisa foi da variação *CART_Speaker_VAR+*, que obteve um coeficiente de Dice de 0,84 e Acurácia de 0,57. Este resultado é maior do que o apresentado pelos algoritmos vencedores dos desafios Belz e Gatt (2007) e TUNA REG 2008 (GATT; BELZ; KOW, 2008). Em Belz e Gatt (2007), o algoritmo vencedor registrou coeficiente de Dice de 0,74. Já no desafio TUNA 2008, o algoritmo vencedor registrou coeficiente de Dice de 0,73 e Acurácia de 0,56.

Capítulo 7

Conclusão

Este trabalho apresentou um estudo em nível de mestrado sobre a variação humana na tarefa de GER. Resultados comprovam a hipótese inicial de que algoritmos de GER que levam em conta a variação humana podem gerar expressões de referência mais próximas a descrições de seres humanos do que algoritmos que não levam esta questão em conta. Além disso, confirmou-se que algoritmos de GER baseados em técnicas de aprendizado de máquina mostram-se superiores a algoritmos de GER consagrados e amplamente utilizados na literatura, como o algoritmo Incremental.

Com relação às contribuições da presente pesquisa, as principais são:

1. A proposta do algoritmo Incremental Estendido (AIE). Este algoritmo de GER é uma extensão do algoritmo Incremental (DALE; REITER, 1995) capaz de gerar descrições relacionais, e foi implementado em duas variações.
2. A proposta do algoritmo *InteliGER*. Esse é um algoritmo de GER que faz uso de técnicas de aprendizado de máquina e foi implementado em 12 variações.
3. A validação das hipóteses do trabalho em sete corpuses de GER.
4. A criação do corpus de GER *Zoom-Pt*.

Com relação à divulgação da presente pesquisa, dois artigos foram publicados com partes do presente estudo (FERREIRA; PARABONI, 2014a, 2014b). Em (FERREIRA; PARABONI, 2014a), apresenta-se uma comparação de desempenho entre algoritmos de GER consagrados e algoritmos de GER baseados em técnicas de aprendizado de máquina. Assim como feito nesta pesquisa, esta comparação foi feita entre variações do algoritmo AIE e variações do algoritmo *InteliGER*. Já em (FERREIRA; PARABONI, 2014b), apresenta-se um estudo preliminar sobre a variação humana na tarefa de GER, comparando versões de modelos computacionais de GER que levam a variação humana em conta frente a versões que não levam esta questão em conta.

Como trabalhos futuros, espera-se finalizar o corpus *Zoom*. Além disso, planeja-se aprimorar o algoritmo *InteliGER*, criando um modelo padrão deste algoritmo para

contextos de domínio mais aberto, e possivelmente menos restrito a cada cópula como aqui apresentado. Espera-se também explorar a variação humana em partes de discurso maiores do que uma sentença, abordando o problema do tipo de expressão de referência a ser gerado (nominal, descritiva ou pronominal).

Referências

- ANDERSON, A. H. et al. The HCRC map task corpus. *Language and speech*, SAGE Publications, v. 34, n. 4, p. 351–366, 1991.
- ARECES, C.; KOLLER, A.; STRIEGNITZ, K. Referring expressions as formulas of description logic. In: *Proceedings of the Fifth International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008. (INLG '08), p. 42–49.
- ARTS, A. et al. Overspecification facilitates object identification. *Journal of Pragmatics*, v. 43, p. 361–374, 2011.
- BELZ, A. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, Cambridge University Press, New York, NY, USA, v. 14, n. 4, p. 431–455, out. 2008. ISSN 1351-3249.
- BELZ, A.; GATT, A. The attribute selection for GRE challenge: Overview and evaluation results. In: *Proceedings of UCNLG+ MT: Language Generation and Machine Translation*. Copenhagen: MT Summit XI, 2007. p. 75–83.
- BINSTED, K.; CAWSEY, A.; JONES, R. Generating personalised patient information using the medical record. In: BARAHONA, P.; STEFANELLI, M.; WYATT, J. (Ed.). *Artificial Intelligence in Medicine*. Pavia, Italy: Springer Berlin Heidelberg, 1995, (Lecture Notes in Computer Science, v. 934). p. 29–41. ISBN 978-3-540-60025-1.
- BOHNET, B. IF-FBN, IS-FBS, IS-IAC: The adaptation of two classic algorithms for the generation of referring expressions in order to produce expressions like humans do. In: *Language Generation and Machine Translation (UCNLG+MT) workshop*. Copenhagen: MT Summit XI, 2007. p. 84–86.
- BOHNET, B. The fingerprint of human referring expressions and their surface realization with graph transducers. In: *Proceedings of the Fifth International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008. (INLG '08), p. 207–210.
- BOHNET, B.; DALE, R. Viewing referring expression generation as search. In: *Proceedings of the 19th international joint conference on Artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. (IJCAI'05), p. 1004–1009.
- BREIMAN, L. et al. *Classification and Regression Trees*. Belmont, California, U.S.A.: Wadsworth Publishing Company, 1984. (Statistics/Probability Series).
- COHEN, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, v. 20, n. 1, p. 37–46, 1960.

- CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, Kluwer Academic Publishers, v. 20, n. 3, p. 273–297, 1995. ISSN 0885-6125.
- CROITORU, M.; DEEMTER, K. V. A conceptual graph approach to the generation of referring expressions. In: *Proceedings of the 20th international joint conference on Artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007. (IJCAI'07), p. 2456–2461.
- DALE, R. Cooking up referring expressions. In: *Proceedings of the 27th annual meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1989. (ACL '89), p. 68–75.
- DALE, R.; HADDOCK, N. Generating referring expressions involving relations. In: *Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1991. (EACL '91), p. 161–166.
- DALE, R.; REITER, E. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, Elsevier, v. 19, n. 2, p. 233–263, 1995.
- DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, JSTOR, v. 26, n. 3, p. 297–302, 1945.
- EUGENIO, B. D. et al. The agreement process: An empirical investigation of human–human computer-mediated collaborative dialogs. *International Journal of Human-Computer Studies*, Elsevier, v. 53, n. 6, p. 1017–1076, 2000.
- FABBRIZIO, G. D.; STENT, A. J.; BANGALORE, S. Trainable speaker-based referring expression generation. In: *Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008. (CoNLL '08), p. 151–158. ISBN 978-1-905593-48-4.
- FERREIRA, T. C.; PARABONI, I. Classification-based referring expression generation. In: GELBUKH, A. (Ed.). *Computational Linguistics and Intelligent Text Processing*. Kathmandu, Nepal: Springer Berlin Heidelberg, 2014, (Lecture Notes in Computer Science, v. 8403). p. 481–491. ISBN 978-3-642-54905-2.
- FERREIRA, T. C.; PARABONI, I. Referring Expression Generation: Taking Speakers' Preferences into Account. *Lecture Notes in Artificial Intelligence*, Springer International Publishing Switzerland, v. 8655, p. 539–546, 2014.
- GARDENT, C. Generating minimal definite descriptions. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ACL '02), p. 96–103.
- GARDENT, C.; STRIEGNITZ, K. Generating bridging definite descriptions. In: BUNT, H.; MUSKENS, R. (Ed.). *Computing Meaning*. Netherlands: Springer, 2007, (Studies in Linguistics and Philosophy, v. 83). p. 369–396. ISBN 978-1-4020-5957-5.

