UNIVERSIDADE FEDERAL DE MINAS GERAIS
Faculdade de Letras
Programa de Pós-Graduação em Estudos Linguísticos

André Luiz Rosa Teixeira

DESENVOLVIMENTO DE MÓDULO DE RECURSOS LEXICOGRAMATICAIS BASEADO EM REGRAS PARA REALIZAÇÃO SUPERFICIAL EM TAREFAS DE GERAÇÃO DE LÍNGUA NATURAL EM PORTUGUÊS BRASILEIRO

Belo Horizonte

2022

André Luiz Rosa Teixeira

DESENVOLVIMENTO DE MÓDULO DE RECURSOS LEXICOGRAMATICAIS BASEADO EM REGRAS PARA REALIZAÇÃO SUPERFICIAL EM TAREFAS DE GERAÇÃO DE LÍNGUA NATURAL EM PORTUGUÊS BRASILEIRO

Tese apresentada ao Programa de Pós-graduação em Estudos Linguísticos da Faculdade de Letras da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Linguística Aplicada.
Área de concentração: Linguística Aplicada
Linha de pesquisa: Estudos da Tradução-3B.

Orientadora: Profª. Drª. Adriana Silvina Pagano

Coorientador: Prof. Dr. Thiago Castro Ferreira

Belo Horizonte

2022

T266d Teixeira, André Luiz Rosa.
Desenvolvimento de módulo de recursos lexicogramaticais baseado em regras para realização superficial em tarefas de geração de língua natural em português brasileiro [manuscrito] / André Luiz Rosa Teixeira. – 2022.
1 recurso online (162 p. : il., tabs. (algumas color.) : pdf.

Orientadora: Adriana Silvina Pagano.

Coorientador: Thiago Castro Ferreira.

Área de concentração: Linguística Aplicada.

Linha de Pesquisa: Estudos da Tradução.

Tese (doutorado) – Universidade Federal de Minas Gerais, Faculdade de Letras

Bibliografia: p. 151-154.

Anexos: p. 157-162.

1.Tradução e interpretação – Teses. 2. Funcionalismo (Linguística) – Teses. I. Pagano, Adriana Silvina. II. Ferreira. Thiago Castro. III. Universidade Federal de Minas Gerais. Faculdade de Letras. IV. Título.

CDD : 418.02

Ficha catalográfica elaborada pelo Bibliotecário Israel Jose da Silva – CRB/6-2128

UNIVERSIDADE FEDERAL DE MINAS GERAIS
FACULDADE DE LETRAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTUDOS LINGUÍSTICOS

FOLHA DE APROVAÇÃO

Desenvolvimento de módulo de recursos lexicogramaticais baseado em regras para realização superficial em tarefas de geração de língua natural em português brasileiro

ANDRE LUIZ ROSA TEIXEIRA

Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ESTUDOS LINGUÍSTICOS, como requisito para obtenção do grau de Doutor em ESTUDOS LINGUÍSTICOS, área de concentração LINGUÍSTICA APLICADA, linha de pesquisa Estudos da Tradução.

Aprovada em 17 de fevereiro de 2022, pela banca constituída pelos membros:

Prof(a). Adriana Silvina Pagano - Orientadora
UFMG

Prof(a). Thiago Castro Ferreira - Coorientador
UFMG

Prof(a). Yohan Bonescki Gumiel
PUC-PR

Prof(a). Igor Antônio Lourenço da Silva
UFU

Prof(a). Evandro Landulfo Teixeira Paradela Cunha
UFMG

Prof(a). Kícila Ferreguetti de Oliveira
UFMG

Belo Horizonte, 17 de fevereiro de 2022.

Documento assinado eletronicamente por **Adriana Silvina Pagano**, **Professora do Magistério Superior**, em 21/02/2022, às 11:39, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Evandro Landulfo Teixeira Paradela Cunha**, **Professor do Magistério Superior**, em 21/02/2022, às 11:42, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Igor Antônio Lourenço da Silva**, **Usuário Externo**, em 21/02/2022, às 16:48, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Kicila Ferreguetti de Oliveira**, **Professora Magistério Superior-Substituta**, em 21/02/2022, às 16:59, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Thiago Castro Ferreira**, **Usuário Externo**, em 22/02/2022, às 07:53, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Yohan Bonescki Gumiel**, **Usuário Externo**, em 03/03/2022, às 18:46, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1235150** e o código CRC **9CD66CB7**.

Referência: Processo nº 23072.206818/2022-18 SEI nº 1235150

AGRADECIMENTOS

Esta pesquisa foi realizada com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES).

Agradeço aos Professores e estimados amigos Adriana Silvina Pagano e Thiago Castro Ferreira pela parceria em mais esta empreitada: o aprendizado acumulado em cada encontro com os senhores é inestimável. Agradeço também a todos os colegas do LETRA que, direta ou indiretamente, contribuíram com este trabalho.

Agradeço, ainda, à minha mãe Darci Rosa e irmãs Flávia Rosa e Isabela Rosa por sempre me apoiarem e estarem ao meu lado e servirem de inspiração diária e fonte de força e admiração.

RESUMO

A Geração de Língua Natural (GLN), subárea do Processamento de Língua Natural (PLN), é um tópico que faz parte da agenda, desde o século passado, das Ciências da Computação – Linguística Computacional e, como uma área de pesquisa interdisciplinar por natureza, é abordada por diferentes perspectivas, dentre elas, a Linguística Aplicada. No escopo da Linguística Sistêmico-Funcional (LSF), o campo dos Estudos Multilíngues, proposto por Matthiessen et al. (2008), contempla uma interação entre os Estudos Linguísticos e campos correlatos, como a Linguística Computacional e promovem a integração dos estudos linguísticos aplicados, teóricos e descritivos, ou seja, nos modos **reflexivo**, teorização e descrição visando à comparação entre línguas e **ativo**, visando à aplicação dos resultados alcançados no modo reflexivo, o que enseja a inserção da Linguística Computacional no escopo da LSF. Iniciativas de implementação de recursos lexicogramaticais no âmbito da Geração de Língua Natural remontam ao século passado, e contemplam o desenvolvimento de gramáticas orientadas à geração de língua natural para diversas línguas, dentre elas o inglês, alemão, chinês, espanhol, e português brasileiro. A iniciativa de desenvolvimento de recursos lexicogramaticais para a realização superficial/textual que contempla o português brasileiro ancorada na LSF limita-se aos significados de espacialidade no domínio de textos turísticos e restringe-se aos recursos lexicogramaticais necessários para a construção das orações que compõem o corpus de teste do estudo (OLIVEIRA, 2013), e portanto, não é independente de domínio/tarefa. Esta tese se insere no campo dos Estudos Multilíngues, modelado no escopo da Linguística Sistêmico-Funcional, no modo ativo de investigação e adota a perspectiva da teorização da Linguística Computacional sob a perspectiva dessa teoria linguística. Nesse cenário, esta tese tem como objetivo principal explorar os recursos de Geração de Língua Natural para elucidar processos que dizem respeito à produção de significados, mais especificamente, desenvolver um módulo de realização superficial/textual, baseado em regras e independente de domínio, que contempla a escala de ordens (do morfema à oração) do português brasileiro, para aplicação na tarefa de realização superficial em linhas de produção de sistemas de geração de língua natural. Esta tese tem, ainda, como objetivos secundários: realizar experimentos com testes comparativos de acurácia entre o módulo de recursos lexicogramaticais baseado em regras e resultados das Redes Neurais desenvolvidas no âmbito do projeto CoNLL-SIGMORPHON (COTTERELL et al., 2017; COTTERELL et al., 2018) na tarefa de flexão verbal nos *corpora* de desenvolvimento e teste compilados no âmbito do SIGMORPHON; e realizar experimentos de aplicação das funções do módulo de recursos baseado em regras para a flexão verbal na subtarefa de realização textual na linha de produção de uma instância local do robô-jornalista @DaMataReporter[1]. A programação do realizador textual baseado em regras do português brasileiro independente de domínio foi desenvolvida na linguagem de programação *Python*,

[1] O robô efetivamente em produção está disponível em: https://twitter.com/DaMataReporter

contemplando toda a escala de ordens lexicogramatical do português brasileiro, pautando-se pela perspectiva trinocular: 'de cima', observando-se os significados semânticos realizados no estrato lexicogramatical, tendo como ponto de referência a oração; 'de baixo', observando-se padrões grafológicos do estrato de expressão e como unidades de dada ordem encerram funções na ordem imediatamente superior na escala; e 'ao redor' como os sistemas organizam os significados em cada uma das ordens da escala, com base nas descrições de base Sistêmico-Funcional do português brasileiro disponíveis. O domínio selecionado para a aplicação do realizador superficial baseado em regras desenvolvido nesta tese é o desmatamento da Amazônia Legal no território brasileiro. O @DaMataReporter realiza postagens jornalísticas, apresentando dados sobre o desmatamento na Amazônia Legal, automaticamente, em rede social e faz parte de iniciativas que visam à publicação de dados abertos, disponibilizados por entidades públicas, levando informações sensíveis a amplo público. Esta pesquisa tem potencial de contribuição no âmbito de pesquisas a) **descritivas**: validando as descrições de base Sistêmico-Funcional já desenvolvidas para o português brasileiro; b) **teóricas**: na medida em que testa e valida descrições ancoradas no modelo teórico Sistêmico-Funcional, estabelecendo uma potencial retroalimentação da teoria linguística; c) **aplicadas**: tanto no âmbito de aplicação direta em sistemas de geração de língua natural, a exemplo do @DaMataReporter, quanto no âmbito educacional, oferecendo subsídios para o treinamento de tradutores, suporte na etapa de análise contrastiva de textos em relação de tradução, ensino de língua, descrição e teoria linguística, dentre outras. Dado o objetivo principal da tese, foi possível programar os principais sistemas que organizam: na ordem da palavra – o verbo e o substantivo, e funções para a realização do advérbio e preposições; na ordem do grupo – os principais sistemas que realizam o grupo nominal (taxonomia do Ente, sistemas de DETERMINAÇÃO, CLASSIFICAÇÃO, QUALIFICAÇÃO, e o grupo verbal (TIPO DE EVENTO, AGÊNCIA, FINITUDE, TEMPO SECUNDÁRIO, ASPECTO VERBAL, e DÊIXIS MODAL), bem como funções para realização de frase preposicional e grupo adverbial; na ordem da oração – os principais sistemas que organizam a oração (TRANSITIVIDADE, MODO: modelagem parcial/preliminar, seleção de escolhas mais prototípicas – modo declarativo e interrogativo polar; TEMA: modelagem parcial, restrito à escolha de tema_default e alguns casos de tema_proeminente_papel_transitivo_participante). Os resultados mostraram que a modelagem de recursos lexicogramaticais para a realização superficial do português brasileiro, sob uma perspectiva Sistêmico-Funcional, e baseada em regras, pode ser uma alternativa produtiva a longo prazo, pois possibilita maior controle nesta fase do processo de geração, especialmente em sistemas de geração que não tenham arquiteturas *end-to-end*.

Palavras-chave:Processamento de Língua Natural (PLN). Geração de Língua Natural (GLN). Estudos Multilíngues. Linguística Sistêmico-Funcional.

ABSTRACT

Natural Language Generation (NLG), a sub-area of Natural Language Processing (NLP), is a research area that has been on the agenda of both Computer Sciences and Linguistics for nearly a century. As an area of interdisciplinary research by nature, NLP draws on different disciplines, one of which is Applied Linguistics. Drawing on Systemic-Functional Linguistics, the area of Multilingual Studies as proposed by Matthiessen et al. (2008) contemplates an integration of Linguistics and related fields of investigation, such as Computational Linguistics. This field of investigation promotes the articulation of modes of integration: **reflexive** – theorizing and description of language production aiming at the contrast between languages and **active** – aiming at the application of the findings in the reflexive mode (such as the development of NLG programs), allowing the investigation of Computational Linguistics within the scope of SFL. The programming of lexicogramatical resources from different languages, such as English, German, Chinese, Spanish, and Brazilian Portuguese, for Natural Language Generation, dates back to the last century. Drawing on SFL, in Brazilian Portuguese, one initiative is available that models the spatial language in the domain of tourist texts, and models the lexicogramatical resources necessary for the realization of the clauses in the corpus of the study (see Oliveira (2013)). The resources developed in Brazilian Portuguese, are, thus, not domain independent. This thesis draws on Computational Linguistics within a Systemic-Functional Linguistics framework and Multilingual Studies (active mode of investigation) to explore Natural Language Generation as a resource to investigate meaning production processes. More specifically, this thesis aims primarily at developing a rule-based domain independent textual realization module that covers the lexicogramatical rank scale of Brazilian Portuguese, for applications in NLG. Furthermore, this thesis also aims to carry out experiments that contrast the acuracy of the ruled based system and artificial neural networks developed within CoNLL-SIGMORPHON (COTTERELL et al., 2017; COTTERELL et al., 2018) shared task for verbal inflection in domain independent dev and test corpora of verbs; also, to apply the rule based module for the sub-task of textual realization of verbs in the pipeline of a local instance of the robot journalist @DaMataReporter. The programming of the lexicogramatical resources for textual realization of Brazilian Portuguese was carried out in Python programming language. The development made use of a trinocular perspective: "from above"- examining semantic figures realized by clauses in the lexicogramatical stratum; "from below" - examining graphological patterns in the stratum of expression and modeling the constituency patterns along the rank scale, whereby units of a given order function in the order immediately above on the scale; and "from roundabout" - modeling the systems that organize the meanings in each order of the rank scale. The programming of the resources for superficial realization drew on available Systemic-Functional descriptions of Brazilian Portuguese, and on relatively congruent systems of

English when such descriptions were not available. The domain selected for the application of the rule-based module developed in this thesis is the deforestation of the Legal Amazon in the Brazilian territory, a sensitive and pressing matter broadly discussed internationally. @DaMataReporter, a robot-journalist which posts data on the deforestation Of the Amazon, is part of initiatives that aim at generating text from publicly available data, to raise awareness about sensitive matters. This research has the potential to contribute to a) descriptive research: validating the Systemic-Functional descriptions of systems in Brazilian Portuguese; b) theoretical research: insofar as it tests and validates descriptions that draw on Systemic-Functional theory; c) applied research: in the educational field – offering results that inform translator training; basis for contrastive analysis of texts in contact through translation; language teaching; description and linguistic theory. This research enabled the computational implementation of the main systems that organize units in the lexicogrammar of Brazilian Portuguese: at word rank – the verb and the noun, and functions for the realization of adverb and prepositions; at group rank, the main systems that organize the nominal group: the taxonomy that organize the *Ente* (Thing), and the systems of *DETERMINAÇÃO* (DETERMINATION), *CLASSIFICAÇÃO* (CLASSIFICATION), *QUALIFICAÇÃO* (QUALIFICATION) and *QUANTIFICAÇÃO* (QUANTIFICATION); and the verbal group: *TIPO DE EVENTO* (EVENT TYPE), *AGÊNCIA* (AGENCY), *FINITUDE* (FINETENESS), *TEMPO SECUNDÁRIO* (SECONDARY TENSE), *ASPECTO VERBAL* (ASPECT), and *DÊIXIS MODAL* (MODAL DEIXIS), as well as functions for the construction of prepositional phrases and adverbial groups; at clause rank, the main systems: *TRANSITIVIDADE* (TRANSITIVITY); *MODO* (MOOD: partially implemented – declarative and polar interrogative options); *TEMA* (THEME: partially implemented – options for *tema_default* (default theme)). Results showed that rule-based development of lexicogramatical resources for domain independent textual realization of Brazilian Portuguese, in the scope of Systemic-Functional theorization of Computational Linguistics, can be a long-term productive alternative, as it allows a greater control at this stage of the language generation process in the pipeline, specially in the application in sensitive domains.

Keywords: Natural Language Processing (NLP). Natural Language Generation (NLG). Multilingual Studies. Systemic-Functional Linguistics (SFL).

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

LSF – Linguística Sistêmico-Funcional

PLN – Processamento de Língua Natural

GLN – Geração de Língua Natural

SUMÁRIO

1 INTRODUÇÃO

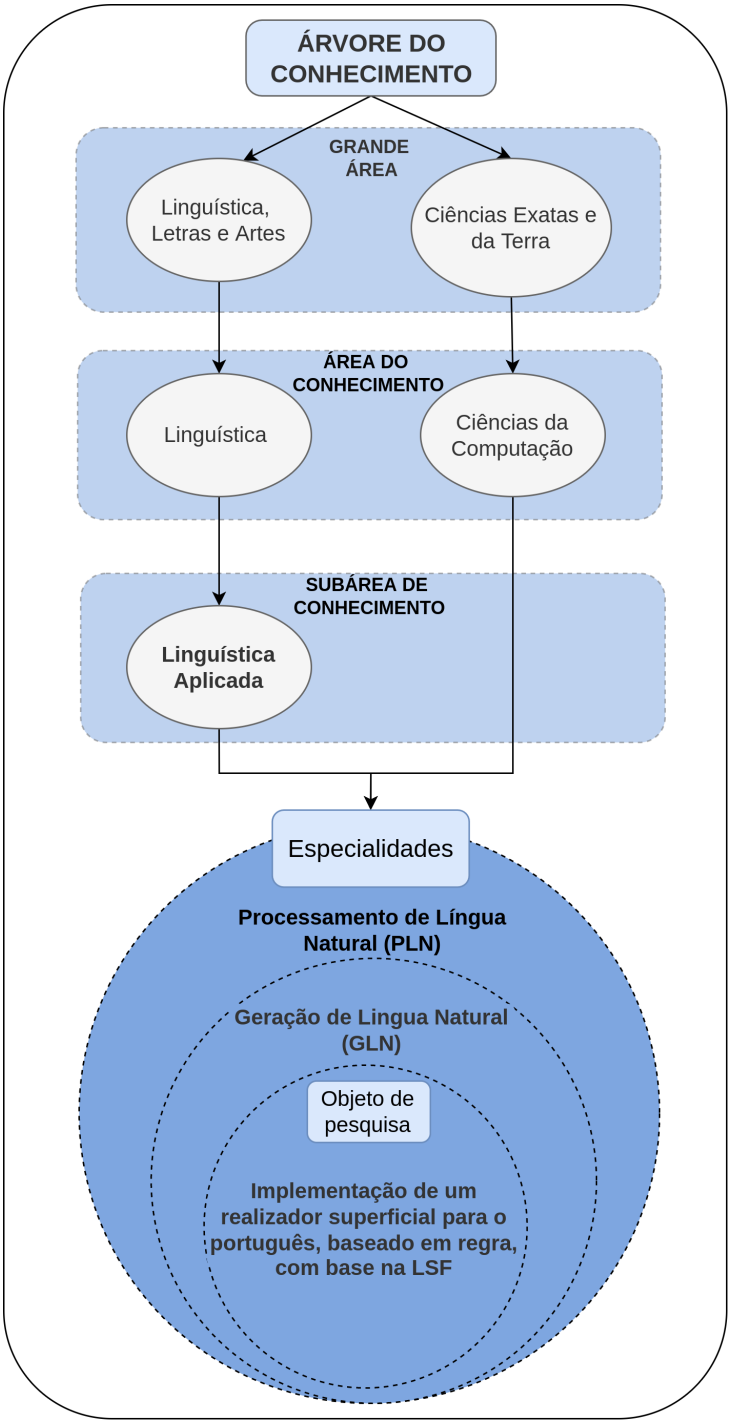
A área de Processamento de Língua Natural (PLN) busca a elaboração de sistemas computacionais para, por um lado, a Interpretação de Língua Natural e, por outro, a Geração de Língua Natural (GLN), ou seja, a produção de textos em línguas naturais, inteligíveis por humanos. Mais especificamente, a Geração de Língua Natural é a produção automática de enunciados em língua natural, a partir da entrada de alguma representação de informação inteligível por computadores – dados; texto; ou outras formas abstratas de representação de informações (TEICH, 1999a; REITER; DALE, 2000).

Nas últimas décadas, são observados avanços na implementação de sistemas computacionais para o Processamento de Língua Natural que auxiliam tarefas das mais variadas. As aplicações desenvolvidas variam de assistentes computacionais inteligentes (e.g., *Google Now, Cortana, Siri, Alexa* etc.), sistemas multilíngues de geração (e.g., *KPML, Multex*), interação homem-máquina na tradução automática (através de algoritmos de aprendizado de máquina), análise e revisão de textos (eg., *Grammarly, Microsoft Word* etc.), implementação de sistemas conversacionais (*chatbots*) (e.g., o *Chatfuel, Insomnobot 3000* da *Casper* etc.) e sua popularização através de plataformas para sua construção (e.g., *Dialogflow*), desenvolvimentos de robôs assistentes domésticos humanoides (o robô *Pepper*, criado no Japão pela *SoftBank*, em 2014), até plataformas de computação cognitiva (*IBM Watson*), dentre outras.

Como uma área de pesquisa interdisciplinar, o Processamento de Língua Natural articula conhecimentos provenientes de várias áreas, dentre as quais, a Computação e a Linguística. Os avanços das últimas décadas nessas duas áreas do conhecimento introduzem cada vez mais sucesso nas tarefas de interpretação e geração de língua natural. A Figura 1 destaca o caráter interdisciplinar desta pesquisa e enseja a localização do objeto de estudo no mapa amplo de áreas do conhecimento.

Amparados pelo modelo multidimensional proposto pela Linguística Sistêmico-Funcional, Matthiessen et al. (2008) propõem um campo disciplinar, os Estudos Multilíngues, que postula dois modos, potencialmente integrados, dos estudos da linguagem: o **reflexivo** – padrões que permitem a descrição e comparação entre línguas; e o **ativo** – promoção da aplicação dos resultados alcançados no modo reflexivo em problemas do 'mundo real'. A Linguística Sistêmico-Funcional (LSF), uma teoria linguística abrangente, com potencial de aplicação, que oferece um modelo de linguagem organizado em dimensões articuladas com o contexto, pode oferecer contribuições para a resolução de problemas inerentes às tarefas da área de PLN. A LSF é uma teoria linguística que reforça, desde sua origem, a integração da Linguística com campos como os Estudos da Tradução, Estudos Descritivos da Linguagem, e campos correlatos como a Linguística Computacional,

Figura 1 – Localização do objeto de estudo no mapa de Áreas do conhecimento

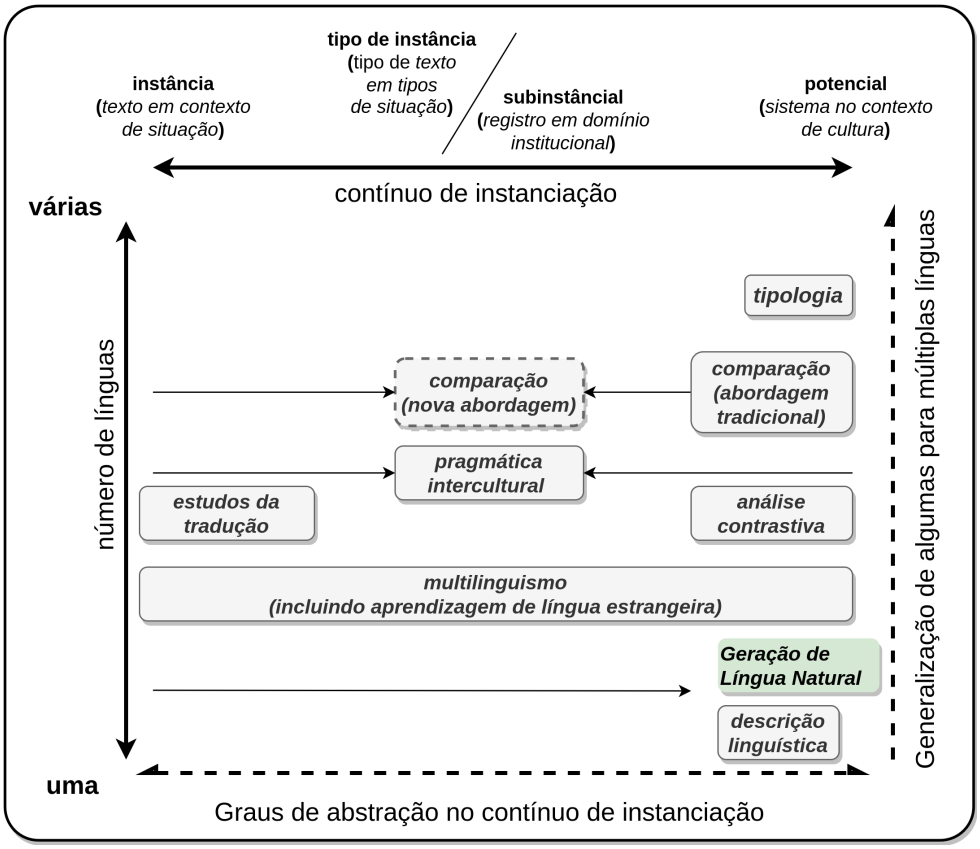


Fonte: desenvolvida com base na tabela de Áreas de Conhecimento da CAPES disponível em <shorturl.at/cejC0>.

Linguística de Corpus etc. e busca acomodar a modelagem do Processamento de Língua Natural, reivindicando o papel da Linguística Aplicada no desenvolvimento desta subárea de pesquisa.

A Figura 2 apresenta a topologia do campo dos Estudos Multilíngues, ensejando a localização da subárea de Geração de Língua Natural nesse mapa semiótico em relação às áreas correlatas, especialmente à descrição linguística, localizando-se em direção ao polo potencial do contínuo de instanciação e envolvendo uma língua.

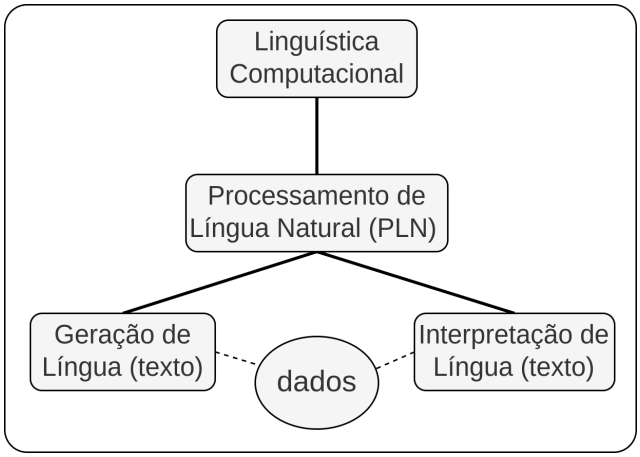
Figura 2 – Estudos multilíngues: contínuo de instanciação e número de línguas



Fonte: adaptado de Matthiessen et al. (2008, p. 149)

Aderindo-se à Linguística Sistêmico-Funcional, uma teoria linguística abrangente e com potencial de aplicação, e na condição de uma pesquisa na área de Linguística Aplicada, esta tese ancora-se na perspectiva da Linguística Computacional no escopo dessa teoria linguística, localizando-se na conceitualização proposta por Halliday e Webster (2009, p. 250). A Figura 3 apresenta um esboço inicial da Linguística Computacional sob perspectiva da LSF, que acomoda o Processamento de Língua Natural (PLN) - Geração de Língua (GLN), tarefa a qual é objeto de estudos desta tese.

Figura 3 – Linguística Computacional no escolo da LSF



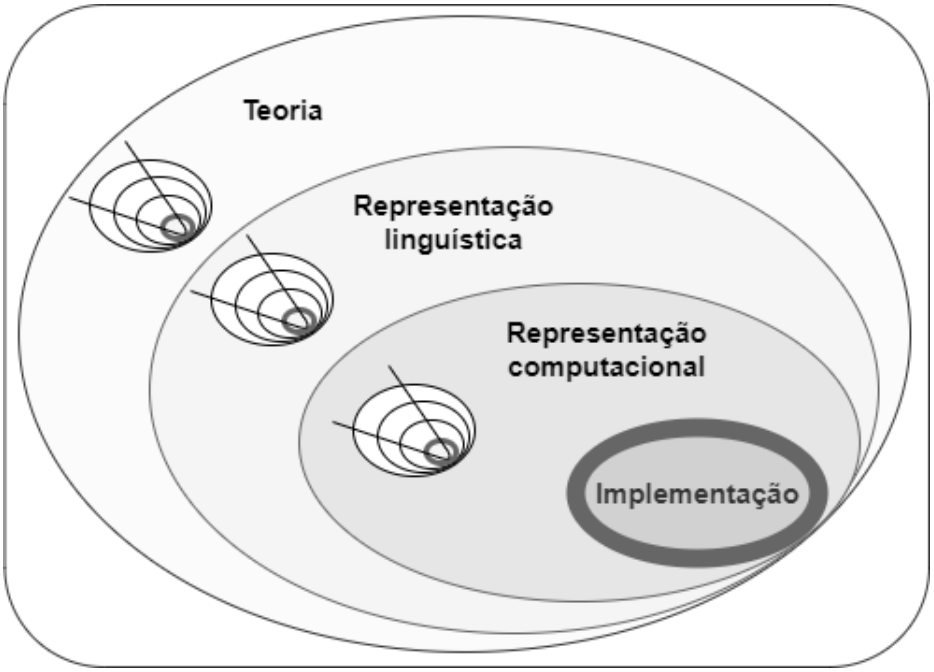
Fonte: traduzido e adaptado de Halliday e Webster (2009, p. 250).

Teich (1999a) apresenta o ambiente metalinguístico no qual se ancora a modelagem computacional de recursos linguísticos, em que se insere esta pesquisa, circunscrevendo a teoria linguística, a descrição e a implementação de recursos computacionais, organizando-o em uma escala de ordens organizada pelo princípio de realização. A **teoria linguística** (Sistêmico-Funcional) é realizada pelas **descrições**, por sua vez, realizadas pela **representação computacional**, que é realizada pela **implementação**. Esse ambiente metalinguístico enseja a localização ainda mais precisa desta pesquisa, a saber, no estrato de implementação dos recursos linguísticos, realizando os estratos imediatamente superiores na escala. A Figura 4 dispõe os estratos de Teoria, Representação linguística, Representação computacional e de **Implementação**, em um relação de **realização**. A modelagem do ambiente metalinguístico apresentado por Teich consolida, no ambiente semiótico, os modos **reflexivo** e **ativo**, promovendo sua integração por meio do princípio de realização. A Figura 4 apresenta esse ambiente semiótico que integra Teoria (modo reflexivo) e Implementação computacional para geração de língua natural (modo ativo).

A integração dos modos reflexivo e ativo dos estudos sobre a linguagem, ou seja, a modelagem computacional dos recursos lexicogramaticais descritos e sua aplicação para a geração de língua natural demanda subsídios provenientes dos estudos descritivos do português brasileiro e, no sentido delineado anteriormente, funcionam como teste e contribuem tanto com os estudos descritivos quanto teóricos, de base sistêmico-funcional. Em outras palavras, a prova de uma teoria linguística está na sua capacidade de explicar o funcionamento da língua e possibilitar a descrição de línguas particulares; a prova de uma descrição linguística está na capacidade potencial de modelagem de sistemas para gerar língua.

Iniciativas de implementação de recursos computacionais para a Geração de Língua Natural, amparadas no modelo Sistêmico-Funcional, remontam ao século passado, tendo

Figura 4 – Estratificação da teorização da Linguística Computacional no escopo da Linguística Sistêmico-Funcional



Fonte: Traduzido e adaptado de Teich (1999a).

início com a implementação da gramática do inglês (*NIGEL*) na tradição de desenvolvimento do sistema de geração de sentenças na língua inglesa PENMAN (MATTHIESSEN; MANN, 1985) - e seu sucessor, *KPML (KOMET-Penman MultiLingual)*, uma plataforma para engenharia linguística de larga escala e orientada à geração multilíngue (BATEMAN et al., 1991; BATEMAN, 1996; BATEMAN, 1997; MATTHIESSEN et al., 1998; BATEMAN et al., 1999; BATEMAN; ZOCK, 2012).

Ancorada no escopo dessa teoria linguística, com subsídios aplicados na área de Geração de Língua Natural, apenas uma iniciativa de implementação de recursos gramaticais para a geração com o português brasileiro foi desenvolvida. Oliveira (2013) aplica, com sucesso, a *GUM (Generalized Upper Model)* - uma ontologia superior, orientada a tarefas gerais e independente de domínio, motivada linguisticamente, empregada no processamento de língua natural - para a modelagem de domínio experiencial, de significados espaciais em textos turísticos, em português brasileiro. Em paralelo, o autor desenvolveu os recursos lexicogramaticais parciais do português brasileiro para a geração de língua natural no sistema de geração *KPML (KOMET-Penman MultiLingual)*. Como Oliveira (2013) aponta, seu objetivo se restringiu a testar a aplicabilidade da ontologia superior (*GUM*) como sistema para a modelagem da representação de conhecimento (base de ideação–*ideation base*) para a geração de língua natural, com o português brasileiro. Nesse sentido, algumas alterações na estrutura da ontologia foram implementadas pelo autor para contemplar o domínio explorado em seu estudo: textos rotulados comumente como **'textos**

turísticos'. Contudo, no que tange à implementação dos outros recursos lexicogramaticais para a geração, isto é, base de interação (*interaction base*) e base textual (*text base*), Oliveira se limitou a um aspecto específico da produção de significado, a espacialidade nos 'textos turísticos', ou seja, não desenvolveu um modelo que pudesse gerar sentenças em outros domínios, tomando a decisão metodológica de modelar os recursos necessários para a geração das orações do seu *corpus*.

