UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

ANDRE COSTA MARIOTTI

**Geração de Expressões de Referência em situações de comunicação com
restrição de tempo**

São Paulo

2017

ANDRE COSTA MARIOTTI

**Geração de Expressões de Referência em situações de comunicação com
restrição de tempo**

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.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 13 de setembro de 2017. 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

2017

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

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

Mariotti, Andre Costa

Geração de expressões de referência em situações de comunicação com restrição de tempo / Andre Costa Mariotti ; orientador, Ivandré Paraboni. – 2017.

68 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. Inteligência artificial. 2. Processamento de linguagem natural. I. Paraboni, André, orient. II. Título.

CDD 22.ed.– 006.3

Dissertação de autoria de Andre Costa Mariotti, sob o título “**Geração de Expressões de Referência em situações de comunicação com restrição de tempo**”, 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, na área de concentração Metodologia e Técnicas da Computação, aprovada em 13 de setembro de 2017 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Ivandré Paraboni

Universidade de São Paulo

Presidente

Profa. Dra. Helena de Medeiros Caseli

Universidade Federal de São Carlos

Profa. Dra. Karina Valdivia Delgado

Universidade de São Paulo

Prof. Dr. Marcelo de Souza Lauretto

Universidade de São Paulo

Agradecimentos

Registro aqui minha sincera gratidão à toda a família Bertolin Mariotti que me motivou a perseguir o título de mestre. Especialmente, agradeço a minha mãe Ivete e minha esposa Myrian, por todo apoio e paciência neste período em que tive de estar focado nos trabalhos acadêmicos. Também sou grato a meu grande amigo Caio Trucolo e meu orientador desde a época da graduação e iniciação científica, Ivandré Paraboni. Esses amigos foram fundamentais na minha vida e ajudaram a abrir meus olhos para o mundo acadêmico, fazendo-me ver a ciência como algo realmente especial.

“Eu não estou preocupado com a morte, mas com a vida, para que ela não seja banal...”
(Mario Sérgio Cortella)

Resumo

Mariotti, Andre Costa **Geração de Expressões de Referência em situações de comunicação com restrição de tempo** 2017. 68 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2017.

Este documento apresenta uma pesquisa a nível de mestrado acadêmico, cujo o foco é a tarefa computacional de Geração de Expressões de Referência (GER), uma parte fundamental da comunicação que é estudada na Geração de Linguagem Natural (GLN). Mais especificamente, foram estudados os aspectos da linguagem que se manifestam em contextos de comunicação com restrição de tempo, e com base nisso foi proposto um modelo computacional de GER para produzir expressões de referência com o nível de superespecificação parametrizável. Além disso, considerando-se as características de adaptabilidade do modelo proposto, também foi sugerida uma generalização deste para outros domínios, como os que compreendem contextos de comunicação além dos que possuem restrição de tempo.

Palavras-chaves: Geração de Linguagem Natural; Geração de Expressões de referência; Restrição de Tempo.

Abstract

Mariotti, Andre Costa **Referring Expression Generation in time-constrained situations of communication** 2017. 68 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2017.

This document presents a MSc research that focused on the computational subtask of Referring Expression Generation (REG), an important component of Natural Language Generation (NLG) systems. More specifically, this work analyzes how time-restricted contexts of communication may affect language production and a computational model of GER was proposed to produce reference expressions with parameterizable superspecification. Furthermore, given the adaptability characteristics of the proposed model, it has also been suggested a generalization to other domains, which includes communication contexts besides those that have time constraints.

Keywords: Natural Language Generation; Referring Expression Generation; Time restrictions;

Lista de figuras

Figura 1 – Arquitetura de pipeline de GLN	14
Figura 2 – Exemplo de contexto visual	15
Figura 3 – Exemplo de contexto visual com objetos rotulados	20
Figura 4 – Exemplo de estímulo visual utilizado no experimento Stars2	25
Figura 5 – Exemplo de estímulos para descrições relacionais	26
Figura 6 – Exemplo de estímulos para descrições atômicas	27
Figura 7 – Exemplo de estímulo visual do domínio TUNA Furniture	28
Figura 8 – Exemplo de estímulo visual do domínio TUNA People	28
Figura 9 – Exemplo de estímulo visual do corpus b5-ref	29
Figura 10 – Contexto com um objeto-alvo B e distraidores ao redor	34
Figura 11 – Grafo representando objetos e propriedades observados na Figura 2	35
Figura 12 – Sub-grafo obtido após a execução do algoritmo de Krahmer	36
Figura 13 – Um cenário do ambiente virtual GIVE - os rótulos com nomes de cores foram adicionados para fins ilustrativos	40
Figura 14 – Ilustração da tela inicial do experimento realizado como jogo	45
Figura 15 – Telas do locutor (esquerda) e do ouvinte (direita)	46
Figura 16 – Médias de Dice obtidas pelo algoritmo IAL com a variação do grau de superespecificação	62

Lista de algoritmos

Algoritmo 1 – Algoritmo Incremental	33
Algoritmo 2 – Algoritmo Incremental Estendido	37
Algoritmo 3 – Classification-based Approach (SVM)	39
Algoritmo 4 – Geração de relações espaciais para ambiente GIVE	41
Algoritmo 5 – Algoritmo IAL	52

Lista de tabelas

Tabela 1 – Estatísticas descritivas dos corpos utilizados na avaliação	25
Tabela 2 – Resultados do estudo exploratório com diferenças estatisticamente sig- nificativas destacadas em negrito	47
Tabela 3 – Resultados	57
Tabela 4 – Resultados em outros domínios. Diferenças estatisticamente significati- vas são destacadas	61
Tabela 5 – Sequências de imagens utilizadas no Stars2	68

Lista de abreviaturas e siglas

AIE	Algoritmo Incremental Estendido
GER	Geração de Expressões de Referência
GLN	Geração de Língua Natural
IA	Algoritmo Incremental
IAL	Algoritmo Incremental Superespecificado
PLN	Processamento de Linguagem Natural

Sumário

1	Introdução	14
1.1	<i>Objetivo geral</i>	18
1.2	<i>Objetivos específicos</i>	18
1.3	<i>Hipótese</i>	19
1.4	<i>Organização da Monografia</i>	19
2	Conceitos Fundamentais	20
2.1	<i>Expressões de Referência</i>	20
2.2	<i>A Geração Computacional de Expressões de Referência</i>	21
2.3	<i>Produção de expressões de referência em condições críticas</i>	23
2.4	<i>Córpus para GER</i>	24
2.4.1	<i>Corpus Stars2</i>	25
2.4.2	<i>Córpus TUNA</i>	28
2.4.3	<i>Córpus b5-ref</i>	29
2.5	<i>Avaliação de algoritmos de GER</i>	30
3	Algoritmos de GER	31
3.1	<i>algoritmo Full Brevity</i>	31
3.2	<i>Algoritmo Greedy</i>	31
3.3	<i>Algoritmo Incremental</i>	32
3.4	<i>Algoritmo Baseado em Grafos</i>	35
3.5	<i>Algoritmo Incremental Estendido</i>	36
3.6	<i>Modelo de GER baseado em Aprendizado de Máquina e Variação Humana</i>	38
3.7	<i>Geração de relações espaciais em mundos virtuais</i>	39
3.8	<i>Considerações</i>	42
4	Estudo Exploratório	43
4.1	<i>Procedimento</i>	44
4.2	<i>Materiais</i>	46
4.3	<i>Sujeitos</i>	47
4.4	<i>Resultados</i>	47

4.5	<i>Discussão</i>	48
5	Modelo computacional de GER para situações de comunicação com restrição de tempo	49
5.1	<i>Algoritmo IAL</i>	49
6	Avaliação	54
6.1	<i>Objetivos</i>	54
6.2	<i>Modelos computacionais avaliados</i>	54
6.3	<i>Hipóteses</i>	55
6.4	<i>Conjunto de dados</i>	56
6.5	<i>Procedimento</i>	56
6.6	<i>Resultados</i>	56
6.7	<i>Discussão</i>	57
7	Generalização para situações de comunicação sem restrição de tempo	59
7.1	<i>Objetivo</i>	59
7.2	<i>Modelos computacionais avaliados</i>	59
7.3	<i>Hipóteses</i>	60
7.4	<i>Conjuntos de dados</i>	60
7.5	<i>Procedimento</i>	60
7.6	<i>Resultados</i>	61
7.6.1	Análise complementar: O papel do parâmetro n	61
8	Conclusão	63
	Referências ¹	64
	Anexo A – Imagens do Corpus Stars2	68

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Uma das principais tarefas envolvidas no ato de se comunicar é produção de textos com significado linguístico (BECHARA, 2009). Esta tarefa pode ser implementada computacionalmente por meio de sistemas de Geração de Linguagem Natural (GLN). Sistemas deste tipo permitem que entradas não-linguísticas sejam transformadas em saídas de texto ou voz compreensíveis para os humanos (REITER; DALE, 2000).

A Geração de Linguagem Natural envolve tipicamente três etapas: Planejamento do Documento, Microplanejamento e Realização superficial, conforme demonstra a Figura 1. Estas etapas são definidas na arquitetura de pipeline de GLN em Reiter e Dale (2000) e tem como objetivo, respectivamente, criar uma representação do domínio sobre o qual se está falando, decidir sobre a melhor representação semântica do conteúdo a ser apresentado, e por fim realizar a conversão deste conteúdo semântico em uma forma textual.

Neste trabalho será tratado especificamente da etapa de microplanejamento, responsável pela seleção do conteúdo semântico a ser comunicado em forma textual, dentre outras tarefas.

Figura 1 – Arquitetura de pipeline de GLN

Fonte: (PEREIRA; PARABONI, 2008)

Sistemas de GLN podem beneficiar inúmeras áreas que demandam habilidades humanas para transformar dados em texto, tal como sistemas capazes de gerar diagnósticos médicos a partir de dados numéricos produzidos por equipamentos hospitalares (BINSTED K.; CAWSEY, 1995; PORTET, 2007), sistemas que geram boletins de previsão do tempo a partir da leitura de satélites (KITTREDGE, 1994; BELZ, 2008), histórias narrativas (WALKER et al., 2011a; WALKER et al., 2011b), legendas para imagens (KARPATHY; FEI-FEI, 2015; XU et al., 2015), poesia automática (ZHANG; LAPATA, 2014), e também

a crescente indústria de jogos eletrônicos, que busca formas de trazer mais realismo para interação entre homem e máquina.

Na Figura 2 pode-se observar um cenário de videogame para o qual um sistema de GLN pode ser aplicado para gerar os diálogos das personagens de forma dinâmica e sensível ao contexto, e não meramente baseado em roteiros ou regras pré-definidas. Por exemplo, um sistema de GLN que receba informações sobre o estado atual do jogo, poderia gerar mensagens altamente customizadas, tais como: “Essa foi por pouco! tente pular mais alto para escapar do monstro”.

Figura 2 – Exemplo de contexto visual

Fonte: Imagem capturada do jogo Super Mario World da Nintendo®

Em situações dinâmicas como a observada neste exemplo, é possível que a linguagem se manifeste com características particulares dado o contexto da comunicação. Por exemplo, caso o motivo da comunicação suscite certas emoções, é possível que a personagem se expresse de maneira distinta em comparação a uma situação de comunicação neutra. Assim, uma personagem que em condições normais poderia dizer algo como “Você deve escalar os blocos para chegarmos ao alto daquele monte”, em condições adversas poderia dizer algo diferente, como: “Suba no bloco, rápido!”.

A relação entre emoções e produção de linguagem ainda é um tópico de pesquisa relativamente pouco explorado, possivelmente por conta da dificuldade em reproduzir emoções de forma controlada para então estudar seus efeitos na comunicação. Por exemplo, o estudo proposto em Krahmer (2014) procura provocar diversas emoções em sujeitos humanos por meio de vídeos de conteúdo alegre ou triste, para que em uma etapa seguinte os participantes do experimento realizem a tarefa de produzir linguagem natural. As conclusões apresentadas mostram que foi possível notar uma influência da emoção sobre a tarefa em questão, mas não foi possível saber com clareza como cada tipo de emoção influenciou a produção de linguagem.

Um método conveniente para observação dos efeitos causados por certos contextos de comunicação na tarefa de produzir linguagem seria por meio de um jogo computacional capaz de simular essas situações de maneira controlada. Em um ambiente “gameficado” é possível modelar de forma objetiva determinadas características do contexto de comunicação. Em especial, a criticidade e urgência de uma situação podem ser modeladas por meio de uma *restrição de tempo*, possibilitando o estudo da produção de língua natural em situações de comunicação adversas comparando-se a maneira como um indivíduo humano se expressa normalmente com a forma como se expressaria sob pressão (e.g., com restrição de tempo).

A produção de linguagem natural em condições de comunicação com restrição de tempo é o tema do presente trabalho. Entretanto, como forma de melhor observar os aspectos da produção de linguagem que podem ser afetados pela restrição de tempo, este trabalho tratará, ao invés da geração de texto de forma irrestrita, apenas do caso específico da subtarefa de Geração de Expressões de Referência (GER), que é um dos componentes essenciais do microplanejamento na arquitetura de sistemas de GLN (REITER; DALE, 2000).

Expressões de referência são formas linguísticas usadas para descrever os objetos sobre os quais se falam em um discurso (KRAHMER; DEEMTER, 2012). Por exemplo, para descrever as personagens e demais objetos presentes na Figura 2, pode-se utilizar expressões tais como:

- “O homem que possui bigode”;
- “O bloco amarelo”;
- “O monstrinho acima do monte”.

Essas expressões podem consistir de descrições definidas, pronomes, nomes próprios, e outras. Neste trabalho será tratado apenas o caso das descrições definidas como nos exemplos acima e, além disso, considerando apenas o problema da seleção de conteúdo semântico dessas expressões, sem discutir a sua realização superficial na forma de texto. Quando não houver risco de confusão, neste trabalho expressões de referência serão chamadas simplesmente de “descrições”, e a tarefa computacional de determinação do conteúdo dessas expressões será designada como a tarefa de GER.

Construções linguísticas como expressões de referência aparecem constantemente em diálogos reais, mas apesar de serem comuns e intuitivas para humanos, sua geração não representa uma tarefa simples do ponto de vista computacional (KRAHMER; DEEMTER,

2012). Para tratar desta tarefa, um algoritmo de GER recebe como entrada um conjunto de objetos e suas propriedades semânticas (como a cena visual da figura 2, composta de personagens de vários tipos, cores, formas e tamanhos), e também um objeto-alvo que se deseja distinguir dos demais elementos da cena. Por conseguinte, um algoritmo de GER tem a tarefa de encontrar um subconjunto de propriedades do objeto-alvo capaz de distingui-lo dos demais elementos da cena sem risco de ambiguidade. Exemplos de descrições que seriam possíveis saídas de um algoritmo de GER, incluem “O monstrinho marrom mais ao alto” e “O que está acima do monte”.