- GATT, A.; BELZ, A.; KOW, E. The TUNA challenge 2008: overview and evaluation results. In: *Proceedings of the Fifth International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008. (INLG '08), p. 198–206.
- GATT, A.; BELZ, A.; KOW, E. The TUNA-REG challenge 2009: overview and evaluation results. In: *Proceedings of the 12th European Workshop on Natural Language Generation*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. (ENLG '09), p. 174–182.
- GINI, C. Variabilit e mutabilita. *Memorie di metodologia statistica*, Libreria Eredi Virgilio Veschi, Rome, 1912.
- GRICE, H. P. Logic and conversation. In: COLE, P.; MORGAN, J. L. (Ed.). *Syntax and Semantics: Vol. 3: Speech Acts*. San Diego, CA: Academic Press, 1975. p. 41–58.
- GUPTA, S.; STENT, A. J. Automatic evaluation of referring expression generation using corpora. In: *Proceedings of the 1st Workshop on Using Corpora in Natural Language Generation (UCNLG)*. Birmingham: Corpus Linguistics 2005, 2005. p. 1–6.
- IACOVELLI, D.; GALINDO, M. R.; PARABONI, I. Lausanne: a Framework for Collaborative online NLP Experiments. In: *11th International Conference on Computational Processing of Portuguese (PROPOR-2014)*. São Carlos: Springer, 2014. p. 280–285.
- JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 31, n. 8, p. 651–666, jun. 2010. ISSN 0167-8655.
- KELLEHER, J. D.; KRUIJFF, G.-J. M. Incremental generation of spatial referring expressions in situated dialog. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. (ACL-44), p. 1041–1048.
- KITTREDGE, R. et al. Sublanguage engineering in the fog system. In: *Proceedings of the Fourth Conference on Applied Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994. (ANLC '94), p. 215–216.
- KNERR, S.; PERSONNAZ, L.; DREYFUS, G. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: *Neurocomputing*. Paris, France: Springer Berlin Heidelberg, 1990, (NATO ASI Series, v. 68). p. 41–50.
- KNUTH, D. E. *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998. ISBN 0-201-89685-0.
- KRAHMER, E.; DEEMTER, K. van. Computational generation of referring expressions: A survey. *Computational Linguistics*, MIT Press, v. 38, n. 1, p. 173–218, 2012.
- KRAHMER, E.; ERK, S. van; VERLEG, A. Graph-based generation of referring expressions. *Computational Linguistics*, MIT Press, v. 29, n. 1, p. 53–72, 2003.

- KRAHMER, E.; THEUNE, M. Efficient context-sensitive generation of referring expressions. In: *Information sharing: Reference and presupposition in language generation and interpretation*. California: CSLI Publications, 2002. v. 143, p. 223–263.
- LUCENA, D. J. de; PEREIRA, D. B.; PARABONI, I. From semantic properties to surface text: The generation of domain object descriptions. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, Asociacion Espanhola para la Inteligencia Artificial, v. 14, n. 45, p. 48–58, 2010.
- NOVAIS, E. M. de; PARABONI, I. Portuguese text generation using factored language models. *Journal of the Brazilian Computer Society*, Springer-Verlag, v. 19, n. 2, p. 135–146, 2013. ISSN 0104-6500.
- PARABONI, I.; DEEMTER, K. van. Reference and the facilitation of search in spatial domains. *Language and Cognitive Processes*, DOI 10.1080/01690965.2013.805796, 2013.
- PARABONI, I.; DEEMTER, K. van; MASTHOFF, J. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, MIT Press, v. 33, n. 2, p. 229–254, 2007.
- PARABONI, I. et al. Generating underspecified descriptions of landmark objects. *Lecture Notes in Artificial Intelligence*, Springer International Publishing Switzerland, v. 8655, p. 76–83, 2014.
- PASSONNEAU, R. Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*. Valletta, Malta: [s.n.], 2006. p. 831–836.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.
- PEREIRA, D. B.; PARABONI, I. A language modelling tool for statistical NLP. In: *5th Workshop on Information and Human Language Technology (TIL-2007)*. Anais do XXVII Congresso da SBC. Rio de Janeiro: Sociedade Brasileira de Computação (SBC), 2007. p. 1679–1688.
- PEREIRA, D. B.; PARABONI, I. Statistical surface realisation of portuguese referring expressions. *LNAI*, Springer-Verlag, v. 5221, p. 383–392, 2008.
- PEREIRA, H. et al. Corpus-based referring expressions generation. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), 2012. ISBN 978-2-9517408-7-7.
- PORTET, F. et al. Automatic generation of textual summaries from neonatal intensive care data. In: BELLAZZI, R.; ABU-HANNA, A.; HUNTER, J. (Ed.). *Artificial Intelligence in Medicine*. Amsterdam, The Netherlands: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4594). p. 227–236. ISBN 978-3-540-73598-4.
- QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

- REITER, E.; DALE, R. *Building natural language generation systems*. New York, NY, USA: Cambridge University Press, 2000. ISBN 0-521-62036-8.
- SIDDHARTHAN, A.; COPESTAKE, A. Generating referring expressions in open domains. In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. (ACL '04).
- SILVA, D. dos S.; PARABONI, I. Geração de expressões de referência usando relações espaciais. In: *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*. Fortaleza, CE, Brazil: Sociedade Brasileira de Computação (SBC), 2013. p. 88–97.
- TEIXEIRA, C. V. M. et al. Generating relational descriptions involving mutual disambiguation. *Lecture Notes in Computer Science*, Springer, v. 8403, p. 492–502, 2014.
- VIETHEN, H. A. E. *The Generation of Natural Descriptions: Corpus-Based Investigations of Referring Expressions in Visual Domains*. Tese (Doutorado) — Macquarie University, Sydney, Australia, 2011.
- VIETHEN, J.; DALE, R. Algorithms for generating referring expressions: do they do what people do? In: *Proceedings of the Fourth International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. (INLG '06), p. 63–70. ISBN 1-932432-72-8.
- VIETHEN, J.; DALE, R. Evaluation in natural language generation: Lessons from referring expression generation. *Traitement Automatique des Langues*, v. 48, n. 1, p. 141–160, 2007.
- VIETHEN, J.; DALE, R. The use of spatial relations in referring expression generation. In: *Proceedings of the Fifth International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008. (INLG '08), p. 59–67.
- VIETHEN, J.; DALE, R. Speaker-dependent variation in content selection for referring expression generation. In: *Proceedings of the Australasian Language Technology Association Workshop 2010*. Melbourne, Australia: [s.n.], 2010. p. 81–89.
- VIETHEN, J.; DALE, R. GRE3D7: A corpus of distinguishing descriptions for objects in visual scenes. In: *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*. Edinburgh, Scotland: Association for Computational Linguistics, 2011. p. 12–22.
- VIETHEN, J.; MITCHELL, M.; KRAHMER, E. Graphs and spatial relations in the generation of referring expressions. In: *Proceedings of the 14th European Workshop on Natural Language Generation*. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 72–81.

Apêndice A

Tabela 26 – Possíveis valores dos atributos para anotação das expressões de referência da cena do *cópus Zoom* representada na figura 18

Objeto	Atributo	Valores possíveis
objeto-alvo	id	rest3
objeto-alvo	tipo	restaurant, other
objeto-alvo	nome	other
objeto-alvo	outros	other
objeto-alvo	em	str6, str7, str26, neig1, other
objeto-alvo	à direita de	chur1, gov1, other
objeto-alvo	à esquerda de	chur1, gov1, other
objeto-alvo	entre/esquina	str6, str7, other
objeto-alvo	próximo a	str6, str7, str26, neig1, chur1, gov1, other
objeto-alvo	em frente a	chur1, gov1, other
objeto-alvo	atrás de	chur1, gov1, other
1º ponto de referência	id	str6, str7, str26, neig1, chur1, gov1, other
1º ponto de referência	tipo	street, neighborhood, church, building, other
1º ponto de referência	nome	largo de sao carlos, rua capelo, rua anchieta, chiado, governo civil, other
1º ponto de referência	outros	end, beginning, corner, other
2º ponto de referência	id	str6, str7, str26, neig1, chur1, gov1, other
2º ponto de referência	tipo	street, neighborhood, church, building, other
2º ponto de referência	nome	largo de sao carlos, rua capelo, rua anchieta, chiado, governo civil, other
2º ponto de referência	outros	end, beginning, corner, other
3º ponto de referência	id	str6, str7, str26, neig1, chur1, gov1, other
3º ponto de referência	tipo	street, neighborhood, church, building, other
3º ponto de referência	nome	largo de sao carlos, rua capelo, rua anchieta, chiado, governo civil, other
3º ponto de referência	outros	end, beginning, corner, other
4º ponto de referência	id	str6, str7, str26, neig1, chur1, gov1, other
4º ponto de referência	tipo	street, neighborhood, church, building, other
4º ponto de referência	nome	largo de sao carlos, rua capelo, rua anchieta, chiado, governo civil, other
4º ponto de referência	outros	end, beginning, corner, other