É neste cenário mais amplo, no campo dos Estudos Multilíngues e da teorização da Linguística Computacional, no escopo da Linguística Sistêmico-Funcional, sob o modo ativo de investigação, que a proposta de trabalho apresentada nesta tese se insere. O **objetivo principal** desta pesquisa é explorar os recursos de Geração de Língua Natural visando a elucidar processos que dizem respeito à produção de significados no português brasileiro, através da implementação de um módulo de recursos lexicogramaticais, baseado em regras e independente de domínio, contemplando a escala de ordens da lexicogramática do português brasileiro (do morfema à oração), para a execução da subtarefa de **realização textual/linguística**[1] (*linguistic realization*) na linha de produção de aplicações de geração de língua natural.

Como já delineado, a pesquisa localiza-se na interface entre a Linguística Aplicada e a Linguística Computacional, mais especificamente, tomando um posicionamento da Linguística Computacional no escopo da Linguística Sistêmico-Funcional. Por ser uma pesquisa que parte da Linguística Aplicada, optou-se pelo desenvolvimento de um modelo baseado em regras, ou seja, a realização superficial é baseada em funções desenvolvidas manualmente, partindo da unidade mais baixa na escala, o morfema, até a unidade mais superior na escala de ordens, a oração. Uma abordagem baseada em regras vai em direção contrária à tendência de abordagens estado da arte, baseadas redes neurais. Este posicionamento justifica-se no fato de que a pesquisa parte, de maneira geral, da Linguística Aplicada, e mais especificamente, posicionando-se no arcabouço da teoria Linguística Sistêmico-Funcional. Nesse sentido, uma abordagem baseada em regras permite o teste de aplicação da teoria no desenvolvimento de sistemas de geração de língua, ensejando o teste do arcabouço teórico. Ademais, sob a perspectiva de aplicação do realizador baseado em regras para a realização superficial em domínios sensíveis, como será explicado mais detidamente, uma abordagem baseada em regras permite maior controle das tarefas.

No escopo do objetivo principal, esta tese tem como **objetivos secundários**: avaliar o módulo de realização superficial baseado em regras, por meio de experimentos de acurácia, comparando os resultados obtidos pelo realizador baseado em regras a resultados obtidos por modelos baseados em redes neurais, na tarefa de flexão verbal nos *corpora* de

[1] Os termos realização textual e realização linguística serão utilizados intercambiavelmente ao longo desta tese. Ambos dizem respeito à mesma subtarefa do submódulo de realização superficial de sistemas de geração.

desenvolvimento e teste compilados no âmbito do projeto de tarefas compartilhadas do CoNLL-SIGMORPHON (COTTERELL et al., 2017; COTTERELL et al., 2018). Note que redes neurais não foram desenvolvidas no âmbito desta pesquisa. Os resultados do realizador baseado em regras são comparados aos resultados alcançados pelos modelos de redes neurais mais bem colocados na execução da tarefa de flexão verbal, para o português brasileiro, no âmbito do projeto de tarefas compartilhadas do CoNLL-SIGMORPHON (edições de 2017 e 2018); avaliar, por meio de experimentos, a aplicação das funções do módulo de recursos lexicogramaticais baseado em regras, especialmente as funções desenvolvidas para a realização dos verbos (que realizam as flexões verbais) e grupo verbal, em verbos extraídos do banco lexical do robô-jornalista (incluindo verbos similares aos elementos do léxico[2]) e na subtarefa de realização textual na linha de produção de uma instância local do robô-jornalista @DaMataReporter.

A realização textual/linguística é uma subtarefa do submódulo de realização superficial em sistemas de GLN, responsável, de maneira geral, por mapear configurações abstratas de informações em construções gramaticais, organizar as estruturas de acordo com as organizações nos sistemas, inserir palavras com funções gramaticais e realizar as flexões onde se aplicam.

Cabe ressaltar aqui uma distinção necessária entre o desenvolvimento do arsenal potencial de recursos para a produção de significado, ou seja o realizador baseado em regras e **independente de domínio** e a sua aplicação no domínio específico - textos rotulados como 'notícia jornalística' sobre o desmatamento da Amazônia. Por um lado, os recursos lexicogramaticais implementados computacionalmente localizam-se em direção ao polo potencial do contínuo de instanciação (ver Figura 10). Em outras palavras, o desenvolvimento do realizador superficial toma a organização sistêmica das unidades, e portanto, tem potencial de ser aplicado em qualquer domínio. Por outro lado, as funções desenvolvidas são testadas para realização de material textual de um domínio e tipo de texto específicos - notícias jornalísticas com dados sobre o desmatamento da Amazônia - localizando-se mais intermediariamente no contínuo de instanciação (ver Figura 10). Assim, é necessário distinguir entre os aspectos metodológicos de desenvolvimento do realizador baseado em regras ancorado no arcabouço teórico e descrições de base sistêmico-funcional e o aspecto de aplicação do realizador em domínios específicos. Nesse sentido, esta pesquisa oferece subsídios mais completos dos que os apresentados por Oliveira (2013), na medida em que apresentam recursos que não se restringem a um domínio específico, ou seja, o realizador baseado em regras dispõe de recursos independentes de domínio.

Destaca-se, ainda, o carácter metodológico inerente a esta pesquisa. Como será abordado mais detidamente no Capítulo 2, existe uma forte ancoragem do desenvolvimento

[2] Os verbos similares são extraídos do vocabulário do modelo de vetores de palavras *Gensim*, através de sua função nativa, a '*most_similar*'. Essa função retorna as entradas consideradas mais semelhantes, dada uma entrada de interesse.

do realizador baseado em regras na arquitetura semiótica e organização da Linguística Sistêmico-Funcional. As funções desenvolvidas seguem a lógica de organização em sistemas, de maneira que os conceitos básicos balizam a implementação computacional. Nesse sentido, o principal resultado desta pesquisa é o sistema desenvolvido - o realizador baseado em regras - propriamente dito. Os experimentos realizados servem de mecanismo de melhoria e validação das funções e testes de acurácia das funções na tarefa de realização superficial.

A relevância deste trabalho reside em seu potencial de contribuição no âmbito teórico e descritivo, no escopo da Linguística Sistêmico-Funcional, bem como no âmbito de potencial de aplicação. Sob a perspectiva de modo reflexivo no ambiente multilíngue, os estudos descritivos se ancoram em categorias teóricas que são gerais a todas as línguas, e evoluem de acordo com a construção do modelo abstrato da linguagem como sistema semiótico superior e em metodologias para a descrição de sistemas ainda não descritos. (HALLIDAY, 2002; HALLIDAY, 2003).

A adoção de uma perspectiva científica do estudo dos fenômenos humanos significa, em primeira instância, dar à linguagem um status de inerente centralidade. Em outras palavras, estudos teóricos sobre os fenômenos humanos são construtos construídos por meio da linguagem e que têm os humanos, produtores de significados, como o objeto de estudos. Em segunda instância, ao tomar-se a língua, um sistema semiótico, como objeto de investigação científica, constitui-se uma metalinguagem, ou seja, um construto teórico construído na linguagem e sobre a linguagem. Nesse sentido, uma metalinguagem é um sistema semiótico, podendo ser modelado seguindo os mesmos princípios de modelagem da linguagem como sistema semiótico. Assim, a implementação computacional para a geração de língua natural fecha o ciclo metalinguístico (e metateórico) apresentado pela autora. Note-se, na medida em que as implementações são baseadas na representação computacional, que por sua vez realizam a representação (meta)linguística e a (meta)teórica, quando bem-sucedidas, testam e validam tanto as descrições linguísticas, quanto a teoria que as subjaz.

Destacam-se, para a implementação das funções computacionais desenvolvidas nesta tese, as contribuições descritivas do português brasileiro desenvolvidas por Figueredo (2007, 2011) com as descrições da estrutura do grupo nominal e do perfil metafuncional do português brasileiro; Sá (2016), que oferece uma descrição do verbo e do grupo verbal, Ferreguetti (2018), com um estudo sobre a frase preposicional em textos em relação de tradução, Paula (2018), apresenta a descrição do processo verbal e Braga (2016), contribuições sobre as circunstâncias no português brasileiro.

Além da contribuição para os estudos descritivos e teóricos nos termos delineados anteriormente, esta pesquisa tem potencial no âmbito aplicado: educacional, oferecendo subsídios que podem ser empregados para o treinamento de tradutores, oferecendo suporte na etapa de análise contrastiva de textos em relação de tradução, ensino de língua,

ensino sobre descrição e teoria linguística (de base sistêmica); no âmbito da interação homem-máquina, oferece subsídios potenciais para implementação em sistemas de tradução automática em domínios específicos, aplicação em sistemas de geração de língua natural, no submódulo de realização superficial–textual.

Para além das contribuições delineadas nos parágrafos anteriores, este trabalho se soma às iniciativas que visam à divulgação dos dados abertos disponíveis de uma forma mais inteligível por humanos. Nesse sentido, tem-se em perspectiva a geração de textos mais diversificados, em domínios experienciais de grande interesse do grande público, como é o exemplo do domínio experiencial em que se insere o robô-jornalista @DaMataReporter, qual seja, o desmatamento da Amazônia Legal em território brasileiro. O desmatamento da Amazônia é um domínio sensível e que vem recebendo crescente atenção internacional, devido ao aumento das taxas de desmatamento e em face às novas diretrizes e necessidades de uma economia cada vez mais preocupada com a sustentabilidade, e graças à disponibilização dos dados pelas instituições públicas brasileiras.

Para a consecução dos objetivos apresentados, a pesquisa foi conduzida em etapas metodológicas inter-relacionadas: desenvolvimento (programação) dos recursos lexicogramaticais baseados em regras para a realização textual (linguística) do português brasileiro, independente de domínio; experimento de desenvolvimento, envolvendo testes de acurácia e aplicação e aprimoramento do módulo de recursos lexicogramaticais para realização linguística; experimentos de validação, visando à comparação dos resultados de acurácia do modelo baseado em regras e redes neurais desenvolvidas no âmbito do projeto CoNLL-SIGMORPHON[3], na tarefa de flexão verbal; e teste de aplicação do realizador textual baseado em regras na linha de produção, em uma instância local do robô-jornalista @DaMataReporter.

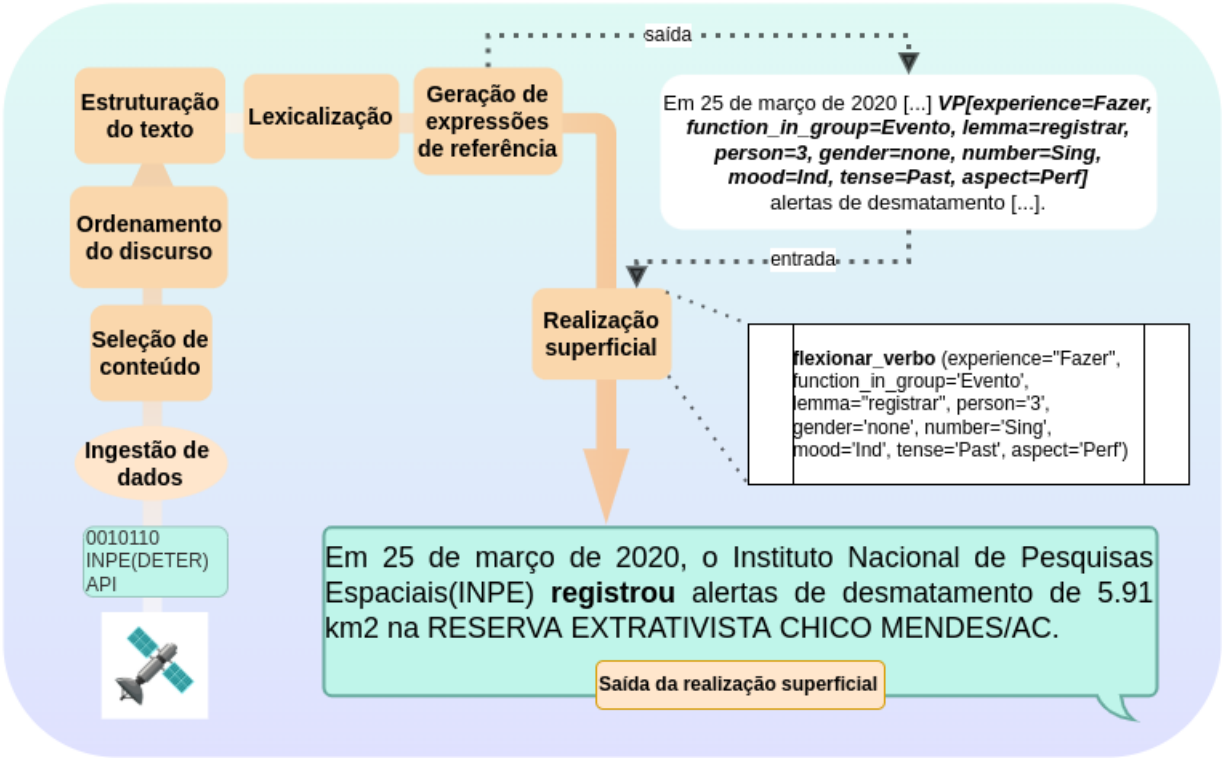
O submódulo de realização textual foi programado em *Python* e contempla toda a escala de ordens lexicogramatical do português brasileiro, guiados pela perspectiva trinocular 'de cima' – observando-se as figuras semânticas realizadas pela oração no estrato lexicogramatical; 'de baixo' – observando-se padrões grafológicos do estrato de expressão e como unidades de dada ordem encerram funções na ordem imediatamente superior na escala; e 'ao redor' – observando como os sistemas organizam os significados em cada uma das ordens da escala, com base nas descrições de base Sistêmico-Funcional do português brasileiro disponíveis (nos casos de sistemas do português ainda não descritos, recorre-se às descrições de sistemas considerados análogos, e.g., do inglês, num nível menor de delicadeza).

A Figura 5 apresenta o ponto em que se localiza, no pipeline do robô-jornalista, a função de flexão verbal, no submódulo de Realização superficial. Como destacado, a função de flexão verbal do modelo baseado em regras toma como entrada a saída do

[3] Ver Cotterell et al. (2017), Cotterell et al. (2018)

submódulo de Geração de expressões de referência, que contém a *tag* **VP[<parâmetros para flexão>]** e repassa os valores à função de flexão verbal para a realização superficial do verbo flexionado.

Figura 5 – Localização da função de realização verbal na linha de produção do robô-jornalista



Fonte: elaborada para fins desta pesquisa

Esta tese estrutura-se da seguinte maneira. Após esta Introdução, no Capítulo 2, na seção 2.1 será apresentado o campo dos Estudos Multilíngues, e como a modelagem computacional dos recursos lexicogramaticais se insere nesse ambiente. Em seguida, na seção 2.2 serão apresentados os fundamentos básicos da Linguística Sistêmico-Funcional: a perspectiva linguística da cognição (subseção 2.2.1); conceitos teóricos básicos da LSF e arquitetura geral da linguagem, destacando-se as dimensões da linguagem (subseção 2.2.2); serão destacados os critérios teóricos que balizam a programação do módulo de realização textual. Então, na seção 2.3 serão apontados os conceitos básicos sobre a Geração de Língua Natural, arquitetura tradicional de sistemas de geração e a arquitetura do robô-jornalista @DaMataReporter, que é ancorada nessa arquitetura tradicional (subseção 2.3.1). No Capítulo 3, serão apresentados os métodos e ferramentas utilizados para a programação do realizador textual/linguístico, independente de domínio, para o português brasileiro. Nesse sentido, será apresentado como os conceitos básicos da teoria linguística balizaram o desenvolvimento das função do realizador baseado em regras (estabelecendo assim uma conexão com a Fundamentação teórica) (seção 3.1); serão apresentados os experimentos de

desenvolvimento - ou experimentos dev - (de acurácia – realizado no *corpus* de desenvolvimento do projeto de tarefa compartilhada desenvolvido no CoNLL-SIGMORPHON-2017, e de aplicação – realizado nos verbos extraídos do corpus do @DaMataReporter e as palavras similares extraídas do *Gensim*) (subseção 3.2.1) e os experimentos de validação - ou experimentos teste - (de acurácia – no *corpus* de teste do CoNLL-SIGMORPHON-2018, e de aplicação – com aplicação das funções de flexão verbal do realizador textual baseado em regras na linha de produção de uma instância local do robô-jornalista @DaMataReporter) (subseção 3.2.2). No Capítulo 4, serão apresentados os resultados dos experimentos de desenvolvimento e dos experimentos de validação realizados e o funcionamento do realizador textual baseado em regras. Por fim, no Capítulo 5, serão apresentadas as considerações finais e sintetização dos resultados.

2 FUNDAMENTAÇÃO TEÓRICA

A integração entre a Linguística (de base Sistêmico-Funcional), e a Geração de Língua Natural, que subjaz este trabalho, parte do potencial de interação e aproximação entre campos correlatos, como argumentam Matthiessen et al. (2008). No escopo do campo que propõem, os Estudos Multilíngues, os autores modelam o campo fenomênico em termos de **reflexão (teoria)**, e **ação (aplicação)**, o primeiro orientado às investigações sobre a organização geral dos sistemas linguísticos (no contínuo sistema-texto) e o contraste entre eles; o segundo voltado às aplicações dos subsídios resultantes das reflexões teóricas no desenvolvimento e implementação de sistemas de geração de língua natural, de programas de tradução automática, treinamento de tradutores, ensino de língua, dentre outros.

As duas perspectivas estabelecem um ciclo, de maneira que os subsídios teóricos alimentam as iniciativas de aplicação, e os subsídios das iniciativas de aplicação realimentam a teoria. Sob essa perspectiva de aplicação, como apontam Matthiessen et al. (2008), destaca-se o potencial de aplicação da modelagem dos sistemas linguísticos voltada à geração de língua natural como uma base para a investigações de como os seres humanos aprendem a produzir significados linguísticos em um idioma diferente de sua língua materna (podendo auxiliar no ensino/aprendizado de idiomas). Dentro do escopo de aplicação, ainda destaca-se o potencial de aplicação de subsídios de modelagem de sistemas linguísticos para a geração de língua natural em contextos de treinamento de tradutores, implementação de ferramentas de auxílio à tradução, sistemas de tomada de decisão por profissionais da área da saúde etc. Salienta-se, ainda, a aplicação em linhas de produção de robô-jornalismo, como o caso de aplicação que será explorado nesta tese.

A seção 2.1 apresenta os Estudos Multilíngues, embasando as discussões com relação ao potencial de colaboração e integração entre a Linguística e as variadas áreas correlatas. Na seção 2.2 são apresentados os fundamentos da teoria Sistêmico-Funcional que subjazem e orientam o desenvolvimento dos recursos do realizador textual baseado em regras.

2.1 ESTUDOS MULTILÍNGUES

Visando a uma potencialização e integração de pesquisas em Linguística e Linguística Aplicada, Matthiessen et al. (2008) propõem um novo campo disciplinar - os Estudos Multilíngues. O novo campo tem como objetivo geral, por um lado, interpretar o domínio de multilinguismo com base na teoria Sistêmico-Funcional, e por outro, examinar a natureza do engajamento científico com esse domínio, com base na teoria Sistêmico-Funcional, em outras palavras, como achados sob as diferentes perspectivas teóricas podem ser inter-relacionados, colocando em contato a Tipologia, Estudos de

Descrição Linguística, Estudos sobre a Tradução etc. Matthiessen et al. (2008, p. 146) baseiam sua argumentação discorrendo sobre o isolamento de áreas afins, como o que se estabelece entre a Tradução e o Ensino de línguas, e a linguística Contrastiva, por exemplo, que historicamente tendiam a manter uma relação de cooperação entre si, mas em algum momento de seu desenvolvimento deixaram de dispor de subsídios das áreas de pesquisa e campos correlatos.

Os Estudos Multilíngues propõem a integração dos estudos em tradução, multilinguismo, linguística contrastiva etc., em um espaço (meta) teórico multidimensional, a arquitetura geral da linguagem segundo a LSF, que possibilita a teorização dos fenômenos de linguagem em contexto e a descrição, ancorada no modelo teórico, de línguas particulares. A integração é apresentada, inicialmente, com relação à dimensão de instanciação: duas línguas em relação podem ser relacionadas em qualquer ponto do contínuo, em termos do produto das seleções sistêmicas, ou seja, os textos resultantes das escolhas, ou em termos do potencial disponível e empregado, dinamicamente, na instanciação. Outro eixo de integração nesse ambiente multidimensional é o número de línguas envolvidas, que percorre de uma a várias línguas envolvidas. A interseção dos contínuos de instanciação e número de línguas modela uma matriz que habilita localizar e integrar os campos e áreas afins (ver Figura 2).

Tradicionalmente, apesar da necessidade de se considerarem os sistemas que instanciam os textos, o campo dos Estudos da Tradução, por exemplo, localiza-se no polo instancial do contínuo, e envolve, pelo menos, duas línguas. Localizam-se neste ponto da matriz pois, de maneira geral, investigam instâncias textuais de mais de duas línguas em relação de tradução, em contato no ambiente multilíngue. Em contrapartida, os estudos em Tipologia tendem a ser localizados no polo potencial do contínuo, na medida em que investigam o potencial de analogia entre sistemas de línguas diferentes, na maioria dos casos, mais de duas línguas simultaneamente. Num ponto intermediário, localiza-se a linguística comparada, interessada no contraste entre sistemas que instanciam os subpotenciais (registros) para a construção do significado. Daí decorre o potencial de integração dessas áreas afins, na medida em que observa-se uma relação de complementaridade dos resultados obtidos em cada campo.

Esse novo campo propõe dois modos de integração dos estudos de linguagem: o **modo reflexivo** (*reflexive mode*) e o **modo ativo** (*active mode*). O **modo reflexivo** visa a investigar os padrões que ensejam o contraste entre línguas, enfocando as similaridades e diferenças que podem ser estabelecidas entre os sistemas (ou instâncias textuais) em contato no ambiente multilíngue. O **modo ativo** promove, por um lado, a implementação de subsídios alcançados em investigações no modo reflexivo, e por outro, retroalimenta investigações do modo reflexivo com base nos resultados alcançados. Exemplo de interação (integração) entre os modos ativo e reflexivo, segundo os autores, são o ensino de língua

estrangeira, treinamento de tradutores, tradução automática, geração multilíngue de textos (MATTHIESSEN et al., 2008). Destacam-se, então, no **modo ativo** de integração no contexto dos Estudos Multilíngues, iniciativas de interface com a área de pesquisa (multidisciplinar por natureza) de Processamento de Língua Natural (PLN), via sistemas de modelagem de significado (sistemas de conhecimento); desenvolvimento de programas de geração e análise sintática integrados, incluindo programas de geração de texto mono e multilíngues.

Sobre o potencial de aplicação dos recursos de sistemas de geração de língua natural, Matthiessen et al. (2008, p. 205), buscando uma integração dos modos **ativo** e **reflexivo**, e uma consolidação da integração de esforços em áreas afins, argumentam que:

> a modelagem de sistemas multilíngues originalmente desenvolvidos no contexto de geração de língua natural pode ser aplicada como a base para examinar-se o aprendizado e produção de significados em línguas diferentes da língua materna e como essa capacidade pode incluir a habilidade de trocar ou misturar 'códigos'. Pode-se ainda, vislumbrar um novo tipo de 'análise contrastiva' emergindo da interseção do trabalho de modelagem de sistemas multilíngues e a análise de textos com a linguística comparativa.[1]

Nesse sentido, esta tese insere-se no campo dos Estudos Multilíngues, modelado no escopo da Linguística Sistêmico-Funcional e adota a perspectiva do *modo ativo* de integração dos estudos sobre a linguagem e correlatos. Essa perspectiva teórico-metodológica possibilita a uma localização mais precisa no ambiente multidisciplinar e enseja uma abordagem da Geração de Língua Natural sob a perspectiva da Linguística Sistêmico-Funcional, com impacto metodológico na implementação de recursos computacionais que realizam o sistema linguístico. Na seção 2.2, serão apresentados os conceitos básicos e a arquitetura geral da linguagem (subseção 2.2.2).

2.2 FUNDAMENTOS DA LINGUÍSTICA SISTÊMICO-FUNCIONAL

2.2.1 Abordagem Sistêmico-Funcional do processo cognitivo

Em seu prefácio de "*Construing experience through meaning*", Halliday e Matthiessen (1999) apresentam a perspectiva na qual ancoram a sua representação da linguagem, sob uma perspectiva da linguagem como semiótica social. Eles definem a linguagem como um conjunto de recursos para a construção de significados através do qual os grupos sociais humanos, e cada membro dessa espécie, "constroem o mapa mental funcional dos fenômenos do mundo, ou seja, o mapa de sua experiência nos processos, que têm lugar

[1] **Tradução de:** The modeling of multilingual systems that was originally developed in the context of text generation could serve as a framework for exploring how people learn how to mean in a language other than their mother tongue and how the ability to mean in another language may also include the ability to switch or mix 'codes', One can imagine a new kind of 'contrastive analysis' emerging at the intersection of the modelling work on multilingual systems and the analysis of texts in the text-based to comparative linguistics

externa, ou internamente à consciência". Tendo em vista a perspectiva da linguagem como semiótica social, Halliday e Matthiessen (1999) apresentam uma abordagem linguística para a modelagem da 'cognição', sob a qual esse fenômeno pode ser melhor investigado como um processo de 'significar', i.e., mais do que um processo de 'pensar'. Os autores argumentam que ao modelar o conhecimento como significado, ele está sendo modelado como uma construção linguística e, portanto, realizada pela lexicogramática, e refletem: "em vez de examinar-se a linguagem através dos processos cognitivos, examinam-se os processos cognitivos através da linguagem".

Essa abordagem, argumentam, é de interesse tanto para áreas afins, como a Linguística Aplicada, Linguística Computacional (Processamento de Língua Natural e Inteligência Artificial), quanto para as Ciências da Cognição. Contudo, assinalam que sua perspectiva é, de certa forma, contraditória àquela que é geralmente proposta pelas Ciências da Cognição, na medida em que trata a informação como significado e não como conhecimento, e o sistema linguístico como um sistema de recursos para a construção do significado e não como um sistema da mente humana. Adotar uma perspectiva linguística do processo cognitivo, possibilita, com base em uma arquitetura linguística abrangente, e organizada sistemicamente, potencialmente acessar o próprio processo cognitivo envolvido na construção do significado em contexto. Sob a perspectiva do Processamento de Língua Natural (PLN), a implementação computacional de recursos lexicogramaticais para a geração de língua natural, sob a perspectiva da teorização da Linguística Computacional no escopo da LSF (como abordaremos a seguir), realiza, e, portanto, é motivada pela arquitetura da linguagem. Nesse sentido, a abordagem do processo cognitivo por uma perspectiva linguística possibilita a modelagem desse processo, subsidiando o exame aos recursos linguísticos envolvidos na construção do significado tanto sob uma perspectiva monolíngue quanto (potencialmente) multilíngue, em particular, dos recursos potencialmente envolvidos no processo de tradução e geração de língua natural. Na subseção 2.2.2 a seguir, serão apresentados os conceitos teóricos básicos e arquitetura geral da linguagem pela perspectiva da Linguística Sistêmico-Funcional.

2.2.2 Linguística Sistêmico-Funcional: conceitos teóricos básicos e arquitetura geral da linguagem

Adotar uma perspectiva científica na análise dos fenômenos humanos, significa, em primeira instância, analisar o papel central que a linguagem exerce nesses fenômenos. As teorias sociológicas, antropológicas etc., têm como fenômenos de interesse os seres humanos enquanto produtores de significado através da língua (FIGUEREDO, 2011; FIRTH, 1957). Em outras palavras, teorias são construtos realizados na linguagem, para examinar fenômenos humanos e, portanto, de linguagem, ou semióticos (i.e., construídos na linguagem). Como aponta Firth (1957), tomar a própria produção linguística como fenômeno de estudo, implica em não só atribuir papel central à linguagem, mas tomar a própria linguagem

como fenômeno. Ao tomar-se a linguagem como fenômeno de investigação, concebe-se uma metalinguagem, ou seja, categorias construídas pela própria linguagem para examinar fenômenos que têm lugar na linguagem, que como tal, são sistemas semióticos (HALLIDAY; MATTHIESSEN, 1999; TEICH, 1999a).

Na condição de sistemas semióticos, os sistemas metalinguísticos têm as mesmas propriedades dos outros sistemas semióticos e podem, portanto, ser modelados seguindo-se os mesmos princípios. Dessa maneira, Teich (1999a) esboça o ambiente, no qual se inserem tanto a teoria linguística Sistêmico-Funcional quanto a sua representação computacional, organizado em termos de abstração, no plano de **estratificação** (*stratification*), cujos estratos se relacionam através do princípio de **realização** (*realization*), no âmbito da teorização da perspectiva computacional da LSF. Partindo desse princípio de estratificação da linguagem, como é concebido pela LSF, pode-se modelar um ambiente metalinguístico que se organiza segundo os mesmos princípios, isto é, em metaestratos.

Segundo Teich (1999a) (desenvolvendo a partir do trabalho de Bateman et al. (1994)), o ambiente metalinguístico no qual se insere a modelagem computacional de recursos linguísticos, sob a perspectiva da LSF, articula-se nos seguintes metaestratos, organizando-se em graus de abstração: o metaestrato superior é o de **teoria**, no qual são organizados significados conceituais de cunho mais geral, ou seja, de alto nível, adotados por dada teoria. No caso da LSF, tratam-se de conceitos básicos, como a linguagem como semiótica social, linguagem como recursos potenciais de produção de significado etc.; adjacente a este, de forma descendente, verifica-se o metaestrato de **representação linguística** *(linguistic representation)*, o qual realiza o metaestrato imediatamente superior em uma metagramática, que recodifica os conceitos abstratos da teoria em termos linguísticos, disponibilizando a metalinguagem necessária para descrições linguísticas, por exemplo, estratificação, sistema, ordenamento etc.; movendo-se mais um passo ao longo do plano metaestratal, encontra-se o metaestrato de **representação computacional**, o qual estabelece uma camada adicional, codificando as representações (meta)linguísticas em termos computacionais; no estrato mais inferior, o de **implementação**, a representação computacional é recodificada em linguagens de programação (e.g., *Python, Php, Lisp* etc.)(ver Figura 4).

O trabalho de **implementação computacional**, i.e., a programação dos recursos linguísticos para composição do realizador superficial–textual aqui proposto se pauta pelo construto teórico da Linguística Sistêmico-Funcional (pautando decisões gerais da implementação computacional propriamente dita nos fundamentos teóricos. i.e., realizando-os), e, consequentemente, em descrições do português brasileiro disponíveis na etapa de desenvolvimento. Como antecipado, a teoria geral da linguagem e as descrições linguísticas estabelecem uma relação intrínseca, de maneira que a teoria apresenta conceitos teóricos que são realizados por representações linguísticas, i.e., categorias, que servem de base para as diversas descrições. Segundo Caffarel et al. (2004, p.8):

> teoria e descrição são ontologicamente distintas sob a perspectiva da Linguística Sistêmico-Funcional: teoria diz respeito à teoria da linguagem humana (ou mais especificamente, por extensão, dos sistemas semióticos em geral); descrição diz respeito a descrições de línguas em particular (ou, por extensão, de sistemas semióticos em particular). Ambas são recursos – recursos para a construção da linguagem (teoria) e línguas (descrição)[2,3].

Halliday (2002, p. 38) afirma que teoria e descrição são conjuntos de esquemas inter-relacionados em graus diferentes de abstração e aponta:

> a teoria relevante consiste em um esquema de categorias inter-relacionadas que é estabelecido para explicar os dados, e um conjunto de escalas de abstração as quais relacionam as categorias aos dados e entre si [...] Descrições consistem em relacionar o texto (corpus) às categorias estabelecidas pela teoria. Os métodos utilizados envolvem vários processos de abstração que variam em tipo e grau. É a teoria que determina a relação desses processos de abstração entre si e com a teoria[4].