Algoritmos de GER tradicionais produzem descrições em situações de comunicação ditas normais. Em situações em que haja restrição de tempo, por outro lado, não está claro como estas descrições seriam produzidas. Além disso, os algoritmos existentes para tarefa computacional de GER tendem a ser pouco parametrizáveis, e são assim naturalmente difíceis de adaptar a situações de comunicação mais específicas como a da restrição de tempo de que trata esta pesquisa.

Uma possível abordagem neste caso seria fazer uso de métodos de aprendizagem de máquina para construir modelos de GER a partir de exemplos reais do uso da linguagem. Modelos deste tipo foram empregados com sucesso na tarefa de GER em (FERREIRA; PARABONI, 2014). Entretanto, o uso prático de aprendizado de máquina em GER é limitado pela necessidade de se possuir muitos exemplos linguísticos produzidos no domínio de interesse. Em outras palavras, exige a construção de recursos linguístico-computacionais especializados - ou *cópus* para GER - para cada domínio de interesse (e, havendo restrição de tempo, contemplando este aspecto da comunicação também). O custo de desenvolvimento de recursos deste tipo é evidentemente elevado e, como agravante, precisa ser repetido para cada nova aplicação de interesse.

Com base nestas observações, este documento apresenta os resultados de pesquisa a nível de mestrado acadêmico enfocando o problema computacional da Geração de Expressões de Referência em situações de comunicação com restrição de tempo. Este projeto envolve um estudo exploratório sobre como sujeitos humanos produzem descrições nestas condições, a proposta de um algoritmo de GER com parâmetros adicionais que permitam exercer maior controle sobre a saída produzida, e a avaliação da proposta como alternativa a algoritmos tradicionais da área.

1.1 Objetivo geral

O objetivo da pesquisa é o estudo e desenvolvimento de modelos computacionais de GER em situações de comunicação com restrição de tempo, de modo a produzir descrições mais próximas do comportamento humano do que as que seriam produzidas por algoritmos de GER tradicionais.

1.2 Objetivos específicos

- Realização de um estudo para comparar expressões de referência produzidas em condições de comunicação ditas normais com expressões produzidas sob restrição de tempo por meio de um experimento de produção de linguagem natural envolvendo sujeitos humanos.
- Criação de um novo corpus de GER com descrições definidas produzidas por sujeitos humanos em situações de comunicação com restrição de tempo.
- Projetar e implementar um algoritmo de GER customizável que seja capaz de gerar expressões de referência tal como um indivíduo humano produziria em situações com restrição de tempo.

1.3 Hipótese

A partir de estudos como em Arts et al. (2011) e também o do estudo exploratório a ser descrito no capítulo 4 observa-se que, em situações de comunicação atípicas, as expressões de referência produzidas por locutores humanos tendem a ser mais superespecificadas, ou seja, contendo mais informação do que o estritamente necessário para a desambiguação (PECHMANN, 1989). Por outro lado, algoritmos de GER tradicionais como o Algoritmo Incremental (DALE; REITER, 1995) e muitos de seus sucessores não costumam acrescentar informação redundante, já que esta tende a provocar falsas implicações lógicas (GRICE, 1975). Por exemplo, em um contexto onde todos monstros são marrons (Como na Figura 2), a informação sobre a cor em “o monstro marrom” seria redundante, e pode suscitar no leitor dúvidas sobre o motivo pelo qual está sendo incluída. Assim, este trabalho estabelece a seguinte hipótese geral de pesquisa:

Um algoritmo de GER que produza descrições com grau de superespecificação parametrizável produz descrições mais próximas das que seriam produzidas por locutores humanos em situações de restrição de tempo do que algoritmos tradicionais de GER.

1.4 Organização da Monografia

O restante desta monografia está estruturado da seguinte forma: O capítulo 2 apresenta conceitos fundamentais de GER. O capítulo 3 apresenta algoritmos tradicionais utilizados nesta tarefa computacional. No capítulo 4 é apresentado um estudo exploratório realizado com sujeitos humanos para investigar a geração de linguagem em situações de comunicação com restrição de tempo. No capítulo 5 é proposto um novo modelo computacional para tratar a tarefa de GER sob restrição de tempo. O capítulo 6 apresenta avaliações do modelo proposto. O capítulo 7 propõe uma generalização do modelo computacional para outras situações de comunicação. E por fim, o capítulo 8 apresenta a conclusão do trabalho.

2 Conceitos Fundamentais

Neste capítulo são discutidos conceitos relacionados à tarefa computacional de GER, a produção de descrições em condições críticas e aos corpos utilizados no estudo exploratório e testes do algoritmo desenvolvido.

2.1 Expressões de Referência

Durante o ato de se comunicar, tipicamente, existe um objeto ao qual o emissor da mensagem está se referindo e para o qual ele construirá uma determinada descrição que serve para distinguir esse objeto perante outros que também estão presentes no contexto da comunicação (REITER; DALE, 2000). Ou seja, participantes de um diálogo costumam fazer descrições do objeto sobre o qual falam para serem compreendidos, sendo que a essas descrições é dado o nome de Expressão de Referência. Por exemplo, participantes de um diálogo que conversam sobre a cena representada na figura 3 podem produzir expressões de referência tais como “O monstinho marrom ao lado dos cactos” para se referir a personagem rotulada como R5, ou “O bloco amarelo” para se referir ao objeto de rótulo R4.

Figura 3 – Exemplo de contexto visual com objetos rotulados

Fonte: Imagem capturada do jogo Super Mario World da Nintendo®

O objeto principal que motiva essas descrições é chamado de *objeto-alvo*. Além deste, também podem existir outros objetos no mesmo contexto, os quais são designados *distraindores*. Desta forma, a descrição “A bala de canhão preta” possui o R8 como objeto-

alvo e os demais personagens e objetos da cena como distraidores (R1, R2, R3, R4, R5, R6 e R7).

Em modelos computacionais de GER, tanto o objeto-alvo quanto seus distraidores são tipicamente representados por um conjunto de propriedades que consistem de pares atributo-valor, tais como $\langle \text{tipo, bala} \rangle$ ou $\langle \text{cor, preta} \rangle$. Além disso, quando uma propriedade se refere a uma característica do próprio objeto, esta é chamada de *Atômica* (Como no exemplo da cor preta usada para descrever a personagem). Já quando a propriedade se refere a um segundo objeto do contexto, aqui denominado *landmark* ou ponto de referência, a mesma é chamada de propriedade *relacional*, como $\langle \text{ao-lado, cacto} \rangle$. Assim, uma expressão que contém propriedades relacionais consiste de uma descrição do objeto-alvo via referência a outros objetos do contexto, tal como na expressão “O monstinho ao lado do cacto”, na qual o monstro é o alvo principal e o cacto é o objeto auxiliar usado como ponto de referência.

2.2 A Geração Computacional de Expressões de Referência

A tarefa computacional de gerar expressões de referência envolve decidir quais propriedades semânticas expressar, e em uma etapa subsequente transformar as propriedades selecionadas em uma forma textual. Em outras palavras, estes algoritmos resolvem o problema de gerar uma expressão por meio de uma seleção de conteúdo seguida de uma realização superficial (PEREIRA; PARABONI, 2008; LUCENA; PARABONI; PEREIRA, 2010; NOVAIS; PARABONI, 2012). Neste trabalho trataremos exclusivamente da sub tarefa de seleção de conteúdo, aqui denominada GER quando não houver risco de confusão com a etapa subsequente de realização superficial.

Algoritmos de GER recebem como entrada um contexto contendo o objeto-alvo e distraidores representados por propriedades (pares atributo-valor), e produzem como saída *expressões de referência* capazes de distinguir o objeto-alvo de todos os distraidores. Além disso, estas expressões devem ser, tipicamente, livres de ambiguidade, breves e psicologicamente plausíveis (KRAHMER; DEEMTER, 2012), pois durante essa tarefa computacional também existe a preocupação de produzir linguagem tal como um locutor humano faria (GATT; BELZ, 2007).

Em linhas gerais, um algoritmo de GER recebe como entrada um objeto-alvo r acompanhado de possíveis distraidores, ambos pertencentes a um dado contexto C . O algoritmo examina as propriedades de r e produz um conjunto L de propriedades que distingue r de todos os distraidores em C . Por exemplo, uma representação do contexto ilustrado na figura 3 poderia ser:

$R1 = (<\text{tipo}, \text{monstro}>, <\text{cor}, \text{marrom}>, <\text{acima}, \text{monte}>)$
 $R2 = (<\text{tipo}, \text{placa}>, <\text{cor}, \text{branca}>)$
 $R3 = (<\text{tipo}, \text{homem}>, <\text{cor}, \text{vermelha}>, <\text{tamanho}, \text{pequeno}>)$
 $R4 = (<\text{tipo}, \text{bloco}>, <\text{cor}, \text{amarela}>)$
 $R5 = (<\text{tipo}, \text{monstro}>, <\text{cor}, \text{marrom}>, <\text{a-esquerda}, \text{cactos}>)$
 $R6 = (<\text{tipo}, \text{cacto}>, <\text{cor}, \text{verde}>)$
 $R7 = (<\text{tipo}, \text{monstro}>, <\text{cor}, \text{marrom}>, <\text{a-direita}, \text{cactos}>)$
 $R8 = (<\text{tipo}, \text{bala}>, <\text{cor}, \text{preta}>, <\text{tamanho}, \text{grande}>)$

Neste contexto, dado o objetivo de descrever o objeto-alvo $R3$, um algoritmo de GER poderia selecionar um conjunto de propriedades de $R3$ como $L = < \text{cor}, \text{vermelha} >$, que posteriormente poderia ser realizado na forma de uma descrição definida como “o homem de vermelho”.

Algoritmos de GER como os que serão discutidos no capítulo 3 apresentam inúmeras estratégias para seleção das propriedades que vão compor a descrição de saída L . Em especial, é comum que estes algoritmos levem em conta o poder discriminatório da propriedade. Dado um objeto-alvo r e uma propriedade $< a, v >$ de r , o poder discriminatório é o número de distraidores que possuem um valor diferente de v para o mesmo atributo a que r . Por exemplo, no contexto ilustrado na Figura 3, para $r = R3$, o poder discriminatório de $< \text{cor}, \text{vermelha} >$ é 7, porque neste contexto há 7 objetos que não possuem a cor vermelha, e são eles $R1, R2, R4, R5, R6, R7$ e $R8$.

Diferentes estratégias de GER podem produzir descrições L de tamanho variado, e com diferentes graus de especificação. Quando L não é suficiente para distinguir o objeto-alvo r de todos os distraidores em C , esta expressão é dita ambígua ou subespecificada, como na expressão “O monstrinho” que não permite saber sobre qual das personagens se está falando quando se observa a cena representada na Figura 3. Quando L possui o menor número possível de propriedades capaz de distinguir r de seus distraidores, a expressão é dita mínima, como na descrição “A placa”, usada para descrever $R2$. Já quando L possui mais propriedades do que o estritamente necessário para desambiguação, a expressão é

designada como superespecificada. Um exemplo de superespecificação ocorre na expressão “o homem de bigode”, construída para descrever a personagem R1 observada na Figura 3. Neste caso a expressão “o homem” seria suficiente para que se identificasse o objeto-alvo.

A superespecificação não representa necessariamente uma característica de estilo, e possui funções particulares na linguagem, ajudando a distinguir objetos com maior facilidade (PECHMANN, 1989; ARTS et al., 2011). Estudos iniciais sobre o tema indicaram que a superespecificação pode ocorrer por diversos fatores, sendo comumente encontrada em situações nas quais existem muitos objetos distraidores (ARNOLD, 2007; KOOLEN et al., 2011).

Como motivador para superespecificação, em Arnold (2008) são apontadas estratégias para geração de língua incrementais que podem favorecer descrições com mais propriedades do que o necessário. Isso acontece porque os emissores da mensagem podem complementar suas descrições a ponto de mitigar o risco de serem mal compreendidos, e eles o farão de forma incremental até terem um sinal de sucesso no entendimento. Nestes cenários, a superespecificação aparece como um mecanismo de baixo custo cognitivo para o falante tornar sua mensagem mais clara e precisa (ENGELHARDT; BAILEY; FERREIRA, 2006).

2.3 Produção de expressões de referência em condições críticas

Em Newell (1972) é observado que o contexto de comunicação pode afetar a forma como as pessoas reagem a um problema e conseqüentemente como se comunicam. Em Arts et al. (2011) esse fenômeno é observado por um viés mais específico que trata da produção humana de expressões de referência em condições ditas críticas, que guardam uma certa afinidade com as situações de restrição de tempo de que trata o presente trabalho. As condições críticas consideradas em Arts et al. (2011) são definidas como situações nas quais especificar corretamente o objeto alvo era altamente importante para o sucesso da ação subjacente. Para tal, o experimento proposto em (ARTS et al., 2011) contou com dois grupos, dos quais o primeiro realizou a tarefa de produzir descrições em condições normais, e o segundo realizou a tarefa sob a orientação de que seriam corresponsáveis por uma cirurgia, na qual as descrições fornecidas serviriam para orientar um médico que estava realizando um procedimento remotamente e de alto risco.

Os resultados apresentados em Arts et al. (2011) foram significativos para confirmar que o contexto em que ocorre a comunicação e a criticidade envolvida na tarefa afetam de modo determinante a linguagem e, mais especificamente, causam a superespecificação de expressões de referência. Isso acontece porque, dada a preocupação de concluir a tarefa com êxito, participantes de um discurso acrescentam informação redundante a suas expressões. Além disso, segundo Arts et al. (2011), em situações de comunicação ditas como críticas, a superespecificação evidencia a preferência por propriedades mais salientes como formato, cor e tamanho.