Apêndice B

Tabela 27 – *Exemplo de anotação de contexto do corpus GRE3D7*

id	type	col	size	loc	on-top-of	next-to	right-of	left-of	under
c1	cube	blue	large	left-hand		b1		b1	
c2	cube	blue	large	right-hand	c3	c3			
c3	cube	green	large	right-hand		c2			c2
c4	cube	blue	small	center		b3			b3
b1	ball	green	large	left-hand		c1	c1		
b2	ball	green	large	left-hand,front					
b3	ball	blue	small	center	c4	c4			

Apêndice C

Tabela 28 – Resultado do experimento para os corpos GRE3D3/7

Modelo	GRE3D3			GRE3D7		
	Dice	MA SI	Acurácia	Dice	MA SI	Acurácia
AIE-	0,6848	0,4581	0,2413	0,8838	0,7510	0,6141
AIE+	0,6843	0,4517	0,2365	0,8835	0,7471	0,6083
<i>SVM.All.VAR-</i>	0,7803	0,6071	0,4635	0,8848	0,7481	0,6060
<i>SVM.All.VAR+</i>	0,9220	0,8286	0,7381	0,9398	0,8556	0,7667
<i>SVM.Speaker.VAR-</i>	0,8185	0,6461	0,4921	0,9098	0,7927	0,6723
<i>SVM.Speaker.VAR+</i>	0,8839	0,7458	0,6095	0,9237	0,8206	0,7156
<i>SVM.Profile.VAR-</i>	0,8255	0,6585	0,5143	0,8881	0,7452	0,5926
<i>SVM.Profile.VAR+</i>	0,9278	0,8362	0,7460	0,9414	0,8583	0,7714
<i>CART.All.VAR-</i>	0,7730	0,5892	0,4254	0,8906	0,7638	0,6346
<i>CART.All.VAR+</i>	0,9000	0,7847	0,6746	0,9202	0,8086	0,6942
<i>CART.Speaker.VAR-</i>	0,7695	0,6032	0,4794	0,8584	0,7184	0,5904
<i>CART.Speaker.VAR+</i>	0,8814	0,7827	0,7175	0,8684	0,7545	0,6549
<i>CART.Profile.VAR-</i>	0,8193	0,6502	0,4968	0,8936	0,7617	0,6239
<i>CART.Profile.VAR+</i>	0,8908	0,7682	0,6524	0,9226	0,8152	0,7051

Tabela 29 – Resultado do experimento para os corpos Stars e Stars2

Modelo	Stars			Stars2		
	Dice	MA SI	Acurácia	Dice	MA SI	Acurácia
AIE-	0,6593	0,3273	0,1641	0,6381	0,3935	0,2558
AIE+	0,7615	0,4447	0,2370	0,6596	0,4121	0,2608
<i>SVM.All.VAR-</i>	0,7246	0,3866	0,1797	0,6618	0,4434	0,3032
<i>SVM.All.VAR+</i>	0,7273	0,4921	0,2943	0,7621	0,5334	0,3605
<i>SVM.Speaker.VAR-</i>	0,7496	0,5395	0,3724	0,6757	0,4618	0,3223
<i>SVM.Speaker.VAR+</i>	0,7494	0,5479	0,3932	0,7442	0,5344	0,3746
<i>SVM.Profile.VAR-</i>	0,7181	0,3820	0,1745	0,6759	0,4489	0,3032
<i>SVM.Profile.VAR+</i>	0,7309	0,5072	0,3229	0,7651	0,5383	0,3696
<i>CART.All.VAR-</i>	0,7420	0,4146	0,2292	0,6611	0,4408	0,2998
<i>CART.All.VAR+</i>	0,8470	0,6697	0,5339	0,7975	0,5821	0,4261
<i>CART.Speaker.VAR-</i>	0,4810	0,3202	0,2135	0,6421	0,4454	0,3248
<i>CART.Speaker.VAR+</i>	0,5983	0,4621	0,3802	0,7283	0,5232	0,3704
<i>CART.Profile.VAR-</i>	0,7291	0,4025	0,2161	0,6744	0,4461	0,3015
<i>CART.Profile.VAR+</i>	0,8777	0,7072	0,5807	0,7962	0,5803	0,4219

Tabela 30 – Resultado do experimento para os domínios do córpus TUNA

Modelo	TUNA Furniture			TUNA People		
	Dice	MA SI	Acurácia	Dice	MA SI	Acurácia
AIE-	0,7297	0,4484	0,2857	0,6096	0,2825	0,0861
AIE+	0,7522	0,5242	0,3286	0,7187	0,4545	0,2417
<i>SVM_All_VAR-</i>	0,7471	0,5174	0,3095	0,4631	0,2079	0,0222
<i>SVM_All_VAR+</i>	0,8443	0,6281	0,3976	0,7332	0,4984	0,2972
<i>SVM_Speaker_VAR-</i>	0,8353	0,6103	0,3762	0,6832	0,4136	0,2139
<i>SVM_Speaker_VAR+</i>	0,8477	0,6350	0,4119	0,7043	0,4408	0,2361
<i>SVM_Profile_VAR-</i>	0,7317	0,4799	0,2357	0,5654	0,3083	0,0944
<i>SVM_Profile_VAR+</i>	0,8579	0,6518	0,4357	0,7393	0,4972	0,2833
<i>CART_All_VAR-</i>	0,7230	0,4438	0,1952	0,4874	0,1961	0,0250
<i>CART_All_VAR+</i>	0,8324	0,6200	0,4119	0,6966	0,4433	0,2639
<i>CART_Speaker_VAR-</i>	0,7908	0,5342	0,2881	0,6407	0,3665	0,1722
<i>CART_Speaker_VAR+</i>	0,8925	0,7406	0,5762	0,8411	0,6997	0,5694
<i>CART_Profile_VAR-</i>	0,7205	0,4626	0,2119	0,5010	0,2337	0,0500
<i>CART_Profile_VAR+</i>	0,8252	0,6081	0,3952	0,7037	0,4472	0,2556

Tabela 31 – Resultado do experimento para o córpus Zoom-Pt

Modelo	Zoom-Pt		
	Dice	MA SI	Acurácia
AIE-	0,5258	0,2059	0,0402
AIE+	0,5691	0,2420	0,0743
<i>SVM_All_VAR-</i>	0,5084	0,2832	0,1523
<i>SVM_All_VAR+</i>	0,5462	0,3136	0,1790
<i>SVM_Speaker_VAR-</i>	0,4532	0,2362	0,1181
<i>SVM_Speaker_VAR+</i>	0,4576	0,2365	0,1218
<i>SVM_Profile_VAR-</i>	0,5203	0,2837	0,1510
<i>SVM_Profile_VAR+</i>	0,5478	0,3159	0,1827
<i>CART_All_VAR-</i>	0,5084	0,2794	0,1754
<i>CART_All_VAR+</i>	0,5243	0,2948	0,1730
<i>CART_Speaker_VAR-</i>	0,4201	0,2129	0,1157
<i>CART_Speaker_VAR+</i>	0,4658	0,2365	0,1255
<i>CART_Profile_VAR-</i>	0,5047	0,2770	0,1742
<i>CART_Profile_VAR+</i>	0,5228	0,2929	0,1705