Esse esquema de categorias é modelado em um ambiente semiótico, uma arquitetura geral: categorias de organização básicas, dimensões da linguagem (globais e locais), e os princípios que as organizam, realizados no estrato de representação linguística, permitindo a modelagem da linguagem como sistema potencial/processo semiótico para a produção de significados. A representação linguística, como antecipado, habilita as descrições de línguas particulares (descrições são ancoradas no construto teórico) e a sua realização no estrato de representação computacional e implementação. Nesse sentido, à medida que as categorias e dimensões forem sendo apresentadas, será examinado o impacto que as categorias teóricas têm na nos aspectos metodológicos da implementação, ou seja, serão explorados em termos da localização do objeto de pesquisa no mapa semiótico geral: na matriz de função-ordem *(function-rank matrix)*, e na matriz de estratificação-instanciação (*stratification-instantiation matrix*) e como essas relações são realizadas pela implementação computacional. Após o delineamento teórico preliminar, serão apresentadas as categorias fundamentais e dimensões que compõem a arquitetura geral da linguagem e como elas pautam a implementação computacional dos recursos para a realização textual/linguística: estratificação (*stratification*); espectro metafuncional (*metafunctional spectrum*): ideacional (*ideational*), interpessoal (*interpersonal*), e textual (*textual*); contínuo de instanciação

[2] Todas as traduções são de minha autoria

[3] **Tradução de** : Theory and description are ontologically quite distinct in systemic functional linguistics: theory is the theory of human language (or indeed, by extension, of semiotic systems in general); descriptions are descriptions of particular languages (or, by extension, of particular semiotic systems). Both theory and description are resources – resources for construing language (theory) and languages (descriptions)

[4] **Tradução de**: The relevant theory consists of a scheme of interrelated categories which are set up to account for the data, and a set of scales of abstraction which relate the categories to the data and to each other (...)Description consists in relating the text to the categories of the theory. The methods by which this is done involve a number of processes of abstraction, varying in kind and variable in degree. It is the theory that determines the relation of these processes of abstraction to each other and to the theory

(*cline of instatiation*), eixos (*axis*): sistema (*system*), estrutura (*structure*) e escala de ordens (*rank*). No Quadro 1 são sintetizadas as dimensões, em termos de **tipos**, **relações**, e **ordens**.

Quadro 1 – Dimensões semióticas da linguagem em contexto

	Dimensão	**Tipo**	**Relação**	**Ordens**
Dimensões globais	Estratificação	hierarquia	realização	contexto ∼ semântica ∼ lexicogramática ∼ fonologia ∼ fonética
	Metafunção	espectro	confluência	ideacional(lógica ∼ experiencial)-interpessoal-textual
	Instanciação	contínuo	instanciação	potencial ∼ subpotencial ∼ tipo de instância ∼ instância
Dimensões locais	Eixos: Paradigmático / Sistema	hierarquia: contínuo	realização: (interna): delicadeza	gramática ∼ léxico (lexicogramática)
	Eixos: Sintagmático / Estrutura	hierarquia: hierarquia	realização: (interna): sequência / expansão / confluência	oração ∼ grupo ∼ frase ∼ palavra ∼ morfema (lexicogramática)
	(Escala de) Ordens	hierarquia	ordenamento (composicional)	

Fonte: traduzido e adaptado de Halliday e Matthiessen (2014), Matthiessen et al. (2010, p.32)

As categorias teóricas de ordem mais fundamental, que servem de base para a organização da linguagem, são as categorias de **unidade** (***ranking*** – ordenamento na hierarquia composicional), **estrutura** (***structure*** – encadeamento dos elementos nas unidades), classe (class), e **sistema** (***system***). Segundo Halliday (2002, p. 41):

> são categorias da ordem mais alta de abstração: são estabelecidas e inter-relacionadas, na teoria. Por que essas quatro categorias, e não três, ou cinco? Pois a linguagem funciona assim, e essas quatro categorias e nenhuma outra são necessárias para explicar os dados: ou seja, para explicar todos os padrões que emergem de generalizações a partir dos dados.[5]

Halliday (2002, p. 41) aponta que esses quatro conceitos fundamentais são intrinsecamente inter-relacionados, derivando-se um do outro de forma cíclica, o que faz com que não seja possível estabelecer-se uma relação de primazia entre eles. A definição desses conceitos teóricos básicos é feita sempre em referência ao sistema como um todo. Seguindo o raciocínio, Halliday (2002, p. 42) aponta que

> na condição de atividade que carrega padrões de organização de significado (padrão gramatical), a linguagem apresenta padrões de regularidades

[5] **Tradução de**: These are categories of the highest order of abstraction: they are established, and interrelated, in the theory. If one asks: "why these four, and not three, or five, or another four?", the answer must be: because language is like that – because these four, and no others, are needed to account for the data: that is, to account for all grammatical patterns that emerge by generalization from the data.

> que se apresentam em segmentos específicos de enunciado. Uma característica desses segmentos é que além de serem de tamanhos diferentes, parecem ocorrer um dentro do outro.

Assim, **unidades** são segmentos linguísticos que encerram padrões gramaticais, de tamanhos variáveis, caracterizados pela organização em uma hierarquia composicional. Segundo Matthiessen et al. (2010), são "domínios de organização sistêmica e estrutural ordenadas pela escala de ordens de um dado estrato, da ordem mais alta para a mais baixa"[6]. Pode-se, então, estabelecer uma relação entre **unidade**, **sistema** e **estrutura**. Por um lado, da perspectiva do sistema, a **unidade** constitui a condição de entrada para as redes do **sistema** que organizam a estrutura que realiza dado elemento realizador da unidade. Por exemplo, tomando a unidade da oração como ponto de referência, a condição de entrada para o sistema é a 'oração'; dada esta condição de entrada, abrem-se duas opções ('oração maior', ou 'menor'); ao selecionar-se o termo 'oração maior', abrem-se as seleções nos sistemas de TEMA, MODO e TRANSITIVIDADE que organizam a oração maior. Por outro lado, da perspectiva da **estrutura**, as **unidades** "são o domínio da realização estrutural", formando o que os Matthiessen et al. (2010) chamam de 'declarações de **realização'** (*realization statements*) que são associados aos termos das redes dos sistemas, ou seja, cada unidade encerra uma estrutura (ou combinações estruturais) que a realiza, e como são organizadas hierarquicamente, uma dada unidade é constituída (composicionalmente) por elementos da estrutura da unidade imediatamente inferior na escada de ordens. A 'oração', por exemplo, é a confluência de estruturas temática, de modo e de transitividade, que têm realizações associadas aos sistemas de TEMA, MODO e TRANSITIVIDADE, respectivamente. Por sua vez, tomando-se o 'grupo verbal' - que realiza o Processo na unidade imediatamente superior na escala de ordens, a oração - como condição de entrada, abrem-se opções em subsistemas de TIPO DE EVENTO - Ser, Fazer, Sentir, AGÊNCIA - abrindo opções de agenciamento ativo ou passivo etc., e TEMPO SECUNDÁRIO; o grupo verbal é então organizado composicionalmente por palavras verbais, confluindo estruturas de agenciamento e tempo primário e secundário etc (ver Figura 6).

A **estrutura**, como afirma Halliday (2002, p. 46), é, na gramática, "a categoria que explica a relação entre eventos semelhantes que ocorrem sucessivamente [...] como um arranjo de **elementos** organizados em **posições**, concatenados em uma progressão linear, organizadas, teoricamente, em termos de ordens"[7]. Sob a perspectiva sistêmica a **estrutura**, correspondente ao **eixo sintagmático**, é como a 'saída' decorrente de todas as escolhas selecionadas na travessia do sistema, ao serem percorridos cada un dos níveis de delicadeza.

[6] **Tradução de**: Domains of systemic and structural organization ordered by the rank scale of a stratum from the most extensive to the least extensive.

[7] Tradução de: In grammar the category set up to account for likeness between events in successivity is the structure. (...) A structure is thus an arrangement of elements ordered in places. (...)A structure is made up of elements which are graphically represented as being in linear progression; but the theoretical relation among them is one of order.

Como destacado na Figura 6, a metáfora de 'saída' fica mais clara, dado que a saída de uma dada função do sistema de realização textual é uma estrutura de dado material linguístico, de uma dada unidade na escala de ordens, decorrente das escolhas sistêmicas (classes desta unidade), que é construída por unidades de uma ordem imediatamente inferior, e realiza um potencial de funções gramaticais dentro de uma unidade imediatamente superior. Cada uma dessas relações tem contrapartida de implementação computacional nas funções desenvolvidas no realizador textual baseado em regras desenvolvido nesta tese.

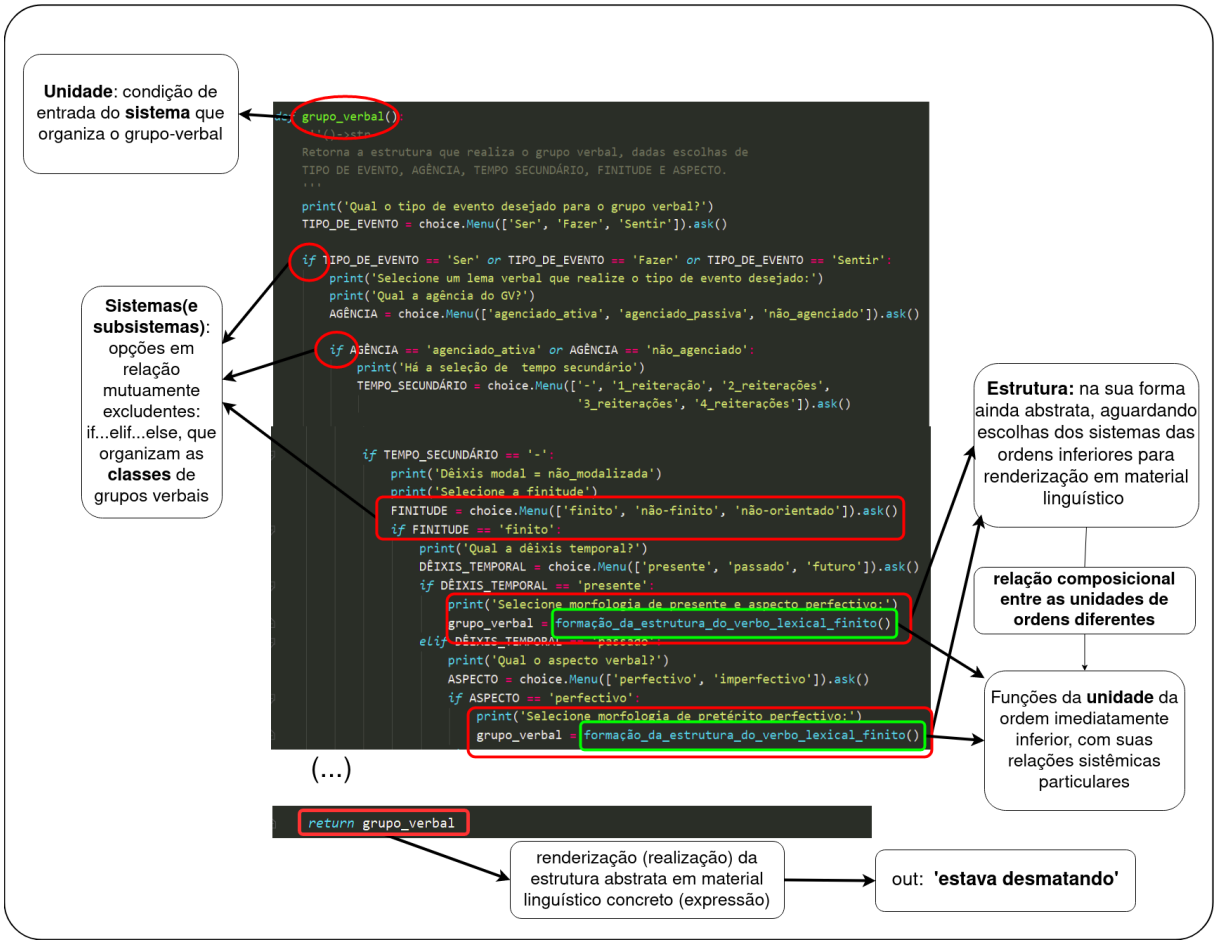
A **classe**, por sua vez, pode ser definida com referência à **estrutura** (e consequentemente à **unidade**). Dada a relação de composição estabelecida entre as unidades de ordens distintas, observa-se um agrupamento de membros, de uma dada ordem, com o mesmo potencial de operação na ordem imediatamente superior na escala, ou seja, elementos de unidades relativamente inferiores se agrupam de acordo com o potencial funcional que encerram nas unidades relativamente imediatamente superiores. Desse princípio decorre que o modo de organização das **classes** restringe seu potencial de funções dentro da estruturas na ordem relativamente superior (HALLIDAY, 2002, p. 49). Matthiessen (1995) aponta que a classe deriva da organização sistêmica da linguagem. Dada uma **unidade** na escala de ordens, verificam-se **classes** gramaticais básicas (de primeira, segunda, e terceira ordem, em diferentes níveis de delicadeza) organizadas em forma de **sistema**, daí decorre que cada unidade da escala de ordens é o ponto de partida de redes de sistema que a organizam, ou seja, são a condição de entrada para o sistema.

No que diz respeito ao **sistema**, que por sua vez corresponde ao **eixo paradigmático**, ele denota a organização das escolhas (classes) potenciais, dada uma **unidade** como condição de entrada, abrindo uma seleção dentre eventos semelhantes mas opostos num dado nível de delicadeza. De acordo com Halliday (2002, p. 40), de maneira geral, um **sistema** é "um conjunto que tem um número finito de opções mutuamente excludentes e mutuamente definíveis, de maneira que o significado de todos muda de acordo com a inserção de um novo item no sistema".

A Figura 6 a seguir, uma captura de tela da interface com interpretador de linguagem *Python*, mostra a implementação computacional, da função em fase de desenvolvimento para a realização de material textual de um grupo verbal: uma função desenvolvida para a realização textual, tendo como condição de entrada a unidade do grupo-verbal (e sua relação com a as outras unidades da escala de ordens), e um exemplo de geração da unidade (saída do realizador textual).

A Figura 6 sintetiza as relações estabelecidas pelas categorias fundamentais da organização da linguagem segundo a LSF, no desenvolvimento das funções computacionais que compõem o módulo de recursos lexicogramaticais para a realização textual, desenvolvidos em linguagem *Python*. O processo realizado pela função apresentada como exemplo na Figura 6 apresenta a realização textual de um grupo verbal do português.

Figura 6 – Implementação computacional do sistema que organiza o grupo verbal em português brasileiro (categorias teóricas realizadas na implementação)



Fonte: elaborada para fins deste estudo.

Note-se, aqui, a conexão que se estabelece entre os conceitos teóricos e o impacto metodológico no desenvolvimento das funções de realização textual. Por isso, no Capítulo 2, de fundamentação teórica, são abordados os conceitos teóricos básicos da LSF e como sua organização baliza os aspectos metodológicos da modelagem computacional. Por outro lado, Capítulo 3, a conexão entre teoria e implementação será abordada com uma orientação mais metodológica.

A implementação computacional dos recursos lexicogramaticais demandou subsídios provenientes dos estudos descritivos do português brasileiro, de base sistêmico-funcional, e, no sentido delineado anteriormente, funcionam como teste para as descrições. Destacam-se as contribuições descritivas do português brasileiro desenvolvida por Figueredo (2007, 2011), com as descrições da estrutura do grupo nominal e o perfil metafuncional do português brasileiro; Sá (2016), que oferece uma descrição do verbo e do grupo verbal; Ferreguetti (2014), com considerações sobre orações existenciais no par inglês-português e Ferreguetti (2018), com a frase preposicional com função de Qualificador no Grupo Nominal; Paula (2018), apresenta a descrição do processo verbal; e Braga (2016), com

contribuições sobre as circunstâncias no português brasileiro.

As subseções a seguir apresentam as dimensões semióticas da linguagem em contexto, como previamente apresentadas no Quadro 1.

2.2.2.1 Estratificação

De acordo com Matthiessen et al. (2010) a estratificação é "uma dimensão global, que organiza a linguagem em contexto em subsistemas (estratos, organizados pelo princípio de realização), de acordo com o grau de abstração simbólica".

Na condição de sistema de quarta ordem superior, ou seja, semiótico, a linguagem precisa ser estratificada em pelo menos dois estratos para cumprir o potencial de produção de significados – **conteúdo** (*content*), responsável pelo significado e **expressão** (*expression*), responsável por realizar esse significado em forma de sons, escrita ou sinais. Os sistemas de quarta ordem, de maneira geral, estabelecem uma relação unívoca entre conteúdo e expressão. No desenvolvimento da linguagem, na linha temporal de cada indivíduo (ontogênese), observa-se que nos estágios iniciais desenvolve-se uma proto linguagem, na qual o estrato de conteúdo e expressão realizam relações unívocas.

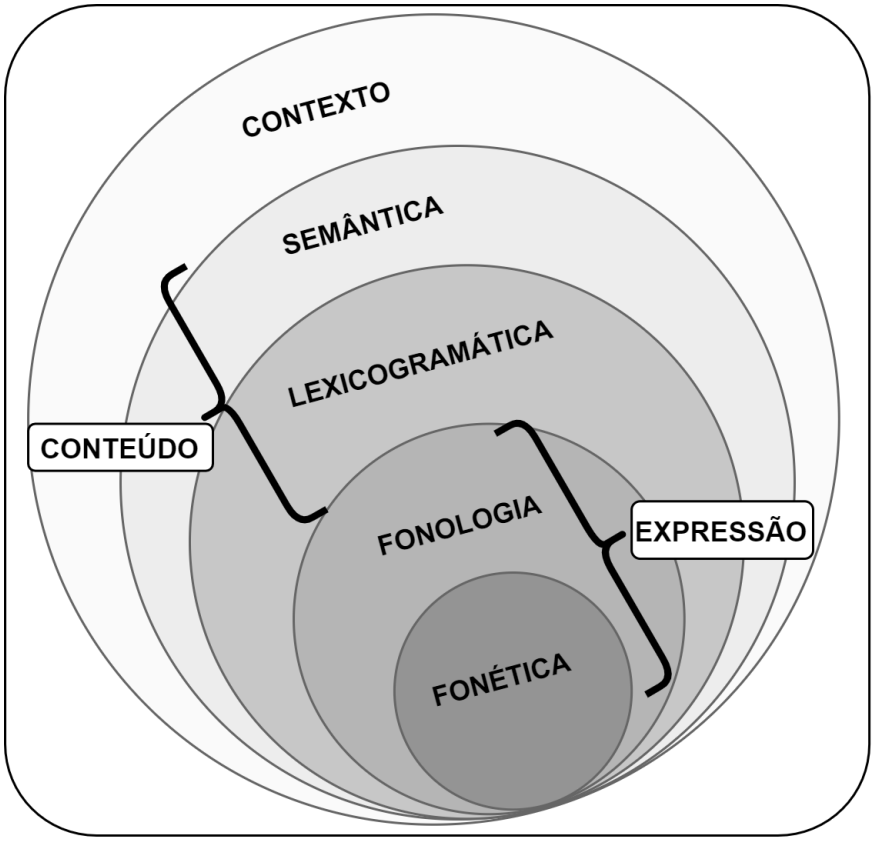
Devido às necessidades potenciais da linguagem, há uma expansão dos estratos de conteúdo e expressão, num estágio mais avançado do desenvolvimento da língua. O estrato do conteúdo e de expressão se dividem em dois: o conteúdo se expande para semântica e lexicogramática; e a expressão se expande para fonologia e fonética. Halliday e Matthiessen (2014, p. 25) apontam que a expansão dos estratos é um reflexo das funções que a linguagem encerra: "seres humanos utilizam a linguagem para fazer sentido das suas experiências e encenar as suas relações com os membros de sua comunidade, construindo essas relações através dos significados, construindo suas relações interpessoais e fazendo sentido da sua experiência em forma de significado, no estrato semântico". Em um segundo momento, esse significado precisa ser realizado em forma de fraseado, o que é realizado pelo estrato lexicogramatical. O mesmo processo acontece no plano da expressão (*expression plane*). A relação entre conteúdo e expressão depende do modo de expressão: no modo fônico, "o estrato fonético faz a interface com o ambiente, no caso, os recursos do corpo para a fala e escuta, e o estrato fonológico, a organização do som em estruturas formais e sistemas". No modo gráfico, a codificação se dá através da grafologia e da grafética, além da linguagem de sinais e outras formas de expressão.

Os estratos de conteúdo e expressão são imbricados no contexto e realizam-no. A constante tensão entre os estratos de conteúdo e expressão faz com que mudanças no estrato lexicogramatical operem mudanças de interpretações no estrato semântico; e mudanças no estrato semântico operem mudanças de realização na lexicogramática, o que possibilita a criação de significados. Como aponta Figueredo (2011), essa relação enseja que a linguagem crie as próprias variáveis contextuais. As variáveis (meta) contextuais

criadas pela linguagem e pelas quais ela deve ser compreendida são, de acordo com Halliday e Matthiessen (2014, p. 33): "**i) campo (*field*)**: o que acontece na situação (organização da realidade e conhecimento); a natureza da atividade (o assunto ou tópico). **ii) sintonia (*tenor*)**: quem participa da situação; estabelecimento e manutenção das relações interpessoais em termos do papel ocupado: institucionais, status de poder entre os falantes etc. **iii) modo (*mode*)**: organização dos significados de campo e sintonia em unidades textuais. O princípio que organiza a dimensão da **estratificação** é a **realização (*realization*)**, que implica que [o contexto é realizado na [semântica [que é realizada na lexicogramática [que é realizada na fonologia [que é realizada na fonética]]]]].

A Figura 7 a seguir mostra a dimensão da **estratificação**, apresentando os estratos desde o contexto (o extrato mais abstrato), até o estrato fonético.

Figura 7 – Estratificação da linguagem em contexto

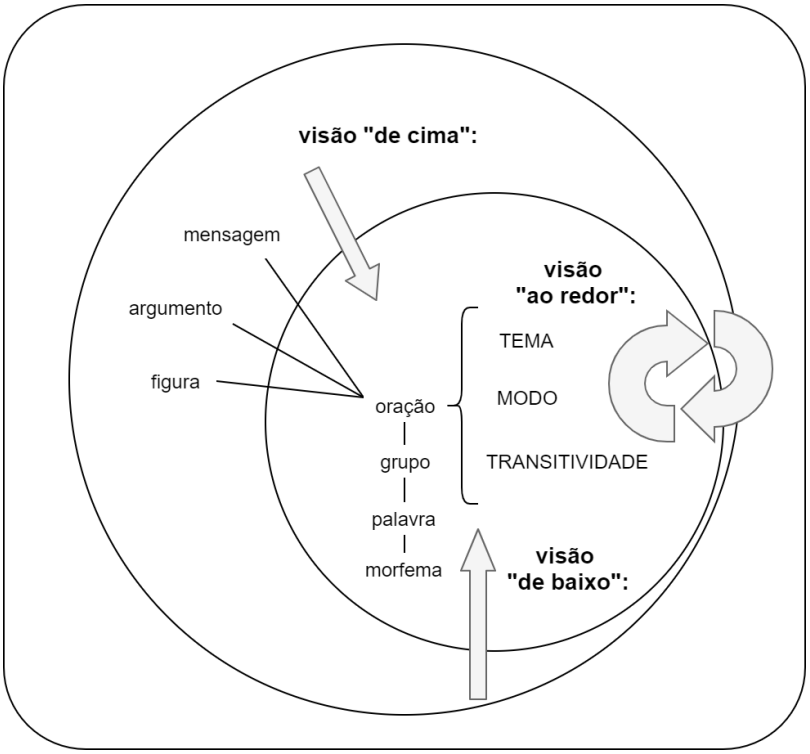


Fonte: traduzida e adaptada de Halliday e Matthiessen (2014)

Dada esta dimensão da linguagem e os princípios que a organizam, é possível estabelecer o como esse ambiente é representado no realizador textual, sob uma perspectiva trinocular de análise: **de cima** (*from above*), os significados do baixo estrato semântico, através do espectro metafuncional, são realizados pela lexicogramática, isto é, como a **figura** - sistema semântico de CONFIGURAÇÃO, o **argumento** - sistema semântico de FUNÇÕES DISCURSIVAS, e a **mensagem** - sistema semântico de PROGRESSÃO confluem na realização

da **oração** (*clause*), através dos sistemas lexicogramaticais de TRANSITIVIDADE, MODO e TEMA, respectivamente; **ao redor** (*from roundabout*), as redes de sistemas organizam os significados que constroem a oração, e consequentemente as ordens relativamente inferiores; e **de baixo** (*from below*), como as unidades inferiores na escala de ordens, e que constituem a oração, são concatenadas para a realização de sua estrutura (ver Figura 8).

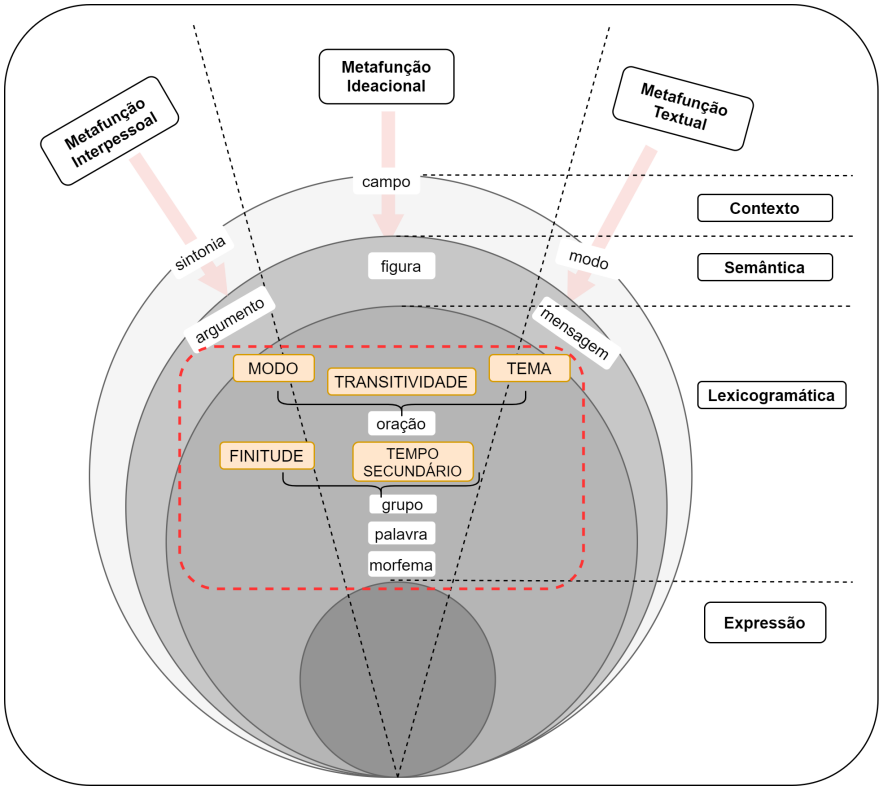
Figura 8 – Perspectiva trinocular



Fonte: Traduzida e adaptada de Halliday e Matthiessen (2014) e Matthiessen et al. (2010)

Como unidade multifuncional da lexicogramática, a oração realiza significados semânticos derivados das três metafunções (*metafunctions*) (ver a subseção 2.2.2.3). Assim, **de cima**, sob a perspectiva da metafunção ideacional, examina-se como a parte superior do estrato lexicogramatical, na sua ordem mais abrangente, a oração, através do sistema de TRANSITIVIDADE (Processos, Participantes e Circunstâncias). Nesse sentido, realiza o sistema semântico de CONFIGURAÇÃO, **figura [quantum de mudança]**: processos, participantes e possíveis circunstâncias, construindo assim o **domínio experiencial**. Ainda sob a perspectiva **de cima**, sob a metafunção interpessoal, examina-se a parte inferior do estrato semântico, **argumento [quantum de interação]**, através do sistema semântico de FUNÇÕES DISCURSIVAS, realizado pelo sistema de MODO, que organiza a oração na lexicogramática em termos de modo: Indicativo (Interrogativo/Declarativo), Imperativo etc. Por fim, sob a perspectiva da metafunção textual, a linguagem constrói significados em termos de **mensagem [quantum de informação]**, através do sistema semântico de PROGRESSÃO, e sua realização no sistema lexicogramatical de TEMA (ver Figura 9).

Figura 9 – Localização da pesquisa na matriz de função-ordem e na matriz de função-estratificação



Fonte: Traduzida e adaptada de Halliday e Matthiessen (2014), Matthiessen et al. (2010)

A subseção 2.2.2.2 a seguir apresenta a dimensão de **instanciação**.

2.2.2.2 Instanciação

A segunda dimensão global a ser apresentada é o **contínuo de instanciação**. Matthiessen et al. (2010) definem esta dimensão como

> o contínuo que se estende do potencial à instância, com pontos intermediários em sua extensão subpotencial e tipo de instância, dependendo da perspectiva de observação [...] que opera em sistemas de todas as ordens (física, biológica, social, e semiótica), e possibilita estabelecer relações entre instâncias e o potencial linguístico [8]

A "comparação dos textos instanciados entre si, e entre estes e potencial, pode ser realizada utilizando-se critérios de qualquer um dos estratos, desde que sistemáticos e explícitos" (HALLIDAY; MATTHIESSEN, 2014, p. 29). É, então, possível estabelecer uma interseção entre estratificação e instanciação, abordando a distinção estabelecida entre o estrato contextual e os de conteúdo e expressão, ao longo do contínuo de instanciação, estabelecendo a matriz de estratificação/instanciação. O ambiente modelado pela matriz

[8] **Tradução de**: Cline extending from potential to instance, with intermediate points along the cline—subpotential and instance type [...] that operates in systems of all orders (physical, biological, social and semiotic); it relates observable instances to the potential that lies behind them.

de estratificação/instanciação possibiliza a localização do objeto de estudo, instâncias, sistemas etc., em suas coordenadas.

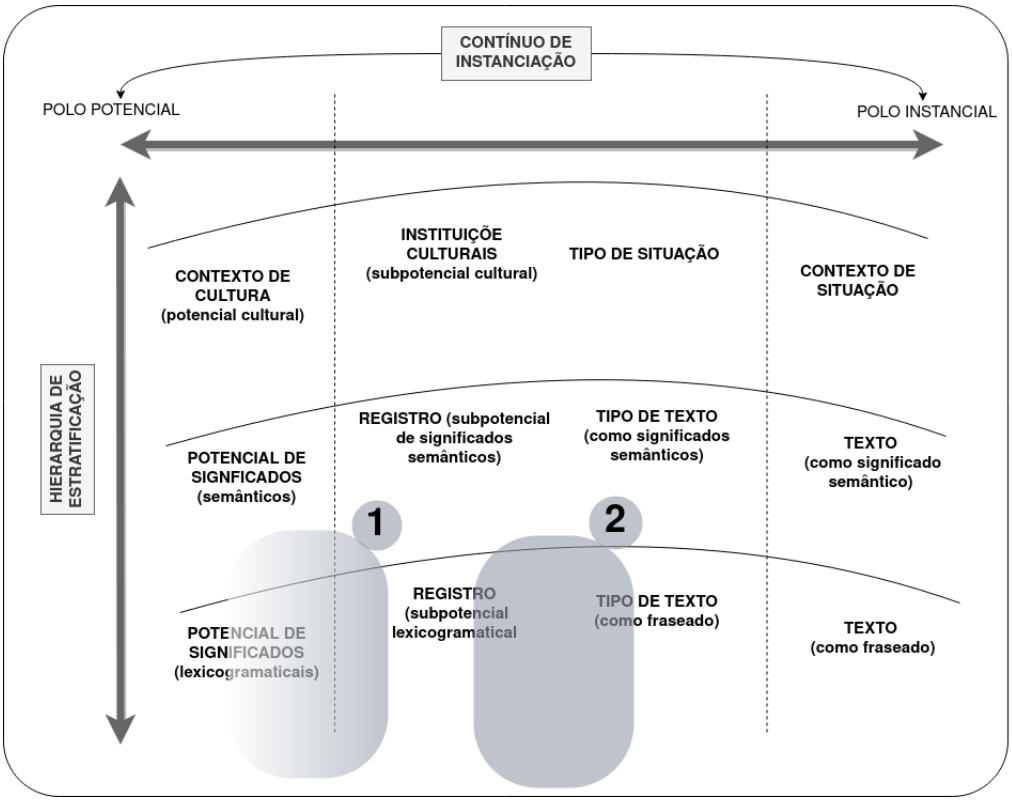
Sob o ponto de vista dos estratos linguísticos de conteúdo e expressão, especificamente o estrato lexicogramatical, é possível interpretar o contínuo de instanciação de duas perspectivas distintas. Por um lado, a linguagem é abordada como sistema potencial para a produção de significado, e por outro, do ponto de vista do polo instancial, a linguagem é abordada como texto, ou seja, as probabilidades sistêmicas que realmente foram empregadas na produção do significado, instanciando o texto. É importante ressaltar que, segundo Halliday e Matthiessen (2014, p. 27), a língua como potencial e a língua como texto não devem ser vistas como dois objetos distintos. Um movimento horizontal no contínuo possibilita um exame a pontos intermediários e a observação de padrões: partindo-se do polo potencial, esses padrões intermediários são **registros**: conjuntos restritos de ajustes de probabilidades do potencial, ou seja, "um conjunto particular, subpotencial, de probabilidades sistêmicas" (HALLIDAY; MATTHIESSEN, 2014, p. 29). Por outro lado, partindo-se do polo instancial, o ponto intermediário é formado por padrões recorrentes, destilados a partir de instâncias textuais que operam em tipos de situações semelhantes, formado agrupamentos, ou **tipos de texto**.