Por além do contexto de comunicação explorado em Arts et al. (2011), em Koolen et al. (2016) são apontados possíveis efeitos da restrição de tempo no processo cognitivo de superespecificação de expressões de referências. Segundo Koolen et al. (2016), participantes de um discurso fazem a seleção de conteúdo em dois estágios. No primeiro deles, mais prematuro, as expressões são formadas por propriedades salientes, ou que de alguma forma se destacam no contexto. Já no segundo estágio, que é mais tardio, pode haver superespecificação, pois locutores possuem mais tempo para visualizar a cena e acrescentar propriedades a essas expressões. A fim de analisar melhor essa hipótese, em Koolen et al. (2016) é proposto um experimento no qual os participantes são convidados a montar descrições para sequências de imagens que desaparecem da tela após um intervalo de tempo restrito para cada estímulo. No entanto, importante ressaltar que o foco do estudo em (Koolen et al., 2016) não é a questão da urgência ou criticidade da comunicação, e sim uma investigação da sequência temporal em que ocorre a seleção de conteúdo e na produção de formas linguísticas.

2.4 Córpus para GER

Um *córpus* para GER consiste de um conjunto de estímulos (por exemplo, cenas) contendo objetos diversos (um dos quais é o objeto-alvo de interesse naquele contexto) e uma coleção de descrições produzidas por locutores humanos participantes do experimento de coleta de dados. Uma vez coletados estes dados, segue-se a anotação semântica (tanto dos contextos como das descrições obtidas) para que o conjunto de dados possa ser manipulado por algoritmos de GER.

Pesquisas em GLN e áreas correlatas têm desenvolvido uma série de corpú de expressões de referência. Exemplos desses corpú incluem o corpú TUNA (GATT; SLUIS; DEEMTER, 2007), GRE3D3 (DALE; VIETHEN, 2009) e GRE3D7 (VIETHEN; DALE, 2011), b5-ref (PARABONI; MONTEIRO; LAN, 2017) e Stars2 (PARABONI; GALINDO; IACOVELLI, 2017). Nesta seção serão apresentados especificamente os corpú Stars2, que foi utilizado no estudo exploratório deste trabalho, e os corpú b5-ref e TUNA, que foram utilizados em testes subsequentes para generalização do modelo computacional proposto.

A tabela 1 relaciona os corpú a serem utilizados no trabalhos, sua extensão em número de descrições coletadas, número de participantes e tipos de contextos considerados.

Tabela 1 – Estatísticas descritivas dos corpú utilizados na avaliação

Córpus	descrições	Participantes	Cenas
TUNA Furniture	420	60	7
TUNA People	360	60	6
Stars2	1204	76	64
b5-ref	1900	152	12

2.4.1 Corpus Stars2

O corpú Stars2 (PARABONI; GALINDO; IACOVELLI, 2017) é um conjunto de expressões de referência atômicas e relacionais produzidas por sujeitos humanos a partir de cenas simples, e que teve as propriedades semânticas de suas expressões anotadas para uso na pesquisa em GER. Os estímulos utilizados no corpú Stars2 são imagens representando conjuntos de formas geométricas. Por exemplo, a Figura 4 ilustra uma das cenas apresentadas aos participantes para condução do experimento.

Figura 4 – Exemplo de estímulo visual utilizado no experimento Stars2

Fonte: (PARABONI; GALINDO; IACOVELLI, 2017)

As situações de referência consideradas no *cópus* Stars2 foram projetadas de modo que determinadas propriedades do objeto-alvo fossem especialmente salientes, e assim provavelmente selecionadas com mais frequência em detrimento de certas preferências estabelecidas na literatura em Ciências Cognitivas, como o uso de cor em detrimento de outras propriedades atômicas (PECHMANN, 1989) ou o uso de propriedades atômicas em detrimento de relações espaciais (VIETHEN, 2010). Além disso, o *cópus* faz uso de cenas contendo um número relativamente alto de distraidores (14) que frequentemente exigem referência à até três objetos para desambiguação do alvo, como na expressão “O cone ao lado do cubo e próximo a esfera”.

No experimento, um grupo de imagens representava situações de referência em que o uso de uma descrição relacional era altamente provável, como na expressão “O cone ao lado da caixa”. Estas situações ilustradas pela Figura 5 constituem as condições ditas relacionais do experimento Stars2.

A diferença entre as quatro condições relacionais está no tipo de propriedade do objeto *landmark* mais próximo ao objeto-alvo, que é disponibilizado para o caso de o locutor desejar produzir uma descrição superespecificada.

Figura 5 – Exemplo de estímulos para descrições relacionais

Fonte: (PARABONI; GALINDO; IACOVELLI, 2017)

Um segundo grupo de imagens representa situações em que não era necessário o uso de descrições relacionais, como em “A esfera vermelha” usada para descrever o objeto-alvo da condição 8 na F 6. Estas situações constituem as condições de referência atômicas do experimento Stars2. Novamente, a diferença entre as condições atômicas reside no tipo de propriedade que pode ser superespecificada: uma relação projetiva (05), topológica (06), o atributo tamanho (07) ou cor (08).

Figura 6 – Exemplo de estímulos para descrições atômicas

Fonte: (PARABONI; GALINDO; IACOVELLI, 2017)

Ao total foram coletadas 1204 expressões de referência válidas que dão insumos para investigação da superespecificação e uso de expressões relacionais (PARABONI; GALINDO; IACOVELLI, 2017), servindo também de base para comparação entre modelos de GER. Assim, na seção de estudo exploratório do capítulo 4 será apresentada uma variação desse experimento que visa investigar a tarefa de GER realizada sob restrição de tempo.

2.4.2 Crpus TUNA

O Crpus TUNA (GATT; SLUIS; DEEMTER, 2007) constitudo de um conjunto de expresses de referncia obtidas de sujeitos humanos a partir de estmulos visuais cujo os temas foram itens de moblia e fotografias de personalidades da Matemtica.

O domnio TUNA Furniture, que trata do conjunto de estmulos visuais de moblia, possui objetos com os atributos de tipo, cor, localizao, tamanho e coordenadas X,Y da tela onde eram exibidos. J o domnio TUNA People consiste de fotografias em preto e branco de matemticos famosos, cujo os atributos podem ser, por exemplo, idade, cor do cabelo, localizao, tipo, etc. A ttulo de exemplo, a seguir tem-se duas imagens usadas como estmulos visuais do crpus TUNA para ambos os domnios.

Figura 7 – Exemplo de estmulo visual do domnio TUNA Furniture

Fonte: (FABBRIZIO; STENT; BANGALORE, 2008)

Figura 8 – Exemplo de estmulo visual do domnio TUNA People

Fonte: (FABBRIZIO; STENT; BANGALORE, 2008)

Cada imagem do crpus TUNA contm um objetos-alvo circulado de vermelho e acompanhado de seis distraidores posicionados de maneira aleatria. Estes estmulos foram apresentados a 60 participantes do experimento que produziram ao todo 2280 expresses de referncia, sendo 1200 do domnio TUNA Furniture e 1080 do domnio TUNA People.

O TUNA foi amplamente utilizado na pesquisa em GER (FABBRIZIO; STENT; BANGALORE, 2008; GATT; SLUIS; DEEMTER, 2007) para teste de diversos modelos computacionais, e portanto tambm estar contemplado neste trabalho. Alm disso, o

córpus foi a base de três competições (a série de TUNA Challenges) nos anos 2007-2009 (GATT; BELZ, 2007; GATT; BELZ; KOW, 2008; GATT; BELZ; KOW, 2009).

2.4.3 Córpus b5-ref

O Córpus b5-ref apresentado por Paraboni, Monteiro e Lan (2017) visa o estudo da geração de expressões de referência a partir de indivíduos humanos e os seus respectivos traços de personalidade. Este córpus oferece suporte a diferentes estudos de identificação de traços de personalidade a partir de textos no caso de expressões de referência, como também a tarefa de Geração de Expressões de Referência com base em traços de personalizada de um determinado locutor-alvo de interesse (PARABONI; MONTEIRO; LAN, 2017).

A coleta dessas descrições que compõem o córpus foi realizada a partir de um conjunto de 12 estímulos visuais contendo rostos de pessoas, tais como no exemplo da Figura 9. Cada uma das 12 imagens (PARABONI; MONTEIRO; LAN, 2017) possui 6 rostos pertencentes a pessoas com traços físicos diferentes e/ou com diferentes expressões faciais, sendo que o quadro contornado em vermelho representa o objeto-alvo a ser descrito.

Figura 9 – Exemplo de estímulo visual do córpus b5-ref

Fonte: (PARABONI; MONTEIRO; LAN, 2017)

Para identificação dos traços de personalidade, foi utilizado o modelo de Cinco Grandes Fatores (JOHN LAURA P. NAUMANN, 2008). Com base neste modelo, as características de personalidade foram estimadas por meio de inventários de personalidade preenchidos pelos próprios participantes do estudo exploratório realizado no b5-ref. No presente trabalho, entretanto, serão utilizadas apenas as expressões de referência produzidas, não fazendo uso dos inventários de personalidade.

Ao todo, foram produzidas cerca de 1900 expressões de referências e, assim como o TUNA (GATT; SLUIS; DEEMTER, 2007), o cópua b5-ref (PARABONI; MONTEIRO; LAN, 2017) se limita ao caso de referência com uso de descrições atômicas. Porém, como fator diferencial, o cópua b5-ref possui um domínio de considerável complexidade, com um total de 27 atributos anotados, e frequentemente sem poder discriminatório (como o formato da cabeça das pessoas na Figura 9, que para efeito de referência seriam todos muito parecidos). Estas características sugerem que o domínio b5-ref oferece desafios adicionais para algoritmos de GER típicos, já que o poder discriminatório é normalmente um fator fundamental na seleção de conteúdo (DALE; REITER, 1995).

2.5 Avaliação de algoritmos de GER

Para verificar o desempenho de um algoritmo de GER costuma-se comparar as expressões geradas pelo algoritmo contra expressões geradas por sujeitos humanos, de forma que se possa analisar quão próximo o resultado computacional foi do comportamento humano. Desta forma, a avaliação precisa ter essencialmente dois conjuntos básicos, sendo um deles com expressões de referência geradas por sujeitos humanos e outro conjunto com expressões geradas pelo algoritmo a ser testado.

Para comparar as expressões desses conjuntos, o método mais comum consiste de utilizar o coeficiente de Dice (DICE, 1945). Trata-se de uma medida de similaridade entre conjuntos que consiste da razão entre as propriedades em comum nos dois conjuntos e o total de propriedades de ambas as expressões, conforme a equação abaixo na qual A e B são conjuntos de propriedades a serem comparados.

$$DICE(A, B) = \frac{2 * |A \cap B|}{|A| + |B|} \quad (1)$$

3 Algoritmos de GER

Neste capítulo serão apresentados alguns algoritmos tradicionais que implementam a tarefa computacional de GER.

3.1 *algoritmo Full Brevity*

Existem diferentes estratégias para selecionar o conteúdo e realizar a produção de linguagem. A primeira classe de algoritmos a ser tratada neste trabalho explora o conceito da Máxima Brevidade introduzido em Grice (1975), o qual diz que participantes de um discurso procuram falar somente o necessário para atingir o objetivo da comunicação.

A Máxima Brevidade é melhor entendida como uma classe de algoritmos porque representa modelos computacionais cujo objetivo é obter expressões com um conjunto mínimo de propriedades para descrever um objeto (DEEMTER et al., 2011). A partir desses modelos, o algoritmo Full Brevity foi formalmente descrito em Dale (1989), sendo referenciado de forma recorrente na área, pois a brevidade é pertinente para se gerar expressões com mais naturalidade (DEEMTER et al., 2011).

O algoritmo Full Brevity procura, inicialmente, encontrar uma expressão formada por uma única propriedade que permita distinguir o objeto-alvo dos seus distraidores. Caso não seja possível, procura uma combinação de duas propriedades, que juntas distinguirão o objeto-alvo. Novamente, se ainda não for possível distinguir o alvo, o processo é repetido até encontrar um conjunto mínimo que descreva o objeto de forma única (DALE, 1989). Entretanto, dado que essa abordagem não possui uma solução ótima, já que a sua complexidade seria exponencial ao número de propriedades da expressão (REITER; DALE, 2000), também existem implementações gulosas para tratar a máxima brevidade.

3.2 *Algoritmo Greedy*

O algoritmo Greedy (DALE, 1989) consiste numa abordagem heurística para implementar o conceito de Máxima Brevidade que, em seu funcionamento, seleciona as propriedades uma a uma para compor a expressão, tendo como critério de escolha a propriedades com maior poder discriminatório.

O algoritmo Greedy compõe a expressão de forma incremental até alcançar sua condição de parada, que se dá quando o objeto-alvo pode distinguir-se dos demais, ou quando todas as propriedades do objeto foram selecionadas. Nesses casos, o algoritmo pode apresentar todas as propriedades para compor a expressão, ou nenhuma delas.

O diferencial deste algoritmo está no uso do valor discriminatório em resposta ao problema de complexidade presente no algoritmo Full Brevity. Essa abordagem, em certo sentido, favorece a seleção de propriedades que melhor descrevem o objeto-alvo na composição de uma expressão mínima.

Embora eficiente na maioria dos casos, o algoritmo Greedy nem sempre retorna uma expressão mínima. Além disso, estudos posteriores demonstraram que as pessoas não necessariamente produzem expressões mínimas em contextos reais de comunicação. Ou seja, em contextos reais de comunicação sujeitos humanos também produzem descrições superespecificadas (PECHMANN, 1989; DALE; REITER, 1995; ENGELHARDT; BAILEY; FERREIRA, 2006).

3.3 Algoritmo Incremental

O algoritmo Incremental em Dale e Reiter (1995) representa uma abordagem amplamente estudada para tratar da geração de expressões de referência, e que motivou diversos trabalhos posteriores. Basicamente, este algoritmo seleciona as propriedades uma a uma para complementar a descrição do objeto-alvo de forma incremental, e faz a seleção conforme uma ordem de prioridade fixa dos atributos, considerando as propriedades que contribuem para a desambiguação. Segundo Dale e Reiter (1995), esta ordem de preferência é definida pelo domínio. Esta estratégia é motivada por estudos da área de psicolinguística, que sugerem a existência de uma ordem em que essas propriedades são usualmente empregadas na comunicação humana, podendo ser tomada por base para construção de uma descrição (PECHMANN, 1989). Além disso, em Dale e Reiter (1995) é observado que, na comunicação entre humanos, a realização textual se inicia mesmo antes da seleção de conteúdo ser concluída. Ou seja, ao falar sobre um objeto, o locutor começa a descrevê-lo mesmo antes de observar a totalidade de suas características e ponderar sobre qual supostamente seriam as propriedades mais relevante para compor a descrição.