Apêndice D

Tabela 32 – Resultados das Máquinas de Vetores de Suporte do córpus GRE3D3

GRE3D3									
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Speaker	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Profile	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
TG_Colour	SVM - All	0,78	1,00	0,88	0,16	0,94	0,97	0,95	0,94
	SVM - Speaker	0,94	0,96	0,95	0,89	0,95	0,97	0,96	0,92
	SVM - Profile	0,78	1,00	0,88	0,53	0,94	0,97	0,95	0,94
TG_Size	SVM - All	0,91	0,86	0,88	0,73	0,95	0,93	0,94	0,98
	SVM - Speaker	0,87	0,58	0,69	0,76	0,91	0,64	0,75	0,87
	SVM - Profile	0,91	0,86	0,88	0,81	0,94	0,94	0,94	0,98
TG_Location	SVM - All	0,00	0,00	0,00	0,15	0,60	0,20	0,30	0,66
	SVM - Speaker	0,00	0,00	0,00	0,60	0,50	0,27	0,35	0,50
	SVM - Profile	0,00	0,00	0,00	0,26	0,80	0,27	0,40	0,76
LM_Type	SVM - All	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Speaker	1,00	0,92	0,96	1,00	1,00	0,89	0,94	1,00
	SVM - Profile	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
LM_Colour	SVM - All	0,72	0,95	0,82	0,26	0,92	0,91	0,92	0,91
	SVM - Speaker	0,85	0,85	0,85	0,73	0,89	0,85	0,87	0,74
	SVM - Profile	0,70	0,94	0,80	0,51	0,92	0,92	0,92	0,92
LM_Size	SVM - All	1,00	0,58	0,73	0,58	0,98	0,64	0,77	0,89
	SVM - Speaker	0,73	0,61	0,66	0,60	0,79	0,74	0,77	0,66
	SVM - Profile	1,00	0,58	0,73	0,83	0,92	0,73	0,81	0,89
LM_Location	SVM - All	0,74	0,76	0,75	0,70	0,90	0,41	0,57	0,88
	SVM - Speaker	0,48	0,26	0,34	0,54	0,51	0,39	0,44	0,49
	SVM - Profile	0,74	0,76	0,75	0,86	0,86	0,52	0,65	0,81
Relation	SVM - All	0,35	0,10	0,15	0,39	0,96	0,79	0,87	0,95
	SVM - Speaker	0,83	0,63	0,71	0,82	0,95	0,84	0,89	0,75
	SVM - Profile	0,80	0,54	0,65	0,49	0,96	0,85	0,90	0,97

Tabela 33 – Resultados das Árvores de Classificação e Regressão do corpus GRE3D3

GRE3D3							
		VAR-			VAR+		
Classificador		P	R	F_1	P	R	F_1
TG_Type	<i>CART - All</i>	1,00	1,00	1,00	1,00	1,00	1,00
	<i>CART - Speaker</i>	1,00	0,90	0,95	1,00	0,89	0,94
	<i>CART - Profile</i>	1,00	1,00	1,00	1,00	1,00	1,00
TG_Colour	<i>CART - All</i>	0,79	0,91	0,85	0,96	0,96	0,96
	<i>CART - Speaker</i>	0,92	0,86	0,89	0,99	0,93	0,96
	<i>CART - Profile</i>	0,84	0,84	0,84	0,96	0,96	0,96
TG_Size	<i>CART - All</i>	0,91	0,86	0,88	0,90	0,87	0,88
	<i>CART - Speaker</i>	0,85	0,85	0,85	1,00	0,99	0,99
	<i>CART - Profile</i>	0,91	0,86	0,88	0,89	0,88	0,88
TG_Location	<i>CART - All</i>	0,00	0,00	0,00	0,56	0,33	0,42
	<i>CART - Speaker</i>	0,07	0,33	0,11	0,15	0,67	0,25
	<i>CART - Profile</i>	0,00	0,00	0,00	0,54	0,47	0,50
LM_Type	<i>CART - All</i>	1,00	1,00	1,00	1,00	1,00	1,00
	<i>CART - Speaker</i>	1,00	0,76	0,86	1,00	0,73	0,85
	<i>CART - Profile</i>	1,00	1,00	1,00	1,00	0,97	0,99
LM_Colour	<i>CART - All</i>	0,70	0,83	0,76	0,90	0,94	0,92
	<i>CART - Speaker</i>	0,85	0,78	0,82	0,90	0,80	0,85
	<i>CART - Profile</i>	0,75	0,89	0,82	0,94	0,91	0,93
LM_Size	<i>CART - All</i>	1,00	0,58	0,73	0,84	0,77	0,80
	<i>CART - Speaker</i>	0,58	0,71	0,64	0,77	0,89	0,83
	<i>CART - Profile</i>	0,83	0,76	0,79	0,83	0,82	0,82
LM_Location	<i>CART - All</i>	0,74	0,76	0,75	0,73	0,87	0,79
	<i>CART - Speaker</i>	0,55	0,70	0,62	0,62	0,87	0,72
	<i>CART - Profile</i>	0,74	0,76	0,75	0,73	0,89	0,80
Relation	<i>CART - All</i>	0,46	0,32	0,38	0,83	0,81	0,82
	<i>CART - Speaker</i>	0,70	0,72	0,71	0,88	0,93	0,90
	<i>CART - Profile</i>	0,79	0,60	0,68	0,79	0,77	0,78

Tabela 34 – Resultados das Máquinas de Vetores de Suporte do corpus GRE3D7

GRE3D7									
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Speaker	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Profile	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
TG_Colour	SVM - All	0,99	1,00	0,99	0,35	0,99	1,00	0,99	0,85
	SVM - Speaker	0,99	1,00	0,99	0,82	0,99	0,99	0,99	0,84
	SVM - Profile	0,99	1,00	0,99	0,56	0,99	1,00	0,99	0,87
TG_Size	SVM - All	0,75	0,73	0,74	0,67	0,87	0,88	0,88	0,92
	SVM - Speaker	0,80	0,82	0,81	0,84	0,84	0,84	0,84	0,88
	SVM - Profile	0,74	0,67	0,71	0,71	0,87	0,88	0,88	0,92
TG_Location	SVM - All	0,00	0,00	0,00	0,43	0,00	0,00	0,00	0,76
	SVM - Speaker	0,18	0,05	0,08	0,80	0,32	0,15	0,20	0,71
	SVM - Profile	0,00	0,00	0,00	0,55	0,32	0,07	0,12	0,83
LM_Type	SVM - All	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Speaker	1,00	0,77	0,87	1,00	1,00	0,75	0,86	1,00
	SVM - Profile	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
LM_Colour	SVM - All	0,87	1,00	0,93	0,34	0,92	0,95	0,93	0,80
	SVM - Speaker	0,91	0,75	0,82	0,50	0,92	0,74	0,82	0,48
	SVM - Profile	0,87	1,00	0,93	0,43	0,92	0,96	0,94	0,76
LM_Size	SVM - All	0,54	0,86	0,66	0,52	0,83	0,85	0,84	0,88
	SVM - Speaker	0,77	0,75	0,76	0,53	0,79	0,74	0,77	0,55
	SVM - Profile	0,73	0,57	0,64	0,61	0,83	0,87	0,85	0,89
LM_Location	SVM - All	0,00	0,00	0,00	0,32	0,00	0,00	0,00	0,80
	SVM - Speaker	0,04	0,20	0,07	0,33	0,02	0,10	0,03	0,37
	SVM - Profile	0,00	0,00	0,00	0,48	0,00	0,00	0,00	0,87
Relation	SVM - All	0,00	0,00	0,00	0,50	0,84	0,68	0,75	0,91
	SVM - Speaker	0,73	0,41	0,53	0,85	0,87	0,56	0,68	0,87
	SVM - Profile	0,99	0,18	0,31	0,72	0,86	0,70	0,77	0,95