Um movimento na direção do estrato contextual, possibilita que se relacione uma instância textual a um **contexto de situação** particular, envolvendo atividades e indivíduos realizando papéis específicos. Contudo, como acontece sob o ponto de vista do estrato lexicogramatical, o contínuo de instanciação se estende do polo potencial (**contexto de cultura**) ao instancial (**contexto de situação**). No polo potencial, o **contexto de cultura** é o potencial de produção de significado no qual operam vários sistemas semióticos, como a linguagem, a dança, a arquitetura etc. No polo instancial, **contexto de situação**, observa-se um número de probabilidades de configurações, de atividade e indivíduos, mais restrito, consistindo das opções de variáveis contextuais mais prováveis de serem selecionadas para a produção de significado num tipo abstrato de situação. Como ocorre sob a perspectiva do estrato lexicogramatical, o movimento horizontal no contínuo de instanciação revela pontos intermediários: partindo do polo instancial em direção ao potencial, padrões recorrentes de contextos de situação (que instanciam situações semelhantes) se agrupam formando tipos de situação; um passo mais em direção ao potencial revela padrões recorrentes de agrupamentos de tipos de situações, que formam as instituições culturais, ou seja, conjuntos de situações que compartilham o emprego de um conjunto de recursos selecionados do potencial.

Dados os objetivos deste trabalho, a delimitação das dimensões da estratificação e da instanciação dentro do espaço semiótico geral possibilita a localização (semiótica) do módulo de realização textual na matriz de estratificação/instanciação. Como apresentado, o principal objetivo deste trabalho é implementar os recursos lexicogramaticais do por-

tuguês brasileiro computacionalmente, na construção de um módulo com funções para a realização textual, independente de domínio. Como objetivos secundários, esta tese visa a realizar testes comparativos de acurácia entre o realizador textual baseado em regras e os resultados alcançados por redes neurais (no âmbito do CoNLL-SIGMORPHON) na tarefa de flexão verbal em *corpora* de verbos independentes de domínio; aplicar o módulo na linha de produção, subtarefa de realização textual (flexão verbal), de uma instância local do robô-jornalista (o @DaMataReporter). Tendo estes objetivos em perspectiva, o módulo de recursos linguísticos para realização textual se localiza no estrato lexicogramatical, e orienta-se ao polo potencial do contínuo de instanciação (índice 1 - Figura 10). Mais especificamente, o módulo implementa recursos que se aproximam do potencial lexicogramatical do português brasileiro, computacionalmente. Por outro lado, o robô-jornalista opera no domínio de desmatamento da Amazônia Legal no território brasileiro, relatando notícias jornalísticas, e portanto, localiza-se intermediariamente no contínuo de instanciação, ou seja, um subpotencial (registro - tipo de texto) lexicogramatical envolvido na construção do domínio em questão. (índice 2 - Figura 10)

Figura 10 – Localização do objeto de estudo na matriz estratificação-instanciação



Fonte: Traduzida e adaptada de Matthiessen (2013)

A subseção 2.2.2.3 a seguir apresenta a dimensão de **metafunção**.

2.2.2.3 Metafunção

Halliday (1978, p. 2) aponta que entender-se a realidade social, que é denominada cultura, como um conjunto de potenciais semióticos no qual a linguagem ocupa papel central (pois serve de potencial para a construção (codificação) de outros sistemas semióticos) viabiliza a abordagem da linguagem como semiótica social. Partindo dessa perspectiva, para o estudo da linguagem, a LSF apresenta subsídios teóricos para a modelagem/teorização do **contexto**, que apresenta dois polos de abstração (no contínuo de instanciação), como apresentado na subseção 2.2.2.2: o **contexto de cultura** e o **contexto de situação**. Como foi apontado, o **contexto de cultura** é o potencial semiótico, e é organizado em termos das variáveis meta-contextuais de **campo, sintonia** e **modo**. Portanto, o **contexto de cultura** é o potencial de variáveis contextuais, (campo, sintonia , e modo), do qual são selecionados os aspectos que delimitam a produção dos significados linguísticos, e, consequentemente, o **contexto de situação** é um subconjunto de escolhas dentro do potencial de variáveis contextuais relevantes para a produção de uma dada instância linguística, ou **situação** (HALLIDAY, 1978; HALLIDAY; MATTHIESSEN, 2014).

A LSF postula que a linguagem é um sistema semiótico potencial para a construção de significados linguísticos e, o texto, apesar de ser realizado pela lexicogramática, é uma unidade semântica que realiza as variáveis meta-contextuais (ver a subseção 2.2.2.1). Sob essa perspectiva, segundo Halliday e Matthiessen (2014), esse sistema potencial presta a "funções básicas em relação ao nosso ambiente ecossocial: representação da nossa experiência (**campo**), encenação das nossas relações sociais (**sintonia**), e organização desses significados em forma de texto (**modo**)". Essas funções gerais são evidenciadas na organização geral da linguagem sob o espectro semânticos metafuncional: ideacional, interpessoal e textual, respectivamente. A **metafunção ideacional** (*ideational metafunction*) é subdividida em dois componentes: o experiencial (*experiential*) e o lógico (*logical*). Sob o componente experiencial, habilita o falante a construir suas representações de 'conteúdo' dos fenômenos de sua experiência em seu ambiente ecossocial, bem como sua experiência dentro da própria consciência; sob o componente lógico, organiza o fluxo de experiências em relação de subordinação, coordenação, elaboração, expansão, projeção etc., ou seja, organiza os significados em estruturas lógicas. O principal sistema lexicogramatical que realiza essa metafunção é o sistema de TRANSITIVIDADE. Além de construir as experiências de mundo e organizá-las em relações lógicas, os falantes precisam, para se consolidarem como indivíduos dentro do seu grupo social, estabelecer, encenar continuamente suas relações, através da encenação de relações de hierarquia e poder, encenando assim perguntas, pedidos, informações, persuasão etc. A metafunção que habilita esse tipo de organização do significado é a **metafunção interpessoal** (*interpersonal metafunction*). O principal sistema que realiza essa metafunção na lexicogramática é o sistema de MODO. Por último, a língua, além de habilitar a construção das experiências dos fenômenos do mundo real

e da nossa consciência, organizando-os de maneira lógica, e habilitar a construção dos significados interpessoais, precisa de uma metafunção que habilite a organização desses significados ideacionais e interpessoais em forma de texto. A **metafunção textual** (*textual metafunction*) realiza esse trabalho organizando os elementos ideacionais e interpessoais em unidades de informação (Dado e Novo) e colocando em destaque elementos que têm mais importância na elocução (Tema e Rema).

O objeto de estudo deste trabalho, mais especificamente o módulo de realização textual implementado, modela os espectros metafuncionais por duas perspectivas, sempre em interseção com o contínuo de instanciação. O realizador textual superficial, por representar, computacionalmente, o potencial para a construção de significados (realizados na oração) do português brasileiro, se localiza mais em direção ao polo potencial do contínuo de instanciação, e cobre todo o espectro metafuncional. Como a unidade mais alta na escala de ordens da lexicogramática, a oração engloba as três metafunções e pode ser compreendida pela perspectiva trinocular, como já antecipado, '**de cima**', '**ao redor**', '**de baixo**'. '**De cima**', sob uma perspectiva da hierarquia de estratificação, a oração conflui significados semânticos de mensagem (textual), de figura (ideacional) e de argumento (interpessoal), com possíveis realinhamentos através de metáfora gramatical[9]; '**ao redor**', no estrato lexicogramatical, a unidade da oração é a condição de entrada para vários sistemas, sob as três metafunções, que trabalham em confluência para a sua organização: sistema de TEMA (metafunção textual), de TRANSITIVIDADE (ideacional), e de MODO (interpessoal); e '**de baixo**', como as unidades relativamente inferiores à oração, na escala de ordens, se organizam na sua estruturação (MATTHIESSEN et al., 2010; HALLIDAY; MATTHIESSEN, 2014)). Dessa maneira, como já apontado, a implementação dos recursos linguísticos para a geração de língua natural (independente de domínio) precisa considerar todo o espectro metafuncional.

Na subseção 2.2.2.4 a seguir, será apresentada a dimensão de **eixo (paradigmático e sintagmático)**

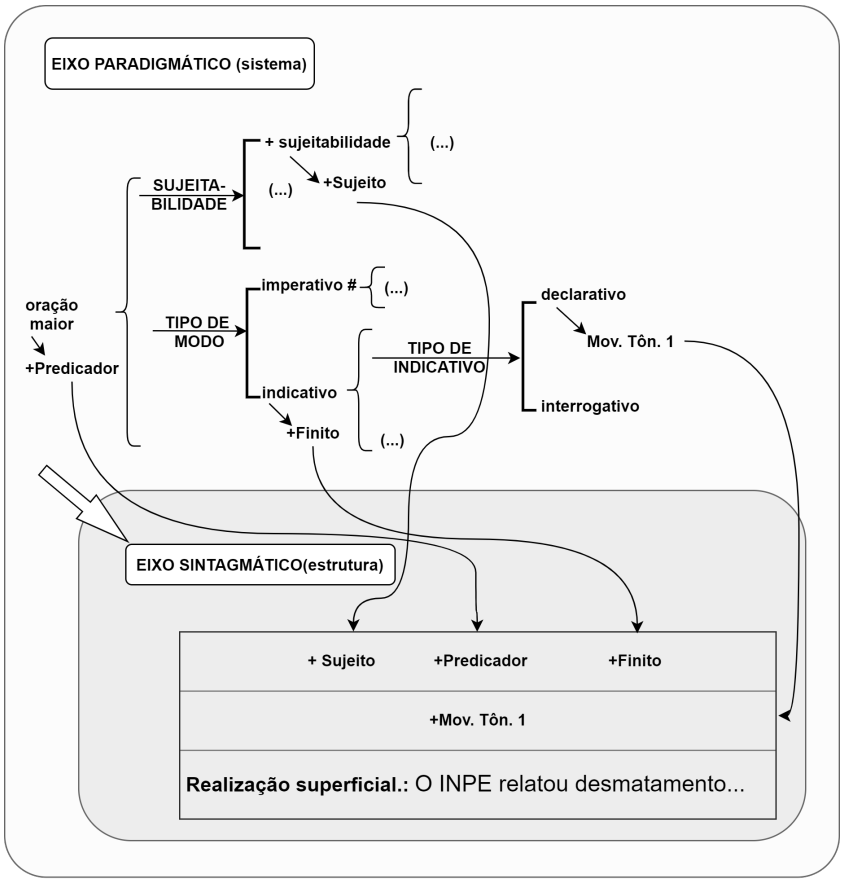
2.2.2.4 Eixos

A hierarquia de **eixos** é a distinção entre o eixo **paradigmático** e o eixo **sintagmático**, organizada em graus de abstração. Assim, como apontam Matthiessen et al. (2010), "são hierarquicamente organizados em graus de abstração, onde a organização paradigmática é localizada acima da organização sintagmática e se relacionam através do **princípio de realização**: o eixo sintagmático **realiza** o eixo paradigmático". Essa relação pode ser observada em redes de sistemas, nas quais uma opção no sistema é associada a uma declaração de realização (*realization statement*). Uma declaração de realização é uma

[9] A metáfora gramatical não será contemplada na implementação computacional, e, portanto, não será abordada neste estudo.

especificação, ainda em estado abstrato, de como a estrutura sintagmática para uma dada escolha sistêmica deve ser renderizada pelo estrato de expressão. A Figura 11 a seguir destaca como a relação de realização entre os eixos se estabelece.

Figura 11 – Relação de realização entre os eixos paradigmático e sintagmático



Fonte: Traduzido e adaptado de Matthiessen et al. (2010) e com aplicão de sistemas descritos por Figueredo (2011) na anotação de um exemplo de oração extraído do @DaMataReporter

Como observa-se na Figura 11, as escolhas por opções no sistema da oração maior, nos subsistemas de SUJEITABILIDADE, TIPO DE MODO:indicativo, culminam em uma especificação que traz informações de como a realização textual deve ser renderizada. No exemplo destacado na Figura 11, o Sujeito é realizado pelo grupo nominal 'O INPE', o Predicador/Finito (ou Núcleo) pelo verbo lexical 'relatar' (com morfema interpessoal de Pretérito Perfectivo I). Além disso, o indicativo declarativo é realizado pelo Movimento Tônico do tipo 1 (ver Figueredo (2011)). A subseção 2.2.2.4.1 e subseção 2.2.2.4.2 a seguir apresentam mais detidamente cada um dos eixos, **ordem sintagmática: estrutura** e **eixo paradigmático**.

2.2.2.4.1 Estrutura (Ordem sintagmática)

A dimensão da **estrutura** compreende o aspecto composicional da linguagem, sendo organizada, segundo Halliday e Matthiessen (2014, p. 21), pelo "princípio de **ordenamento**:

algumas camadas composicionais organizadas pelo princípio de **partes de**".

Essa dimensão compreende diferentes hierarquias composicionais, divididas em domínios específicos dentro da arquitetura, a saber: no som, na escrita, no verso e na gramática e é guiada pelo princípio de **esgotamento**, ou seja, unidades de uma dada ordem superior são formadas por um conjunto finito de unidades de uma ordem imediatamente inferior (HALLIDAY; MATTHIESSEN, 2014, p. 21). Ao tomarmos um dos domínios, por exemplo o da gramática, a hierarquia no português, inglês e outras línguas é constituída de oração∼grupo/frase∼palavra∼morfema, de maneira que uma oração é composta por um conjunto de grupos, os grupos por um conjunto de palavras e assim por diante. Por esta razão, a estrutura é responsável por estabelecer a ordem sintagmática.

Até este ponto, o objeto de estudos do presente trabalho foi localizado em termos de hierarquia de estratificação, localizando o módulo de recursos linguísticos baseado em regras para realização textual na lexicogramática e o domínio experiencial do robô jornalista @DaMataReporter na interface entre estrato semântico e o lexicogramatical. Em termos de contínuo de instanciação, o módulo de recursos linguísticos é localizado em um ponto mais próximo ao potencial de produção de significados linguísticos, e o domínio em um ponto mais intermediário (subpotencial).Metafuncionalmente, o módulo de recursos linguísticos é localizado na confluência dos espectros metafuncionais (ideacional, interpessoal e textual), e o domínio experiencial no espectro metafuncional ideacional, modelando a base de ideação que serve de plataforma para aplicação do módulo de recursos linguísticos para a geração, mais especificamente, o domínio de desmatamento da Amazônia legal em reportagem jornalística. Em termos de estrutura, a geração da oração (e como consequência, das unidades relativamente inferiores na escala de ordens) pelo módulo de recursos linguísticos foi modelada com base no princípio de composição das ordens da escala, ou seja, a oração é constituída por elementos da unidade da ordem imediatamente inferior [grupos], que, por sua vez, são constituídos por elementos de unidade da ordem imediatamente inferior [palavras], que são constituídos por elementos da unidade da ordem imediatamente inferior [morfemas], e dessa forma, o módulo computacional de recursos abrange toda a escala de ordens e suas estruturas inerentes. A subseção 2.2.2.4.2 apresenta mais detidamente o eixo paradigmático.

2.2.2.4.2 Sistema (Eixo paradigmático)

Ao contrário da ordem sintagmática (estrutura), na qual unidades de uma ordem inferior se concatenam para compor a ordem imediatamente superior, no eixo paradigmático (sistema) temos conjuntos de opções em oposição, ou seja, a realização de uma opção em contraponto a opções que poderiam ser selecionadas, mas não são. Matthiessen et al. (2010, p. 211) definem a categoria de **sistema** (*system*) como:

> categoria central de representação da organização **paradigmática** do significado, em qualquer estrato: fonológico, gramatical ou semântico. Os sistemas consistem em (1) uma declaração de contraste entre dois **termos**, representados por **opções** e (2) uma **condição de entrada** que especifica onde o contraste se estabelece [10].

Segundo Halliday e Matthiessen (2014), diferentemente da dimensão da estrutura, que se baseia em um tipo de relacionamento de 'partes de', o sistema é baseado no relacionamento de 'tipos de'. Exemplifica-se com o sistema de POLARIDADE: uma oração seleciona obrigatoriamente a POLARIDADE:positiva ou POLARIDADE:negativa, logo, uma oração negativa é um 'tipo de oração'. Além disso, pode-se tomar mais um passo adiante, selecionando entre tipos de negativa, dando assim mais um contraste paradigmático, o segundo sendo mais refinado que o primeiro, isto é, um passo a mais na **delicadeza** (*delicacy*) do sistema. A travessia no sistema de organização da polaridade até esse nível de delicadeza estabelece uma relação de 'um tipo de um tipo de' oração.

Cabe apontar-se que opções menos delicadas nos sistemas tendem a ser mais genéricas, e à medida em que caminha-se em delicadeza nos sistemas, ou seja, repetindo o procedimento de escolhas paradigmáticas, as escolhas vão se especializando, tornando-se cada vez mais diferentes entre si, até o polo mais delicado, a realização. Isso confere o status de contínuo à lexicogramática: um contínuo que vai da gramática, escolhas menos delicadas (polo gramatical) ao léxico, escolhas mais delicadas (polo lexical).

Um princípio importante, que possibilita evidenciar e testar padrões dentro dos sistemas, é a **agnação** (*agnation*). De acordo com Matthiessen et al. (2010), a agnação é "uma propriedade de organização do sistema (eixo paradigmático): relação entre opções paradigmáticas, representadas como termos nos sistemas. Termos relacionados (em oposição nos sistemas) são considerados opções agnatas". Matthiessen (2001, p. 80) aponta que qualquer expressão estabelece infinitas relações de agnação, definidas através de ordens de vários ambientes semióticos e, nesse sentido, é multidimensional.

Como mencionado na subseção 2.2.2, o desenvolvimento do módulo de recursos lexicogramaticais para a realização textual tem lastro no construto teórico da Linguística Sistêmico-Funcional, e portanto, pauta decisões gerais da implementação computacional nos fundamentos da teoria, realizando-os. Foram apresentados os conceitos básicos e as dimensões da linguagem de acordo com a teoria, ensejando a localização do objeto de estudos desta tese nesse ambiente semiótico. Neste ponto faz-se necessária a discussão de alguns pontos. Por se tratar de um trabalho que tem como objetivo o desenvolvimento computacional de recursos lexicogramaticais, os conceitos teóricos nos quais se pauta a implementação têm impacto direto nas decisões metodológicas. Por isso, a relação entre a

[10] **Tradução de**: The central category for representing **paradigmatic** organization at any stratum—phonological, grammatical or semantic. It consists of (1) a statement of a contrast between two or more **terms**, represented by **features**, and (2) an **entry condition**, which specifies where the contrast holds.

teoria (conceitos base da abordagem sistêmico-funcional) e a implementação computacional é apresentada de forma difusa nesta tese: por um lado, no Capítulo 2 apresentam-se os conceitos básicos e dimensões da linguagem e a localização do trabalho nesse ambiente semiótico; por outro lado, no Capítulo 3 apresenta-se como a articulação dos conceitos teóricos são realizados e impactam o desenvolvimento dos recursos computacionais para a realização textual.

A seção 2.3 a seguir apresenta o Processamento de Língua Natural (PLN), particularmente a subárea de Geração de Língua Natural (GLN). Serão apresentados os conceitos básicos e arquitetura geral de sistemas de GLN e sua aplicação em um robô-jornalista; será abordada a perspectiva da GLN dentro do escopo da Linguística Sistêmico-Funcional.

2.3 PROCESSAMENTO DE LÍNGUA NATURAL (PLN): GERAÇÃO DE LÍNGUA NATURAL (GLN)

Esta seção visa a apresentar a base teórica sobre a Geração de Língua Natural (GLN) que pauta esta pesquisa. O conceito de Geração de Língua Natural será definido de maneira mais ampla, e serão apresentados a arquitetura de sistemas de geração e os conceitos básicos gerais da área. Será destacada a abordagem da GLN no escopo da Linguística Sistêmico-Funcional.

Devido ao seu caráter multidisciplinar, o Processamento de Língua Natural é abordado de diferentes perspectivas, resultando numa diversidade de definições, com claros pontos de sobreposição entre elas, a depender da perspectiva disciplinar e institucional do observador. Segundo Jurafsky (2000, p. 9), historicamente recebe tratamento distinto por diferentes áreas do conhecimento e

> engloba uma série de campos diferentes, mas sobrepostos, nos diferentes departamentos: Linguística Computacional na Linguística, Processamento de Linguagem Natural na Ciência da Computação, Reconhecimento de Fala na Engenharia Elétrica, Psicolinguística Computacional na Psicologia.[11]

Esse caráter difuso resulta em uma alternância de nomes para a área. A Associação para Linguística Computacional (*Association for Computational Linguistics*) menciona que, frequentemente, referem-se à área de pesquisa ora como **Linguística Computacional**, ora como **Processamento de Língua Natural (PLN)** e define:

> Linguística Computacional é o estudo científico da linguagem sob uma perspectiva computacional. Linguistas computacionais estão interessados

[11] **Tradução de:** speech and language processing encompasses a number of different but overlapping fields in these different departments: computational linguistics in linguistics, natural language processing in computer science, speech recognition in electrical engineering, computational psycholinguistics in psychology

> em fornecer modelos computacionais dos vários tipos de fenômenos linguísticos.[12,13]

A *ACL* ressalta que os modelos podem ser desenvolvidos manualmente, ou baseados em dados (modelos estatísticos e empíricos). Ainda segundo a associação, a motivação para a Linguística Computacional pode partir de uma motivação teórica (científica) ou aplicada: i)teórica, na medida em que pode ser motivada pelo interesse de explicar algum fenômeno linguístico com uma abordagem computacional; ou ii)aplicada, sendo motivada pela necessidade de desenvolvimento de módulo funcionais em sistemas de processamento de língua natural.

Na mesma linha, Sociedade Brasileira de Computação (SBC) aponta a alternância no nome da área e a define:

> a área de pesquisa de Processamento da Linguagem Natural (PLN), também denominada Linguística Computacional ou, ainda, Processamento de Línguas Naturais, lida com problemas relacionados à automação da interpretação e da geração da língua humana em aplicações como Tradução Automática [...] (SANTOS, 2020).

Reiter e Dale (2000, p. 1) definem **Geração de Língua Natural** como:

> uma subárea da Inteligência Artificial e Linguística Computacional que enfoca na engenharia de sistemas computacionais para a produção de textos em línguas naturais inteligíveis por humanos. A partir de representações não linguísticas da informação, os sistemas de geração usam conhecimentos sobre as línguas e o domínio de aplicação para produzir variados tipos de textos automaticamente.[14]

Na qualidade de uma pesquisa na área de Linguística Aplicada, e adotando a Linguística Sistêmico-Funcional como uma teoria linguística com potencial de aplicação, esta tese adere à perspectiva de GLN no escopo da LSF. Halliday e Webster (2009, p. 250) propõem uma conceitualização que insere o Processamento de Língua Natural (PLN) como uma subárea da Linguística Computacional, segmentada como Interpretação e Geração de Língua Natural (GLN), e contempla ainda o Aprendizado de Máquina, como uma técnica que prevê a automação do desenvolvimento dos recursos para a geração e interpretação de língua natural (ver Figura 12). A LSF postula que a GLN é a produção automática de enunciados em dada língua natural, a partir da entrada de alguma representação de informação

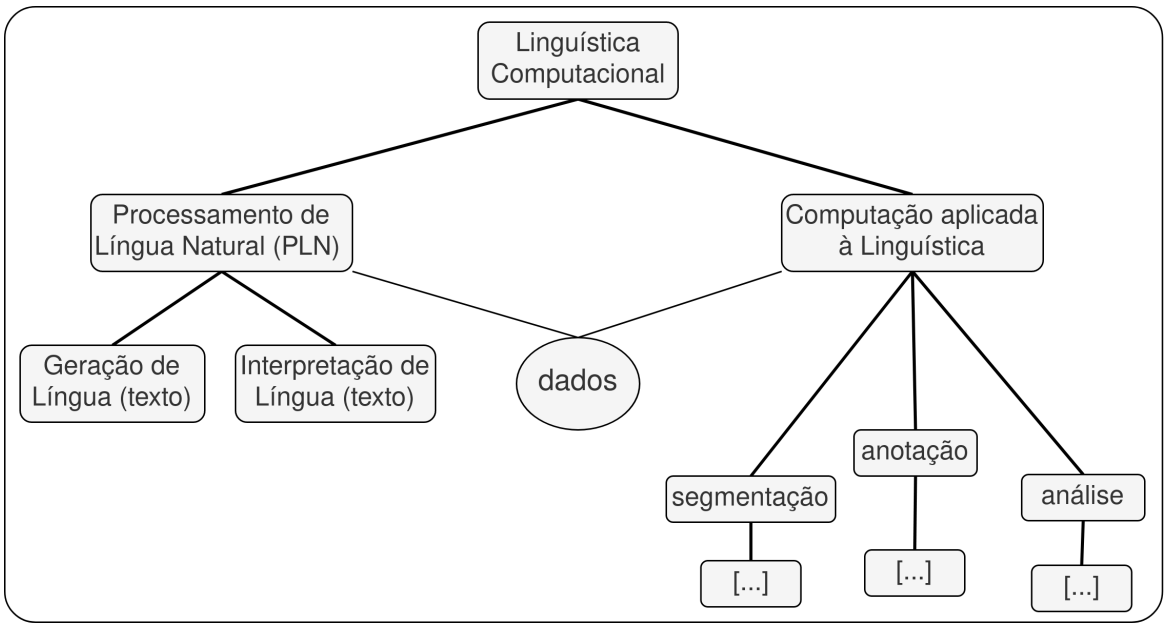
[12] *Computational linguistics* is the scientific study of language from a computational perspective. Computational linguists are interested in providing computational models of various kinds of linguistic phenomena.

[13] **Disponível em:** https://www.aclweb.org/portal/what-is-cl

[14] **Tradução de:** Natural Language Generation (NLG) is the subfield of artificial intelligence and computational linguistics that focuses on computer systems that can produce understandable texts in English or other human languages. Typically starting from some nonlinguistic representation of information as input, NLG systems use knowledge about language and the application domain to automatically produce documents

inteligível para o computador (dados; texto; ou outras formas abstratas de representações de informações, comumente carregando alguma intenção comunicativa)(BATEMAN et al., 1999; BATEMAN; ZOCK, 2012; TEICH, 1999a).

Figura 12 – Mapeamento da Linguística Computacional na LSF



Fonte: traduzido e adaptado de Halliday e Webster (2009, p. 250).

Iniciativas de modelagem de recursos computacionais para a geração de textos, amparadas no modelo Sistêmico-Funcional, remontam ao século passado, tendo início com a implementação computacional da gramática do inglês, NIGEL, na tradição de desenvolvimento do sistema de geração de sentenças na língua inglesa, o *PENMAN* (MATTHIESSEN; MANN, 1985). Esse sistema tem como sucessores os ambientes de desenvolvimento de recursos linguísticos para a geração de língua natural, o *KPML* (*KOMET-Penman Multi-Lingual*), uma plataforma para engenharia linguística de larga escala e orientado à geração multilíngue, o qual, dadas especificações em nível semântico como arquivo de entrada, gera sentenças, através de um módulo lexicogramatical implementado, e o *MULTEX*. Atualmente o *KPML* é mantido na faculdade de Linguística e Literatura, na Universidade de Bremen, e tem recursos desenvolvidos para uma variedade de línguas, incluindo inglês, alemão, holandês, chinês, espanhol, português dentre outras (BATEMAN, 1996; BATEMAN, 1997; BATEMAN et al., 1991; BATEMAN et al., 1999; BATEMAN; ZOCK, 2012; MATTHIESSEN et al., 1998; MATTHIESSEN; BATEMAN, 1992).

A inserção da GLN no escopo da LSF, como discorre Teich (1999a, p. 55), é uma resposta à necessidade de se empregarem, no Processamento de Língua Natural, subsídios (teóricos e descritivos) provenientes de uma teoria de linguagem bem estabelecida para o desenvolvimento dos recursos de geração, pois são teorias linguísticas que se ocupam de questões particulares da produção de significado através da linguagem. Nesse sentido,

um modelo de linguagem robusto, como a Linguística Sistêmico-Funcional, pode oferecer a base para a resolução de problemas comuns em sistemas de processamento de língua natural, e reivindica para o escopo da Linguística um papel mais ativo, abrangente, na área de Linguística Computacional.

A subseção 2.3.1 apresenta os conceitos que básicos que constituem arquiteturas tradicionais para a geração de língua natural.

2.3.1 Arquitetura tradicional de sistemas de Geração de Língua Natural

Tradicionalmente, a tarefa de GLN converte uma representação não linguística em texto ou voz a partir de subsequentes módulos, que transformam a entrada. Esta arquitetura clássica de GLN, que se assemelha a uma linha de produção de textos, recebe o nome de *pipeline*. O Quadro 2 abaixo resume a arquitetura clássica desenvolvida por Reiter e Dale (2000).

Quadro 2 – Módulos gerais de sistemas de geração de língua natural

Macroplanejamento (*macroplanning*) - Planejamento do documento	Seleção de conteúdo (*content determination*)	Ordenamento do discurso (e estruturação do texto) (document structuring)
Microplanejamento (*microplannig*)	-Lexicalização e gramaticalização (*lexicalization-grammaticalization*); -Geração de expressões de referência (*referring expression generation*);	Agregação lógica (aggregation)
Realização superficial (*surface realization*)	Realização linguística (*linguistic realization*)	Realização estrutural (*structure realization*)
Formatação	Diagramação, pontuação (...) apresentação física	

Fonte: traduzido e adaptado de Reiter e Dale (2000) e Bateman e Zock (2012) (grifo meu).

Como verifica-se no Quadro 2, os sistemas de GLN são divididos em quatro grandes módulos (macroplanejamento, microplanejamento, realização superficial e formatação). A seguir, explicaremos cada um destes macro-módulos, como proposto por Reiter e Dale (2000), e exemplificaremos o seu funcionamento através da linha de produção do robô-jornalista @DaMataReporter, um sistema que gera relatórios sobre o desmatamento da Amazônia Legal. A arquitetura detalhada do @DaMataReporter foi descrita em Campos et al. (2020) e Rosa et al. (2020). Os *corpora* anotados e código estão disponíveis publicamente em repositório na internet[15]. As postagens geradas pelo @DaMataReporter podem ser acessadas no *Twitter*[16].

Seguindo a arquitetura tradicional, o @DaMataReporter é subdividido em três módulos principais e os submódulos correspondentes, organizado hierarquicamente em uma

[15] Disponível em: https://github.com/BotsDoBem/DEMO_INPE_COVID
[16] https://twitter.com/DaMataReporter

estrutura pipeline: o módulo de **macroplanejamento**: mais especificamente submódulos de **seleção de conteúdo, ordenamento do discurso** e **estruturação do texto**; módulo de **microplanejamento**: mais especificamente os submódulos de **lexicalização** e **geração de expressões de referência**; e, por fim, o módulo de **realização superficial–textual**. A Figura 13 dispõe a estrutura em *pipeline* do robô jornalista @DaMataReporter em um diagrama. A seguir, serão explicados cada um desses macro-módulos, como proposto por Reiter e Dale (2000) e transcorreremos toda a linha de produção do @DaMataReporter, apontando o conteúdo de entrada e de saída de cada um dos submódulos.

Figura 13 – Arquitetura do robô jornalista @DaMataReporter

Fonte: traduzido e adaptado de Rosa et al. (2020, p. 2)

2.3.1.1 Macroplanejamento: seleção de conteúdo, ordenamento do discurso e estruturação do texto

O módulo de Macroplanejamento da arquitetura clássica de GLN é responsável pelo planejamento do documento textual a ser gerado. Como explicitado no Quadro 2, o módulo divide-se em duas tarefas: seleção de conteúdo e estruturação geral do documento.

2.3.1.1.1 Seleção de conteúdo

Como apontam Reiter e Dale (2000, p. 96), a seleção de conteúdo se resume em "uma tarefa na qual se decidem as informações a serem comunicadas pelo produto da geração, e que pode ser interpretada como o aspecto de conteúdo do dentro do submódulo de **planejamento do documento**" . Esse conteúdo, apontam os pesquisadores, pode ser proveniente da organização interna do próprio sistema de geração ou da aplicação

que o hospeda. A escolha das informações relevantes para a geração é dependente de alguns fatores, destacando-se: distinção entre **objetivos comunicacionais**; adequação do conteúdo a determinados públicos-alvo; restrições quanto à formatação/tamanho do produto da geração; e a fonte de informação (representação do conhecimento) subjacente, que oferece os subsídios para a geração: a relevância do que será gerado dependera do conteúdo da fonte de informações para a geração. De maneira geral, o papel restritivo de cada um destes fatores será dependente do domínio ao qual a aplicação é associada, ou seja, as informações que são selecionadas para a geração de linguagem natural em contextos jornalísticos esportivos, serão diferentes de contextos de geração de documentos com previsão do tempo, sob todos os fatores.