Com base nessas observações, em Dale e Reiter (1995) foi formulado o algoritmo Incremental (IA), cujo o pseudocódigo pode ser visto no Algoritmo 1.

Algoritmo 1 Algoritmo Incremental

```

1: IncrementalAlgorithm(r,D,P)
2:  $L \leftarrow null$ 
3:  $C \leftarrow D - (r)$ 
4: foreach  $i$  in list  $P$ , do
5:    $V = Value(r, i)$ 
6:   if  $rulesOut(C, < i, V >) > 0$  then
7:      $L \leftarrow L \cup < i, V >$ 
8:      $C \leftarrow C - rulesOut(C, < i, V >)$ 
9:   end if
10:  if  $|C| == 0$ , then
11:    return  $L$ 
12:  end if
13: end for
14: return  $L$ 
15:
16: rulesOut(Context  $C$ , Proprety  $p$ )
17:  count = 0
18:  foreach object in Context , do
19:    if object.getProperty(p.i) != p.v then
20:      count++
21:       $C.remove(object)$ 
22:    end for
23: return count

```

Fonte: (DALE; REITER, 1995)

O algoritmo IA recebe como entrada um objeto-alvo r , o domínio de objetos D , e uma ordem de preferência P para utilização dos atributos (passo 1). O algoritmo inicia construindo uma descrição vazia L (passo 2) e removendo r do conjunto de elementos à considerar como distraidores para atribuir esse subconjunto ao contexto C (passo 3). A cada iteração, o algoritmo adiciona propriedades segundo a ordem de preferência definida P e, se a propriedade considerada é discriminatória - verificada através da função $rulesOut$ (passo 6), ela é incluída na saída L (passo 7) e o conjunto de elementos do contexto é atualizado removendo-se os objetos eliminados com a propriedade selecionada (passo 8). Ao final de cada iteração, o algoritmo verifica se já possui uma descrição capaz de diferenciar o objeto-alvo de seus distraidores (passo 10), e caso não consiga, realiza uma nova iteração.

Por exemplo, caso o algoritmo fosse gerar uma expressão de referência para o cubo B representado pela Figura 10, receberia como entrada o objeto-alvo B, os distraidores A, C e

D, e uma lista de preferência $P = \{\text{forma, cor, tamanho, posição}\}$. Em sua primeira iteração, o algoritmo produziria $L = \{< \text{forma, cubo} >\}$, pois a forma geométrica é o primeiro atributo da lista de preferências, e a sua seleção permite descartar os distraidores A e C que não são cubos. Num passo seguinte, verifica-se que ainda existem objetos no contexto a serem eliminados (D), dando sequência a uma nova iteração. A seguir, a descrição L não seria incrementada novamente, pois não seria possível distinguir o objeto-alvo do distraidor restante (D) dado que ambos possuem a mesma cor. Por fim, em uma última iteração, o algoritmo Incremental selecionaria o atributo tamanho e obteria a descrição $L = \{< \text{forma, cubo} >, < \text{cor, verde} >, < \text{tamanho, grande} >\}$ que permite diferenciar o objeto-alvo (B) de seus distraidores (A, C e D), encontrando portanto a sua condição de parada. Tipicamente, a saída produzida pelo algoritmo poderia ser realizada de forma textual, como na descrição “O cubo grande”.

Figura 10 – Contexto com um objeto-alvo B e distraidores ao redor

Fonte: Andre Mariotti, 2017

Em termos de custo computacional, em Dale e Reiter (1995) é discutido que o Algoritmo Incremental apresenta um tempo polinomial melhor do que os tempos de Full Brevity e Greedy. Além disso, segundo Dale e Reiter (1995) gerar expressões de referência seguindo uma ordem de prioridade fixa por domínio oferece um resultado mais próximo do real na maioria dos casos.

Embora amplamente estudado e considerado geralmente eficiente, o fato do algoritmo IA possuir uma ordem de prioridade fixa para seleção das propriedades o torna pouco flexível em alguns cenários, como no caso de situações atípicas de comunicação no qual o uso dos atributos pode variar. Além disso, em Dale e Reiter (1995) é discutido que participantes de um discurso podem utilizar propriedades conforme sua conveniência, não sendo elas necessariamente úteis para desambiguação. Assim, com o uso do algoritmo IA, as expressões geradas podem apresentar superespecificação de forma espúria, já que uma

propriedade selecionada por ser discriminatória pode se tornar redundante com as seleções subsequentes.

3.4 Algoritmo Baseado em Grafos

Em Krahmer, Erk e Verleg (2003) é apresentado um algoritmo que permite favorecer o uso de propriedades relacionais na tarefa de GER, no qual o contexto de entrada é representado por meio de um grafo. Nesta estrutura de dados, os vértices representam os objetos e as arestas representam propriedades. O grafo é direcionado e laços representam propriedades atômicas do objeto, enquanto arestas para outros vértices representam propriedades relacionais.

O ganho deste algoritmos está na facilidade de percorrer a estrutura de dados para obter descrições com propriedades relacionais, tais como “A cadeira ao lado da mesa” que faz referência a propriedades atômicas e relacionais do objeto-alvo (cadeira).

Para exemplificar o funcionamento deste algoritmo, pode-se observar na Figura 11 uma representação de alguns objetos da Figura 2 do Capítulo 1 e suas respectivas propriedades, dispostos na forma de grafo.

Figura 11 – Grafo representando objetos e propriedades observados na Figura 2

Fonte: Andre Mairiotti, 2017

Dado um contexto representado por um grafo como na Figura 11, o algoritmo percorre a estrutura de dados a partir do objeto-alvo que se deseja descrever na tentativa de encontrar um sub-grafo que o identifica de forma única. Esta lógica caracteriza um problema análogo ao isomorfismo de sub-grafos (KRAHMER; ERK; VERLEG, 2003). Assim, como possível exemplo de resultado na execução, tem-se o sub-grafo da Figura 12, que posteriormente poderia ser realizado na forma superficial “O homem de bigode”.

Figura 12 – Sub-grafo obtido após a execução do algoritmo de Krahmer

Fonte: Andre Mariorri, 2017

O algoritmo baseado em grafos tira proveito de algoritmos tradicionais da Teoria dos Grafos para encontrar descrições ótimas. Além disso, atribuindo-se custos as arestas do grafo é possível reproduzir o comportamento de diversos algoritmos de GER. Por exemplo, uma escolha cuidadosa de custos permite implementar a ordem de preferência do Algoritmo Incremental (cf. Seção 3.3), e a definição de arestas de custo zero favorece a superespecificação referencial.

3.5 Algoritmo Incremental Estendido

Apesar de amplamente utilizado para tarefa computacional de GER, o Algoritmo Incremental (DALE; REITER, 1995) apresentado na Seção 3.3 possui a limitação de não gerar expressões de referência relacionais. Uma primeira resposta a essa limitação do algoritmo IA surgiu em Krahmer e Theune (2002) e foi posteriormente estendida em KELLEHER J. D.; KRUIJFF (2006). Em ambos os casos, os modelos computacionais propostos tinham a funcionalidade de incrementar a expressões de referência com o uso propriedades relacionais, mas priorizavam o uso de propriedades atômicas. Entretanto, em alguns domínios, como o GRE3D3 e GRE3D7 (DALE; VIETHEN, 2009; VIETHEN; DALE, 2011) apresentados no Capítulo 2, é possível notar grande incidência de propriedades relacionais, tal como na expressão de referência “O cubo abaixo do cone e próximo à esfera”.

Visando oferecer um modelo de GER capaz de gerar expressões de referência que favorecem o uso de propriedades relacionais, em Ferreira e Paraboni (2017) é apresentado uma variação do algoritmo IA, denominada AIE (Algoritmo Incremental Estendido). O pseudocódigo deste modelo é apresentado a seguir, no algoritmo 2 :

Algoritmo 2 Algoritmo Incremental Estendido

```

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

```

Fonte: Adaptado de Ferreira e Paraboni (2017)

O algoritmo AIE (FERREIRA; PARABONI, 2017) é essencialmente uma extensão do algoritmo IA (DALE; REITER, 1995) para manipular propriedades relacionais, e não apenas atômicas. Para este fim, o algoritmo faz uso de uma função auxiliar *eliminaDistraidores* que é uma generalização da função *rulesOut* do algoritmo original. Esta função implementa um procedimento de teste de ambiguidade que leva em conta as relações entre objetos, que é mais complexo do que a simples comparação de propriedades atômicas feita no algoritmo IA original. Detalhes deste procedimento são discutidos em Ferreira e Paraboni (2017).

O algoritmo AIE faz uso de chamadas recursivas (passo 16) para descrever tanto o objeto-alvo como os pontos de referência (ou landmarks) a ele relacionados. A chamada é

realizada sempre que o valor do atributo selecionado faz referência a um objeto do domínio (como acima-B) ao invés de referenciar um valor atômico (como cor-azul) (passo 12).

Adicionalmente, o algoritmo AIE precisa tratar o problema de circularidade no uso de relações, de forma a impedir a geração de expressões do tipo “a bola ao lado da caixa que está ao lado da bola”. Este tratamento é feito por meio do controle dos objetos que estão ativos (passos 4 e 13), evitando que os mesmos sejam novamente referenciados.

3.6 Modelo de GER baseado em Aprendizado de Máquina e Variação Humana

Grande parte da literatura sobre GER é focada em soluções algorítmicas para realizar a tarefa computacional de selecionar conteúdo e produzir expressões de referência. Entretanto, esse paradigma vem sendo aprimorado com o surgimento de novas abordagens que fazem uso de Aprendizagem de Máquina (AM) (FERREIRA; PARABONI, 2014).

Em Ferreira e Paraboni (2014) é apresentada uma abordagem de GER que faz uso aprendizagem por meio de Máquinas de Vetor Suporte (SVM). Neste algoritmo, a SVM constitui uma abordagem supervisionada para classificação das decisões de seleção de cada atributo do objeto-alvo. Assim, a partir de um corpus para GER pré-existente, o modelo faz uso de classificadores individuais para decidir quando selecionar cada propriedade atômica ou relacional. Os classificadores de propriedades atômicas são do tipo binário, e o classificador de propriedades relacionais é do tipo multiclasse, podendo assumir valores como “acima”, “abaixo”, etc. e o valor especial “norelation” indicando que uma relação não é recomendada para aquela situação.

Os modelos utilizados por Ferreira e Paraboni (2014) consideram características de aprendizagem que representam o contraste dos objetos da cena (tais como o número de objetos que compartilham a mesma cor ou formato do objeto-alvo) e atributos do próprio sujeito autor da expressão tomada como exemplo (como sua identificação, gênero, faixa etária e a lista de propriedades mais frequentes que este utiliza), tirando assim proveito da noção de variação humana segundo a qual diferentes pessoas podem selecionar conjuntos de propriedades distintas quando descrevendo um mesmo objeto.

Um exemplo do pseudocódigo apresentando por Ferreira e Paraboni (2014) pode ser visualizado no Algoritmo 3. O algoritmo recebe um objeto-alvo t , um conjunto de atributos a ser considerado, e previsões de atributos obtidas com uso do conjunto de classificadores.

Para cada previsão positiva, o algoritmo adiciona o atributo correspondente através da função `getObjectDescription` e, num passo subsequente, testa a possibilidade de utilizar um atributo relacional. Caso a previsão indique “não relacional”, então o algoritmo retorna uma expressão atômica e, do contrário, adiciona a referência para o outro objeto que será referenciado de forma recursiva.

Algoritmo 3 Classification-based Approach (SVM)

```

1: function getDescription(t, attributes, predictions)
2:    $L \leftarrow getObjectDescription(t, null, attributes[target], predictions[target])$ 
3:   if  $predictions[relation] \neq norelation$  then
4:      $landmark \leftarrow value(t, predictions[relation])$ 
5:      $L \leftarrow L \cup (Ai, )$ 
6:      $L \leftarrow getObjectDescription(V, L, attributes[landmark], predictions[landmark])$ 
7:   end if
8: end if
9: return L
10: end function
11:
12: function getObjectDescription(t, L, attributes, objectPredictions)
13:    $L \leftarrow [null]$ 
14:   foreach  $Ai$  in attributes do
15:      $v \leftarrow value(t, Ai)$ 
16:     if  $objectPredictions[Ai] = positive$  then
17:        $L[t] \leftarrow L[t] \cup (Ai, v)$ 
18:     end if
19:   end for
20:   return L
21: end function

```

Fonte: Adaptado de Ferreira e Paraboni (2014)

Em comparações realizadas entre a abordagem baseada em classificação SVM e o algoritmo AIE discutido na seção anterior, o modelo que faz uso de aprendizagem se mostrou superior na maioria dos casos. Entretanto, cabe destacar que abordagens deste tipo só são factíveis quando um conjunto de exemplos de descrições no domínio de interesse pode ser disponibilizado para fins de treinamento.

3.7 Geração de relações espaciais em mundos virtuais

Em Silva e Paraboni (2015) é apresentado um algoritmo capaz de produzir relações espaciais em ambientes virtuais do tipo GIVE (Sigla em inglês para Geração de Instruções em Ambientes Virtuais). GIVE é um *Framework* para avaliação de sistemas Geradores de

Linguagem Natural em mundos virtuais interativos, e foi usado em uma série de desafios de geração de linguagem (BYRON et al., 2007). A Figura 13 apresenta uma imagem ilustrativa do ambiente virtual GIVE.

Figura 13 – Um cenário do ambiente virtual GIVE - os rótulos com nomes de cores foram adicionados para fins ilustrativos

Fonte: (BYRON et al., 2007)

Grande parte da literatura sobre GER explora a geração de expressões em ambientes bidimensionais, fazendo pouca distinção entre propriedades atômicas e relacionais, bem como não abordam o caso particular do uso de relações espaciais para desambiguação. No trabalho proposto em Silva e Paraboni (2015), o ambiente GIVE é assim usado para investigar quando relações espaciais são empregadas na construção de expressões de referência e como estas se relacionam com propriedades atômicas.

A solução de GER proposta em Silva e Paraboni (2015) é composta de quatro módulos, sendo que o primeiro destes tem por objetivo extrair uma cena do ambiente virtual contendo os objetos do contexto e suas respectivas propriedades. Nesta etapa, todas as relações espaciais possíveis entre todos os pares de objetos da cena são computadas com uso das equações de distância absolutas e relativas propostas em Kelleher e Costello (2009).