Tabela 35 – Resultados das Árvores de Classificação e Regressão do corpus GRE3D7

GRE3D7							
		VAR-			VAR+		
Classificador		P	R	F_1	P	R	F_1
TG_Type	<i>CART - All</i>	1,00	1,00	1,00	1,00	1,00	1,00
	<i>CART - Speaker</i>	1,00	0,94	0,97	1,00	0,90	0,95
	<i>CART - Profile</i>	1,00	1,00	1,00	1,00	1,00	1,00
TG_Colour	<i>CART - All</i>	0,99	1,00	0,99	0,99	0,99	0,99
	<i>CART - Speaker</i>	0,99	0,93	0,96	0,99	0,90	0,95
	<i>CART - Profile</i>	0,99	1,00	0,99	0,99	0,99	0,99
TG_Size	<i>CART - All</i>	0,78	0,76	0,77	0,84	0,81	0,82
	<i>CART - Speaker</i>	0,81	0,77	0,79	0,83	0,84	0,84
	<i>CART - Profile</i>	0,79	0,68	0,73	0,85	0,83	0,84
TG_Location	<i>CART - All</i>	0,00	0,00	0,00	0,44	0,40	0,42
	<i>CART - Speaker</i>	0,07	0,32	0,12	0,10	0,60	0,18
	<i>CART - Profile</i>	0,00	0,00	0,00	0,38	0,40	0,39
LM_Type	<i>CART - All</i>	1,00	1,00	1,00	1,00	1,00	1,00
	<i>CART - Speaker</i>	1,00	0,63	0,77	1,00	0,63	0,77
	<i>CART - Profile</i>	1,00	1,00	1,00	1,00	1,00	1,00
LM_Colour	<i>CART - All</i>	0,87	1,00	0,93	0,92	0,89	0,91
	<i>CART - Speaker</i>	0,91	0,62	0,74	0,93	0,64	0,76
	<i>CART - Profile</i>	0,87	1,00	0,93	0,92	0,89	0,91
LM_Size	<i>CART - All</i>	0,60	0,65	0,62	0,80	0,81	0,81
	<i>CART - Speaker</i>	0,69	0,66	0,68	0,77	0,77	0,77
	<i>CART - Profile</i>	0,67	0,59	0,63	0,81	0,81	0,81
LM_Location	<i>CART - All</i>	0,00	0,00	0,00	0,40	0,40	0,40
	<i>CART - Speaker</i>	0,03	0,40	0,05	0,04	0,50	0,07
	<i>CART - Profile</i>	0,00	0,00	0,00	0,30	0,30	0,30
Relation	<i>CART - All</i>	0,00	0,00	0,00	0,68	0,70	0,69
	<i>CART - Speaker</i>	0,41	0,59	0,49	0,59	0,74	0,65
	<i>CART - Profile</i>	0,99	0,18	0,31	0,68	0,72	0,70

Tabela 36 – Resultados das Máquinas de Vetores de Suporte do *córpus Stars*

		Stars							
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Speaker	1,00	1,00	1,00	1,00	1,00	0,98	0,99	1,00
	SVM - Profile	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
TG_Colour	SVM - All	0,00	0,00	0,00	0,27	0,75	0,64	0,69	0,87
	SVM - Speaker	0,75	0,73	0,74	0,81	0,63	0,73	0,68	0,82
	SVM - Profile	0,00	0,00	0,00	0,27	0,70	0,64	0,67	0,87
TG_VPos	SVM - All	0,00	0,00	0,00	0,23	0,57	0,40	0,47	0,76
	SVM - Speaker	0,62	0,40	0,48	0,72	0,45	0,45	0,45	0,57
	SVM - Profile	0,00	0,00	0,00	0,26	0,57	0,40	0,47	0,77
TG_HPos	SVM - All	0,00	0,00	0,00	0,23	0,83	0,33	0,47	0,82
	SVM - Speaker	0,70	0,54	0,61	0,73	0,68	0,63	0,63	0,68
	SVM - Profile	0,00	0,00	0,00	0,39	0,83	0,33	0,47	0,80
LM_Type	SVM - All	0,98	1,00	0,99	0,31	1,00	1,00	1,00	0,76
	SVM - Speaker	1,00	0,99	0,99	0,94	1,00	0,97	0,98	0,87
	SVM - Profile	0,98	1,00	0,99	0,41	1,00	1,00	1,00	0,74
LM_Colour	SVM - All	0,62	0,55	0,58	0,43	0,80	0,78	0,79	0,82
	SVM - Speaker	0,83	0,85	0,84	0,79	0,84	0,86	0,85	0,79
	SVM - Profile	0,62	0,56	0,59	0,43	0,80	0,80	0,80	0,83
LM_VPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM_HPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM2_Type	SVM - All	0,97	1,00	0,99	0,14	0,97	0,99	0,98	0,07
	SVM - Speaker	0,98	0,75	0,85	0,30	0,98	0,77	0,86	0,28
	SVM - Profile	0,97	1,00	0,99	0,15	0,97	0,99	0,98	0,27
LM2_Colour	SVM - All	0,40	0,13	0,20	0,30	0,78	0,60	0,68	0,82
	SVM - Speaker	0,67	0,47	0,55	0,50	0,77	0,57	0,65	0,45
	SVM - Profile	0,40	0,13	0,20	0,30	0,78	0,60	0,68	0,81
LM2_VPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM2_HPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Relation	SVM - All	0,97	1,00	0,98	0,22	0,97	0,99	0,98	0,65
	SVM - Speaker	0,98	0,99	0,99	0,55	0,98	0,99	0,99	0,55
	SVM - Profile	0,97	0,98	0,97	0,74	0,98	0,98	0,98	0,88
LM_Relation	SVM - All	0,00	0,00	0,00	0,34	0,91	0,61	0,73	0,89
	SVM - Speaker	0,85	0,77	0,81	0,78	0,81	0,76	0,78	0,77
	SVM - Profile	0,00	0,00	0,00	0,52	0,91	0,61	0,73	0,89

Tabela 37 – Resultados das Árvores de Classificação e Regressão do córpus Stars

		Stars					
		VAR-			VAR+		
Classificador		P	R	F ₁	P	R	F ₁
TG_Type	CART - All	1,00	1,00	1,00	1,00	0,99	0,99
	CART - Speaker	1,00	0,67	0,80	1,00	0,57	0,73
	CART - Profile	1,00	1,00	1,00	1,00	0,98	0,99
TG_Colour	CART - All	0,00	0,00	0,00	0,80	0,73	0,76
	CART - Speaker	0,13	0,52	0,20	0,15	0,73	0,25
	CART - Profile	0,00	0,00	0,00	0,78	0,76	0,77
TG_VPos	CART - All	0,00	0,00	0,00	0,81	0,65	0,72
	CART - Speaker	0,09	0,60	0,16	0,11	0,85	0,19
	CART - Profile	0,00	0,00	0,00	0,93	0,65	0,76
TG_HPos	CART - All	0,00	0,00	0,00	0,90	0,61	0,73
	CART - Speaker	0,24	0,56	0,34	0,29	0,84	0,43
	CART - Profile	0,00	0,00	0,00	0,80	0,61	0,69
LM_Type	CART - All	0,98	1,00	0,99	1,00	1,00	1,00
	CART - Speaker	0,99	0,65	0,79	0,98	0,58	0,73
	CART - Profile	0,98	1,00	0,99	1,00	0,99	0,99
LM_Colour	CART - All	0,66	0,78	0,72	0,85	0,93	0,89
	CART - Speaker	0,72	0,75	0,73	0,78	0,76	0,77
	CART - Profile	0,66	0,78	0,72	0,87	0,94	0,90
LM_VPos	CART - All	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Speaker	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Profile	0,00	0,00	0,00	0,00	0,00	0,00
LM_HPos	CART - All	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Speaker	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Profile	0,00	0,00	0,00	0,00	0,00	0,00
LM2_Type	CART - All	0,97	1,00	0,99	0,97	0,97	0,97
	CART - Speaker	0,97	0,48	0,64	0,97	0,49	0,65
	CART - Profile	0,97	1,00	0,99	0,97	0,97	0,97
LM2_Colour	CART - All	0,46	0,20	0,28	0,72	0,77	0,74
	CART - Speaker	0,45	0,43	0,44	0,45	0,33	0,38
	CART - Profile	0,46	0,20	0,28	0,72	0,77	0,74
LM2_VPos	CART - All	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Speaker	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Profile	0,00	0,00	0,00	0,00	0,00	0,00
LM2_HPos	CART - All	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Speaker	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Profile	0,00	0,00	0,00	0,00	0,00	0,00
Relation	CART - All	0,97	1,00	0,98	0,98	0,99	0,99
	CART - Speaker	0,98	0,88	0,92	0,89	0,99	0,94
	CART - Profile	0,97	0,97	0,97	0,98	0,99	0,99
LM_Relation	CART - All	0,00	0,00	0,00	0,79	0,77	0,78
	CART - Speaker	0,32	0,70	0,44	0,33	0,85	0,47
	CART - Profile	0,00	0,00	0,00	0,75	0,77	0,76