Como descrito em Campos et al. (2020), no contexto do @DaMataReporter, o submódulo é responsável por escolher as intenções de fala relevantes a serem realizadas textualmente, dados estímulos específicos detectados no banco de dados pelo algoritmo de seleção de conteúdo. Essa fase na linha de produção é dividida em três subtarefas: **ingestão de dados, análise de dados** e **interpretação de dados**.

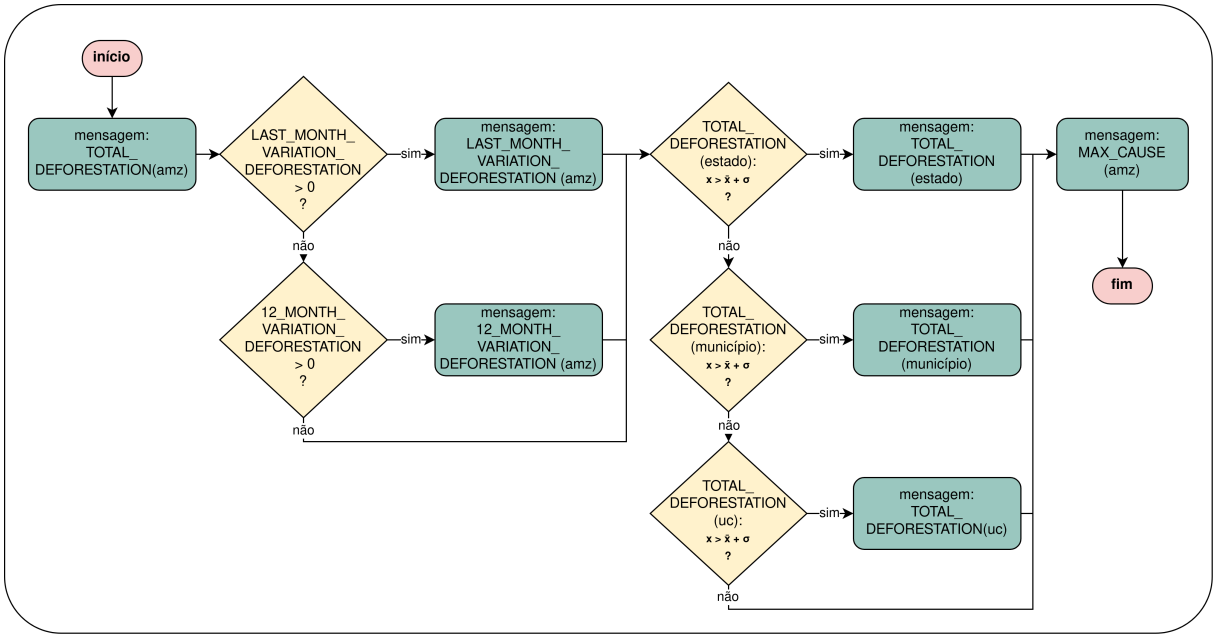
A tarefa de **ingestão de dados** é realizada automaticamente: o sistema busca dados brutos oficiais do DETER, um sistema desenvolvido pelo Instituto Nacional de Pesquisas Espaciais (INPE), que faz "levantamento rápido de alertas de evidências de alteração da cobertura florestal na Amazônia"[17] e disponibiliza os dados publicamente[18]. Os dados são salvos em um banco de dados e são disponibilizados para a entrada da tarefa seguinte. O sistema extrai dados para postagens mensais e diárias. Para as postagens mensais, o sistema acessa o banco de dados diariamente, buscando pelos dados consolidados do mês anterior. Se os dados estiverem disponíveis, o robô seleciona os campos relevantes para passar às próximas fases. No caso das postagens diárias, o sistema acessa o banco diariamente e busca por municípios ou áreas protegidas que tenham as taxas mais elevadas de desmatamento e que ainda não tenham sido selecionadas para postagem no mês sob análise.

O algoritmo de **análise** extrai conteúdos-chave/objetivos comunicacionais (*key facts*) sobre o desmatamento no amazônia legal: objetivos comunicacionais contêm dados de extensão, em quilômetros quadrados, de área desmatada no mês e a variação mensal e anual. A extração de conteúdos-chave leva em consideração o local (município, Área Protegida), principais causas de desmatamento (mineração, corte raso etc.), além de dados da data (ano, mês, dia).

Os objetivos comunicacionais extraídos na fase de análise são então interpretados pelo algoritmo, como mostra a Figura 14 com uma árvore de decisão para a interpretação de um exemplo com dados mensais.

[17] DETER — Coordenação-Geral de Observação da Terra (inpe.br)
[18] Disponível em: http://terrabrasilis.dpi.inpe.br/homologation/file-delivery/download/deter-amz/daily

Figura 14 – Árvore de decisão - interpretação dados mensais



Fonte: Traduzido e adaptado de Campos et al. (2020, p. 5)

A Figura 15 apresenta a entrada e saída do processo de seleção de conteúdo, após a passagem pelo algoritmo de análise e interpretação, com um exemplo para dados diários.

Figura 15 – Entrada e saída – Seleção de conteúdo

Estímulo: *Atualização dos dados de desmatamento do sistema (DETER)*

⇓

```
CAUSE(cause="MINERACAO",city="None",location="deter-amz",month="3",state="AC",
uc="RESERVA EXTRATIVISTA CHICO MENDES", year="2020")
DAILY_ALERT(area="5.91",city="None",daily_accumulation="8",day="25",location=
"deter-amz",month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES"
,year="2020")
TOTAL_DEFORESTATION(area="10.52",city="None",location="deter-amz",month="3",
state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
```

Fonte: adaptada de Rosa et al. (2020)

A Figura 15 apresenta os objetivos comunicacionais para alerta diário (*DAILY_ALERT*) referente a uma Área Protegida (i.e., Unidade de Conservação), com valores para os respectivos atributos: área (*area*), dias acumulados com desmatamento (*daily_accumulation*) etc.; *TOTAL_DEFORESTATION* e atributos/valores: área (*area*), local *location* área protegida (uc) etc.; e *CAUSE* e os valores respectivos: causa (*cause*), área protegida (uc) etc. A saída desta fase na linha de produção será alimentada à próxima fase, o ordenamento do discurso, como veremos na subseção 2.3.1.1.2.

2.3.1.1.2 Ordenamento do discurso e estruturação do texto

O submódulo de **ordenamento do discurso (e estruturação do texto)** (*document structuring*), como o próprio nome já aponta, é responsável por estabelecer um ordenamento estrutural coerente às informações que o sistema julga relevantes para a geração, dados o objetivo comunicacional, público-alvo e tamanho esperado para o produto da geração. Essa organização estrutural do texto, de forma geral, se articula hierarquicamente e pode ser orientada por diferentes critérios: como uma organização temporal, sequencial, ou diferentemente de acordo com o que os autores chamam de "relações discursivas (*discourse relations*)", a depender dos fatores já apontados para o módulo de determinação de conteúdo (REITER; DALE, 2000, p. 101). Uma maneira de abordar a estruturação do documento, adotada por iniciativas sob a perspectiva da LSF, é o modelo de estruturas retóricas (*Rhetorical Structure Theory - RST*) (BATEMAN; ZOCK, 2012).

Ao definirem este submódulo, Reiter e Dale (2000, p. 80) apontam que, dependendo do tipo da aplicação, o macroplanejamento é considerado o de maior importância. Os autores argumentam que se as informações selecionadas como relevantes para a geração forem organizadas coerentemente, uma maior adesão di público-alvo é garantida, pois ele tende a não se concentrar em possíveis deficiências decorrentes de operações nos outros submódulos, que são responsáveis pela realização superficial do texto (e.g. organização sintática e mecanismos coesivos de referência). Por outro lado, se o sistema seleciona informações não relevantes e sua organização geral no texto é realizada sem coerência, o produto não tem potencial de uso por humanos, independente da qualidade de organização lexicogramatical das orações individualmente.

No contexto do @DaMataReporter, dados os objetivos comunicacionais (*key-facts*), que são a saída da fase de seleção de conteúdo, a subtarefa de ordenamento do discurso é, então, responsável por organizar as mensagens na ordem que serão alimentadas à fase de estruturação do texto (CAMPOS et al., 2020, p. 6). A Figura 16 apresenta a entrada e saída da etapa de ordenamento do discurso.

Como pode-se verificar na Figura 16, os objetivos comunicacionais, ou mensagens, selecionadas na fase de seleção de conteúdo são organizadas como serão apresentadas no texto gerado. Nesse caso específico, a as mensagens são ordenadas na sequência **DAILY_ALERT**, **TOTAL_DEFORESTATION**, e **CAUSE**. A saída desta fase, ou seja, as mensagens ordenadas, é, por sua vez, alimentada à próxima fase da linha de produção, a estruturação do texto.

Figura 16 – Entrada e saída – Ordenamento do discurso

Realiza o ordenamento das intenções de fala de acordo com o formato esperado no texto final gerado pelo robô

```
CAUSE(cause="MINERACAO",city="None",location="deter-amz",month="3",
state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES", year="2020")
DAILY_ALERT(area="5.91",city="None",daily_accumulation="8",day="25",location=
"deter-amz",month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",
year="2020")
TOTAL_DEFORESTATION(area="10.52",city="None",location="deter-amz",
month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
```

```
DAILY_ALERT(area="5.91",city="None",daily_accumulation="8",day="25",location=
"deter-amz",month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",
year="2020")
TOTAL_DEFORESTATION(area="10.52",city="None",location="deter-amz",
month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
CAUSE(cause="MINERACAO",city="None",location="deter-amz",month="3",
state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
```

Fonte: adaptada de Rosa et al. (2020)

A subtarefa de **estruturação do texto** é diretamente relacionada com o ordenamento do discurso, recebendo as mensagens ordenadas e realizando a sua estruturação em parágrafos e sentenças (CAMPOS et al., 2020, p. 6). A Figura 17 apresenta a entrada e resultado dessa etapa.

Como pode-se verificar na Figura 17, a saída da etapa de estruturação do texto retorna a organização em parágrafo e as respectivas sentenças, através das *tags* (*<paragraph> e <sentence>*). Essa organização será, por sua vez, alimentada à próxima fase da linha de produção, a **lexicalização**, como veremos na subseção 2.3.1.2.1 a seguir.

Figura 17 – Entrada e saída – Estruturação do texto

Faz a estruturação das intenções de fala em parágrafos e sentenças:

```
DAILY_ALERT(area="5.91",city="None",daily_accumulation="8",day="25",location=
"deter-amz",month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
TOTAL_DEFORESTATION(area="10.52",city="None",location="deter-amz",
month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
CAUSE(cause="MINERACAO",city="None",location="deter-amz",month="3",
state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
```

⇓

```
<PARAGRAPH>
    <SENTENCE>
        DAILY_ALERT(area="5.91",city="None",daily_accumulation="8",
        day="25",location="deter-amz",month="3",state="AC",uc="RESERVA
        EXTRATIVISTA CHICO MENDES",year="2020")
    </SENTENCE>
    <SENTENCE>
        TOTAL_DEFORESTATION(area="10.52",city="None",location="deter-amz",
        month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
    </SENTENCE>
    <SENTENCE>
        CAUSE(cause="MINERACAO",city="None",location="deter-amz",month="3",
        state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
    </SENTENCE>
</PARAGRAPH>
```

Fonte: adaptada de Rosa et al. (2020)

Na subseção 2.3.1.2 a seguir, será apresentado o submódulo de microplanejamento e suas subtarefas.

2.3.1.2 Microplanejamento: Lexicalização, Geração de expressões de referência e Agregação lógica

Uma vez finalizadas as tarefas do módulo de macroplanejamento - a seleção de conteúdo e a estruturação do documento a ser verbalizado - dá-se sequência na linha de produção executando-se o módulo de microplanejamento, que é responsável pelo planejamento das sentenças do documento. De acordo com Reiter e Dale (2000), o microplanejamento compreende a **lexicalização** (*lexicalization*); **geração de expressões de referência** (*referring expression generation*); **agregação lógica** (*aggregation*).

Antes de explorar-se cada uma das tarefas desse submódulo, vale apontar-se um panorama geral de como os recursos desenvolvidos nesta pesquisa se enquadram no desenho tradicional de módulos de sistemas de geração, em específico no de **microplanejamento**. Como já foi reiterado ao longo do texto até este ponto, no presente trabalho propõe-se desenvolver recursos restritos à realização superficial da oração. Mais especificamente, se restringe à modelagem dos recursos lexicogramaticais para realização textual, desde a unidade da ordem do morfema até a unidade da ordem da oração. Em outras palavras, o

módulo visa a permitir a travessia nos sistemas, que culmina em uma saída do material linguístico que realiza as escolhas cumulativas para os sistemas inerentes a cada uma das unidades na escala de ordens, até a unidade da oração.

Desta maneira, reitera-se que o módulo de recursos linguísticos desenvolvido na presente pesquisa contempla a escala de ordens do português brasileiro (oração–grupo/frase–palavra–morfema) e, portanto, não se compromete a explorar particularmente os recursos que dizem respeito à organização geral de potenciais documentos a serem gerados por sistemas de geração. Nesse aspecto, limita-se a apresentar a arquitetura do robô-jornalista, relacionando-a à arquitetura geral de sistemas de geração, incluindo os submódulos apresentados nesta seção. Assim, o presente trabalho não explora a modelagem e implementação recursos de coesão no módulo de realização textual baseado em regras, como é o caso da tarefa de geração de **expressões de referência**, já que a coesão se trata de recurso textual não estrutural, que extrapola os limites da unidade da oração.Contudo, o realizador textual baseado modela recursos lexicogramaticais responsáveis pela realização de elementos que, por sua vez, são potencialmente recuperados coesivamente, como, por exemplo, através de rastreamento de participantes, como é o caso de recursos lexicogramaticais realizados por pronomes, determinantes etc.

O realizador textual desenvolvido nesta tese também não explora formas automáticas de **agregação lógica**, pois se restringe à realização da oração, e, portanto, não extrapola para o complexo oracional. Contudo, aspectos relativos a esta subtarefa são modelados no robô jornalista que serve de plataforma de aplicação para o módulo de recursos lexicogramaticais desenvolvido no presente trabalho. A tarefa de **agregação lógica** é realizada na fase estruturação do documento, adotando as configurações apresentadas por Ferreira et al. (2019).

2.3.1.2.1 Lexicalização

De acordo com Reiter e Dale (2000), a **lexicalização** é a subtarefa responsável por atribuir elementos lexicais e sintáticos às mensagens, que são provenientes da etapa de planejamento geral do documento (texto) no módulo de macroplanejamento. Esta etapa pode ser realizada de maneiras distintas, segundo os autores, a depender do sistema. No âmbito do robô-jornalista @DaMataReporter, esta etapa, como apresentam Campos et al. (2020, p. 7), é responsável por selecionar o texto (frases e palavras) para verbalizar os objetivos comunicacionais estruturados na fase anterior e segue uma abordagem de *templates*, segundo a abordagem apresentada por Ferreira et al. (2019 apud CAMPOS et al., 2020, p. 7). Além da seleção dos textos que irão verbalizar as intenções, são utilizadas *tags* (e.g., *TOTAL_DEFORESTATION, CAUSE*), que serão substituídas pelos valores ingeridos e selecionados nas fontes de dados na etapa de ingestão dos dados, para os respectivos

atributos (selecionados e interpretados na fase de seleção de conteúdo). A Figura 18 apresenta a entrada e saída para esta fase na linha de produção do @DaMataReporter.

Figura 18 – Entrada e saída – Lexicalização

Faz escolhas sintáticas e de vocabulário para realizar as intenções de fala:

```
<PARAGRAPH>
    <SENTENCE>
        DAILY_ALERT(area="5.91",city="None",daily_accumulation="8",
        day="25",location="deter-amz",month="3",state="AC",uc="RESERVA
        EXTRATIVISTA CHICO MENDES",year="2020")
    </SENTENCE>
    <SENTENCE>
        TOTAL_DEFORESTATION(area="10.52",city="None",location="deter-amz",
        month="3",state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
    </SENTENCE>
    <SENTENCE>
        CAUSE(cause="MINERACAO",city="None",location="deter-amz",month="3",
        state="AC",uc="RESERVA EXTRATIVISTA CHICO MENDES",year="2020")
    </SENTENCE>
</PARAGRAPH>
```

⇓

```
<PARAGRAPH>
    <SENTENCE>
        Em DAY_1 de MONTH_1 de YEAR_1, INSTITUTE_1 VP[...] registrar NN[...] alerta
        de desmatamento de AREA_1 em UC_1 .
    </SENTENCE>
    <SENTENCE>
        A Unidade de Conservação VP[...] acumular AREA_1 de
        desmatamento em MONTH_1 de YEAR_1 .
    </SENTENCE>
    <SENTENCE>
        A principal causa de desmatamento VP[...] ser a CAUSE_1 .
    </SENTENCE>
</PARAGRAPH>
```

Fonte: adaptada de Rosa et al. (2020)

Como mostra a Figura 18, as intenções de fala estruturadas na fase anterior são lexicalizadas de acordo com os *templates*. Além disso, identificam-se as *tags* que serão substituídas por valores recuperados nas intenções de fala (*AREA, CAUSE, MONTH* etc.), bem como *tags* do tipo *VP[]*, com parâmetros que serão necessários para a flexão verbal. A saída desta fase da linha de produção é então passada à próxima fase, a geração de expressões de referência (subseção 2.3.1.3).

2.3.1.3 Geração de expressões de referência

Como apresentam Campos et al. (2020, p. 7), esta etapa é responsável por substituir as *tags* de entidades e realizar as respectivas mudanças de parâmetros de pessoa, gênero e número. Seguindo o exemplo apresentado, a Figura 19 apresenta a entrada e saída para esta fase.

A saída desta fase na linha de produção é alimentada à próxima fase, a realização textual, como apresenta a subseção 2.3.1.5.

Figura 19 – Entrada e saída – Geração de expressões de referência

Gera expressões de referência para valores dos atributos das intenções de fala:

```
<PARAGRAPH>
    <SENTENCE>
        Em DAY_1 de MONTH_1 de YEAR_1, INSTITUTE_1 VP[...] registrar NN[...] alerta
        de desmatamento de AREA_1 em UC_1 .
    </SENTENCE>
    <SENTENCE>
        A Unidade de Conservação VP[...] acumular AREA_1 de
        desmatamento em MONTH_1 de YEAR_1 .
    </SENTENCE>
    <SENTENCE>
        A principal causa de desmatamento VP[...] ser a CAUSE_1 .
    </SENTENCE>
</PARAGRAPH>
```

⇓

```
<PARAGRAPH>
    <SENTENCE>
        Em 25 de março de 2020, o Instituto Nacional de Pesquisas
        Espaciais(INPE) VP[aspect=
        simple,tense=past,voice=active,person=3rd,number=singular]
         registrar NN[...]alerta de
        desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO
        MENDES/AC.
    </SENTENCE>
    <SENTENCE>
        A Unidade de Conservação VP[aspect=simple,tense=past,voice=active,
        person=3rd,number=
        singular] acumular 10.52 km2 de desmatamento em março de 2020.
    </SENTENCE>
    <SENTENCE>
        A principal causa de desmatamento VP[...] ser a mineração.
    </SENTENCE>
</PARAGRAPH>
```

Fonte: adaptada de Rosa et al. (2020)

2.3.1.4 Agregação lógica

A terceira tarefa dentro do módulo de microplanejamento é a **agregação lógica** (*aggregation*), uma tarefa de estruturação dentro deste módulo. De acordo com Reiter e Dale (2000, p. 123), de maneira geral, a tarefa de agregação lógica toma a saída do mapeamento de mensagens com as especificações proto-frásicas e analisa seu potencial de agregação, distribuindo, assim, mensagens em sentenças. Os autores apontam que "a agregação é mais explorada no seu aspecto de formação de sentenças, mas pode ser explorada em termos de estruturação de parágrafos". Na formação de sentenças, "a combinação de mensagens em apenas uma sentença pode ser ancorada em mecanismos linguísticos". A agregação então pode ser operacionalizada via: **conjunção simples** (*simple conjunction*); **conjunção via participantes compartilhados** (*conjuction via shared participants*); **conjunção via estruturas compartilhadas** (*conjunction via shared structures*); **encaixe sintático** (*syntactic embedding*).

Como apontado anteriormente, esta tese se limita ao desenvolvimento de recursos para a realização textual, contemplando a escala de ordens da lexicogramática e, portanto, a subtarefa de agregação lógica foge ao seu escopo. Contudo, no âmbito da linha de produção

do @DaMataReporter, esta tarefa é realizada no submódulo de macroplanejamento, estruturação do documento, de acordo com abordagem apresentada por Ferreira et al. (2019) (ver também Campos et al. (2020)).

A subseção 2.3.1.5 apresenta o submódulo de **realização superficial**.

2.3.1.5 Realização superficial

As subseções anteriores apresentaram os submódulos de macro e microplanejamento. Nesta subseção será apresentado o submódulo de **realização superficial** (*surface realization*), que é objeto específico desta tese. Mais especificamente a subtarefa de realização textual do submódulo que, de forma geral, é subdividido em dois aspectos: estrutural, responsável por mapear as especificações estruturais do documento aos recursos disponíveis para o meio de apresentação do documento; e a realização textual (ou realização linguística), responsável por tomar uma configuração abstrata, saída do módulo de microplanejamento (a mensagem), e mapeá-la em construções gramaticais (e.g., pronomes, verbos, realizando Sujeito, Processo, Complemento etc.), realizar as estruturas de acordo com as opções nos sistemas, inserir palavras 'funcionais' (e.g., preposições), determinar as flexões onde se aplicam (e.g., morfologia do verbo e dos substantivos).

No âmbito do @DaMataReporter, o submódulo de realização superficial toma as últimas decisões para a realização do texto: realiza a flexão verbal de acordo com os parâmetros correspondentes, realiza contrações pertinentes, substitui os *tokens* com os valores das intenções de fala. A Figura 20 apresenta o fluxo de entrada e saída dessa fase, seguindo o exemplo apresentado desde as fases iniciais da linha de produção. Pode-se verificar a saída de material textual, com os verbos flexionados adequadamente e com os dados pertinentes recuperados na fase de seleção de conteúdo já inseridos convenientemente.

O realizador textual baseado em regras desenvolvido nesta tese, então, apresenta funções que realizam as tarefas estruturantes e de realização textual das unidades da escala de ordens do português brasileiro. Como já mencionado anteriormente, ele restringe-se à escala de ordens do português brasileiro e, portanto, não extrapola para funções estruturantes dos complexos oracionais (agregação lógica e ordenamento e estruturação do texto). Contudo, as funções do realizador textual baseado em regras possibilitam a organização estrutural e a realização textual de cada uma das unidades da escala de ordens, desde o funcionamento dos morfemas na constituição das palavras, até o funcionamento dos grupos que constituem e realizam a oração.

Ancorada no escopo da tradição Sistêmico-Funcional de *GLN*, apenas uma iniciativa de implementação de recursos gramaticais para a realização textual na geração com português brasileiro foi desenvolvida. Oliveira (2013) aplica, com sucesso, a *GUM* (*Generalized Upper Model*) – uma ontologia superior, orientada a tarefas gerais e independente de domínio, motivada linguisticamente, empregada no processamento de língua natural – para

a modelagem de domínio experiencial, de significados espaciais em português brasileiro. Em paralelo, desenvolve os recursos lexicogramaticais do português brasileiro para a geração de língua natural no sistema de geração *KPML* (*KOMET-Penman MultiLingual*[19]).

Figura 20 – Entrada e saída – Realização textual

Realiza as escolhas finais para realização textual: verbos, substantivos e adjetivos (e suas flexões adequadas):

```
<PARAGRAPH>
    <SENTENCE>
        Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE)
        VP[aspect=simple,tense=past,voice=active,person=3rd,number=singular]
        registrar NN[...]alerta de desmatamento de 5.91 km2 na RESERVA
        EXTRATIVISTA CHICO MENDES/AC.
    </SENTENCE>
    <SENTENCE>
        A Unidade de Conservação VP[aspect=simple,tense=past,voice=active,person
        =3rd,number=singular] acumular 10.52 km2 de desmatamento em março de 2020.
    </SENTENCE>
    <SENTENCE>
        A principal causa de desmatamento VP[...] ser a mineração.
    </SENTENCE>
</PARAGRAPH>
```

⇓

```
"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais (INPE)
registrou alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO
MENDES/AC. A Unidade de Conservação acumulou 10.52 km2 de desmatamento
em março de 2020. A principal causa de desmatamento foi a mineração."
```

Fonte: adaptada de Rosa et al. (2020)

Como Oliveira (2013) aponta, seu objetivo foi testar a aplicabilidade da ontologia superior (*GUM*) como sistema para a modelagem da representação de conhecimento, **base de ideação** (*ideation base*) para a geração de língua natural, em português brasileiro, e algumas alterações na estrutura da ontologia foram necessárias para contemplar o domínio explorado em seu estudo – textos turísticos. Contudo, na implementação dos outros recursos gramaticais para a geração, ou seja, **base de interação** (*interaction base*) e **base textual** (*text base*), reitera-se que Oliveira (2013) se limitou a um aspecto específico da produção, a espacialidade, e, portanto, não implementou recursos lexicogramaticais que se aproximasse do potencial disponível para o português brasileiro.

Oliveira (2013) argumenta que não visava a apresentar um módulo de recursos lexicogramaticais para a geração que contemplasse todo o potencial sistêmico do português brasileiro, e, portanto, toma a decisão metodológica de modelar os recursos restritos à geração das orações que ocorriam em seu corpus. Dessa forma, os recursos gramaticais desenvolvidos por Oliveira (2013), por serem restritos ao domínio de textos turísticos, enfocando espacialidade, não oferecem subsídios para uma geração independente de domínio. Em outras palavras, os recursos lexicogramaticais implementados pelo autor modelam subpotenciais sistêmicos (registros específicos) e não se orientam ao potencial do português

[19] Para mais detalhes sobre o KPML, ver Reiter e Dale (2000)

brasileiro para produção de significado.

O realizador textual baseado em regras desenvolvido neste trabalho foi motivada pela organização das opções gramaticais em forma de sistema, mas sua implementação não foi propriamente realizada em forma de redes de sistemas, como possibilitado no ambiente do *KPML*. A implementação do realizador textual baseado em regras, que modela a realização computacional dos sistemas lexicogramaticais, foi realizada através do desenvolvimento de estruturas condicionais em *Python*. Detalhes da implementação do módulo de recursos lexicogramaticais baseado em regras serão abordados mais detidamente no Capítulo 3, de Metodologia.

Na seção 2.4 a seguir, apresentaremos noções sobre o domínio experiencial construído pela linguagem, e mais especificamente o domínio experiencial realizado no @DaMataReporter.

2.4 DOMÍNIO EXPERIENCIAL – A BASE DE IDEAÇÃO (*IDEATION BASE*)

Um dos pressupostos básicos da Linguística Sistêmico-Funcional é o de que a linguagem não "reflete", mas "constrói" a nossa realidade semioticamente, sendo resultado da "permanente dialética presente entre o material e o semiótico na existência humana"(HALLIDAY, 2003). A linguagem humana atua como um dos sistemas semióticos, operando conjuntamente a sistemas como a paralinguagem (gestos e expressões faciais que acompanham a linguagem falada, qualidade da voz, timbre, ritmo e outros sistemas semióticos construídos pelo corpo), além de outros como a dança, música, desenho, pintura e arquitetura para a "construção" da nossa realidade (HALLIDAY; MATTHIESSEN, 2014, p. 33).

Como já abordado anteriormente, da perspectiva do modelo Sistêmico-Funcional de linguagem como semiótica social, proposto por Halliday (1978), a nossa realidade social é entendida como um conjunto de potenciais semióticos, que denomina-se cultura, no qual a linguagem ocupa papel central. O '**contexto**' é modelado em dois polos de abstração em termos de variáveis contextuais que delimitam a variação do potencial, passando por subpotenciais, até alcançar, no polo instancial, as variáveis instanciadas em textos: as variáveis de **campo** (*field*), organizada em domínio experiencial (assunto ou tópico), e a natureza da atividade, ou atividade sociossemiótica (ação ou reflexão); **sintonia** (*tenor*), ou seja, interlocutores envolvidos na atividade em termos de estabelecimento e manutenção de relações interpessoais em papeis institucionais, relação de poder entre os interlocutores etc.; e **modo** (*mode*) organização dos significados de campo e sintonia em unidades textuais (HALLIDAY; MATTHIESSEN, 2014). Para realizar esse trabalho, como já apontado anteriormente, o sistema linguístico desempenha três componentes funcionais gerais, os espectros metafuncionais, que organizam a construção semiótica

(semogênese), através das metafunções ideacional (experiencial e lógica), interpessoal, e textual, respectivamente. Tomemos apenas a perspectiva da metafunção ideacional (em seu componente experiencial), parâmetro contextual de campo. Sob o parâmetro contextual de campo, nossa experiência é construída semioticamente, de maneira que os acontecimentos tanto do mundo material, quanto do mundo de nossa consciência são construídos como significado, tendo como interface o (inter)estrato contextual (realizando a interface entre o mundo material e o semiótico). Esses significados são organizados em termos de atividades sociossemióticas, ou seja, as atividades sociais e semióticas nas quais os interlocutores se inserem, e o domínio experiencial, o 'assunto' ou 'tópico' dessas interações.

As **atividades sociossemióticas** nas quais indivíduos operam separam-se, em primeiro grau, em **ação** e **reflexão**; as atividades de reflexão, subdividem-se, em um grau secundário, em sete atividades. Em primeiro lugar, nós humanos interagimos no sentido de **fazer** (*doing*) em termos de colaboração e/ou comando de alguma atividade social, na qual a linguagem presta um papel **ancilar** (*auxiliar*), ou seja, a atividade não necessita de verbalização ostensiva, em nenhum meio ou modo. Além dessa atividade básica, essencial para a organização social, a linguagem é utilizada, em maior ou menor grau, para organizar o nosso mundo simbólico, realizando sete atividades secundárias: a linguagem é usada para **explicar** os fenômenos do mundo, criando taxonomias, categorizando os fenômenos, documentando e disseminando o conhecimento (

3 MÉTODOS

Como apresentado no Capítulo 1, o objetivo principal deste trabalho foi desenvolver (implementar) um módulo de realização textual/linguística baseado em regras independente de domínio, para aplicação em sistemas de GLN, contemplando a escala de ordens da lexicogramática do português brasileiro, com potencial de aplicação na linha de produção de sistemas de geração de língua natural (e.g., em textos representativos da atividade sociossemiótica de 'relatar', tipo de texto rotulado como reportagem jornalística, no domínio de desmatamento da Amazônia legal no território brasileiro). Com esse objetivo em perspectiva, o desenvolvimento desta pesquisa foi conduzido em módulos metodológicos inter-relacionados: **a)** desenvolvimento das funções que compõem o módulo de recursos lexicogramaticais baseado em regras para a realização textual do português brasileiro, independente de domínio; **b)** experimentos de desenvolvimento: experimentos de acurácia das funções de flexão verbal do módulo baseado em regras em um *corpus* de desenvolvimento (padrão ouro de flexão verbal); aplicação das funções de flexão verbal do realizador baseado em regras na flexão de verbos extraídos da base lexical do robô-jornalista @DaMataReporter, visando à verificação e aprimoramento das funções do módulo de recursos lexicogramaticais baseado em regras; **c)** experimentos: testes comparativos de acurácia entre o módulo baseado em regras e resultados alcançados por redes neurais (no âmbito do CoNLL-SIGMORPHON - 2017 e 2018 [1]) na tarefa de flexão verbal em um *corpus* de teste; **d)** aplicação do das funções do realizador baseado em regras na subtarefa de realização linguística (restrito à flexão verbal) em uma instância local do robô jornalista @DaMataReporter. A Figura 21 apresenta um fluxograma com uma visão geral do trajeto metodológico:

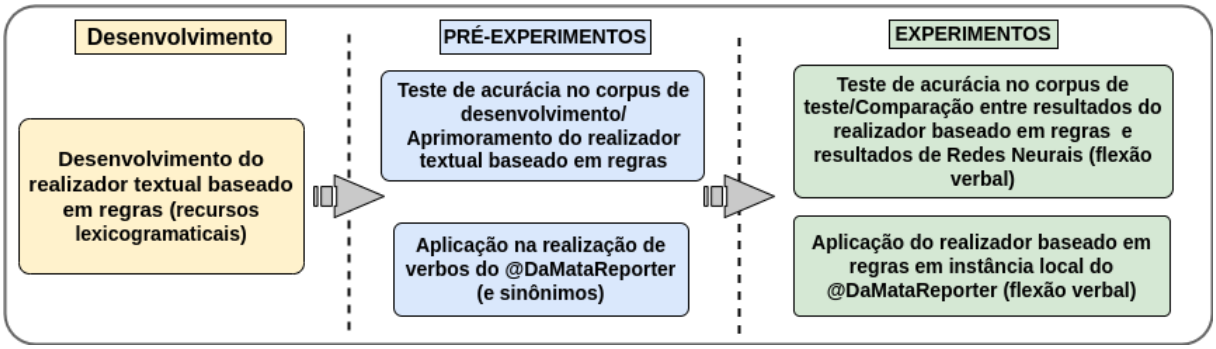
As seções a seguir apresentam cada uma das etapas metodológicas.