Em seguida, o módulo denominado *classificador* faz uso do cópulus GIVE-2 (KOLLER et al., 2010) para aplicar um filtro que ajuda a selecionar tipos de referências que ocorrem no uso prático da língua. O terceiro módulo (de seleção), calcula o potencial que as propriedades relacionais têm de desambiguar o objeto, e escolhe a mais adequada para compor a descrição naquela situação específica. Por fim, o quarto e último módulo

implementa uma extensão do Algoritmo Incremental (DALE; REITER, 1995) capaz de lidar com propriedades atômicas e relacionais, cuja a ordem de prioridade é obtida com o conjunto de treinamento.

O quarto módulo de GER é responsável pela construção da descrição propriamente dita, e tem como entradas um objeto-alvo r , um contexto C representando distraidores do ambiente GIVE, e uma ordem de preferência P para uso das propriedades, tal como no Algoritmo Incremental. Este procedimento é ilustrado no algoritmo 4.

Algoritmo 4 Geração de relações espaciais para ambiente GIVE

```

1: Describe( $C, r, P$ ):
2:  $L \leftarrow null$ 
3:  $History \leftarrow null$ 
4:  $t \leftarrow r$ 
5:  $Lt \leftarrow null$ 
6:  $hasRelation \leftarrow False$ 
7:  $Distractors \leftarrow C$ 
8: foreach  $a$  in  $P$  do:
9:    $v \leftarrow value(t, a)$ 
10:  if  $|RulesOut(a, v)| > 0$  and  $v \notin History$  then
11:    if  $IsRelational(a) = false$  OR  $hasRelation = false$  then
12:       $Lt \leftarrow Lt \cup \langle a, v \rangle$ 
13:       $Distractors \leftarrow Distractors - RulesOut(a, v)$ 
14:      if  $IsRelational(a)$  then  $hasRelation \leftarrow True$ 
15: until  $|Distractors| = 0$ 
16:  $L \leftarrow L \cup Lt$ 
17:  $t \leftarrow landmark(Lt)$ 
18: if  $t = null$  then return  $L$ 
19:  $History \leftarrow History \cup t$ 
20: repeat 04

```

Fonte: (SILVA; PARABONI, 2015)

O algoritmo inicializa com uma descrição vazia L para o objeto-alvo r e uma lista para histórico dos objetos já mencionados em L . Ao executar o laço principal, o algoritmo constrói uma descrição para cada objeto t pertencente ao domínio, sendo t o próprio objeto-alvo na primeira iteração. Dentro do laço, o algoritmo verifica se a propriedade é discriminatória e, caso seja relacional, se o *landmark* ainda não existe no histórico de objetos já citados, de modo a evitar a referências circulares (e.g., “O botão ao lado da porta ao lado do botão”). Tratamentos adicionais nas linhas subsequentes impedem que múltiplas relações sejam concatenadas num mesmo objeto, como em “O botão próximo a porta e ao lado do vaso”. A cada interação é atualizada a contagem de distraidores restantes, até a obtenção de uma descrição não ambígua.

Os resultados apresentados em Silva e Paraboni (2015) sugerem que, em se tratando em ambientes tridimensionais, o uso de relações espaciais não aparece como último recurso para a descrição de objetos. Esta observação contradiz a literatura convencional em GER (DALE; HADDOCK, 1991). Além disso, foi observado que o uso de um classificador para definir a prioridade no uso de propriedades aumentou a presença de expressões relacionais, tornando as expressões de referência geradas mais próximas daquelas observadas no corpus GIVE-2 (KOLLER et al., 2010).

3.8 Considerações

Por além dos algoritmos tratados nas sessões anteriores, existem diversas outras abordagens para tratar o problema de GER. Contudo, nenhum dos modelos investigados trata da tarefa de GER considerando as situações de comunicação críticas ou com restrição de tempo, tais como as situações apresentadas em (ARTS et al., 2011; KOOLEN et al., 2016). Desta forma, pode-se afirmar que a implementação de um algoritmo que produza expressões de referência realistas nestas situações permanece um problema de pesquisa em aberto.

4 Estudo Exploratório

A fim de observar os possíveis efeitos da restrição de tempo na tarefa de produzir linguagem, foi elaborado um experimento exploratório para coleta de expressões de referência produzidas por sujeitos humanos. Este experimento trata de uma reprodução do experimento Stars2 (PARABONI; GALINDO; IACOVELLI, 2017) adaptado para que a sua dinâmica ocorresse na forma de jogo e considerando uma condição de comunicação crítica modelada de forma objetiva por meio de uma restrição de tempo.

O experimento teve por objetivo comparar as expressões originalmente produzidas sem restrição de tempo no experimento Stars2 com expressões produzidas sob situações idênticas, porém com restrição de tempo. Além disso, o estudo apresentado em Arts et al. (2011) demonstrou que expressões de referência tendem a ser mais superespecificadas quando a comunicação é considerada uma tarefa crítica por parte dos locutores e, por esse motivo, pressupõe-se no presente estudo exploratório que o mesmo princípio pode ser válido no caso da restrição de tempo. Desta forma, espera-se obter evidências de possíveis diferenças entre a linguagem natural produzida em condições críticas e não críticas.

A hipótese a ser investigada neste experimento é a de que expressões de referência geradas em condições de comunicação com restrição de tempo tendem a apresentar certas diferenças quantitativas e qualitativas em relação a expressões geradas em condições ditas normais. Em termos quantitativos, espera-se que expressões produzidas sob restrição de tempo sejam em média mais extensas (e portanto superespecificadas) do que as produzidas em condições ditas normais, em concordância assim com os achados em (ARTS et al., 2011) referentes à produção de descrições em condições críticas.

Em termos qualitativos, espera-se que as expressões com restrição de tempo façam uso de conjuntos de propriedades semânticas diferentes dos que seriam utilizados em condições normais, especialmente no que diz respeito ao uso dos atributos cor e tamanho do objeto-alvo. Estudos como Pechmann (1989) demonstram que o atributo cor é frequentemente acrescentado (de forma redundante) para produção de expressões superespecificadas, e pode assim ser mais frequente em uma situação de restrição de tempo. Estudos como Gompel et al. (2014) demonstram também que o atributo tamanho, por ser relativo aos outros objetos do contexto (e.g., ‘grande’, ‘maior do que’ etc.) exige esforço cognitivo considerável, e pode assim ser menos frequente em situações de restrição de tempo.

Esta análise será conduzida comparando-se o conteúdo semântico das expressões produzidas no experimento com restrição de tempo, e as expressões originais do *Stars2*. De forma mais específica, a comparação entre os dois tipos de condições levará em conta a medição de seis variáveis de interesse: (i) a quantidade de informação presente nestas expressões (*length*, representada pela quantidade de propriedades anotadas), (ii) a quantidade de relações espaciais empregada (*rel-count*), (iii) o grau de superespecificação da descrição do objeto-alvo (*over.tg*, representado pelo número de propriedades semânticas que poderiam ser removidas da descrição sem ambiguidade), (iv) o grau de superespecificação da descrição do primeiro *landmark* (*over.lm1*, medido de forma análoga à *over.tg*), (v) a quantidade de referências à cor do objeto-alvo (*colour.use*) e (vi) a quantidade de referências ao tamanho do objeto-alvo (*size.use*).

4.1 Procedimento

O experimento fez uso de uma ferramenta desenvolvida especialmente para coleta de expressões de referência com base em um conjunto de sequências de estímulos visuais utilizados no experimento *Stars2*, porém implementando recursos adicionais para reforçar a noção de restrição de tempo imposta à tarefa de geração destas descrições. Além disso, o experimento teve uma etapa inicial pela qual todos os participantes passaram para que pudessem compreender o objetivo da tarefa de forma efetiva.

Como diferencial para simulação da condição crítica, os participantes foram envolvidos na problemática simulada de ter que desarmar uma bomba durante o jogo. De forma mais específica, ao iniciar o experimento os participantes foram orientados por uma personagem do jogo de que, para cumprir essa tarefa, deveriam fornecer descrições para uma sequência de imagens que seria apresentada antes que o tempo acabasse, pois do contrário a bomba iria explodir. A título de exemplo, a Figura 14 ilustra a tela inicial do jogo.

Figura 14 – Ilustração da tela inicial do experimento realizado como jogo

Fonte: Andre Mariotti, 2017

As imagens utilizadas como estímulos visuais foram apresentadas nas mesmas sequências em que foram aplicadas no experimento original Stars2 (PARABONI; GALINDO; IACOVELLI, 2017). Cada imagem continha um objeto-alvo apontado por uma seta azul, a ser descrito pelo participante, e objetos distraidores ao redor, totalizando 15 objetos por imagem. Em uma tela, participantes no papel de locutor forneciam descrições para os objetos solicitados e, em outra tela, participantes no papel de ouvintes tinham a tarefa de identificar o objeto a partir da expressão de referência fornecida pelo locutor.

Para a tarefa de identificação do objeto-alvo, os ouvintes tinham de escolher um objeto dentre os demais que apareciam rotulados por uma letra na tela, ou clicar em “Não entendi”, caso não fosse possível localizar o objeto-alvo a partir da descrição fornecida. Uma vez que o ouvinte selecionasse a letra correta que correspondia ao objeto-alvo, o jogo avançava e uma nova imagem era exibida até que se chegasse ao final da sequência de 20 imagens. Já quando o ouvinte indicava o objeto errado, ou quando clicava em “Não Entendi”, o jogo penalizava os participantes exibindo a mesma imagem e reforçando a ideia de que o tempo para cumprir a tarefa estava se esgotando. Os participantes (locutor e ouvinte) não podiam ver a tela um do outro, sendo que a sua interação se limitou à troca de descrições e o feedback sobre o entendimento das descrições fornecidas (c. f. Figura 15). Para realização do experimento proposto, foi utilizado o paradigma experimental director-matcher (KRAUSS; WEINHEIMER, 1966) no qual o papel de diretor foi desempenhado

pelos participantes locutores do experimento, e o papel de assistente (ou ouvinte) foi desempenhado pelo autor desta pesquisa.

Figura 15 – Telas do locutor (esquerda) e do ouvinte (direita)

Fonte: Andre Mariotti, 2017

Durante toda a execução do experimento, os participantes visualizaram a imagem de um explosivo em contagem regressiva. Além disso, os participantes não sabiam quantas imagens ainda precisariam descrever para concluir a tarefa. Com o passar do tempo, uma personagem aparecia oportunamente para reforçar que era necessário se apressar, exercendo mais pressão sobre os participantes.

4.2 Materiais

Foram utilizadas 24 instâncias completas do experimento Stars2 (PARABONI; GALINDO; IACOVELLI, 2017), na mesma ordem de apresentação em que ocorreram na construção do corpus. Essas sequências de imagens estão registradas no anexo A. Cada sequência consiste de 20 imagens de estímulo semelhantes à da Figura 4 do capítulo 2. Assim como em Paraboni, Galindo e Iacovelli (2017), metade dos estímulos (ou seja, 10 imagens) representava contextos que suscitam o uso de relações espaciais, e a outra metade era do tipo em que não havia necessidade de usar relações espaciais, sendo possível identificar o objeto-alvo simplesmente com o uso de uma descrição atômica, como em “a esfera vermelha”.

4.3 Sujeitos

Participaram 23 voluntários ao todo, sendo 12 do sexo feminino e 11 do sexo masculino. Os voluntários possuíam uma média de idade de 32 anos e escolaridade de nível superior completa.

4.4 Resultados

Foram realizadas 23 sessões individuais do experimento com diferentes participantes, obtendo-se um total de 368 descrições produzidas sob restrição de tempo. Este conjunto de expressões constitui o *córpus* que será aqui denominado Stars2T. Essas descrições foram anotadas por dois juízes seguindo o mesmo esquema adotado em Paraboni, Galindo e Iacovelli (2017).

A Tabela 2 apresenta estatísticas descritivas do experimento, divididas em seis grupos representando cada variável de interesse: *length*, *rel.count*, *over.tg*, *over.lm1*, *colour.use* e *size.use*. Para cada variável, são apresentados os valores de média, variância e a quantidade *n* de itens considerados tanto na condição normal (observada no *córpus* Stars2) quanto na condição dita crítica (observada no *córpus* Stars2T com descrições produzidas sob restrição de tempo).

O valor de *n* corresponde ao total de descrições coletadas (368), exceto para os testes que distinguem descrições relacionais e atômicas (*over.tg* e *over.lm1*). Nestes casos, as variáveis foram analisadas apenas com base nos estímulos visuais em que esse tipo de descrição era esperado, o que corresponde a metade das descrições (184).

Tabela 2 – Resultados do estudo exploratório com diferenças estatisticamente significativas destacadas em negrito

	length		rel-count		over.tg		over.lm1		colour.use		size.use	
	normal	crítico	normal	crítico	normal	crítico	normal	crítico	normal	crítico	normal	crítico
média	3,14	3,34	0,90	0,94	0,52	0,79	0,74	0,73	0,18	0,26	0,19	0,20
var	2,84	2,47	0,66	0,56	0,39	0,39	0,31	0,33	0,14	0,23	0,15	0,20
n	368	368	368	368	184	184	184	184	368	368	368	368

Fonte: Andre Mariotti, 2017

Na análise dos resultados, foram utilizados testes do tipo ANOVA entre sujeitos comparando-se a variável de interesse nas condições sem e com restrição de tempo. Observa-se em primeiro lugar que o grau médio de superespecificação do objeto-alvo *over.tg* em

condições de restrição de tempo é superior ao das mesmas descrições em situações ditas normais. A diferença é significativa ($F(1, 101) = 15.947, p < 0.05$). Este resultado é consistente com o estudo em (ARTS et al., 2011) sobre a produção de expressões de referência em condições críticas.

Além disso, observa-se que a referência à cor do objeto (*colour.use*) aumenta quando há restrição de tempo. A diferença também é significativa ($F(1, 101) = 6.039, p < 0.05$). Este resultado é explicada pelo fato de que, do domínio Stars2, a informação sobre a cor do objeto tende a ser redundante, e possivelmente correlacionado à superespecificação do objeto-alvo.

Os demais testes não apresentaram diferença significativa entre as duas condições, o que em alguns casos pode ser explicado pelo número reduzido de descrições que fazem uso de relações espaciais no conjunto de dados coletado.

4.5 *Discussão*

Embora modestos, os resultados do experimento podem ser vistos como um indício de que, sob restrição de tempo, locutores produzem expressões de referências distintas tanto no volume como teor de informação. Esta observação é em princípio consistente com os resultados apresentados em Arts et al. (2011) para situações de comunicação críticas, e serão tomadas por base para um estudo mais detalhado sobre este tema, a ser descrito no próximo capítulo.