Tabela 38 – Resultados das Máquinas de Vetores de Suporte do *córpus Stars2*

Stars2									
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	0,99	1,00	0,99	0,27	1,00	1,00	1,00	0,88
	SVM - Speaker	0,99	0,99	0,99	0,50	0,99	0,97	0,98	0,63
	SVM - Profile	0,99	1,00	0,99	0,30	1,00	1,00	1,00	0,84
TG_Colour	SVM - All	0,70	0,72	0,71	0,80	0,79	0,82	0,81	0,94
	SVM - Speaker	0,83	0,80	0,81	0,47	0,80	0,78	0,79	0,75
	SVM - Profile	0,70	0,72	0,71	0,86	0,72	0,84	0,77	0,92
TG_Size	SVM - All	0,13	0,01	0,02	0,79	0,27	0,04	0,07	0,96
	SVM - Speaker	0,64	0,29	0,39	0,48	0,63	0,43	0,51	0,66
	SVM - Profile	0,00	0,00	0,00	0,70	0,65	0,27	0,38	0,96
TG_VPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
TG_HPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM_Type	SVM - All	0,99	1,00	0,99	0,15	0,99	1,00	0,99	0,44
	SVM - Speaker	0,99	0,98	0,98	0,40	0,99	0,97	0,98	0,40
	SVM - Profile	0,99	1,00	0,99	0,20	0,99	1,00	0,99	0,26
LM_Colour	SVM - All	0,91	0,78	0,84	0,84	0,88	0,89	0,88	0,96
	SVM - Speaker	0,85	0,90	0,88	0,30	0,87	0,89	0,88	0,44
	SVM - Profile	0,91	0,78	0,84	0,87	0,91	0,88	0,89	0,94
LM_Size	SVM - All	0,66	0,92	0,78	0,92	0,83	0,86	0,85	0,96
	SVM - Speaker	0,00	0,00	0,00	0,25	0,50	0,09	0,15	0,29
	SVM - Profile	0,66	0,95	0,78	0,95	0,88	0,81	0,84	0,96
LM_VPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM_HPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM2_Type	SVM - All	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	SVM - Speaker	1,00	0,74	0,85	1,00	1,00	0,71	0,83	1,00
	SVM - Profile	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
LM2_Colour	SVM - All	0,00	0,00	0,00	0,36	0,50	0,27	0,35	0,91
	SVM - Speaker	0,17	0,64	0,27	0,63	0,18	0,64	0,28	0,64
	SVM - Profile	0,00	0,00	0,00	0,68	0,50	0,27	0,35	0,91
LM2_Size	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM2_VPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
LM2_HPos	SVM - All	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Speaker	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	SVM - Profile	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Relation	SVM - All	1,00	0,83	0,90	0,85	0,97	0,93	0,95	0,97
	SVM - Speaker	0,93	0,88	0,91	0,91	0,94	0,94	0,94	0,91
	SVM - Profile	0,99	0,92	0,95	0,84	0,97	0,93	0,95	0,98
LM_Relation	SVM - All	0,00	0,00	0,00	0,40	0,73	0,30	0,43	0,70
	SVM - Speaker	0,48	0,33	0,39	0,63	0,63	0,45	0,53	0,68
	SVM - Profile	0,52	0,24	0,33	0,54	0,68	0,43	0,53	0,78

Tabela 39 – Resultados das Árvores de Classificação e Regressão do *cópus Stars2*

		Stars2					
		VAR-			VAR+		
Classificador		P	R	F ₁	P	R	F ₁
TG_Type	<i>CART - All</i>	0,99	1,00	0,99	1,00	1,00	1,00
	<i>CART - Speaker</i>	0,99	0,94	0,97	0,99	0,94	0,97
	<i>CART - Profile</i>	0,99	1,00	0,99	1,00	0,99	1,00
TG_Colour	<i>CART - All</i>	0,70	0,72	0,71	0,79	0,81	0,80
	<i>CART - Speaker</i>	0,74	0,79	0,76	0,73	0,84	0,78
	<i>CART - Profile</i>	0,70	0,72	0,71	0,81	0,78	0,79
TG_Size	<i>CART - All</i>	0,48	0,42	0,45	0,75	0,78	0,76
	<i>CART - Speaker</i>	0,61	0,76	0,67	0,64	0,81	0,71
	<i>CART - Profile</i>	0,52	0,49	0,50	0,75	0,78	0,76
TG_VPos	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
TG_HPos	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
LM_Type	<i>CART - All</i>	0,99	1,00	0,99	0,99	0,99	0,99
	<i>CART - Speaker</i>	0,99	0,95	0,97	0,99	0,87	0,92
	<i>CART - Profile</i>	0,99	1,00	0,99	0,99	0,99	0,99
LM_Colour	<i>CART - All</i>	0,91	0,78	0,84	0,90	0,88	0,89
	<i>CART - Speaker</i>	0,88	0,89	0,88	0,79	0,87	0,83
	<i>CART - Profile</i>	0,91	0,78	0,84	0,89	0,89	0,89
LM_Size	<i>CART - All</i>	0,66	0,95	0,78	0,86	0,89	0,87
	<i>CART - Speaker</i>	0,81	0,84	0,83	0,66	0,77	0,71
	<i>CART - Profile</i>	0,66	0,95	0,78	0,87	0,90	0,89
LM_VPos	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
LM_HPos	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
LM2_Type	<i>CART - All</i>	1,00	1,00	1,00	1,00	1,00	1,00
	<i>CART - Speaker</i>	1,00	0,62	0,76	1,00	0,62	0,76
	<i>CART - Profile</i>	1,00	1,00	1,00	1,00	1,00	1,00
LM2_Colour	<i>CART - All</i>	0,00	0,00	0,00	0,50	0,55	0,52
	<i>CART - Speaker</i>	0,17	0,73	0,28	0,14	0,64	0,23
	<i>CART - Profile</i>	0,00	0,00	0,00	0,50	0,55	0,52
LM2_Size	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
LM2_VPos	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
LM2_HPos	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
Relation	<i>CART - All</i>	1,00	0,83	0,90	0,94	0,95	0,95
	<i>CART - Speaker</i>	0,96	0,95	0,95	0,95	0,95	0,95
	<i>CART - Profile</i>	0,99	0,92	0,95	0,94	0,95	0,94
LM_Relation	<i>CART - All</i>	0,14	0,22	0,17	0,35	0,71	0,47
	<i>CART - Speaker</i>	0,36	0,42	0,38	0,42	0,63	0,50
	<i>CART - Profile</i>	0,21	0,29	0,24	0,35	0,70	0,47

Tabela 40 – Resultados das Máquinas de Vetores de Suporte do domínio TUNA Furniture

TUNA Furniture									
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	0,92	1,00	0,96	0,37	0,95	0,99	0,97	0,71
	SVM - Speaker	0,95	0,98	0,97	0,72	0,96	0,98	0,97	0,73
	SVM - Profile	0,92	1,00	0,96	0,51	0,95	0,99	0,97	0,81
TG_Colour	SVM - All	0,85	1,00	0,92	0,36	0,89	0,97	0,93	0,77
	SVM - Speaker	0,91	0,96	0,93	0,77	0,91	0,96	0,94	0,71
	SVM - Profile	0,85	1,00	0,92	0,53	0,89	0,97	0,93	0,81
TG_Size	SVM - All	0,59	0,65	0,62	0,70	0,70	0,77	0,74	0,87
	SVM - Speaker	0,68	0,59	0,63	0,72	0,73	0,66	0,69	0,73
	SVM - Profile	0,68	0,18	0,29	0,58	0,76	0,75	0,80	0,90
TG_Orientation	SVM - All	0,52	0,27	0,35	0,58	0,71	0,57	0,63	0,86
	SVM - Speaker	0,63	0,52	0,57	0,71	0,70	0,63	0,66	0,75
	SVM - Profile	0,69	0,21	0,32	0,55	0,72	0,57	0,63	0,85
TG_XDimension	SVM - All	0,00	0,00	0,00	0,38	0,68	0,65	0,66	0,85
	SVM - Speaker	0,72	0,72	0,72	0,79	0,72	0,72	0,72	0,79
	SVM - Profile	0,63	0,05	0,09	0,53	0,74	0,72	0,73	0,86
TG_YDimension	SVM - All	0,00	0,00	0,00	0,41	0,78	0,69	0,73	0,86
	SVM - Speaker	0,78	0,77	0,78	0,83	0,77	0,77	0,77	0,82
	SVM - Profile	0,25	0,01	0,03	0,50	0,76	0,71	0,74	0,87