3.1 DESENVOLVIMENTO – REALIZADOR TEXTUAL BASEADO EM REGRAS: RECURSOS LEXICOGRAMATICAIS DO PORTUGUÊS BRASILEIRO

Nesta subseção, será apresentada uma síntese das propriedades da LSF e como elas balizaram a implementação do módulo de recursos lexicogramaticais baseado em regras para realização textual do português brasileiro independente de domínio. Dado o objetivo principal desta tese, foi possível programar os principais sistemas que organizam

[1] Como destacado na Capítulo 1, não foram desenvolvidos modelos de redes neurais no âmbito desta pesquisa. Os resultados do realizador baseado em regras desenvolvidas nesta pesquisa são comparados aos resultados dos modelos de redes neurais apresentados no âmbito das tarefas compartilhadas do CoNLL-SIGMORPHON - 2017 e 2018 - tarefa de flexão verbal no português brasileiro.

Figura 21 – Fluxograma–trajeto metodológico



Fonte: elaborada para fins deste estudo.

as unidades da escala de ordens da lexicogramática do português brasileiro [2].

Primeiramente, serão apresentados princípios gerais que motivam a lógica de programação das funções, através de exemplos da funções que realizam a unidade mais ampla da lexicogramática, a oração. Em seguida, serão apresentados, de maneira geral, os sistemas lexicogramaticais que guiaram a modelagem referente a cada uma das unidades da escala de ordens da lexicogramática: alguns sistemas serão abordados mais detidamente, ensejando a apresentação das funções programadas em linguagem *Python* que compõem o módulo.

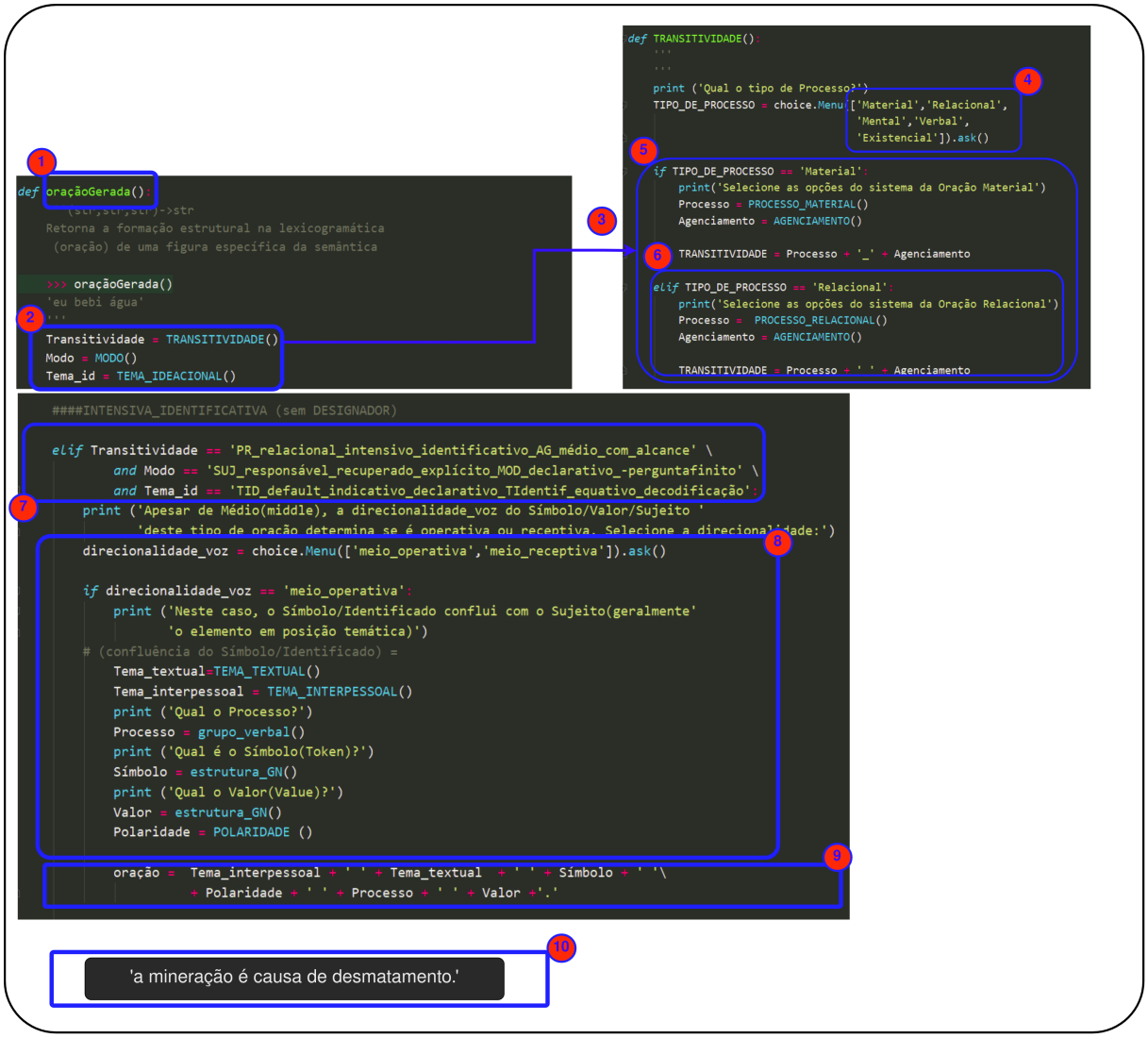
A organização Sistêmico-Funcional dos **eixos** (paradigmático e sintagmático), do espectro **metafuncional**, **estratificação** e **instanciação** balizaram a implementação computacional da lexicogramática, motivando a modelagem dos recursos de **sistema** e de **estrutura**, na confluência das perspectivas metafuncionais na construção da estrutura da oração, como é sintetizado na Figura 22, a seguir, e reiterado em *zooms*, nos recortes, na Figura 23 e Figura 24.

As redes de sistemas que organizam a oração (e as unidades inferiores na escala de ordens) são desenvolvidas em linguagem *Python*, para a modelagem dos recursos para o realizador textual superficial, utilizando estruturas computacionais condicionais e as estruturas linguísticas são a consequência, isto é, o resultado das escolhas nas estruturas condicionais. Essa abordagem de implementação se distingue da empregada na modelagem da gramática do inglês (e as adaptações implementadas por Oliveira (2013) para o português), no *KPML*, no qual a lexicogramática é modelada propriamente como rede de sistemas.

A Figura 23 apresenta a função de geração da oração em estágio inicial de desenvolvimento. No **índice 1**, a condição de entrada do sistema para a organização da estrutura da oração é implementada como uma função em *Python*. Essa condição de entrada implica

[2] Os módulos desenvolvidos para cada unidade da escala de ordens estão disponíveis em <https://github.com/AndreRosaLRT/NLG_BRAZILIAN_PORTUGUESE>

Figura 22 – Categorias da LSF e implementação dos recursos lexicogramaticais do português em *Python*



Fonte: elaborada para fins deste estudo.

em seleções nos sistemas de TRANSITIVIDADE, MODO e TEMA_IDEACIONAL, como mostra o **índice 2**, ou seja, contempla todo o espectro metafuncional. Por decisão metodológica, as escolhas para Tema textual e interpessoal são separadas no fluxo da função desenhada em *Python*, sendo apresentadas como escolhas posteriormente, mas mantendo a função e estrutura.

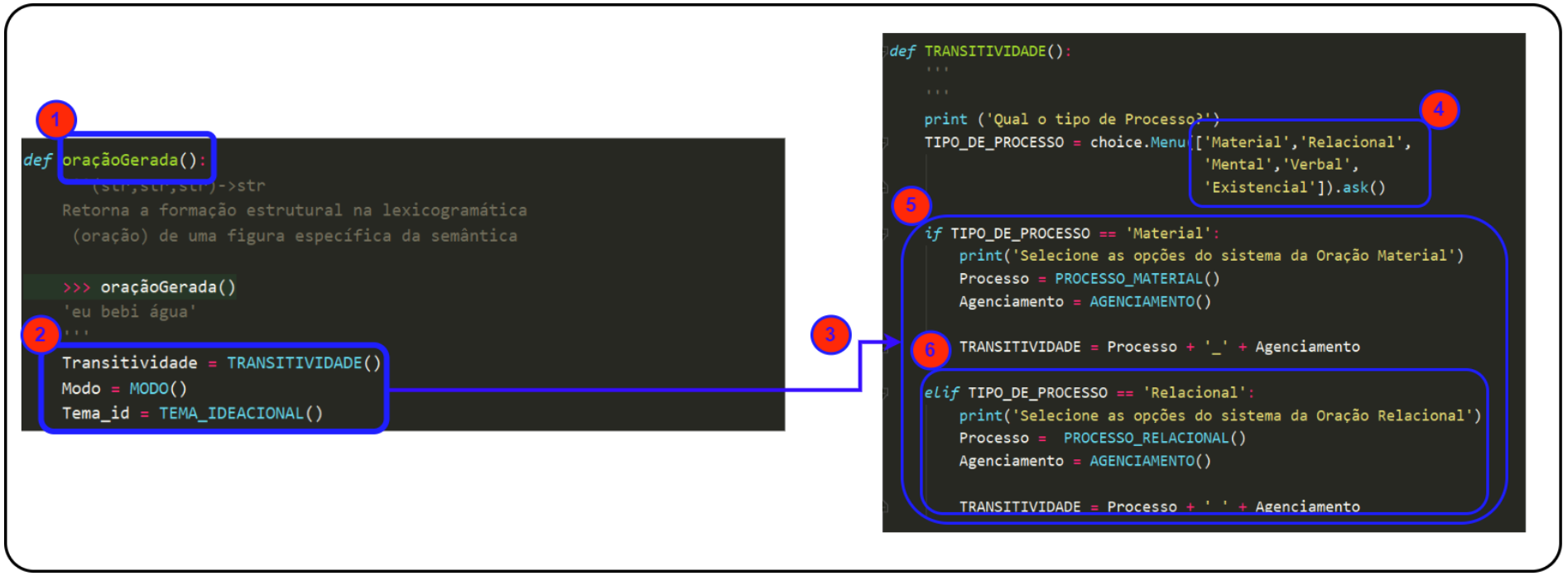
As funções são desenvolvidas em 'camadas aninhadas', partindo-se da unidade mais baixa na escala – morfema, passando por todas as escalas intermediárias – até a ordem superior da lexicogramática, a da oração. Por isso, cada função desenvolvida implica em várias outras funções aninhadas. O **índice 3** mostra a expansão das seleções no sistema de TRANSITIVIDADE. O sistema de TRANSITIVIDADE abre seleções em subsistemas de TIPO DE PROCESSO, com as opções de Material, Relacional etc., como mostra o **índice 4** na

Figura 22. Dada a opção por um tipo de Processo no sistema, abrem-se opções específicas para tipos de tipos de Processos.

O **índice 5** apresenta opções agnatas de TIPO DE PROCESSO, e as opções em subsistemas que as escolhas nesse sistema implicam. Por exemplo, como vê-se no **índice 6**, ao selecionar-se a opção 'Relacional', implicam-se escolhas específicas para o PROCESSO RELACIONAL e abrem-se também seleções no sistema de AGENCIAMENTO. Essas escolhas culminam na estrutura abstrata, resultante de escolhas nos subsistemas de TRANSITIVIDADE: escolhas para os TIPOS DE PROCESSO e AGENCIAMENTO.

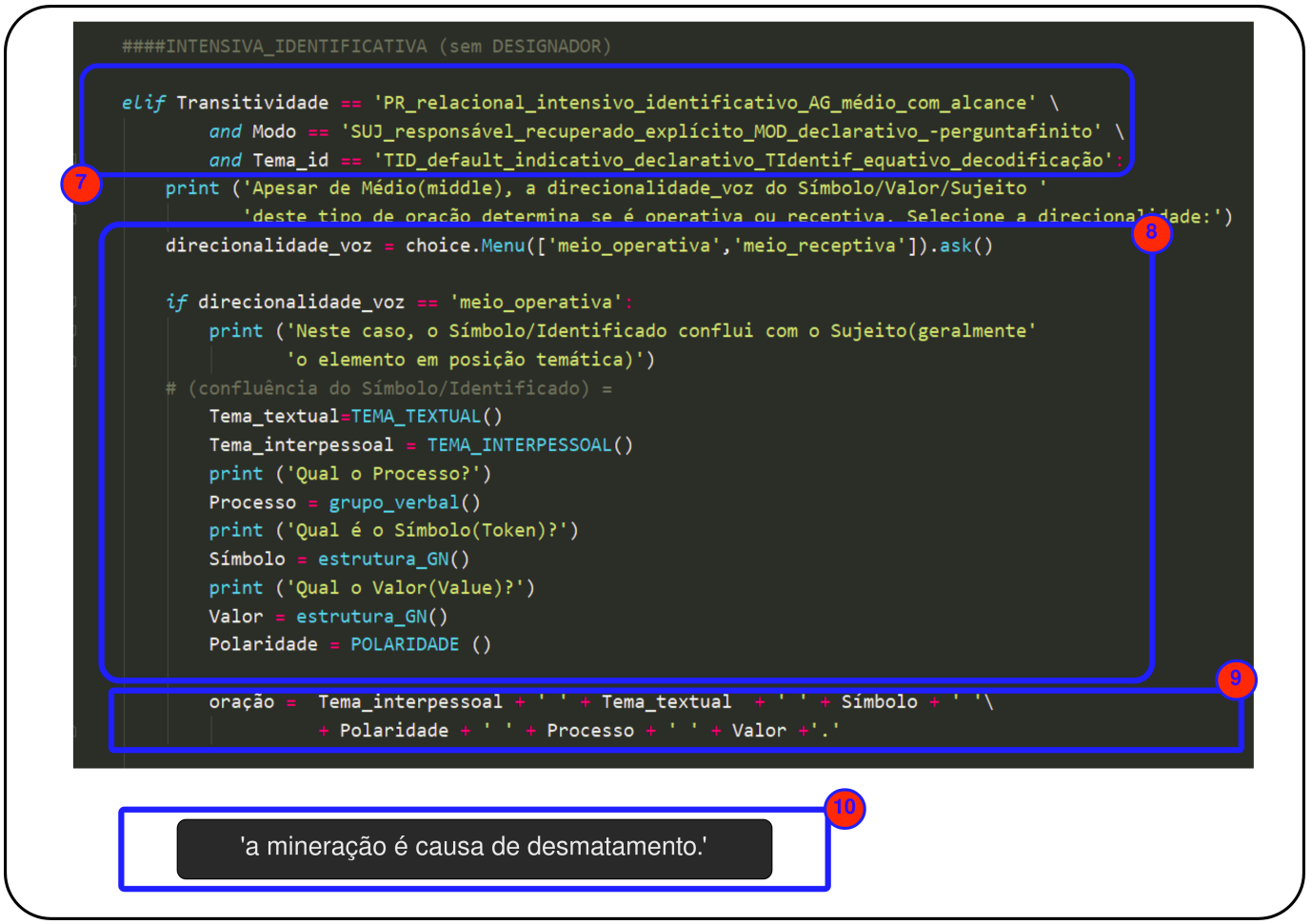
O **índice 7**, na Figura 24 apresenta um conjunto específico de escolhas, confluindo as três metafunções realizadas nos sistemas de TRANSITIVIDADE, MODO e TEMA(considerando as escolhas apontadas nos **índices 4, 5** e **6** na Figura 23). Essas escolhas, por sua vez, abrem opções apresentadas no **índice 8**.

Figura 23 – Categorias LSF e implementação (recorte1)



Fonte: elaborada para fins deste estudo.

Figura 24 – Categorias LSF e implementação (recorte2)



Fonte: elaborada para fins deste estudo.

O **índice 9** da Figura 24 apresenta a estrutura 'abstrata', na qual culminam todas as escolhas sistêmicas abordadas anteriormente. Essa estrutura 'abstrata' representa o que será renderizado na realização textual. Por fim, no **índice 10**, a oração é renderizada textualmente. Esse processo dinâmico com um exemplo de oração, ilustrado na Figura 22, representa o apanhado de escolhas sistêmicas necessárias para a construção/realização e renderização da oração "**a mineração é causa de desmatamento**".

Em suma, são abordadas: **metafunção**: confluência ideacional, interpessoal e textual na construção da oração, a partir da condição de entrada; **sistema**: modelado como estruturas condicionais (*if/else*) em *Python*; **estrutura**: organizada de forma 'abstrata' e aguardando renderização – realização estrutural no material textual. Como já foi apontado, dada uma condição de entrada, subsistemas aninhados são selecionados, até a ordem do morfema na escala de ordens: assim é realizada a organização das unidades na escala de ordens no desenvolvimento do realizador textual. No que diz respeito à dimensão de **estratificação**, o estrato diretamente modelado é o estrato lexicogramatical. Contudo, a renderização, ou seja, a 'saída' do material linguístico, se localiza no estrato de **expressão**. Por fim, a instanciação não é diretamente modelada, mas subentende-se um ponto próximo ao polo potencial do contínuo, dado que o realizador modela o potencial para a realização textual independente de domínio.

A seguir serão apresentadas, para cada unidade da escala de ordens, algumas das funções desenvolvidas de acordo com a descrição disponível, quando se aplica. Também serão apresentados exemplos do funcionamento do módulo de realização textual baseado em regras. O desenvolvimento das funções, isto é, a estrutura das funções desenvolvidas, segue os padrões de desenvolvimento apresentados anteriormente, realizando os conceitos básicos da LSF (unidade, sistema, ordem, classe etc.).

1) **Ordem da palavra**

a)**Palavras verbais_verbos**:

Como descrito por Sá (2016), o verbo pode ser definido pela perspectiva trinocular: 'de baixo', o verbo é constituído pelos morfemas verbais – morfema experiencial (ME), morfema interpessoal (MI), e morfema lógico-semântico (MLS); 'ao redor', é organizado conforme os sistemas de TIPO DE EXPERIÊNCIA, ORIENTAÇÃO INTERPESSOAL e MODIFICAÇÃO DA EXPERIÊNCIA; 'de cima', se agrupam formando classes: lexical (função de Evento), auxiliar (função de Auxiliar) e modal (função de Modal) e todos podem encerrar a função de Núcleo. As seleções segundo esses parâmetros culminam na realização dos verbos no português.

Com base nessa descrição, foram modelados computacionalmente: os morfemas que constituem o verbo – morfema experiencial (ME) e morfema interpessoal (MI); sistemas que organizam o verbo: o sistema de ORIENTAÇÃO INTERPESSOAL, sistema de TIPO DE

EXPERIÊNCIA; foram desenvolvidas, também, funções que levam em consideração a visão 'de cima' do verbo, isto é, de acordo com os tipos de verbo e a função que encerram no grupo verbal: função de Evento, de Modal, e de Auxiliar e Núcleo.

O primeiro exemplo de função implementada será a de formação da estrutura do verbo. A Figura 25, a seguir, é um conjunto de capturas de tela com fatias da função geral que organiza a estrutura do verbo no português brasileiro. Tomando como exemplo o verbo 'desmatar', serão apresentados o processo e saída, dados parâmetros específicos.

Figura 25 – Função–realização do verbo no português brasileiro

Fonte: elaborada para fins deste estudo.

A seguir são apresentadas as etapas da função e um exemplo de sua execução, tendo como exemplo de entrada o lema **'desmatar'**, e os parâmetros para a conjugação em Pretérito Perfectivo I, na 3ª pessoa do singular, retornando o verbo conjugado 'desmatou':

– Ao rodar-se a função 'verbo_geral(<*args*>)', dá-se a condição de entrada para a realização textual do verbo. No (**Índice 1**) podemos verificar os parâmetros necessários para a realização textual do verbo (TIPO_DE_EXPERIENCIA, funcao_no_grupo_verbal, verbo, tipo_de_orientacao, OI_numero, genero, OI_tipo _de_pessoa, padrao_pessoa_morfologia). Para a flexão do verbo 'desmatar', com

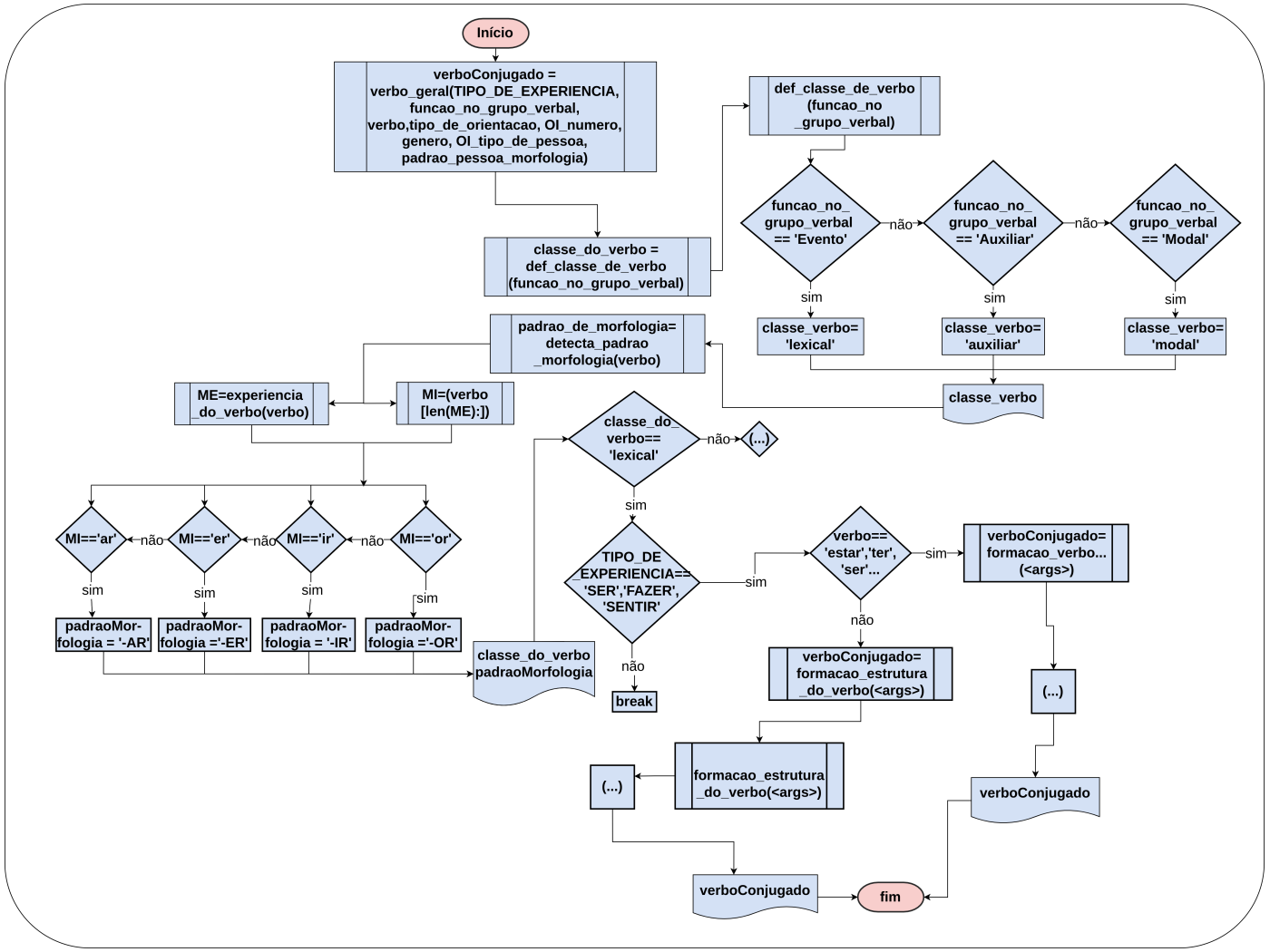
a morfologia apresentada acima, os parâmetros a serem passados à função devem ser 'verbo_geral('Fazer','Evento', 'desmatar', 'pretérito_perfectivo_I', 'singular', None, '3pessoa')(ver **Índice 6**). As primeiras decisões para a realização da função são a seleção de classe do verbo e padrão de morfologia, cujas realizações dependem das subfunções aninhadas – 'def_classe_de_verbo(funcao_no_grupo_verbal) e detecta_padrao_morfologia (ver **Índice 2**). Dada a seleção da classe do verbo, a função seleciona o tipo de experiência (**Índice 3**). Então, a função verifica o lema dado como parâmetro de entrada, e seleciona os casos particulares (e.g., verbos irregulares), como é o caso do verbos lexicais 'trazer' e 'ter', que necessitam de uma modelagem particular (ver **Índice 4**). Esses verbos são realizados por subfunções específicas que são aninhadas na função de flexão verbal (e.g, 'formacao_verbo_trazer' no **Índice 4**). No caso de ser um verbo regular, ou seja, realizado pelas funções gramaticais gerais para a flexão verbal, a função principal seleciona uma subfunção, a formacao_da_estrutura_do_verbo(<*args*>) (ver **Índice 5**) para a flexão do verbo.

– As seleções no sistema de ORIENTAÇÃO INTERPESSOAL (**Índice 4, tipo_de _orientacao, genero, numero**), são passadas às subfunções e ficam a cargo destas para as escolhas dos morfemas pertinentes. Neste ponto, para a realização do verbo flexionado **'desmatou'**, selecionamos a opção 'orientado_finito, pretérito_perfectivo_I, 3a_pessoa, singular'.

Todas essas seleções descritas acima culminam na saída do verbo conjugado '**desmatou**' no console (**Índice 6**).

O fluxograma apresentado na Figura 26 representa um algoritmo que resume a realização do verbo de acordo com a função 'verbo_geral(<args>)', ao ser alimentada pelos parâmetros pertinentes. De maneira geral, partindo do início, o algoritmo recebe os parâmetros para a flexão verbal – tipo_de_experiência, funcao_no_grupo_verbal, verbo, tipo_de_orientacao etc; define a classe do verbo, acordo com a função do verbo no grupo verbal (Evento, Auxiliar etc) – retornando se é um verbo lexical, auxiliar etc; dada a classe do verbo o algoritmo passa à detecção do padrão de morfologia do verbo – separando o morfema experiencial (ME) e o morfema interpessoal (MI); o algoritmo passa à verificação da classe do verbo – se lexical a função então verifica o tipo de experiência dada como entrada; a função então realiza uma verificação pelo tipo de verbo, se regular ou irregular, e repassa os parâmetros para a subfunção de conjugação do verbo e formação da estrutura, dados todos os parâmetros verificados, retornando o verbo conjugado. Por exemplo, segundo os parâmetros ["Fazer", "Evento", "registrar", "Pretérito_Perfectivo_I","singular", None, "3pessoa", None], a função de conjugação verbal retorna o verbo conjugado "registrou".

Figura 26 – Fluxograma algoritmo geração verbo conjugado



Fonte: elaborada para fins deste estudo.

b)**Palavras verbais_preposições**:

Foi desenvolvida uma função básica para a realização das preposições mais comuns. Cabe apontar que, no caso da função de realização das preposições, não foram desenvolvidos, os sistemas lexicogramaticais que organizam seus morfemas. A função desenvolvida em *Python* se restringe a retornar a preposição a partir de uma lista de seleções para que o usuário escolha a preposição desejada. Isso se deve ao fato de que não há descrições disponíveis para a palavra verbal_ preposição, no português brasileiro e pelo fato de que as preposições apresentam um potencial restrito de variabilidade – não flexionam para Gênero, Número como os as palavras nominais, e nem para Temporalidade, Pessoa, Orientação Modal etc., como é o caso do verbo no português brasileiro. A Figura 27 a seguir mostra a função desenvolvida para a realização de preposições do português brasileiro com um exemplo de saída, para a preposição 'para'.

Figura 27 – Função em *Python* para a realização de preposições

```
def preposicao(indice):
    opcoes = ['a', 'ante', 'após', 'até', 'com', 'contra',
              'de', 'desde', 'em', 'entre', 'para',
              'por', 'perante', 'sem',
              'sob', 'sobre', 'trás']
    nums = [x for x in range(len(opcoes))]
    preposicoes = dict(zip(nums, opcoes))

    preposicao=preposicoes[indice]
    return preposicao

print(preposicao(10))
'para'
```

Fonte: elaborada para fins deste estudo.

Como mostra a Figura 27, a função para realização recebe um índice e retorna a respectiva preposição. Nesse exemplo, a função recebe '10' como parâmetro e retorna a preposição correspondente 'para'. Note-se que a lista preposições disponível não é exaustiva, e contém apenas algumas entradas mais potencialmente frequentes.

c)**Palavras nominais**:

Para a classe de palavras nominais, foram desenvolvidas funções que realizam os Numerativos (Ordinais, Percentuais, Cardinais, Indefinidos), Determinantes; bem como funções de detecção e realização do morfema que realiza a experiência nos substantivos comuns, o seu radical, funções que realizam estes substantivos, levando em consideração as terminações mais comuns e as flexões respectivas de gênero e número; funções básicas

com entrada de substantivos próprios, e funções para a flexão de adjetivos considerando-se o gênero e número; funções que realizam pronomes de caso reto, e oblíquo – átono e tônico (e flexões correspondentes de pessoa, gênero e número), pronomes relativos. A seguir, será apresentada a função de 'ordinal(<*args*>)', que retorna o a posição ordinal, dado um número cardinal e o gênero como parâmetros (ver Figura 28) e a função de 'adjetivo(<*args*>)', que realiza a flexão dos adjetivos dados as informações de gênero e número.

Figura 28 – Função – realização numerativo ordinal

```
def ordinal(cardinal,genero):
    """
    :param cardinal:
    :param genero:
    :return: orginal
    """
    num = str(cardinal)
    if genero == 'masculino':
        ordinal = num + 'º'
    else:
        ordinal = num + 'ª'
    return ordinal

print(ordinal(4,'masculino'))
'4º'
```

Fonte: elaborada para fins deste estudo.

A Figura 28 apresenta uma captura de tela da função desenvolvida para a palavra nominal numerativo_ordinal, com um exemplo de realização. Por sua vez, a Figura 29 apresenta uma captura de tela com a função completa de realização do adjetivo, com um exemplo de utilização. Dados os parâmetros lema:'esperto', gênero: 'feminino' e número: 'singular', a função retorna o adjetivo flexionado 'esperta'.

Figura 29 – Função – realização adjetivo

```
def adjetivo(adjModificacao,adjetivo_lematizado,genero,numero):
    """
    Retorna a realizacao de um adjetivo comum dados a experiência_do_adjetivo
     e as flexões_desejadas.
    :param adjModificacao:
    :param adjetivo_lematizado:
    :param genero:
    :param numero:
    :return:
    """
    if adjModificacao == None:
        adj = ''
    else:
        if numero == 'singular':
            if genero == 'masculino':
                morfema_experiencial_do_adjetivo = adjetivo_lematizado[slice(-1)]
                morfema_flexao_adjetivo = 'o'

            elif genero == 'feminino':
                morfema_experiencial_do_adjetivo = adjetivo_lematizado[slice(-1)]
                morfema_flexao_adjetivo = 'a'

            elif genero == 'não-binário':
                morfema_experiencial_do_adjetivo = adjetivo_lematizado
                morfema_flexao_adjetivo = ''


        elif numero == 'plural':
            if genero == 'masculino':
                morfema_experiencial_do_adjetivo = adjetivo_lematizado[slice(-1)]
                morfema_flexao_adjetivo = 'os'


            elif genero == 'feminino':
                morfema_experiencial_do_adjetivo = adjetivo_lematizado[slice(-1)]
                morfema_flexao_adjetivo = 'as'

            elif genero == 'não-binário':
                morfema_experiencial_do_adjetivo = adjetivo_lematizado
                morfema_flexao_adjetivo = 's'

        adj = morfema_experiencial_do_adjetivo + morfema_flexao_adjetivo

    return adj

print(adjetivo("sim",'esperto','feminino','singular'))

'esperta'
```

Fonte: elaborada para fins deste estudo.

d) **Palavras adverbiais**:

Para a classe de palavras adverbiais, foram desenvolvidas (sub)funções para a realização dos principais advérbios: de modo, de intensidade, de lugar, de tempo, de negação, de afirmação, de dúvida. Essas subfunções são introduzidas como subfunção de uma função geral de advérbios (adverbio(<*args*>)). A função geral/principal recebe o tipo de advérbio (de modo, de lugar etc.) e um índice que recupera, dentre uma lista de opções, o advérbio desejado. Cumpre-se ressaltar que, no caso da função de realização do advérbio, não foram explorados, os sistemas lexicogramaticais que organizam seus morfemas. A função geral desenvolvida se restringe a receber qual tipo de advérbio e o índice para retornar o advérbio. Isso se deve ao fato de, em primeiro lugar, o enfoque principal não se orientar a essa subclasse de palavra, e em segundo lugar, pelo fato de os advérbios não flexionarem quanto a gênero e número. A Figura 30 apresenta a função

principal de realização dos advérbios, com um exemplo para a realização do advérbio de Modo 'depressa'. Note-se que há funções para cada subtipo de advérbio aninhadas na função principal.