Além da análise apresentada neste capítulo, o estudo exploratório também contou com uma pesquisa qualitativa informal realizada com os participantes. Nesta pesquisa os participantes manifestaram terem se sentido pressionados ao produzir as descrições, pois entenderam que era importante completar a tarefa antes do tempo limite. Com base nesta entrevista e também nas características de superespecificação das expressões analisadas, parece razoável sugerir que situações de restrição de tempo mantêm uma relação de afinidade com situações de comunicação críticas, tal como as observadas em Arts et al. (2011).

5 Modelo computacional de GER para situações de comunicação com restrição de tempo

Neste capítulo será apresentado um modelo computacional motivado pelo estudo exploratório realizado no capítulo anterior e também pelos resultados apresentados em (ARTS et al., 2011). O modelo proposto - aqui denominado Algoritmo Incremental Parametrizável (IAL) - compreende uma nova modificação do Algoritmo Incremental (DALE; REITER, 1995) para tratar o caso da geração de expressões de referência em situações de comunicação com restrição de tempo.

5.1 Algoritmo IAL

O algoritmo Incremental padrão, apresentado na Seção 3.3, trata o problema de GER sem exercer um controle explícito sobre o tamanho das expressões produzidas e assumindo que receberá uma lista de propriedades preferenciais que é fixa por domínio. Entretanto, os resultados do estudo exploratório do capítulo 4 sugerem que expressões geradas sob restrição de tempo podem variar tanto no tamanho quanto na forma, e tendem assim a apresentar maior grau de superespecificação. Como base nestas observações, foi desenvolvida uma variação do algoritmo Incremental que torna a superespecificação parametrizável, aqui denominado algoritmo IAL.

O objetivo do algoritmo proposto é permitir que a tarefa de GER se adapte a diferentes contextos de comunicação, fazendo com que o tamanho das expressões reflita a estratégia de referência que um locutor humano utilizaria em situações de comunicação tais como as que possuem restrição de tempo. Para tal, a estratégia de superespecificação utilizada pelo algoritmo IAL demanda essencialmente duas novas funcionalidades, nas quais define-se quantas propriedades a mais serão utilizadas na superespecificação (que será definido por meio de um parâmetro n), e quais propriedades serão empregadas na geração da descrição (que serão definidas por meio de uma lista de preferência P customizada). Com esses insumos, o algoritmo proposto realizará a tarefa de GER não apenas produzindo uma expressão livre de ambiguidades como faz o algoritmo IA padrão, mas também ajustando esta expressão ao grau de superespecificação desejado.

O grau de superespecificação n , pode ser definido de forma empírica com base em um conjunto de descrições de treinamento. Neste trabalho, este parâmetro é definido

como o número médio de propriedades utilizadas para uma expressão num dado contexto subtraindo-se o tamanho mínimo que uma expressão precisa ter para que não seja ambígua. Assim, dada uma expressão L de um objeto-alvo r em um contexto C , e dado um conjunto T de descrições de treinamento neste mesmo domínio, o grau de especificação n de L é definido como segue:

$$n = \frac{\sum(|L| - \text{tamMin}(r, C))}{|T|} \quad (2)$$

Além de definir a quantidade de propriedades adicionais que as expressões devem conter, uma estratégia de superespecificação precisa também decidir quais propriedades serão selecionadas e em qual ordem. Originalmente, o algoritmo IA tenta construir uma expressão adicionando propriedades conforme uma lista de preferência P , que é fixa para o domínio em questão. No modelo proposto, por outro lado, são utilizadas listas P customizadas para cada contexto do domínio. Esta estratégia é motivada pela observação prática de que o acréscimo de propriedades redundantes pode não estar sujeito às regras de seleção de propriedades com propósito de desambiguação. Em outras palavras, observa-se que, em atenção à Máxima de Brevidade (GRICE, 1975), o algoritmo IA seleciona somente propriedades com poder discriminatório, critério este que pode não ser relevante na escolha de propriedades redundantes subsequentes.

Para definição da lista P a ser usada em cada contexto, o algoritmo IAL analisa, com base em dados de treinamento, a frequência com que cada propriedade é utilizada. Desta forma, o algoritmo sempre considerará uma lista P que prioriza a seleção das propriedades mais utilizadas na cena para qual ocorre a descrição.

Além destas duas funcionalidades básicas de superespecificação referencial, o algoritmo IAL implementa também um tratamento específico para as propriedades relacionais, permitindo que a descrição produzida faça referência à *landmarks* que também podem ser superespecificados. Esta funcionalidade faz-se necessária para avaliação do algoritmo com base no corpus Stars2T, a ser discutida no próximo capítulo. Por exemplo, para uma cena em que bastaria a descrição “A esfera ao lado do cubo” para distinguir o objeto-alvo, a descrição “A esfera ao lado do *pequeno* cubo *vermelho*” seria considerada superespecificada, sendo que as propriedades adicionais foram acrescentadas ao *landmark* (cubo) e não ao objeto-alvo (esfera). Quando uma propriedade relacional é encontrada na lista P , o algoritmo IAL tenta completar a descrição com propriedades do objeto referenciado, sendo

que a prioridade sempre será a superespecificação do *landmark*. Nos casos em que existem referências espaciais múltiplas, como na descrição “O cone ao lado do cubo e próximo a esfera”, a superespecificação ocorre no último *landmark* (que no caso do exemplo seria a esfera). Além disso, o algoritmo possui ainda um tratamento para evitar que a expressão gerada possua uma relação ciclica, como no caso da descrição “O cone ao lado do cubo que está próximo ao cone”, tal como no algoritmo AIE (FERREIRA; PARABONI, 2017). O pseudocódigo do algoritmo IAL é ilustrado no algoritmo 5 e o link para sua implementação original encontra-se no rodapé desta página¹.

O algoritmo IAL funciona da seguinte forma: Nas linhas 1 e 2 tem-se uma chamada para funções auxiliares que oferecem suporte à execução do modelo computacional proposto. O parâmetro n é computado pela função *defineOverSpec*, cujo objetivo é retornar o número médio de propriedades que as expressões do contexto possuem além do necessário para desambiguação. A lista de preferências P é feita com a função *getPreferences*. Esta função retorna uma lista P computada com base na frequência com que cada propriedade foi utilizada nas descrições da cena em questão, começando pelos atributos com maior incidência no contexto.

Dos passos 4 ao 9, o comportamento do algoritmo IAL é semelhante ao do algoritmo IA (DALE; REITER, 1995). Basicamente, tem-se a criação de uma descrição vazia que será incrementada durante a execução do algoritmo (passo 5), a definição do contexto como sendo os objetos do domínio D menos o objeto-alvo r (passo 6), e a subsequente tentativa de eliminar os objetos distraidores com as propriedades da lista de preferência P fazendo uso da função *rulesOut* tal como no algoritmo IA padrão (passos 7 a 9). Detalhes desse procedimento foram apresentados na seção 3.3.

No passo 10 do algoritmo é possível notar que, adicionalmente, o algoritmo recebeu um teste para verificar se a propriedade que está sendo selecionada é relacional. A ideia neste passo é permitir uma chamada recursiva (passo 12) para complementar a descrição L com informações do *landmark*. A lógica utilizada nesta etapa é a mesma do algoritmo AIE (FERREIRA; PARABONI, 2017) para tratar descrições relacionais.

No passo 18 é possível notar uma importante diferença entre o algoritmo IAL e o original IA, que é a modificação da condição de parada do algoritmo. Originalmente, o algoritmo verificava se a descrição L obtida era suficiente para desambiguação. O algoritmo IAL, por outro lado, verifica se o nível de superespecificação foi atingido. Assim, caso o

¹ <https://github.com/drehmariotti/IAL.git>

Algoritmo 5 Algoritmo IAL

```

1:  $P \leftarrow \text{getPreferences}(C)$ 
2:  $n \leftarrow \text{defineOverSpecLevel}(C)$ 
3:
4:  $\text{IAL}(r, D, P, n)$ 
5:  $L \leftarrow \text{null}$ 
6:  $C \leftarrow D - (r)$ 
7:  foreach  $i$  in list  $P$ , do
8:     $v = \text{Value}(r, i)$ 
9:    if  $\text{rulesOut}(C, \langle i, v \rangle) > 0$  then
10:     if  $v \in D$  then
11:        $L \leftarrow L \cup \text{IAL}(v, C, P, n)$ 
12:        $C \leftarrow C - \text{rulesOut}(C, \langle i, v \rangle)$ 
13:     else
14:        $L \leftarrow L \cup \langle i, v \rangle$ 
15:        $C \leftarrow C - \text{rulesOut}(C, \langle i, v \rangle)$ 
16:     end if
17:   if  $|L| == n$  then
18:     return  $L$ 
19:   end if
20: end for
21:
22: // superespecificação para atingir  $n$ 
23:
24: if  $|L| < n$  then
25:   foreach  $i$  in list  $P$ , do
26:     if  $\text{Value}(r, i) \neq 0$  then
27:       if  $v \in D$  then
28:          $L \leftarrow L \cup \langle i, v \rangle$ 
29:          $C \leftarrow C - \text{rulesOut}(C, \langle i, v \rangle)$ 
30:          $L \leftarrow L \cup \langle \text{"landmark - type"}, \text{Value}(r, i) \rangle$ 
31:       else
32:          $L \leftarrow L \cup \langle i, v \rangle$ 
33:          $C \leftarrow C - \text{rulesOut}(C, \langle i, v \rangle)$ 
34:       end else
35:     end if
36:   end for
37: return  $L$ 
38: end IAL

```

tamanho da superespecificação de L seja inferior a n , o algoritmo tenta complementar a descrição com mais propriedades até atingir o nível de superespecificação desejado para o contexto de comunicação (passos 25 ao 36). Também é importante observar que nesta etapa o algoritmo não seleciona apenas propriedades com poder discriminatório, mas sim propriedades que potencialmente seriam utilizadas por sujeitos humanos segundo a lista P , e que consequentemente podem contribuir para superespecificação.

6 Avaliação

Neste capítulo será apresentada a avaliação do algoritmo IAL aplicado à tarefa de geração de expressões de referência em situações de comunicação em que há restrição de tempo, e sua comparação com o algoritmo Incremental original (DALE; REITER, 1995) nestas mesmas condições. As próximas seções apresentarão uma visão geral sobre os testes realizados (6.1), os modelos computacionais avaliados (6.2), as hipóteses elencadas a cerca do modelo proposto (6.3), o conjunto de dados analisados (6.4), o experimento de avaliação propriamente dito (6.5), os seus resultados (6.6) e considerações (6.7).

6.1 Objetivos

O objetivo da avaliação realizada foi o de demonstrar que o algoritmo IAL proposto na seção anterior é capaz de gerar descrições mais próximas do desempenho humano em situações em que há restrição de tempo, em comparação ao algoritmo Incremental padrão. Para este fim, foram submetidos à avaliação três modelos computacionais: O algoritmo IAL e três sistemas de baseline baseados no algoritmo Incremental (DALE; REITER, 1995). Estes modelos são descritos a seguir.

6.2 Modelos computacionais avaliados

O modelo proposto IAL, conforme apresentado no Capítulo 5, possui o parâmetro n (grau de superespecificação) combinado com a utilização de listas P customizadas por contexto. Assim, além de demonstrar que o desempenho do IAL é superior ao do algoritmo IA, a avaliação procurou demonstrar também que esta melhoria é resultado da combinação das estratégia de superespecificação com listas de preferência customizadas (parâmetros adicionais do modelo proposto IAL). Para este fim, além de usar o algoritmo IA como *baseline*, dois outros sistemas foram avaliados. Uma variação do algoritmo IAL, denominada $IA+$, recebe como parâmetros o valor n , computado com base no tamanho médio do grau de superespecificação de cada contexto do conjunto de treinamento, e uma lista de preferência P , que é fixa para todo o domínio. A segunda variação do algoritmo

IAL é uma modificação do algoritmo padrão IA para utilização de listas P customizadas, porém sem superespecificação, denominada IAP.

6.3 Hipóteses

A avaliação realizada objetivou verificar as três hipóteses h1 e h2 e h3 a seguir:

h1: Em situações de comunicação em que há restrição de tempo, a geração de expressões de referência com grau de superespecificação parametrizável e listas P customizadas por contexto (IAL) produz resultados mais próximos do desempenho humano do que a geração realizada por padrão pelo Algoritmo Incremental (IA).

h2: Em situações de comunicação em que há restrição de tempo, a geração de expressões de referência com grau de superespecificação parametrizável e listas P customizadas por contexto (IAL) produz resultados mais próximos do desempenho humano do que a geração que faz uso apenas do parâmetro adicional n (IA+).

h3: Em situações de comunicação em que há restrição de tempo, a geração de expressões de referência com grau de superespecificação parametrizável e listas P customizadas por contexto (IAL) produz resultados mais próximos do desempenho humano do que o algoritmo IA utilizando listas P customizadas (IAP).

Para verificação destas hipóteses, as expressões de referência produzidas pelos modelos computacionais avaliados serão comparadas com expressões geradas por sujeitos humanos em um conjunto de dados de teste, medindo-se o coeficiente Dice (DICE, 1945).

6.4 Conjunto de dados

A avaliação dos modelos computacionais foi realizada com base no *cópus* Stars2T de expressões produzidas por locutores humanos em situações com restrição de tempo, apresentado no capítulo 4. Estas expressões foram separadas em conjuntos de treinamento e teste. Levou-se em conta a proporção de 80% das expressões para treinamento e 20% para testes, selecionadas aleatoriamente dentro de cada contexto de maneira que houvesse ao menos uma expressão para cada cena do mesmo par.

O conjunto de dados de treinamento contou com 294 expressões.

6.5 Procedimento

Os modelos avaliados (IAL, IA+, IAP e IA) tiveram seus parâmetros definidos com base no conjunto de dados de treinamento, sendo que este procedimento antecede a execução dos algoritmos e foi suportado pelas funções *defineOverSpec* e *getPreferences*, ilustradas no algoritmo 5. No caso do modelo IAL foram utilizadas ambas as funções, no caso do IA+, apenas a função *defineOverSpec*, e para o IAP apenas *getPreferences*.