Tabela 41 – Resultados das Árvores de Classificação e Regressão do domínio TUNA Furniture

TUNA Furniture							
		VAR-			VAR+		
Classificador		P	R	F_1	P	R	F_1
TG_Type	CART - All	0,92	1,00	0,96	0,96	0,93	0,95
	CART - Speaker	0,94	0,96	0,95	0,97	0,98	0,98
	CART - Profile	0,92	1,00	0,96	0,96	0,92	0,94
TG_Colour	CART - All	0,85	1,00	0,92	0,93	0,92	0,93
	CART - Speaker	0,91	0,91	0,91	0,94	0,96	0,95
	CART - Profile	0,85	1,00	0,92	0,93	0,92	0,92
TG_Size	CART - All	0,62	0,41	0,50	0,77	0,59	0,67
	CART - Speaker	0,62	0,57	0,60	0,75	0,79	0,77
	CART - Profile	0,61	0,30	0,40	0,75	0,57	0,65
TG_Orientation	CART - All	0,13	0,03	0,06	0,58	0,43	0,50
	CART - Speaker	0,54	0,49	0,51	0,78	0,86	0,82
	CART - Profile	0,29	0,05	0,09	0,67	0,50	0,58
TG_XDimension	CART - All	0,00	0,00	0,00	0,80	0,84	0,82
	CART - Speaker	0,56	0,56	0,56	0,77	0,88	0,82
	CART - Profile	0,33	0,07	0,12	0,74	0,75	0,75
TG_YDimension	CART - All	0,34	0,22	0,27	0,74	0,82	0,78
	CART - Speaker	0,69	0,69	0,69	0,78	0,87	0,82
	CART - Profile	0,21	0,04	0,06	0,77	0,82	0,79

Tabela 42 – Resultados das Máquinas de Vetores de Suporte do domínio TUNA People

TUNA People									
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	0,83	1,00	0,91	0,44	0,95	0,98	0,96	0,89
	SVM - Speaker	0,95	0,94	0,94	0,89	0,95	0,95	0,95	0,89
	SVM - Profile	0,83	0,99	0,91	0,72	0,95	0,97	0,96	0,92
TG_Orientation	SVM - All	0,00	0,00	0,00	0,21	0,00	0,00	0,00	0,76
	SVM - Speaker	0,40	0,33	0,36	0,65	0,40	0,33	0,36	0,65
	SVM - Profile	0,00	0,00	0,00	0,30	0,00	0,00	0,00	0,55
TG_Age	SVM - All	0,00	0,00	0,00	0,11	0,40	0,22	0,29	0,55
	SVM - Speaker	0,60	0,33	0,43	0,82	0,67	0,44	0,53	0,83
	SVM - Profile	0,00	0,00	0,00	0,36	0,60	0,33	0,43	0,66
TG_HairColour	SVM - All	0,00	0,00	0,00	0,32	0,48	0,22	0,30	0,59
	SVM - Speaker	0,49	0,42	0,45	0,64	0,53	0,44	0,48	0,63
	SVM - Profile	0,29	0,05	0,08	0,47	0,60	0,31	0,41	0,61
TG_HasBeard	SVM - All	0,59	0,47	0,52	0,52	0,82	0,66	0,73	0,86
	SVM - Speaker	0,57	0,68	0,62	0,62	0,62	0,72	0,67	0,63
	SVM - Profile	0,62	0,75	0,68	0,68	0,79	0,69	0,73	0,86
TG_HasHair	SVM - All	0,00	0,00	0,00	0,30	0,00	0,00	0,00	0,47
	SVM - Speaker	0,16	0,10	0,13	0,59	0,21	0,14	0,17	0,57
	SVM - Profile	0,06	0,02	0,03	0,30	0,33	0,02	0,04	0,55
TG_HasGlasses	SVM - All	0,00	0,00	0,00	0,51	0,65	0,69	0,67	0,84
	SVM - Speaker	0,60	0,57	0,59	0,67	0,65	0,66	0,66	0,66
	SVM - Profile	0,57	0,68	0,62	0,61	0,68	0,76	0,71	0,83
TG_HasShirt	SVM - All	0,00	0,00	0,00	0,17	0,00	0,00	0,00	0,86
	SVM - Speaker	0,50	0,33	0,40	0,36	0,50	0,33	0,40	0,30
	SVM - Profile	0,00	0,00	0,00	0,13	0,00	0,00	0,00	0,79
TG_HasTie	SVM - All	0,00	0,00	0,00	0,03	0,00	0,00	0,00	0,24
	SVM - Speaker	0,00	0,00	0,00	0,69	0,00	0,00	0,00	0,67
	SVM - Profile	0,00	0,00	0,00	0,13	0,00	0,00	0,00	0,75
TG_HasSuit	SVM - All	0,00	0,00	0,00	0,03	0,00	0,00	0,00	0,07
	SVM - Speaker	0,00	0,00	0,00	0,57	0,00	0,00	0,00	0,59
	SVM - Profile	0,00	0,00	0,00	0,04	0,00	0,00	0,00	0,27
TG_XDimension	SVM - All	0,00	0,00	0,00	0,38	0,75	0,79	0,77	0,88
	SVM - Speaker	0,70	0,68	0,69	0,79	0,70	0,68	0,69	0,81
	SVM - Profile	0,48	0,12	0,19	0,58	0,76	0,67	0,71	0,87
TG_YDimension	SVM - All	0,00	0,00	0,00	0,40	0,78	0,73	0,75	0,86
	SVM - Speaker	0,73	0,72	0,73	0,83	0,73	0,71	0,72	0,84
	SVM - Profile	1,00	0,19	0,17	0,48	0,76	0,76	0,76	0,86

Tabela 43 – Resultados das Árvores de Classificação e Regressão do domínio TUNA People

TUNA People							
		VAR-			VAR+		
Classificador		P	R	F_1	P	R	F_1
TG_Type	CART - All	0,83	1,00	0,91	0,98	0,97	0,97
	CART - Speaker	0,93	0,94	0,94	0,97	0,90	0,93
	CART - Profile	0,90	0,89	0,89	0,98	0,97	0,97
TG_Orientation	SVM - All	0,00	0,00	0,00	0,43	0,50	0,46
	SVM - Speaker	0,50	0,50	0,50	0,08	0,50	0,13
	SVM - Profile	0,00	0,00	0,00	0,50	0,50	0,50
TG_Age	SVM - All	0,00	0,00	0,00	0,57	0,44	0,50
	SVM - Speaker	0,11	0,22	0,15	0,10	0,44	0,17
	SVM - Profile	0,00	0,00	0,00	0,50	0,44	0,47
TG_HairColour	SVM - All	0,07	0,02	0,03	0,51	0,35	0,41
	SVM - Speaker	0,43	0,41	0,42	0,83	0,90	0,86
	SVM - Profile	0,18	0,06	0,09	0,49	0,39	0,43
TG_HasBeard	SVM - All	0,58	0,46	0,52	0,79	0,75	0,77
	SVM - Speaker	0,62	0,58	0,60	0,91	0,93	0,92
	SVM - Profile	0,60	0,68	0,64	0,82	0,74	0,78
TG_HasHair	CART - All	0,12	0,14	0,13	0,16	0,24	0,19
	CART - Speaker	0,23	0,31	0,27	0,49	0,59	0,54
	CART - Profile	0,12	0,08	0,10	0,20	0,29	0,23
TG_HasGlasses	CART - All	0,30	0,44	0,36	0,35	0,41	0,38
	CART - Speaker	0,43	0,50	0,47	0,87	0,88	0,87
	CART - Profile	0,35	0,44	0,39	0,42	0,42	0,42
TG_HasShirt	CART - All	0,00	0,00	0,00	1,00	0,67	0,80
	CART - Speaker	0,00	0,00	0,00	0,05	0,67	0,09
	CART - Profile	0,00	0,00	0,00	0,50	0,67	0,57
TG_HasTie	CART - All	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Speaker	0,00	0,00	0,00	0,05	0,67	0,09
	CART - Profile	0,00	0,00	0,00	0,50	0,67	0,57
TG_HasSuit	CART - All	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Speaker	0,00	0,00	0,00	0,00	0,00	0,00
	CART - Profile	0,00	0,00	0,00	0,00	0,00	0,00
TG_XDimension	CART - All	0,00	0,00	0,00	0,80	0,87	0,83
	CART - Speaker	0,61	0,61	0,61	0,77	0,93	0,84
	CART - Profile	0,47	0,24	0,32	0,75	0,79	0,77
TG_YDimension	CART - All	0,39	0,27	0,32	0,86	0,85	0,85
	CART - Speaker	0,75	0,74	0,75	0,80	0,90	0,85
	CART - Profile	0,47	0,32	0,38	0,85	0,83	0,84