Figura 30 – Função – realização advérbio

```
def adverbio(tipo_de_adverbio, indice):
    """
    :param tipo_de_adverbio: 'Modo' 'Intensidade' 'Lugar' 'Tempo'
           'Negacao' 'Afirmacao' 'Duvida' 'Adv_relativo'
    :param indice: --
    :return: advérbio
    """


    if tipo_de_adverbio == 'Modo':

        adverbio = adverbio_modo(indice)

    elif tipo_de_adverbio == 'Intensidade':

        adverbio = adverbio_intensidade(indice)

    elif tipo_de_adverbio == 'Lugar':

        adverbio = adverbio_lugar(indice)

    elif tipo_de_adverbio == 'Tempo':

        adverbio = adverbio_tempo(indice)

    elif tipo_de_adverbio == 'Negacao':

        adverbio = adverbio_negacao(indice)

    elif tipo_de_adverbio == 'Afirmacao':
        adverbio = adverbio_afirmacao(indice)

    elif tipo_de_adverbio == 'Duvida':

        adverbio = adverbio_duvida(indice)

    elif tipo_de_adverbio == 'Adv_relativo':

        adverbio = adverbio_relativo(indice)

    return adverbio

print(adverbio("Modo", 6))

'depressa'
```

Fonte: elaborada para fins deste estudo.

2) **Ordem do grupo**

a) **Grupo nominal**: A implementação computacional do grupo nominal do português brasileiro foi baseada na descrição de base sistêmico-funcional realizado por Figueredo (2007). A descrição do grupo nominal toma uma perspectiva trinocular: o exame 'de baixo' permitiu a identificação das classes de palavras que encerram função no grupo nominal; 'ao redor' foram identificados os sistemas que organiza a estrutura do grupo nominal; 'de cima' foram identificados as funções encerradas pelo grupo nominal na ordem imediatamente superior, a oração. Segundo Figueredo (2007), os resultados da descrição apontaram que o grupo nominal no português estrutura-se experiencial e logicamente:

> experiencialmente, os elementos presentes nesta classe de grupo cumprem as funções de ***Qualidade***: Dêiticos: não-seletivo (específico e não-

> específico), seletivo de proximidade, seletivo de pessoa, indefinido, interrogativo. Numerativo: ordenativo, quantitativo, interrogativo. Epíteto: experiencial, interpessoal. Classificador: várias classes (material, origem, característica, etc.). Além destas, há presente também a função do ***Ente***, que pode ser classificado taxonomicamente como: consciente, animal, objeto material, substância, abstração material, instituição, objeto semiótico, abstração semiótica. Na estrutura lógica, o sistema linguístico do português possui um elemento operando como Núcleo e outros elementos operando como Pré- e Pós-modificadores.

Com base nessa descrição do grupo nominal, foi possível modelar computacionalmente os seguintes sistemas: a taxonomia de tipos de Ente; os principais sistemas que organizam o Grupo Nominal: sistemas de DETERMINAÇÃO, CLASSIFICAÇÃO, QUALIFICAÇÃO, QUANTIFICAÇÃO. Note-se, como já apontado, o português brasileiro compreende uma escala de ordens lexicogramatical que vai da unidade mais básica, o morfema, até a mais ampla, a oração. Dessa maneira, funções modeladas para a unidade da palavra (e consequentemente do morfema) são aninhadas na organização das funções que realizam o grupo (nominal). A seguir serão apresentadas a função que realiza o Ente e a função geral que realiza o grupo nominal.[3]

A taxonomia do Ente no português brasileiro, segundo Figueredo (2007) (seguindo critérios aplicados na descrição da taxonomia do Ente no inglês), se organiza segundo 3 vetores, **contagem, generalidade** e **animação**, articulados com a função experiencial do elemento no grupo nominal e são realizados prototipicamente por substantivos comuns, próprios ou pronomes. A articulação desses princípios resulta em uma taxonomia que organiza o Ente no português como segue: o Ente pode ser 'Consciente' ou 'Não-consciente[material[Animal, Objeto(material)], Substância(material), Abstração(material); semiótico[Instituição, Objeto(semiótico), Abstração(semiótico)]] ((FIGUEREDO, 2007, p. 157)). A função de realização do Ente recebe como argumentos de entrada os parâmetros destacados (**Índice 1**) na Figura 31. Para além das informações estritamente relacionadas às escolhas na taxonomia, a função também recebe parâmetros que passam informações de gênero, número, informações sobre o padrão de terminação (importantes para a flexão de gênero e número específicas que apresentam um padrão particular de flexão – e.g., substantivos terminados em 'ão'). O Ente então é realizado por uma das funções correspondentes à ordem da palavra, classe de palavras nominais_substantivo, a depender da necessidade no grupo nominal (substantivo comum, próprio ou pronome)(ver Figura 31, **Índice 2**).

A função que realiza o grupo nominal recebe os parâmetros dispostos no **Índice 1** (ver Figura 32), e combina as perspectivas de organização experiencial e lógica da estrutura, acomodando as subfunções de realização para cada um dos subsistemas do grupo. O **Índice 2** da Figura 32 apresenta a disposição dos elementos que potencialmente compõem um grupo nominal, sendo realizados respectivamente pelas funções aninhadas, desenvolvidas

[3] Para o repositório com todas as funções, acessar: REVISAR INSERIR REPOSITÓRIO

Figura 31 – Função – realização Ente

```
def Ente(tipo_de_Ente=None, tipo_de_nao_consciente=None, tipo_de_nao_consciente_material=None,
    tipo_de_nao_consciente_semiotico=None, classe_palavra_Ente=None, substantivo_lematizado=None
    numero=None, genero=None, tipo_feminino_AO=None, tipo_masc_AO=None, acentTonica=None,
    nomeProprio=None,pessoa_da_interlocucao=None,transitividade_verbo=None, tonicidade=None,
    morfologia_do_pronome=None, reflexivo=None):                                          1
  """""""

  if tipo_de_Ente == 'NA':                                                                2
     Ente = ''
  else:
     if classe_palavra_Ente == 'substantivo_comum':
        Ente = substantivo_comum(substantivo_lematizado, numero, genero,
                                 tipo_feminino_AO, tipo_masc_AO, acentTonica)

     elif classe_palavra_Ente == 'substantivo_próprio':
        Ente = nomeProprio
     elif classe_palavra_Ente == 'pronome_caso_reto':
        Ente = realizacao_pronominal_casoreto(pessoa_da_interlocucao, genero,
                                              numero, morfologia_do_pronome)
     elif classe_palavra_Ente == 'pronome_caso_oblíquo':
        Ente = realizacao_pronome_caso_obliquo(transitividade_verbo, tonicidade,
                                               pessoa_da_interlocucao, numero, genero,
                                               morfologia_do_pronome, reflexivo)

  return Ente
```

Fonte: elaborada para fins deste estudo.

para a realização de cada um dos elementos (i.e., Determinante: realizada pela função Dêixis_geral(<*args*>); Numerativo: realziado pela função Numerativo(<*args*>); Ente: realizado pela função Ente(<*args*>); Qualificador: realizado pela função qualificador(<*args*>) etc.). Dadas as escolhas para cada um dos elementos, a estrutura é concatenada, de acordo com uma organização estrutural genérica, e retorna-se o grupo nominal formatado (retirando-se espaços extras), como mostra o **Índice 3** da Figura 32. O **Índice 4** na Figura 32 mostra a execução da função, dados os parâmetros para a realização de um grupo nominal, com dêixis específica, sem orientação, sem dissociação entre Ente e Núcleo, sem modificação realizada por Qualificadores ou Classificadores; Ente realizado por um substantivo comum, 'desmatamento'. Essa execução retorna o grupo nominal '**o desmatamento**'.

b) **Grupo verbal**:

Segundo Sá (2016), o grupo verbal é descrito sob uma perspectiva trinocular: 'de baixo' é composto por verbos lexicais, auxiliares, e modais que, realizam, respectivamente as funções de Evento, Auxiliar e Modal, além da função de Núcleo, que pode ser realizada por qualquer classe de verbo; 'de cima' o grupo verbal realiza a função de Processo, Finito, Predicador, contribuindo para a "construção da experiência, da transitoriedade, e da negociação"na oração; 'ao redor' o grupo verbal é organizado pelos sistemas de TIPO DE EVENTO, FINITUDE,DÊIXIS TEMPORAL, TEMPO SECUNDÁRIO, ASPECTO VERBAL,

Figura 32 – Função – realização Grupo Nominal

```
def estrutura_GN(dissocEnteNucleo=None, temQualificador=None, tipoQualificador=None, indicePreposicao=None,
            DETERMINAÇÃO_especificidade_beta=None, ORIENTAÇÃO_beta=None, gênero_beta=None, número_beta=None,
            morfologia_do_pronome_beta=None, DETERMINAÇÃO_especificidade_alpha=None, ORIENTAÇÃO_alpha=None,
            gênero_alpha=None, número_alpha=None, morfologia_do_pronome_alpha=None,
            pessoa_da_interlocução_possuidor=None, número_obj_possuído=None, gênero_obj_possuído=None,
            pessoa_da_interlocução_proximidade=None, funcaoNumerativo=None, cardinal=None, genero=None,
            tipo_precisa=None, tipoRealCard=None, milharExtenso=None, centenaExtenso=None,
            dezenaExtenso=None, unidadeExtenso=None, numIndefinido=None,
            tipo_de_Ente=None, tipo_de_nao_consciente=None, tipo_de_nao_consciente_material=None,
            tipo_de_nao_consciente_semiotico=None, classe_palavra_Ente=None, substantivo_lematizado=None,
            numero=None, tipo_feminino_AO=None, tipo_masc_AO=None, acentTonica=None, nomeProprio=None,
            pessoa_da_interlocucao=None, transitividade_verbo=None, tonicidade=None, morfologia_do_pronome=None
            reflexivo=None, epitetoModificacao=None, adjetivo_epiteto=None, classificadorModificacao=None,
            adjetivo_classificador=None):                                                                    1

    if dissocEnteNucleo == None:                                                                             2

        Determinante = Dêixis_geral(DETERMINAÇÃO_especificidade_beta,ORIENTAÇÃO_beta,
               gênero_beta,número_beta,morfologia_do_pronome_beta, DETERMINAÇÃO_especificidade_alpha,
               ORIENTAÇÃO_alpha,gênero_alpha,número_alpha,morfologia_do_pronome_alpha,
               pessoa_da_interlocução_possuidor,número_obj_possuído,
               gênero_obj_possuído,pessoa_da_interlocução_proximidade)

        numerativo = Numerativo(funcaoNumerativo,cardinal,genero,tipo_precisa,tipoRealCard,
              milharExtenso,centenaExtenso,dezenaExtenso,unidadeExtenso,numIndefinido)

        ente = Ente(tipo_de_Ente,tipo_de_nao_consciente,tipo_de_nao_consciente_material,
         tipo_de_nao_consciente_semiotico,classe_palavra_Ente,substantivo_lematizado,numero,
        genero, tipo_feminino_AO, tipo_masc_AO,acentTonica,nomeProprio,pessoa_da_interlocucao,
         transitividade_verbo,tonicidade,morfologia_do_pronome,reflexivo)

        Classificador = adjetivo(classificadorModificacao,adjetivo_classificador,genero,numero)

        Epíteto = adjetivo(epitetoModificacao,adjetivo_epiteto,genero,numero)

        Qualificador = qualificador(temQualificador, tipoQualificador, indicePreposicao,
                            DETERMINAÇÃO_especificidade_beta,
                            ORIENTAÇÃO_beta,gênero_beta, número_beta, morfologia_do_pronome_beta,
                            DETERMINAÇÃO_especificidade_alpha, ORIENTAÇÃO_alpha, gênero_alpha,
                            número_alpha, morfologia_do_pronome_alpha, pessoa_da_interlocução_possuidor,
                            número_obj_possuído, gênero_obj_possuído, pessoa_da_interlocução_proximidade,  #
                            funcaoNumerativo, cardinal, genero, tipo_precisa, tipoRealCard,
                            milharExtenso, centenaExtenso, dezenaExtenso, unidadeExtenso, numIndefinido,
                            tipo_de_Ente, tipo_de_nao_consciente, tipo_de_nao_consciente_material,
                            tipo_de_nao_consciente_semiotico, classe_palavra_Ente, substantivo_lematizado,
                            numero,tipo_feminino_AO, tipo_masc_AO, acentTonica, nomeProprio,
                            pessoa_da_interlocucao,transitividade_verbo, tonicidade, morfologia_do_pronome,
                            reflexivo, epitetoModificacao, adjetivo_epiteto, classificadorModificacao,
                            adjetivo_classificador)

        GN = Determinante + ' ' + numerativo + ' ' + ente + ' ' + Classificador + ' ' + Epíteto + ' ' + Qualificador
    return (re.sub(' +', ' ', GN).strip())                                                                   3
```

```
estrutura_GN(DETERMINAÇÃO_especificidade_alpha='específico', ORIENTAÇÃO_alpha='NA',
          gênero_alpha='masculino',número_alpha='singular', morfologia_do_pronome_alpha='morfologia_terceira_pessoa',
          pessoa_da_interlocução_possuidor='1s', número_obj_possuído='plural', gênero_obj_possuído='masculino',
          genero='não-binário', tipo_de_Ente='não_consciente', tipo_de_nao_consciente='material',
          tipo_de_nao_consciente_material='instituição',classe_palavra_Ente='substantivo_comum',
          substantivo_lematizado='desmatamento', numero='singular')
'o desmatamento'                                                                                             4
```

Fonte: elaborada para fins deste estudo.

AGÊNCIA e DÊIXIS MODAL[4]. Com base nessa descrição do grupo verbal, foi desenvolvida uma função que realiza o grupo, acomodando os (sub)sistemas que o organizam. Uma observação importante sobre o desenvolvimento: como os sistemas de FINITUDE, DÊIXIS TEMPORAL, ASPECTO na ordem do grupo são 'síndromes' de significados que reverberam desde a ordem mais básica, o morfema, não foram desenvolvidas funções particulares para a sua realização. Em outras palavras, os significados construídos por esses sistemas são de certa maneira uma reiteração de decisões que são tomadas pelos sistemas que organizam as palavras

[4] Para a descrição detalhada de cada um dos sistemas, ver Sá (2016)

verbais, isto é, uma síndrome. Assim, os significados que os sistemas do grupo verbal organizam, são carregados desde a ordem do morfema. A função 'grupo_verbal(<*args*>) então organiza a estrutura experiencial e lógica, organizando as funções realizadas pelas palavras verbais (da ordem imediatamente inferior na escala).

A Figura 33 apresenta a função geral de realização do grupo verbal ('grupo_verbal (<*args*>)). Como mostra o **Índice 1** na figura, a função recebe tanto parâmetros específicos para a organização do grupo (tipo de experiência do GV, AGENCIA), quanto parâmetros que dizem respeito à realização das palavras verbais_verbos que compõem o grupo, isto é, esses parâmetros são dados como entrada para a realização dos verbos (de 1 a Evento/verbos_da_passiva) que potencialmente constituem a estrutura do grupo verbal. Note-se, que, potencialmente, a estrutura do grupo verbal tem a organização temporal (secundária) passível de reiterações, e dessa maneira, pode chegar a um número infinito de verbos, realizando vários tempos secundários. A função 'grupo_verbal' desenvolvida no realizador textual restringe-se ao potencial de 4 iterações, ou seja, chega a grupos verbais de até 5 verbos. Como pode-se notar, a estrutura lógica é organizada de acordo com a numeração dos parâmetros ('TIPO_DE_EXPERIENCIA_1, funcao_no_grupo_verbal_1 etc.') em relação às variáveis 'verbo' ('verbo1, verbo2, etc.'). O **Índice 4** na Figura 33 mostra o uso da função 'grupo_verbal(<*args*>)' para a realização de um grupo verbal na voz passiva, na 3–ª pessoa do singular, no pretérito_perfectivo_I, lema 'desmatar'. Dados os parâmetros de entrada, a função retorna o grupo verbal '**foi desmatado**'.

Figura 33 – Função – realização Grupo Verbal


```
def grupo_verbal(TIPO_DE_EXPERIENCIA_GV=None, AGENCIA=None, TIPO_DE_EXPERIENCIA_1=None,
          funcao_no_grupo_verbal_1=None, verbo_1=None, tipo_de_orientacao_1=None,
          OI_numero_1=None,genero_1=None, OI_tipo_de_pessoa_1=None,
          padrao_pessoa_morfologia_1=None, TIPO_DE_EXPERIENCIA_2=None,
          funcao_no_grupo_verbal_2=None, verbo_2=None, tipo_de_orientacao_2=None,
          OI_numero_2=None,genero_2=None, OI_tipo_de_pessoa_2=None,
          padrao_pessoa_morfologia_2=None, TIPO_DE_EXPERIENCIA_3=None,
          funcao_no_grupo_verbal_3=None, verbo_3=None, tipo_de_orientacao_3=None,
          OI_numero_3=None,genero_3=None, OI_tipo_de_pessoa_3=None,
          padrao_pessoa_morfologia_3=None, TIPO_DE_EXPERIENCIA_4=None,
          funcao_no_grupo_verbal_4=None, verbo_4=None, tipo_de_orientacao_4=None,
          OI_numero_4=None,genero_4=None, OI_tipo_de_pessoa_4=None,
          padrao_pessoa_morfologia_4=None,TIPO_DE_EXPERIENCIA_LEX=None,
          funcao_no_grupo_verbal_POS_FINAL=None, verbo_LEX=None,
          tipo_de_orientacao_LEX=None, OI_numero_LEX=None,
          genero_LEX=None, OI_tipo_de_pessoa_LEX=None,
          padrao_pessoa_morfologia_LEX='Morfologia_padrão'):

    if TIPO_DE_EXPERIENCIA_GV == 'Ser' or TIPO_DE_EXPERIENCIA_GV == 'Fazer' \
        or TIPO_DE_EXPERIENCIA_GV == 'Sentir':

       if AGENCIA == 'agenciado_ativa' or AGENCIA == 'não_agenciado':

          verbo1 = verbo_geral(TIPO_DE_EXPERIENCIA_1, funcao_no_grupo_verbal_1, verbo_1,
                  tipo_de_orientacao_1,  OI_numero_1, genero_1,
                  OI_tipo_de_pessoa_1, padrao_pessoa_morfologia_1)
          verbo2 = verbo_geral(TIPO_DE_EXPERIENCIA_2,
                               funcao_no_grupo_verbal_2, verbo_2, tipo_de_orientacao_2,
                               OI_numero_2,
                               genero_2, OI_tipo_de_pessoa_2, padrao_pessoa_morfologia_2)
          verbo3 = verbo_geral(TIPO_DE_EXPERIENCIA_3,
                               funcao_no_grupo_verbal_3, verbo_3, tipo_de_orientacao_3,
                               OI_numero_3,
                               genero_3, OI_tipo_de_pessoa_3, padrao_pessoa_morfologia_3)
          verbo4 = verbo_geral(TIPO_DE_EXPERIENCIA_4,
                               funcao_no_grupo_verbal_4, verbo_4, tipo_de_orientacao_4,
                               OI_numero_4,
                               genero_4, OI_tipo_de_pessoa_4, padrao_pessoa_morfologia_4)
          Evento = verbo_geral(TIPO_DE_EXPERIENCIA_LEX, funcao_no_grupo_verbal_POS_FINAL,
                               verbo_LEX, tipo_de_orientacao_LEX,
                               OI_numero_LEX, genero_LEX, OI_tipo_de_pessoa_LEX
                               , padrao_pessoa_morfologia_LEX)

          grupo_verbal = verbo1 + ' ' + verbo2 + ' ' + verbo3 + ' ' + verbo4 + ' ' + Evento

       else:
          tipo_de_orientacao_LEX = 'particípio'
          verbo_4 = 'ser'
          verbo1 = verbo_geral(TIPO_DE_EXPERIENCIA_1,
                               funcao_no_grupo_verbal_1,
                               verbo_1, tipo_de_orientacao_1,  OI_numero_1,
                               genero_1, OI_tipo_de_pessoa_1, padrao_pessoa_morfologia_1)
          verbo2 = verbo_geral(TIPO_DE_EXPERIENCIA_2,
                               funcao_no_grupo_verbal_2, verbo_2, tipo_de_orientacao_2,
                               OI_numero_2,
                               genero_2, OI_tipo_de_pessoa_2, padrao_pessoa_morfologia_2)
          verbo3 = verbo_geral(TIPO_DE_EXPERIENCIA_3,
                               funcao_no_grupo_verbal_3, verbo_3, tipo_de_orientacao_3,
                               OI_numero_3,
                               genero_3, OI_tipo_de_pessoa_3, padrao_pessoa_morfologia_3)

          verbos_passiva = realizacao_de_AGENCIA_passiva(verbo_4, tipo_de_orientacao_4,
                                                         OI_numero_4, genero_4, OI_tipo_de_pessoa_4,
                                                         padrao_pessoa_morfologia_4, TIPO_DE_EXPERIENCIA_LEX,
                                                         funcao_no_grupo_verbal_POS_FINAL, verbo_LEX,
                                                         tipo_de_orientacao_LEX,
                                                         OI_numero_LEX, genero_LEX, OI_tipo_de_pessoa_LEX,
                                                         padrao_pessoa_morfologia_LEX)


          grupo_verbal = verbo1 + ' ' + verbo2 + ' ' + verbo3 + ' ' + verbos_passiva
    return (re.sub(' +', ' ', grupo_verbal).strip())

grupo_verbal('Fazer', 'agenciado_passiva', None, None, None, None, None, None,
             None, None, None, None, None, None, None, None, None,
             None, None, None, None, None, None, None, None, 'Ser', 'Auxiliar',
             'ser', 'pretérito_perfectivo_I', 'singular', None, '3pessoa',
             'Morfologia_padrão', 'Fazer', 'Evento', 'desmatar', 'particípio', 'singular',
             'masculino', None, 'Morfologia_padrão')

'foi desmatado'
```

Fonte: elaborada para fins deste estudo.

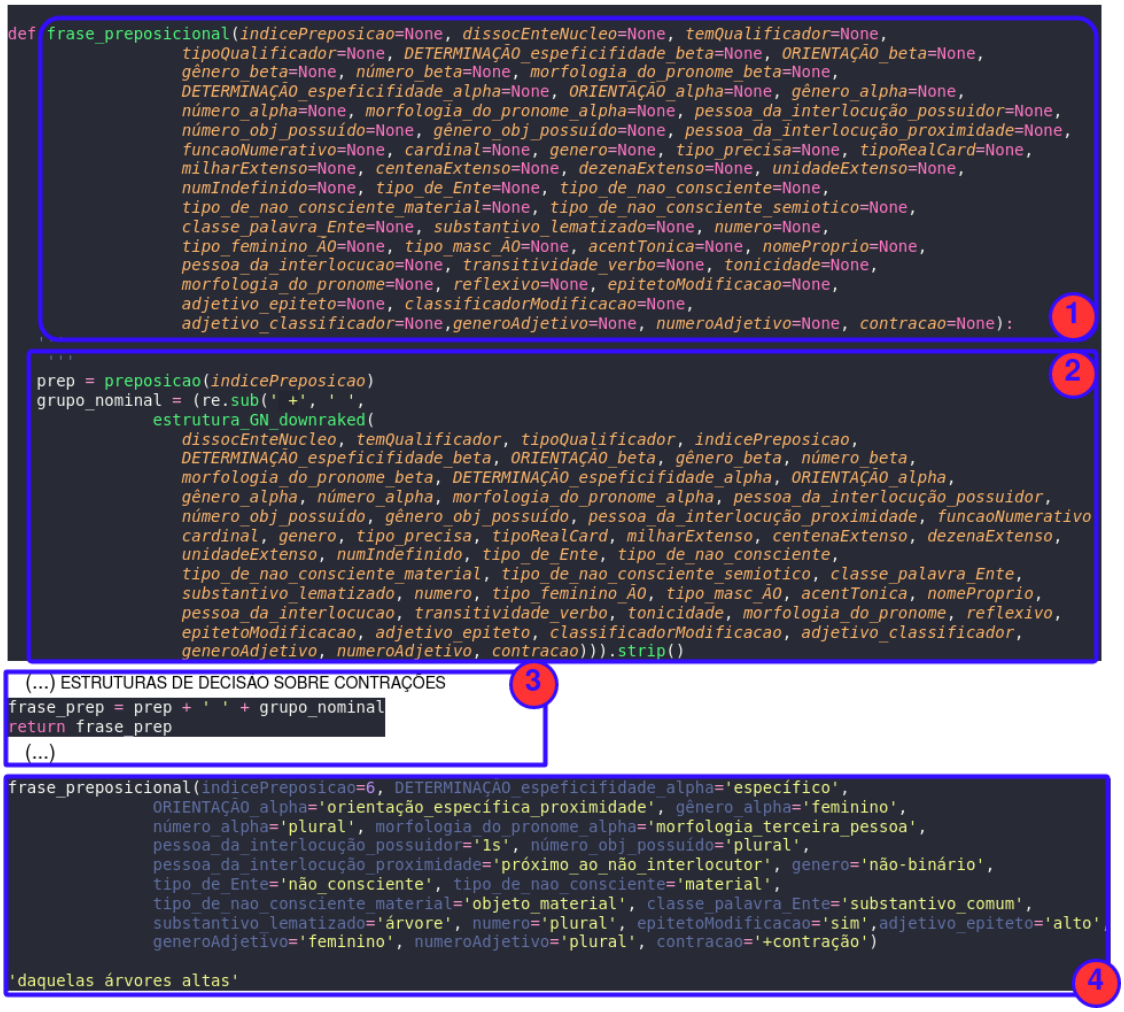
c) **Frase preposicional**:

Sob análise 'de cima', ou seja, função que exerce na ordem imediatamente superior (oração), a frase preposicional realiza, prototipicamente, as Circunstâncias (podendo também, em alguns tipos de configurações, realizar Participantes e modificadores em grupos nominais). 'De baixo', as frases preposicionais são constituídas por uma preposição e um grupo nominal descido de ordem. A estrutura lógica é organizada a partir do Núcleo (a preposição) e modificadores (grupos nominais descidos de ordem) (HALLIDAY; MATTHIESSEN, 2014; FERREGUETTI, 2018)[5]. Essa organização básica da frase preposicional demanda que funções da ordem imediatamente inferior, unidade da palavra: preposição, e uma unidade da mesma ordem, unidade do grupo: nominal, sejam aninhadas para que sua estrutura seja realizada. A Figura 34 mostra partes da função que realiza a frase preposicional [6].

[5] Para uma análise de da frase preposicional em sua função de Qualificador no grupo nominal, ver Ferreguetti (2018)

[6] A partir da ordem do grupo, as funções desenvolvidas começam a ficar muito longas, impedindo sua disposição completa em figuras. Para uma visualização completa, acessar: <https://github.com/AndreRosaLRT/NLG_BRAZILIAN_PORTUGUESE>

Figura 34 – Função – realização Frase Preposicional



Fonte: elaborada para fins deste estudo.

O **Índice 1** da Figura 34 apresenta o conjunto de parâmetros que a função de realização de frase preposicional recebe. Como mencionado, uma frase preposicional é realizada por uma preposição e um grupo nominal descido de ordem. Assim, os parâmetros que essa função recebe serão passados para as subfunções que são alinhadas para realizar os elementos que constituem a frase preposicional, como podemos ver destacado no **Índice 2** da figura. As variáveis 'prep' e 'grupo_nominal' são então formatadas e concatenadas e retornam a frase preposicional como destacado no **Índice 3**. Note-se que um trecho da função de frase preposicional não foi alocada na figura. Esse trecho é responsável pelas formatações necessárias quanto às contrações da proposição com o determinante do grupo nominal, por exemplo. O **Índice 4** destaca um exemplo de execução da função de realização textual de uma frase preposicional que tem como Núcleo a preposição 'de' e é modificada pelo grupo nominal 'aquelas árvores altas', retornando o texto formatado 'daquelas árvores altas'.

d) **Grupo adverbial**:

Como não há uma descrição completa desta classe de grupo para o português brasileiro, serão adotados aspectos descritos para outras línguas que, em um nível menos delicado, se apliquem ao português. Tomando as noções da descrição apresentada por Halliday e Matthiessen (2014) do grupo adverbial do inglês, verifica-se que esta classe de grupo encerra as funções de Adjunto na oração – Adjunto circunstancial ou Adjunto modal (de comentário ou de modo).

A estrutura do grupo adverbial tem uma palavra adverbial como Núcleo, que pode ou não ser modificado por outros elementos. Os grupos adverbiais que desempenham funções de Adjuntos circunstanciais têm um advérbio que denota significados circunstanciais como Núcleo –e.g., de tempo (ontem, amanhã) de modo (bem, mal, rapidamente) etc. Por outro lado, grupos adverbiais com função de Adjunto de modo têm um advérbio que denota avaliação como Núcleo, por exemplo, avaliação de tempo (já, ainda) e de intensidade (apenas, somente). O Núcleo pode ser pré-modificado por advérbios, mas essa modificação é mais restrita em termos do subtipo de advérbios disponíveis (de polaridade, de comparação, e de intensificação) (HALLIDAY; MATTHIESSEN, 2014, p. 421). Levando em consideração que, neste nível de delicadeza, o grupo adverbial no português brasileiro apresenta opções congruentes com o sistema descrito para o inglês, foi desenvolvida uma função de realização do grupo adverbial que aninha a função desenvolvida para realização do advérbio, na ordem da palavra. Como destacado na Figura 35, no **Índice 1**, apresentam-se os argumentos de entrada para a função: como a estrutura do grupo adverbial prevê pré-modificação iterativa por outras palavras da mesma classe, os parâmetros são numerados até 5, prevendo um potencial de estrutura constituída por 5 palavras adverbiais.

No **Índice 2** destacam-se os advérbios que são realizados pela função geral de realização desta classe de palavra, também numerados de acordo com a sua potencial posição na estrutura. Dadas as escolhas de tipo de adverbio e o índice referente ao advérbio específico dentro da lista de tipos de advérbio, a função concatena, formata e retorna o grupo adverbial. Note-se que não houve restrição de qual classe de advérbio pode ser realizado em qual posição da estrutura, ou seja, em cada posição, todo o potencial de realização dos advérbios está disponível para seleção. Essa decisão foi tomada por não haver uma descrição exaustiva desta classe de grupo para o português, ensejando as opções de estruturação em aberto.

No **Índice 3**, destaca-se uma execução da função, que exemplifica a realização de um grupo adverbial que tem um advérbio de 'Modo' como Núcleo ('cuidadosamente') e é pré-modificado por dois advérbios, um de 'Intensidade' e um de 'Polaridade:Negação', realizando o grupo adverbial 'não muito cuidadosamente'.

Figura 35 – Função – realização Grupo Adverbial

Fonte: elaborada para fins deste estudo.

3) **Ordem da oração**:

Como já foi discutido no Capítulo 2, a oração é a unidade da ordem mais elevada na escala de ordens lexicogramatical do português brasileiro e pode ser analisada sob a perspectiva trinocular: como apresentado na Figura 7, 'de cima', a oração conflui significados através de todo o espectro metafuncional, realizando significados do baixo estrato semântico pela lexicogramática. A oração é, assim, a confluência da **figura** sistema semântico de CONFIGURAÇÃO, do **argumento** sistema semântico de FUNÇÕES DISCURSIVAS, e da **mensagem** sistema semântico PROGRESSÃO; 'ao redor', a oração é organizada pelos grandes sistemas lexicogramaticais de TRANSITIVIDADE, MODO e TEMA, respectivamente; ; e 'de baixo', a oração é constituída pelas unidades inferiores na escala de ordens (ver Figura 8).

Sob a perspectiva da metafunção ideacional, o sistema de TRANSITIVIDADE organiza Participantes e Circunstâncias envolvidas em Processos. Nesse sentido, realiza o sistema semântico de CONFIGURAÇÃO, **figura** [**quantum de mudança**]: processos, participantes e possíveis circunstâncias, construindo assim o **domínio experiencial**. Sob a perspectiva interpessoal, a oração realiza o **argumento** [**quantum de interação**], através do sistema semântico de FUNÇÕES DISCURSIVAS, realizado pelo sistema de MODO, que organiza a oração em termos de modo: Indicativo (Interrogativo/Declarativo), Imperativo etc. Por fim, sob a perspectiva textual,a oração realiza a **mensagem** [**quantum de informação**], através do sistema semântico de PROGRESSÃO, e sua realização no sistema lexicogramatical de TEMA (ver Figura 9).