O algoritmo IA, por outro lado, fez uso de uma lista de preferências *P* fixa para todo o domínio. Essa lista foi a mesma utilizada na execução do IA+, sendo:

$$P = \{ \textit{colour}, \textit{size}, \textit{near}, \textit{in-front-of}, \textit{right}, \textit{left}, \textit{below}, \textit{above} \textit{ e } \textit{behind} \}$$

Os modelos IAL, IA+, IAP e IA foram executados com seus devidos parâmetros de entrada para gerar as expressões de referência do conjunto de teste. A seguir, cada expressão gerada foi comparada com o mesmo par no *cópus* Stars2T, calculando-se o coeficiente de Dice (DICE, 1945) de cada par sistema-*cópus*. A significância estatística entre a média de coeficientes Dice de cada algoritmo foi avaliada com uso do teste de Wilcoxon. Adicionalmente, foi computada a acurácia global de cada modelo como a proporção de descrições idênticas às do conjunto de teste.

6.6 Resultados

Os resultados da avaliação estão sumarizados na Tabela 3.

Tabela 3 – Resultados

IA			IA +			IAP			IAL		
Dice			Dice			Dice			Dice		
Média	DP	Acc	Média	DP	Acc	Média	DP	Acc	Média	DP	Acc
0,23	0,16	0	0,26	0,16	0	0,56	0,21	0,03	0,65	0,16	0,06

Com relação à hipótese h1 (IAL vs. IA), os resultados sugerem que o algoritmo IAL pode produzir expressões de referência mais próximas do comportamento humano em situações com restrição de tempo do que o *baseline* IA. Esse resultado é significativo segundo teste de Wilcoxon ($W=2080$, $p<0.0001$)).

Com relação à hipótese h2 (IAL vs. IA+), os resultados obtidos sugerem que o modelo IAL produz resultados mais próximos do desempenho humano do que o modelo IA+. Isso significa que, para superespecificar as expressões de referência, é preciso escolher criteriosamente as propriedades que irão compor a descrição, e não apenas acrescentar um certo número de propriedades à descrição gerada. A diferença entre as médias de coeficiente Dice obtidas para esse teste também é significativa ($w=-1830$, $P<0001$) e oferece suporte à hipótese h2.

Com relação à hipótese h3 (IAL vs. IAP), os resultados obtidos sugerem que o modelo IAL é superior ao *baseline* IAP. Isso significa que, a estratégia de escolher criteriosamente as propriedades não é suficiente caso não seja combinada com o parâmetro de superespecificação n . Esse resultado também é significativo segundo teste de Wilcoxon ($W=586$, $p=0.025$)).

6.7 Discussão

No experimento exploratório detalhado no Capítulo 4, foi constatado que as expressões produzidas sob restrição de tempo são, sob certos aspectos, diferentes das produzidas em condições normais, característica esta que tentou-se reproduzir com o algoritmo IAL.

Ao analisar o modelo proposto em comparação aos sistemas de *baseline*, pode-se verificar que o algoritmo IA padrão apresenta resultados de coeficiente Dice relativamente baixos ao realizar a tarefa de GER em contextos de comunicação com restrição de tempo. Além disso, observou-se que a estratégia de tentar superespecificar expressões de referência de forma irrestrita (apenas adicionando o parâmetro de $rgau$ superespecificação)

não melhora estes resultados, dado que diminui a precisão com que as propriedades adicionais são selecionadas. O uso do algoritmo IA padrão combinado com o uso de listas P customizadas, por outro lado, produziu uma melhora no resultado. Entretanto, essa estratégia apresenta um resultado inferior ao obtido com o modelo computacional IAL, que obteve o melhor resultado ao realizar a tarefa de GER sob restrição de tempo no domínio Stars2T.

7 Generalização para situações de comunicação sem restrição de tempo

A observação de que o algoritmo IAL pode ter se comportamento customizado para situações de comunicação com restrição de tempo nos leva a considerar sua possível customização também para outros tipos de situação. Embora não fizesse parte dos objetivos iniciais desta pesquisa, a possibilidade de uso do algoritmo IAL para situações de comunicação ditas normais (i.e., sem restrição de tempo) pode representar uma alternativa a abordagens de GER baseadas em métodos de aprendizagem de máquina (FERREIRA; PARABONI, 2017) que exigem grandes massas de exemplos de expressões para fins de treinamento. Assim, neste capítulo será apresentada uma tentativa de generalização deste modelo para outros domínios em que não há restrição de tempo, mais ainda assim observa-se superespecificação referencial em níveis distintos do que seria possível reproduzir com uso de um algoritmo de GER tradicional como o algoritmo Incremental (DALE; REITER, 1995).

7.1 *Objetivo*

O objetivo desta avaliação foi o de verificar o desempenho do algoritmo IAL em relação ao algoritmo Incremental (DALE; REITER, 1995) na geração de expressões de referência em domínios em que não há restrição de tempo.

7.2 *Modelos computacionais avaliados*

Nesta avaliação foram utilizados os algoritmos IAL (modelo proposto) e o algoritmo IA utilizado como *baseline*. Ambos os modelos seguiram a mesma parametrização descrita na seção 6.5, ou seja, o algoritmo IAL fez uso do grau de superespecificação e listas de preferência customizadas, e o algoritmo IA utilizou a lista de preferência fixa para todo o domínio.

7.3 Hipóteses

Dado que o modelo computacional apresentado neste trabalho foi desenvolvido para o caso específico de situações de comunicação com restrição de tempo em contextos de comunicação, para sua generalização cabe a elaboração de uma nova hipótese h4.

h4: Em situações de comunicação em que não há restrição de tempo, a tarefa de GER parametrizada com o nível médio de superespecificação e listas P customizadas (IAL) permite resultados mais próximos aos de locutores humanos do que o algoritmo Incremental tradicional (IA).

Para verificação desta hipótese, as expressões de referência produzidas pelos dois modelos computacionais avaliados serão comparadas com expressões geradas por sujeitos humanos, novamente medindo-se o coeficiente Dice (DICE, 1945).

7.4 Conjuntos de dados

Para fins de avaliação foram utilizados três corpuses para GER: TUNA Furniture e TUNA People (GATT; SLUIS; DEEMTER, 2007) e b5-ref (PARABONI; MONTEIRO; LAN, 2017), todos apresentados no Capítulo 2. Os corpuses TUNA Furniture e TUNA People possuem, respectivamente, 1200 e 1080 expressões de referência do tipo singular. O corpus b5-ref possui 1900 expressões de referência. Estes corpuses foram divididos na proporção de 80% para treinamento e 20% para testes, conforme o mesmo procedimento realizado na etapa de avaliação do Capítulo 6.

7.5 Procedimento

Com base nos conjuntos de treinamento considerados, foram obtidas as listas P e o parâmetro n utilizados como entrada para o algoritmo IAL, bem como foi computada uma lista de preferência fixa para o algoritmo IA, tal como descrito na Seção 6.5.

O procedimento utilizado na comparação entre as expressões geradas pelos algoritmos e as expressões dos corpuses foram os mesmos utilizados na etapa de avaliação do capítulo 6, e novamente foi utilizado o coeficiente Dice para comparação dos resultados.

7.6 Resultados

Na Tabela 4 é apresentado o coeficiente Dice médio obtido como resultado na execução do modelo computacional proposto e também do algoritmo de *baseline*, comparando-se ambos com as expressões dos respectivos corpus de teste.

Tabela 4 – Resultados em outros domínios. Diferenças estatisticamente significativas são destacadas

Domínio	IA		IAL		teste de Wilcoxon		
	Dice	DP	Dice	DP	W	z	p
TUNA Furniture	0,82	0,22	0,95	0,13	-15378	-11.36	<0.0001
TUNA People	0,62	0,22	0,94	0,14	-55725	-15.71	<0.0001
b5-ref	0,59	0,08	0,67	0,07	78	3.04	=0.0012

Com base nos resultados apresentados na Tabela 4, pode-se verificar que o algoritmo IAL apresenta um resultado superior ao do algoritmo IA na tarefa de gerar expressões de referência. Esta diferença é significativa nos três domínios considerados segundo o teste de Wilcoxon sobre coeficientes Dice médios. Estes resultados oferecem suporte à hipótese h4.

7.6.1 Análise complementar: O papel do parâmetro n

Após o teste da hipótese h4, foi realizado um teste adicional com o objetivo de melhor ilustrar o papel do parâmetro n no resultado médio do algoritmo IAL. A título de exemplo, foi realizado um pequeno experimento de geração da descrição do contexto 9 do corpus b5-ref (PARABONI; MONTEIRO; LAN, 2017) que, segundo os dados de treinamento considerados, apresenta um tamanho médio de 3 propriedades para este contexto. O experimento consistiu em executar o algoritmo IAL seis vezes, variando-se o parâmetro n de 1 e 6. Os resultados destas seis execuções são ilustrados na Figura 16, e demonstram que o uso do parâmetro n=3 produz a média de coeficiente Dice mais alta dentre todas alternativas.

Figura 16 – Médias de Dice obtidas pelo algoritmo IAL com a variação do grau de superespecificação

Fonte: Andre Mariotti, 2017

Este experimento, embora limitado a apenas uma situação de referência do corpus b5-ref, demonstra que o controle do grau de superespecificação por meio de um parâmetro único como feito pelo algoritmo IAL é uma forma viável de produzir descrições mais próximas do desempenho humano. Cabe destacar também que este parâmetro não necessariamente precisa ser computado a partir de dados de treinamento extensos, já que pôde ser testado de forma empírica como neste experimento. Além disso, estudos como Koolen, Krahmer e Theune (2012) sugerem que é possível obter listas de preferência para algoritmos deste tipo com base em um número muito reduzido de exemplos de descrições. No caso do domínio TUNA, por exemplo, este limite seria inferior a 20 descrições.

8 Conclusão

Esta monografia apresentou um trabalho de pesquisa de mestrado enfocando o problema de gerar expressões de referência em diferentes contextos de comunicação. O projeto contou com a realização de um experimento piloto que ajudou na investigação dos aspectos da linguagem que se manifestam em situações de comunicação atípicas, como no caso da restrição de tempo. Por conseguinte, foi proposto um novo algoritmo para tratar a tarefa de GER sob essas condições, o qual foi avaliado contra um *baseline* estabelecido. Finalmente, foi sugerida uma generalização do modelo computacional proposto para outros domínios, nos quais não há restrição de tempo, mas podendo haver superespecificação referencial.

Este trabalho é de especial interesse da área de GER, pois os resultados obtidos sugerem que o modelo computacional proposto pode se adaptar ao contexto de comunicação e obter resultados mais próximos ao do comportamento humano. Como principais contribuições, tem-se a proposta do algoritmo IAL, que é capaz de produzir uma superespecificação adaptável ao contexto de comunicação, e o corpus Stars2T de descrições produzidas sob restrição de tempo.

Em trabalhos futuros, é possível estender a presente pesquisa com o objetivo de investigar aspectos mais específicos da Geração de Expressões de Referência em contextos de comunicação atípicos. Isso inclui, por exemplo, situações diferentes das que possuem restrição de tempo, e que podem favorecer diferentes estratégias de seleção de conteúdo, tanto em termos quantitativos como qualitativos.

Referências¹

- ARNOLD, J. E. Reference production: Production-internal and addressee-oriented processes. *language and cognitive processes* 23(4). Tilburg: Tilburg University dissertation, p. 495–527, 2008. Citado na página 23.
- ARNOLD, J. E. . Z. M. G. The effect of additional characters on choice of referring expressions: Everyone competes. *Journal of Memory and Language* 56, Tilburg: Tilburg University dissertation, p. 521—536, 2007. Citado na página 23.
- ARTS, A. et al. Overspecification in written instruction. *Journal of Pragmatics*, Tilburg University, v. 43, 2011. Citado 7 vezes nas páginas 19, 23, 24, 42, 43, 48 e 49.
- BECHARA, E. *Moderna Gramática Portuguesa 37ª edição*. [S.l.: s.n.], 2009. 16–17 p. Citado na página 14.
- 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, p. 431, 2008. Citado na página 14.
- BINSTED K.; CAWSEY, A. J. R. Generating personalised patient information using the medical record. *Artificial Intelligence in Medicine*, Pavia, Italy, v. 934, p. 29, 1995. Citado na página 14.
- BYRON, D. et al. Generating instructions in virtual environments (GIVE): A challenge and evaluation testbed for NLG. In: *Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*. [S.l.: s.n.], 2007. Citado na página 40.
- DALE, R. Cooking up referring expressions. *Proceedings of the 27th annual meeting on Association for Computational Linguistic*, Stroudsburg, PA, USA: Association for Computational Linguistics, p. 68—75, 1989. Citado na página 31.
- DALE, R.; HADDOCK, N. J. Content determination in the generation of referring expressions. *Computational Intelligence*, v. 7, n. 4, p. 252–265, 1991. Citado na página 42.
- DALE, R.; REITER, E. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, v. 19, n. 2, p. 233–263, 1995. Citado 12 vezes nas páginas 19, 30, 32, 33, 34, 36, 37, 41, 49, 51, 54 e 59.
- DALE, R.; VIETHEN, J. Referring expression generation through attribute-based heuristics. In: *Proceedings of ENLG-2009*. [S.l.: s.n.], 2009. p. 58–65. Citado 2 vezes nas páginas 25 e 36.
- DEEMTER, K. van et al. Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, v. 36, p. 799—836, 2011. Citado na página 31.
- DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, v. 26, n. 3, p. 297–302, 1945. Citado 4 vezes nas páginas 30, 55, 56 e 60.
- ENGELHARDT, P. E.; BAILEY, K. G. D.; FERREIRA, F. Do speakers and listeners observe the gricean maxim of quantity? *Journal of Memory and Language* 54, p. 554—573, 2006. Citado 2 vezes nas páginas 23 e 32.

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

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. Citado na página 28.

FERREIRA, T. C.; PARABONI, I. Classification-based referring expression generation. In: *Computational Linguistics and Intelligent Text Processing (CICLing-2014), Lecture Notes in Computer Science 8403*. Kathmandu, Nepal: Springer, 2014. p. 481–491. Citado 3 vezes nas páginas 17, 38 e 39.

FERREIRA, T. C.; PARABONI, I. Generating natural language descriptions using speaker-dependent information. *Natural Language Engineering*, Cambridge University press, p. 1–22, 2017. Citado 4 vezes nas páginas 36, 37, 51 e 59.

GATT, A.; BELZ, A. The attribute selection for GRE challenge: Overview and evaluation results. In: *UCNLG+MT: Language Generation and Machine Translation*. [S.l.: s.n.], 2007. Citado 2 vezes nas páginas 21 e 29.

GATT, A.; BELZ, A.; KOW, E. The TUNA challenge 2008: Overview and evaluation results. In: *Fifth International Natural Language Generation Conference (INLG-2008)*. [S.l.: s.n.], 2008. p. 198–206. Citado na página 29.