Tabela 44 – Resultados das Máquinas de Vetores de Suporte do *córpus Zoom-Pt*

		Zoom-Pt							
		VAR-				VAR+			
Classificador		P	R	F_1	AUC	P	R	F_1	AUC
TG_Type	SVM - All	0,95	1,00	0,98	0,25	0,95	1,00	0,97	0,54
	SVM - Speaker	0,95	0,90	0,92	0,49	0,95	0,87	0,91	0,57
	SVM - Profile	0,96	1,00	0,98	0,29	0,96	1,00	0,98	0,40
TG_Name	SVM - All	0,09	0,05	0,07	0,41	0,00	0,00	0,00	0,43
	SVM - Speaker	0,26	0,07	0,11	0,30	0,23	0,08	0,12	0,28
	SVM - Profile	0,10	0,06	0,08	0,55	0,33	0,01	0,02	0,49
TG_Other	SVM - All	0,00	0,00	0,00	0,05	0,00	0,00	0,00	0,11
	SVM - Speaker	0,01	0,50	0,02	0,76	0,01	0,50	0,02	0,50
	SVM - Profile	0,00	0,00	0,00	0,05	0,00	0,00	0,00	0,14
LM_Type	SVM - All	0,93	1,00	0,96	0,44	0,93	1,00	0,96	0,52
	SVM - Speaker	0,93	0,92	0,92	0,57	0,93	0,91	0,92	0,60
	SVM - Profile	0,93	1,00	0,96	0,43	0,93	1,00	0,96	0,57
LM_Name	SVM - All	0,97	1,00	0,98	0,35	0,97	0,99	0,98	0,44
	SVM - Speaker	0,97	0,92	0,95	0,49	0,97	0,90	0,94	0,34
	SVM - Profile	0,97	1,00	0,98	0,33	0,97	0,99	0,98	0,53
LM_Other	SVM - All	0,00	0,00	0,00	0,43	0,00	0,00	0,00	0,35
	SVM - Speaker	0,25	0,18	0,21	0,56	0,23	0,27	0,25	0,51
	SVM - Profile	0,00	0,00	0,00	0,42	0,00	0,00	0,00	0,38
Next-To	SVM - All	0,50	0,24	0,32	0,63	0,60	0,50	0,54	0,72
	SVM - Speaker	0,48	0,43	0,45	0,62	0,52	0,46	0,49	0,64
	SVM - Profile	0,50	0,24	0,32	0,63	0,60	0,50	0,54	0,72
Right-Of	SVM - All	0,00	0,00	0,00	0,28	0,60	0,30	0,40	0,85
	SVM - Speaker	0,01	0,1	0,02	0,53	0,03	0,40	0,06	0,77
	SVM - Profile	0,00	0,00	0,00	0,28	0,60	0,30	0,40	0,84
Left-Of	SVM - All	0,00	0,00	0,00	0,27	0,60	0,30	0,40	0,87
	SVM - Speaker	0,01	0,10	0,02	0,48	0,03	0,40	0,06	0,51
	SVM - Profile	0,00	0,00	0,00	0,28	0,60	0,30	0,40	0,85
In-Front-Of	SVM - All	0,00	0,00	0,00	0,42	0,00	0,00	0,00	0,49
	SVM - Speaker	0,03	0,10	0,05	0,55	0,02	0,07	0,03	0,53
	SVM - Profile	0,00	0,00	0,00	0,45	0,00	0,00	0,00	0,50
Behind-Of	SVM - All	0,00	0,00	0,00	0,17	0,00	0,00	0,00	0,53
	SVM - Speaker	0,01	0,50	0,02	0,77	0,01	0,05	0,02	0,76
	SVM - Profile	0,00	0,00	0,00	0,17	0,00	0,00	0,00	0,56
In	SVM - All	0,60	0,60	0,60	0,61	0,65	0,73	0,69	0,78
	SVM - Speaker	0,54	0,55	0,55	0,63	0,59	0,61	0,60	0,62
	SVM - Profile	0,55	0,62	0,58	0,59	0,65	0,73	0,69	0,78

Tabela 45 – Resultados das Árvores de Classificação e Regressão do *córpus Zoom-Pt*

		Zoom-Pt					
		VAR-			VAR+		
Classificador		P	R	F_1	P	R	F_1
TG_Type	<i>CART - All</i>	0,95	1,00	0,98	0,96	0,95	0,95
	<i>CART - Speaker</i>	0,96	0,77	0,85	0,95	0,80	0,87
	<i>CART - Profile</i>	0,95	1,00	0,98	0,96	0,95	0,95
TG_Name	<i>CART - All</i>	0,44	0,51	0,57	0,60	0,31	0,41
	<i>CART - Speaker</i>	0,24	0,29	0,26	0,57	0,73	0,64
	<i>CART - Profile</i>	0,43	0,50	0,46	0,62	0,32	0,42
TG_Other	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,01	0,50	0,01	0,01	0,50	0,01
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
LM_Type	<i>CART - All</i>	0,93	0,97	0,95	0,94	0,90	0,92
	<i>CART - Speaker</i>	0,93	0,68	0,78	0,93	0,72	0,81
	<i>CART - Profile</i>	0,93	0,97	0,95	0,94	0,90	0,92
LM_Name	<i>CART - All</i>	0,97	0,89	0,93	0,97	0,95	0,96
	<i>CART - Speaker</i>	0,97	0,70	0,81	0,97	0,75	0,85
	<i>CART - Profile</i>	0,97	0,89	0,93	0,97	0,95	0,96
LM_Other	<i>CART - All</i>	0,00	0,00	0,00	0,23	0,18	0,20
	<i>CART - Speaker</i>	0,24	0,41	0,30	0,26	0,48	0,34
	<i>CART - Profile</i>	0,00	0,00	0,00	0,23	0,18	0,20
Next-To	<i>CART - All</i>	0,42	0,45	0,44	0,52	0,50	0,51
	<i>CART - Speaker</i>	0,46	0,50	0,48	0,53	0,70	0,60
	<i>CART - Profile</i>	0,42	0,45	0,44	0,51	0,50	0,51
Right-Of	<i>CART - All</i>	0,00	0,00	0,00	0,21	0,30	0,25
	<i>CART - Speaker</i>	0,01	0,10	0,01	0,05	0,90	0,09
	<i>CART - Profile</i>	0,00	0,00	0,00	0,20	0,30	0,24
Left-Of	<i>CART - All</i>	0,00	0,00	0,00	0,21	0,3	0,25
	<i>CART - Speaker</i>	0,01	0,10	0,01	0,05	0,90	0,09
	<i>CART - Profile</i>	0,00	0,00	0,00	0,20	0,30	0,24
In-Front-Of	<i>CART - All</i>	0,00	0,00	0,00	0,12	0,10	0,11
	<i>CART - Speaker</i>	0,04	0,20	0,06	0,06	0,33	0,10
	<i>CART - Profile</i>	0,00	0,00	0,00	0,12	0,10	0,11
Behind-Of	<i>CART - All</i>	0,00	0,00	0,00	0,00	0,00	0,00
	<i>CART - Speaker</i>	0,01	0,50	0,01	0,01	0,50	0,01
	<i>CART - Profile</i>	0,00	0,00	0,00	0,00	0,00	0,00
In	<i>CART - All</i>	0,52	0,49	0,50	0,61	0,65	0,63
	<i>CART - Speaker</i>	0,54	0,59	0,56	0,62	0,72	0,66
	<i>CART - Profile</i>	0,52	0,49	0,50	0,61	0,65	0,63