GATT, A.; BELZ, A.; KOW, E. The TUNA challenge 2009: Overview and evaluation results. In: *Proceedings of the 12nd European Workshop on Natural Language Generation*. [S.l.: s.n.], 2009. p. 174–182. Citado na página 29.

GATT, A.; SLUIS, I. van der; DEEMTER, K. van. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In: *Proceedings of ENLG-07*. [S.l.: s.n.], 2007. Citado 4 vezes nas páginas 25, 28, 30 e 60.

GOMPEL, R. van et al. Testing computational models of reference generation as models of human language production: The case of size contrast. In: *RefNet Workshop on Psychological and Computational Models of Reference Comprehension and Production*. Edinburgh, Scotland: [s.n.], 2014. Citado na página 43.

GRICE, H. P. Logic and conversation. In: COLE, P.; MORGAN, J. L. (Ed.). *Syntax and semantics*. [S.l.]: New York: Academic Press, 1975. v. 3. Citado 3 vezes nas páginas 19, 31 e 50.

JOHN LAURA P. NAUMANN, C. J. S. O. P. Paradigm shift to the integrative big five trait taxonomy. *Handbook of Personality, Third Edition: Theory and Research*, 2008. Citado na página 29.

KARPATY, A.; FEI-FEI, L. Deep visual-semantic alignments for generating image descriptions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, 2015. p. 3128–3137. Citado na página 14.

KELLEHER, J. D.; COSTELLO, F. J. Applying computational models of spatial prepositions to visually situated dialog. *Computational Linguistics*, v. 35, n. 2, p. 271–306, 2009. Citado na página 40.

- KELLEHER J. D.; KRUIJFF, G.-J. M. Incremental generation of spatial referring expressions in situated dialog. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, p. 1041–1048, 2006. Citado na página 36.
- KITTREDGE, R. e. a. Sublanguage engineering in the fog system. Stroudsburg, PA, USA: Association for Computational Linguistics, p. 215, 1994. Citado na página 14.
- KOLLER, A. et al. Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In: *6th International Natural Language Generation Conference (INLG)*. Dublin: [s.n.], 2010. Citado 2 vezes nas páginas 40 e 42.
- KOOLEN, R. et al. The effect of time pressure on referential overspecification. *Tilburg University*, Aberdeen, United Kingdom, p. 2, 2016. Citado 2 vezes nas páginas 24 e 42.
- KOOLEN, R. et al. Factors causing overspecification in definite descriptions. *Journal of Pragmatics*, Elsevier, v. 43, n. 13, p. 3231–3250, 2011. Citado na página 23.
- KOOLEN, R.; KRAHMER, E.; THEUNE, M. Learning preferences for referring expression generation: Effects of domain, language and algorithm. In: *Proceedings of the Seventh International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. p. 3–11. Citado na página 62.
- KRAHMER, E. Refnet workshop on psychological and computational models of reference comprehension and production. *Edinburgh*, 2014. Citado na página 15.
- KRAHMER, E.; DEEMTER, K. van. Computational generation of referring expressions: A survey. *Computational Linguistics*, v. 38, n. 1, p. 173–218, 2012. Citado 3 vezes nas páginas 16, 17 e 21.
- KRAHMER, E.; ERK, S. van; VERLEG, A. Graph-based generation of referring expressions. *Computational Linguistics*, v. 29, n. 1, p. 53–72, 2003. Citado na página 35.
- KRAHMER, E.; THEUNE, M. Efficient context-sensitive generation of referring expressions. In: DEEMTER, K. van; KIBBLE, R. (Ed.). *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*. Stanford, CA: CSLI Publications, 2002. p. 223–264. Citado na página 36.
- KRAUSS, R. M.; WEINHEIMER, S. Concurrent feedback, confirmation and the encoding of referents in verbal communication. *Journal of Personality and Social Psychology*, v. 4, p. 343–346, 1966. Citado na página 45.
- LUCENA, D. J. de; PARABONI, I.; PEREIRA, D. B. 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. Citado na página 21.
- NEWELL, A. . H. A. S. Human problem solving. englewood cliffs: Prentice-hall. 1972. Citado na página 23.
- 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, 2012. Disponível em: <<http://dx.doi.org/10.1007/s13173-012-0095-1>>. Citado na página 21.

PARABONI, I.; GALINDO, M.; IACOVELLI, D. Stars2: a corpus of object descriptions in a visual domain. *Language Resources and Evaluation*, Springer, v. 51, n. 2, p. 439–462, 2017. Citado 8 vezes nas páginas 25, 26, 27, 43, 45, 46, 47 e 68.

PARABONI, I.; MONTEIRO, D. S.; LAN, A. G. J. Personality-dependent referring expression generation. In: *Text, Speech and Dialogue (TSD-2017)*. Prague, Czech Republic: Springer-Verlag, 2017. Citado 5 vezes nas páginas 25, 29, 30, 60 e 61.

PECHMANN, T. Incremental speech production and referential overspecification. *Linguistics*, v. 27, n. 1, p. 98–110, 1989. Citado 5 vezes nas páginas 19, 23, 26, 32 e 43.

PEREIRA, D. B.; PARABONI, I. Statistical surface realisation of portuguese referring expressions. *Advances in Natural Language Processing*, Springer-Verlag, LNAI 5221, p. 383–392, 2008. Citado 2 vezes nas páginas 14 e 21.

PORTET, F. e. a. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence in Medicine*, Amsterdam, The Netherlands, v. 4594, p. 227, 2007. Citado na página 14.

REITER, E.; DALE, R. *Building natural language generation systems*. New York, NY, USA: Cambridge University Press, 2000. ISBN 0-521-62036-8. Citado 4 vezes nas páginas 14, 16, 20 e 31.

SILVA, D. dos S.; PARABONI, I. Generating spatial referring expressions in interactive 3D worlds. *Spatial Cognition & Computation*, v. 15, n. 03, p. 186–225, 2015. Citado 4 vezes nas páginas 39, 40, 41 e 42.

VIETHEN, H. A. E. *The Generation of Natural Descriptions: Corpus-based Investigations of Referring Expressions in Visual Domains*. Tese (Doutorado) — Macquarie University, 2010. Citado na página 26.

VIETHEN, J.; DALE, R. GRE3D7: A corpus of distinguishing descriptions for objects in visual scenes. In: *Proceedings of UCNLG+Eval-2011*. [S.l.: s.n.], 2011. p. 12–22. Citado 2 vezes nas páginas 25 e 36.

WALKER, M. A. et al. Perceived or not perceived: Film character models for expressive NLG. In: SI, M. et al. (Ed.). *Lecture Notes in Computer Science 7069*. Vancouver, Canada: Springer Verlag, 2011. p. 109–121. Citado na página 14.

WALKER, M. A. et al. Murder in the arboretum: Comparing character models to personality models. In: *Intelligent Narrative Technologies IV, Papers from the 2011 AIIDE Workshop*. Palo Alto, California, USA: The AAAI Press, 2011. Citado na página 14.

XU, K. et al. Show, attend and tell: Neural image caption generation with visual attention. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*. Lille, France: [s.n.], 2015. p. 2048–2057. Citado na página 14.

ZHANG, X.; LAPATA, M. Chinese poetry generation with recurrent neural networks. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 670–680. Citado na página 14.

Anexo A – Imagens do Corpus Stars2

A tabela a seguir relaciona as sequências de estímulos visuais do Stars2 (PARABONI; GALINDO; IACOVELLI, 2017) replicadas no experimento exploratório deste trabalho. Nesta tabela, cada linha representa uma das sequência de imagens do Stars2, enquanto as colunas contém o nome de cada imagem.

Tabela 5 – Sequências de imagens utilizadas no Stars2

sequencia	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
52	05f-t2n	03o-t2n	02o-t2r	07f-t1r	04f-t2n	01o-t2r	06o-t2n	03f-t2n	07o-t1n	08f-t1r	01f-t1r	06f-t1r	04o-t1n	02f-t2n	05o-t1r	08o-t1r
89	05o-t1n	03f-t2r	04o-t1r	02f-t2n	05f-t1r	06f-t1n	06o-t2n	07o-t2r	02o-t2n	01f-t2n	07f-t1r	03o-t2r	01o-t1n	08o-t1r	04f-t2n	08f-t1r
105	03o-t2r	07f-t2n	08f-t1n	04f-t1n	01o-t1r	07o-t2r	02f-t2n	05f-t2n	08o-t1n	03f-t1n	06f-t1r	02o-t2r	01f-t1r	04o-t2r	06o-t2n	05o-t1r
115	08o-t2r	02f-t1n	08f-t1n	01o-t1r	07o-t1n	02o-t2n	06f-t1n	05f-t2r	01f-t2n	03f-t1r	04f-t2n	07f-t1n	05o-t2r	03o-t2r	06o-t2r	04o-t1r
136	05f-t1n	04f-t2n	06o-t1r	03f-t2r	01o-t2r	08f-t1r	03o-t1n	02o-t1r	02f-t1r	07o-t2n	06f-t2n	08o-t1r	07f-t2n	04o-t2r	01f-t1n	05o-t2n
307	02f-t1n	06o-t1r	07o-t2r	07f-t2r	04o-t2r	08f-t1n	04f-t2n	05o-t1n	06f-t2n	01o-t1r	03o-t1r	02o-t2n	03f-t1r	05f-t1n	08o-t2r	01f-t2n
439	06o-t1r	03f-t1r	07o-t2n	04o-t2n	02o-t2n	07f-t1r	04f-t1r	08f-t1r	01o-t1r	02f-t2r	06f-t2n	01f-t2n	05f-t1n	05o-t2n	03o-t2r	08o-t1n
455	04o-t1n	05o-t2n	04f-t2n	02f-t1r	03o-t1r	02o-t2n	01o-t1r	06o-t2n	05f-t2r	07o-t1n	03f-t2n	08f-t1n	06f-t1r	07f-t2r	08o-t2r	01f-t1r
503	03f-t2n	01o-t2r	01f-t1r	08o-t1n	04f-t1n	02o-t1n	07f-t1n	03o-t2r	04o-t2r	05o-t1r	05f-t2n	02f-t2r	06f-t2r	07o-t2n	08f-t1r	06o-t1n
538	01f-t2r	05f-t1r	06o-t1n	05o-t1n	08o-t2r	07f-t2n	03f-t1n	07o-t2r	08f-t2r	04f-t2r	02f-t1r	04o-t1n	03o-t1n	01o-t2n	06f-t1n	02o-t2r
585	07f-t1r	03f-t1r	03o-t2r	08f-t1n	02o-t2n	07o-t1r	04o-t1r	01o-t2n	06f-t2n	08o-t1n	05f-t2r	05o-t2r	01f-t1r	06o-t2n	04f-t1n	02f-t2n
597	06f-t1n	06o-t2n	01o-t1r	04o-t2r	01f-t2n	03f-t2r	07o-t1r	08f-t2r	05f-t1r	02f-t1n	08o-t1n	05o-t1n	07f-t2n	04f-t2n	03o-t1r	02o-t2r
621	03f-t1n	02f-t1r	03o-t1r	05f-t1r	07o-t2r	06f-t2n	01o-t2n	02o-t1r	08f-t1n	01f-t1n	04o-t1r	04f-t2r	06o-t2n	07f-t2n	05o-t2r	08o-t2n
704	03f-t1r	01o-t2n	05f-t2r	07o-t1n	06f-t1r	06o-t2r	08o-t2n	03o-t2r	04f-t1r	08f-t2n	04o-t2r	07f-t1r	05o-t1n	02o-t2n	01f-t1n	02f-t1n
788	06o-t2r	02o-t1n	08o-t2n	04f-t2n	01f-t1n	07o-t2n	01o-t2r	03f-t2r	05o-t1r	08f-t1n	02f-t1r	03o-t2r	05f-t1n	04o-t2n	06f-t1r	07f-t1r
823	05o-t1n	03f-t1r	06f-t2r	03o-t2n	07o-t2n	01f-t1n	08o-t1r	07f-t1n	05f-t2r	06o-t2r	08f-t2n	02o-t2r	04o-t1n	02f-t2r	04f-t1n	01o-t1r
832	03f-t2n	05o-t1n	04f-t2r	08o-t1n	08f-t1r	05f-t2n	07o-t1r	07f-t2r	02f-t2r	01f-t1n	06f-t2r	02o-t1r	06o-t1n	01o-t1r	04o-t2n	03o-t2n
858	03o-t1r	01f-t1r	04f-t2r	08o-t2n	03f-t1n	07f-t2r	06o-t2n	06f-t2n	04o-t1r	05f-t2r	02f-t1n	02o-t2r	01o-t1r	08f-t2n	05o-t1n	07o-t1n
895	01f-t1r	06f-t1n	02o-t1n	08o-t2n	03f-t1r	05o-t1n	04o-t2r	02f-t1n	01o-t2r	06o-t2r	07f-t2r	03o-t1n	08f-t2n	07o-t2r	05f-t2n	04f-t1r
898	02f-t1r	06o-t2r	04o-t2n	03f-t1n	04f-t2n	01o-t1n	08o-t2n	05o-t2r	05f-t2r	03o-t1r	02o-t2r	07o-t1r	06f-t1n	08f-t1n	01f-t2r	07f-t1n
969	01o-t2n	07f-t1r	01f-t2r	05o-t1r	02f-t2n	04o-t1n	08o-t2r	08f-t1n	05f-t1r	03f-t1r	03o-t2n	07o-t1r	02o-t2r	06f-t2n	04f-t2n	06o-t1n
972	08o-t2r	01f-t1r	05o-t2n	08f-t1n	04f-t2r	01o-t1n	02o-t1r	03f-t1n	04o-t2n	03o-t1r	05f-t2n	06f-t1r	07o-t2n	06o-t2r	02f-t1n	07f-t2r
978	04f-t2n	05o-t1r	05f-t2n	07f-t1r	07o-t2r	02o-t1r	03o-t2r	02f-t1r	04o-t1n	01f-t2n	06f-t1n	06o-t1r	01o-t2n	08o-t2r	08f-t1n	03f-t2n
998	04o-t1r	01f-t1n	05f-t2n	04f-t2n	02f-t2n	05o-t2r	03o-t2r	07o-t1r	02o-t1n	07f-t1n	08o-t1n	06f-t1r	06o-t2r	03f-t2r	08f-t2n	01o-t1r

Fonte: Andre Mariotti, 2017