Por realizar computacionalmente a ordem mais abrangente da lexicogramática, a

oração, a função que combina as funções desenvolvidas para todas as unidades desenvolvidas (morfema, palavra, grupo), por meio das relações básicas da linguagem (classe, unidade, estrutura, sistema). Para a realização da oração foram desenvolvidas as funções que realizam os principais sistemas, TRANSITIVIDADE, MODO: modelagem parcial/preliminar, seleção de escolhas mais prototípicas – modo declarativo e interrogativo polar; TEMA IDEACIONAL: modelagem parcial, restrito à escolha de tema_default); TEMA INTERPESSOAL e TEMA TEXTUAL, e respectivos subsistemas: AGENCIAMENTO, SUJEITABILIDADE, AVALIAÇÃO MODAL, TIPOS DE PROCESSO, TIPO DE MODO, AVALIAÇÃO MODAL. Como já mencionado, as funções para a realização textual a partir da ordem do grupo recebem muitos parâmetros (pois precisam passar os argumentos para todas as ordens inferiores na escala) e portanto são funções muito extensas. Isto impossibilita a confecção de figuras que comportem toda sua extensão, especialmente a função que realiza a oração. Assim, a seguir, serão apresentadas algumas figuras com partes das funções [7].

Em um estágio mais avançado de desenvolvimento, foram desenvolvidas funções particulares para cada tipo de Processo: relacional, existencial, material, verbal, mental, devido à diversidade e grande quantidade de parâmetros necessários para a realização das orações. Como mencionado, as funções de realização da oração acumulam os parâmetros necessários para a realização de todas as unidades da escala de ordens abaixo da oração, além dos parâmetros específicos da própria unidade da escala de ordens. Essa característica impôs alguns desafios de desenvolvimento e aplicação/uso das funções, destaca-se o desenvolvimento das funções com potencial de projeção, como é o caso das orações mental e verbal. Esses tipos de oração abrem opções de projeção de outras orações que por sua vez demandam parâmetros particulares, realizando complexos oracionais por meio de parataxe ou hipotaxe. Cada função de realização de oração recebe em média de 350 (Mental, sem previsão de projeção) a 500 (Material). O desenvolvimento de uma função de realização da oração mental que preveja o potencial de uma projeção de oração material necessitaria, assim, de aceitar por volta de 800 parâmetros de entrada. Ou seja, seriam necessários tanto os parâmetros específicos da mental quanto os particulares das potenciais projeções, sem contar ainda o potencial de níveis ainda mais profundos de projeção, por exemplo, 'Eu disse que ela pensou que haviam desmatado todo o território', que demandaria três coleções de parâmetros diferentes para cada oração. Nesse sentido, foram desenvolvidas funções de realização da oração que ainda não preveem esse potencial de complexos oracionais por meio de projeção.

A primeira função a ser apresentada é responsável pelas opções para TEMA IDEACIONAL (ver Figura 36. Seguindo a descrição apresentada por Figueredo (2011), a função 'TEMA_IDEACIONAL(<*args*>) foi desenvolvida, até o momento, restringindo-se às opções mais prototípicas (ORIENTAÇÃO MODAL:orientado; ORIENTAÇÃO TRANSI-

[7] Para uma visualização das funções, acessar:<https://github.com/AndreRosaLRT/NLG_BRAZILIAN_PORTUGUESE>

TIVA:direcional; SELEÇÃO TEMÁTICA:default, opções não-default, TEMA ÂNGULO, ELEMENTAL E PROEMINENTE apresentam apenas o nível menos delicado. No **Índice 1** da Figura 36, destacam-se os parâmetros de entrada. O **Índice 2** apresenta o potencial de saída, dadas as escolhas dadas como entrada pelos parâmetros. Note-se que, por ser realizado por algum elemento experiencial da oração, o tema ideacional é realizado por algum elemento que é, por sua vez, realizado por uma subfunção desenvolvida para a ordem do grupo (nominal, frase preposicional) que encerra, por exemplo, uma função na estrutura experiencial e interpessoal da oração (e.g., Participante 1/Sujeito:grupo nominal). Portanto, a função 'TEMA_IDEACIONAL' é responsável por retornar uma *string* com o acúmulo de opções selecionadas para os subsistemas de tema ideacional. Na função de realização da oração a saída destas escolhas, por sua vez, aponta para o tipo de grupo que irá realizar o tema.

O **Índice 3** destaca as opções menos delicadas de TEMA PROEMINENTE, ANGULO E ELEMENTAL.

O **Índice 4** destaca um exemplo de execução da função 'TEMA_IDEACIONAL' e a saída. As opções cumulativas retornam um 'tema ideacional default indicativo declarativo sem tema identificativo' – ('TID_default_indicativo_declarativo_TIdentif_NA'). No sistema de TEMA do português brasileiro, essa opção de tema conflui com Sujeito e Participante, e portanto, é prototipicamente realizado por um grupo nominal. A subfunção que organiza as escolhas para o TEMA INTERPESSOAL e TEMA TEXTUAL foram desenvolvidas seguindo o mesmo padrão aplicado ao desenvolvimento da subfunção de TEMA IDEACIONAL.

Figura 36 – Função – Tema Ideacional

```
def TEMA_IDEACIONAL(ORIENTACAO_MODAL=None,ORIENTACAO_TRANSITIVA=None,
            SELECAO_TEMATICA=None, TEMA_DEFAULT=None,
            TEMA_DEFAULT_indicativo=None, TEMA_IDENTIFICATIVO=None,
            TEMA_ANGULO=None, TEMA_ELEMENTAL=None,
            TEMA_PROEMINENTE=None):

  if ORIENTACAO_MODAL == 'orientado' and ORIENTACAO_TRANSITIVA == 'direcional'\
        and SELECAO_TEMATICA == 'default':

     if TEMA_DEFAULT == 'imperativo':
        TEMA_IDEACIONAL = 'TID_default_imperativo'

     elif TEMA_DEFAULT == 'indicativo':

        if TEMA_DEFAULT_indicativo == 'declarativo'\
              and TEMA_IDENTIFICATIVO == 'NA':

           TEMA_IDEACIONAL = 'TID_default_indicativo_' \
                        'declarativo_TIdentif_NA'

        elif TEMA_DEFAULT_indicativo == 'interrogativo_polar'\
              and TEMA_IDENTIFICATIVO == 'NA':

           TEMA_IDEACIONAL = 'TID_default_indicativo_interrogativo_polar' \
                        '_TIdentif_NA'

        elif TEMA_DEFAULT_indicativo == 'interrogativo_sujeito_elemental'\
              and TEMA_IDENTIFICATIVO == 'NA':

           TEMA_IDEACIONAL = 'TID_default_indicativo_interrogativo_' \
                        'sujeito_elemental_TIdentif_NA'

        elif TEMA_DEFAULT_indicativo == 'declarativo' \
              and TEMA_IDENTIFICATIVO == 'equativo_decodificação':
           TEMA_IDEACIONAL = 'TID_default_indicativo_declarativo_TIdentif_' \
                        'equativo_decodificação'

        elif TEMA_DEFAULT_indicativo == 'interrogativo_polar' \
              and TEMA_IDENTIFICATIVO == 'equativo_decodificação':

           TEMA_IDEACIONAL = 'TID_default_indicativo_interrogativo_polar' \
                        '_TIdentif_equativo_decodificação'

        elif TEMA_DEFAULT_indicativo == 'interrogativo_sujeito_elemental' \
              and TEMA_IDENTIFICATIVO == 'equativo_decodificação':

           TEMA_IDEACIONAL = 'TID_default_indicativo_interrogativo_sujeito' \
                        '_elemental_TIdentif_equativo_decodificação'


        elif TEMA_DEFAULT_indicativo == 'declarativo' and \
              TEMA_IDENTIFICATIVO == 'equativo_codificação':

           TEMA_IDEACIONAL = 'TID_default_indicativo_declarativo_' \
                        'TIdentif_equativo_codificação'

        elif TEMA_DEFAULT_indicativo == 'interrogativo_polar' and \
              TEMA_IDENTIFICATIVO == 'equativo_codificação':

           TEMA_IDEACIONAL = 'TID_default_indicativo_interrogativo_polar_' \
                        'TIdentif_equativo_codificação'

        elif TEMA_DEFAULT_indicativo == 'interrogativo_sujeito_elemental' and \
              TEMA_IDENTIFICATIVO == 'equativo_codificação':

           TEMA_IDEACIONAL = 'TID_default_indicativo_interrogativo_sujeito_' \
                        'elemental_TIdentif_equativo_codificação'

  elif ORIENTACAO_MODAL == 'não_orientado' and ORIENTACAO_TRANSITIVA == 'direcional'\
        and SELECAO_TEMATICA == 'proeminente':
     TEMA_IDEACIONAL = 'TID_angulo'

  elif ORIENTACAO_MODAL == 'orientado' and ORIENTACAO_TRANSITIVA == 'não_direcional'\
        and SELECAO_TEMATICA == 'default':
     TEMA_IDEACIONAL = 'TID_elemental'

  elif ORIENTACAO_MODAL == 'não_orientado' and ORIENTACAO_TRANSITIVA == 'não_direcional'\
        and SELECAO_TEMATICA == 'proeminente':

     TEMA_IDEACIONAL = 'TID_prominente'

  return TEMA_IDEACIONAL

TEMA_IDEACIONAL(ORIENTACAO_MODAL='orientado',ORIENTACAO_TRANSITIVA='direcional',
           SELECAO_TEMATICA='default', TEMA_DEFAULT='indicativo',
           TEMA_DEFAULT_indicativo='declarativo', TEMA_IDENTIFICATIVO='NA',
           TEMA_ANGULO=None, TEMA_ELEMENTAL=None, TEMA_PROEMINENTE=None)

'TID_default_indicativo_declarativo_TIdentif_NA'
```

Fonte: elaborada para fins deste estudo.

Para além das subfunções de TEMA, também foi desenvolvida uma subfunção que organiza as escolhas para a TRANSITIVIDADE, e a subfunção que organiza o sistema de AGENCIAMENTO e em alguns casos, subfunções para organização das opções de tipo de Processo (e.g., Material e Relacional). A Figura 37 apresenta os destaques da função de escolhas para o AGENCIAMENTO.

O **Índice 1** apresenta o parâmetro de entrada, o 'indice', referente a uma das opções com um caminho no sistema. Note-se que, como para as outras subfunções desenvolvidas para compor a função geral de geração da oração, a função de 'AGENCIAMENTO(indice)' apenas retorna a escolha no sistema, para que a realização dessa escolha seja então realizada em conformidade com as outras escolhas para a organização da oração e seja renderizada por funções ja descritas das ordens inferiores na escala. O parâmetro 'indice' é então passado como chave para uma das opções de um dicionário e retorna o acúmulo de escolhas do sistema de AGENCIAMENTO, como destacado no **Índice 2** da Figura 37. O **Índice 3** da figura destaca um exemplo de execução da função, recebendo como entrada o 'indice:2' que corresponde ao acúmulo de opções 'AG_efetivo_operativo'.

Figura 37 – Função – Agenciamento

Fonte: elaborada para fins deste estudo.

A função de realização de AGENCIAMENTO é, por sua vez, aplicada como subfunção da função de TRANSITIVIDADE, como mostra a Figura 38.

Figura 38 – Função – Transitividade

```
def TRANSITIVIDADE(TIPO_DE_PROCESSO=None,indiceMat=None,
                   indiceAgen=None, indiceRel=None,
                   indiceAtrib=None):

    if TIPO_DE_PROCESSO == 'Material':
        Processo = PROCESSO_MATERIAL(indiceMat)
        Agenciamento = AGENCIAMENTO(indiceAgen)

    elif TIPO_DE_PROCESSO == 'Relacional':
        Processo = PROCESSO_RELACIONAL(indiceRel,indiceAtrib)
        Agenciamento = AGENCIAMENTO(indiceAgen)

    elif TIPO_DE_PROCESSO == 'Existencial':
        Processo = 'PR_Existencial'
        Agenciamento = AGENCIAMENTO(indiceAgen)

    elif TIPO_DE_PROCESSO == 'Verbal':
        Processo = 'PR_Verbal'
        Agenciamento = AGENCIAMENTO(indiceAgen)

    elif TIPO_DE_PROCESSO == 'Mental':
        Processo = 'PR_Mental'
        Agenciamento = AGENCIAMENTO(indiceAgen)

    TRANSITIVIDADE = Processo + '_' + Agenciamento
    return TRANSITIVIDADE
#
TRANSITIVIDADE(TIPO_DE_PROCESSO='Material', indiceMat=0,
               indiceAgen=2,indiceRel=None, indiceAtrib=None)

'PR material transformativo IMPA transitivo AG efetivo operativo'
```

Fonte: elaborada para fins deste estudo.

Na Figura 38, o **Índice 1** destaca os parâmetros de entrada, e.g., o TIPO_DE_PROCESSO, e os índices que são parâmetros para as funções aninhadas (PROCESSO_MATERIAL(<*args*>), PROCESSO_RELACIONAL(<*args*>)). O **Índice 2** apresenta o potencial de escolhas e as respectivas entradas em subfunções (e subsistemas). O **Índice 3** destaca um exemplo de execução, com a respectiva saída de execução. A saída acumula as escolhas de transitividade que serão passadas para a organização da oração. No exemplo, as escolhas retornam a transitividade para Processo material:transformativo_impacto transitivo com Agenciamento: efetivo_operativo.

3.2 EXPERIMENTOS

Os experimentos, tanto de acurácia, quanto de aplicação na realização textual dos verbos no material textual do @DaMataReporter, foram divididos em duas etapas: experimentos de desenvolvimento e experimentos de validação. Os testes realizados nos corpora de desenvolvimento, visando ao aprimoramento do módulo baseado em regras, são

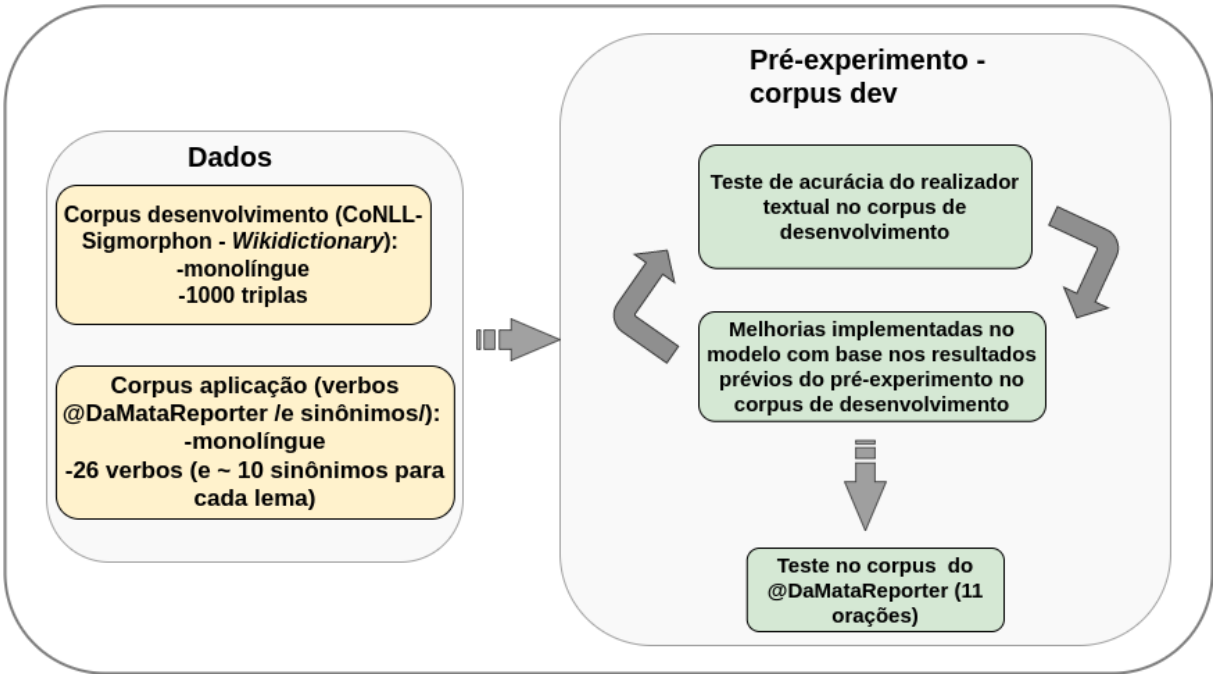
rotulados de experimentos de desenvolvimento (ou experimentos dev), e os experimentos nos *corpora* de teste são chamados de experimentos de validação (ou experimentos teste).

A subseção 3.2.1 a seguir apresenta o experimento de desenvolvimento - teste de acurácia no *corpus* de desenvolvimento e o experimento de de aplicação – realização textual de verbos extraídos do banco lexical de verbos do robô jornalista @DaMataReporter.

3.2.1 Experimentos de desenvolvimento – *dev*

Apos o desenvolvimento das funções que compõem o módulo de recursos lexico-gramaticais, anterior aos experimentos no *corpus* de teste, foram aplicados experimentos dev para avaliar acurácia e aplicação das funções de flexão verbal na realização textual de verbos extraídos do *corpus* do @DaMataReporter, visando ao aprimoramento (quando necessário) das funções. O experimento dev de acurácia consiste em comparar os resultados de flexão verbal do módulo de recursos baseado em regras com um *corpus* que estabelece um padrão ouro de flexão verbal (ver subseção 3.2.1.1.1 para detalhes dos dados que compõem o *corpus* de desenvolvimento). A seções a seguir apresentam um panorama geral da etapa de aprimoramento das funções do modelo baseado em regras. A Figura 39 apresenta uma prévia dos dados (*corpus* e seu formato de entrada) e a organização geral do experimento de desenvolvimento.

Figura 39 – Fluxograma – aprimoramento do realizador textual baseado em regras



Fonte: elaborada para fins deste estudo.

As seções a seguir apresentam mais detidamente os dados que compõem o *corpus* de desenvolvimento e o algoritmo de execução do experimento de desenvolvimento.

3.2.1.1 Experimento de desenvolvimento – acurácia no *corpus* de desenvolvimento e aprimoramento do realizador textual baseado em regras

3.2.1.1.1 Dados

Os dados linguísticos utilizados no experimento de desenvolvimento - acurácia - fazem parte do conjunto de dados compilados do *wikidictionary* para compor a tarefa compartilhada 1 (*shared task 1*), do CoNLL-SIGMORPHON–2017, como descrita em Cotterell et al. (2017). Nas edições de 2017 e 2018, o grupo se concentrou em duas tarefas, envolvendo diferentes línguas, dentre elas o português. A tarefa 1, flexão verbal a partir de um lema, envolveu 52 línguas, como apresentado em Cotterell et al. (2017) e 103 línguas, como descrito em Cotterell et al. (2018). O CoNLL_SIGMORPHON é uma iniciativa da *ACL(Association for Computational Linguistics)* que agrega cientistas de todo o mundo em esforço conjunto para desenvolver sistemas de automatização no âmbito da morfologia e fonologia computacional.

Para a edição de 2017–tarefa 1 do CoNLL_SIGMORPHON, foi compilada uma amostra contendo 12000 triplas, selecionadas aleatoriamente, compostas pelo lema, verbo flexionado e os parâmetros de entrada para a flexão verbal. Essa amostra inicial foi segmentada entre treinamento (de 100 a 10000 triplas, de acordo com o grau de exposição desejado na etapa de treinamento – baixo, médio ou alto); desenvolvimento (1000 triplas); e teste (1000 triplas),ou seja, foram disponibilizados *corpora* de desenvolvimento (para realização de testes durante a fase de implementação e treinamento dos modelos) e de teste (para teste efetivo dos sistemas).´

Nesta tese, a etapa de aprimoramento, experimento dev de acurácia, empregou o *corpus* de desenvolvimento da tarefa 1, no português, do CoNLL_SIGMORPHON–2017, composto de 1000 triplas contendo lema, verbo flexionado, e os parâmetros para a flexão (e.g., relatar/relatou/V;3;PL;IND;PST;PRF)[8]. As triplas são importadas como dispostas no Quadro 3.

Quadro 3 – Exemplos de triplas–corpus de desenvolvimento

lema	**verbo_conjugado**	**classe**	**pes/gên**	**número**	**modo**	**tempo**	**aspecto**
mistificar	mistificaram	V	3	PL	IND	PST	PRF
abraçar	abraces	V	2	SG	IMP	NEG	
sorver	sorvais	V	2	PL	IMP	NEG	

Fonte: Elaborado para fins deste estudo.

A subseção 3.2.1.1.2 apresenta o desenvolvimento do experimento dev de acurácia.

[8] Disponível em: <https://github.com/sigmorphon/conll2017/tree/master/all/task1>

3.2.1.1.2 Desenvolvimento

O experimento dev de acurácia consistiu, de maneira geral, em um algoritmo que realiza: a conversão dos rótulos dos parâmetros para flexão verbal do *corpus* de desenvolvimento, visando a adequá-los às funções do módulo baseado em regras; importação do *corpus* de desenvolvimento (português-dev) do CoNLL_SIGMORPHON–2017(tarefa 1); filtragem dos lemas e parâmetros no corpus; aplicação dos lemas e parâmetros na função **flexionar_verbo(*<args>*)**; comparação dos resultados obtidos com a conjugação do modelo baseado em regras com o padrão ouro estabelecido pelo *corpus* de desenvolvimento - o padrão ouro é estabelecido na coluna **verbo_conjugado**; a métrica de acurácia empregada trata-se do percentual de acerto em comparação com o padrão ouro; filtragem e armazenamento das conjugações corretas e erradas e comparação do percentual de acurácia; implementação de mudanças necessárias nas funções que não tiveram performance adequada. A seguir, são descritas cada uma das etapas desenvolvidas, em maior grau de detalhe:

- Desenvolvimento de uma função para conversão dos rótulos dos parâmetros: Os rótulos empregados na anotação do *corpus* de desenvolvimento seguem um esquema universal, interlinguístico, de parâmetros morfológicos (Sylak-Glassman et al. (2015)). Por sua vez, o módulo baseado em regras desenvolvido não segue esse esquema e, por isso, foi necessário desenvolver uma função de conversão dos parâmetros para que fossem dados como argumentos para a função de flexão verbal baseada em regras (flexionar_verbo(*args*)). Essa função faz a conversão dos parâmetros, passa-os à função de realização do verbo (verbo_geral(*args*)) e retorna o verbo flexionado.

- Importação do *corpus* de desenvolvimento: o arquivo .csv é importado como um *dataframe*, no formato apresentado no Quadro 3.

- Tratamento de valores '*nan*': foi utilizada uma função do *Scikit Learn* (o *SimpleImputer*) para o tratamento desses valores. O *SimpleImputer* provê um ambiente que permite a customização dos valores que substituem os valores *'nan'*. Para facilitar a conversão dos rótulos, valores *nan* foram trocados pela *string* '**none**'.

- Elaboração da lista de verbos conjugados: foi implementada uma estrutura de repetição que percorre o *corpus* e filtra os parâmetros necessários a serem passados para uma função que associa a conversão dos parâmetros e a passagem desses parâmetros convertidos para a subfunção responsável pela flexão do verbo (flexionarVerbo(*args*)). A saída para cada uma das iterações é um verbo conjugado, que é, por sua vez, concatenado à lista de verbos conjugados. A saída final da estrutura de repetição é uma lista completa com 1000 verbos conjugados, que serão confrontados com os

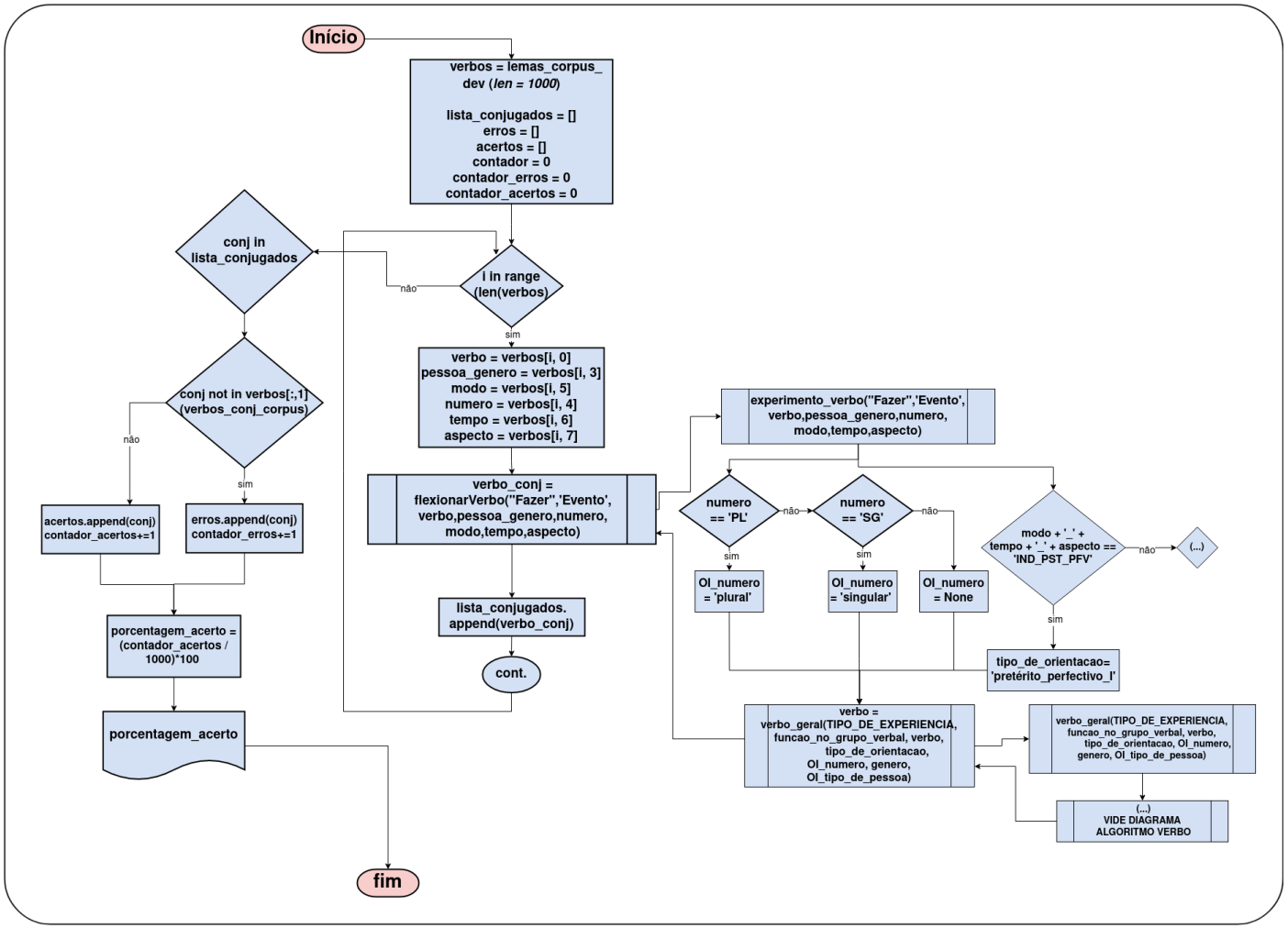
verbos conjugados que servem de padrão ouro no *corpus*, para que sejam extraídas os valores percentuais de acurácia.

- Elaboração de uma estrutura de repetição que percorre toda a lista de verbos conjugados, compara cada um dos itens ao padrão ouro estabelecido pelo *corpus* de desenvolvimento (coluna: verbo_conjugado). Caso a comparação seja positiva, concatena o verbo conjugado corretamente em uma lista de acertos, e estabelece um contador de acertos. Caso contrário, concatena o erro em uma lista própria e também estabelece um contador de erros. O teste de acurácia é uma contagem percentual de acertos em comparação com a coluna **verbo_conjugado**, disponibilizada no corpus disponibilizado pelo CoNLL_SIGMORPHON Cotterell et al. (2017).

- Os contadores de erros e acertos possibilitam a verificação do percentual de acurácia na tarefa de flexão verbal (número de acertos em comparação com o padrão ouro).

A Figura 40 apresenta um diagrama descrevendo o algoritmo do experimento de desenvolvimento - acurácia. Os resultados de acurácia obtidos no experimento dev serão apresentados no Capítulo 4. Como é possível verificar, o algoritmo inicia estabelecendo as entradas que serão alimentados às funções de flexão verbal (lemas extraídos do corpus de desenvolvimento e os respectivos parâmetros para flexão verbal); são estabelecidas listas vazias de verbos conjugados, erros, acertos, e contadores de erros e acertos; o algoritmo então caminha na lista de entradas, repassando os parâmetros para a função de flexão verbal e adicionando os verbos conjugados à lista de conjugados (reiterando até que a lista de entradas para flexão se finalize); o algoritmo então verifica se os verbos conjugados, da lista de conjugados, estão presentes na coluna de verificação de acerto de flexão – o padrão ouro do corpus de desenvolvimento – em caso positivo, o verbo flexionado é adicionado à lista de acertos, caso contrário, ele é adicionado à lista de erros; a cada reiteração os contadores de erros e acertos são atualizados; por fim, são computados os percentuais de acerto e erro.

Figura 40 – Fluxograma algoritmo–experimento dev - acurácia



Fonte: elaborada para fins deste estudo.

O algoritmo foi reiterado visando à verificação de potenciais erros no modelo baseado em regras. Novas funções foram implementadas no modelo, bem como correções e adequações, de acordo com os resultados de cada reiteração. Para além das adequações e adições de funções no módulo, foi necessário realizar alguns ajustes no corpus. O wikidicionário (*wikidictionary*) não faz distinção entre as variantes do português, e consequentemente, mantém algumas grafias mais prototípicas da variante europeia do idioma (provavelmente antecedentes ao novo acordo ortográfico). Dessa forma, alguns sufixos não poderiam ter sido sequer previstos no modelo baseado em regras, o que resulta, em alguns casos, em erro no funcionamento das funções, e em outros, falsos negativos na saídas das funções de flexão verbal. Assim, ajustes foram necessários em dois campos da base de dados. Em primeiro lugar, no campo 'lemas', alguns lemas continham a terminação do tipo '-OR' grafadas com acento circunflexo ('-ÔR'), o que efetivamente não ocorre no português brasileiro. O português brasileiro prevê verbos no infinitivo nas seguintes terminações (ou conjugações): 1ª ('-AR'), 2ª ('-ER' – e o caso particular '-OR', sem acento circunflexo, decorrente do desaparecimento da vogal temática), 3ª ('-IR') (ver Bechara (2012)). Além disso, o campo 'verbo_conjugado', que guarda todos os verbos conjugados de acordo com os parâmetros apontados no corpus, e, portanto, todo o padrão ouro ao qual as saídas do modelo são confrontadas, apresentava alguns verbos com o sufixo que também conservavam características da variante europeia do português, especialmente no 'Pretérito_Perfectivo_I'(1, plural, no indicativo). Nesses casos, falsos negativos influenciavam o percentual de acurácia do modelo. Veja o Quadro 4 para exemplos.

Quadro 4 – Exemplos de triplas–ajuste corpus

lema	**verbo_conjugado**	**classe**	**pes/gên**	**número**	**modo**	**tempo**	**aspecto**
pressagiar	pressagiámos	V	1	PL	IND	PST	PFV
antagonizar	antagonizámos	V	1	PL	IND	PST	PFV
economizar	economizámos	V	1	PL	IND	PST	PFV
oppôr	opporia	V	1	SG	COND		
decompôr	decompostas	V.PTCP	FEM	PL	PST		

Fonte: Elaborado para fins deste estudo.

A seguir, são apresentados os erros mitigados após as iterações no experimento de desenvolvimento. Note-se que, para facilidade

a) **verbos regulares**:

Nesta fase de aprimoramento do módulo baseado em regras, poucos ajustes foram necessários para que as funções genéricas de realização dos morfemas interpessoais pudessem dar conta de prever todo o paradigma de realização para todos os tipos de Orientação Interpessoal (i.e., Presente, Pretérito_Perfectivo, Passado_Volitivo etc.). Erros comuns mitigados com base nos resultados do experimento dev - de acurácia - se restringiram a acentos gráficos, erros de grafia de alguns morfemas interpessoais, erros na implementação

das subfunções.

b) **verbos que demandam mudanças no radical - irregulares** [9]:

No caso dos verbos irregulares, as funções genéricas (padrão) de realização dos morfemas interpessoais não contemplam todo o potencial de flexão verbal. Nesses casos, foram implementadas funções de flexão particulares para cada caso (ou conjunto de casos semelhantes, os quais mantinham uma potencial de regularidade, mesmo dentre os irregulares). Note-se que, por não ser possível abstrair regras de conjugação que contemplem todo o paradigma para verbos irregulares, devido à variabilidade tanto de radical quanto de flexão, para uma modelagem manual, esta é uma área de desafio para o linguista/programador, já que é necessário modelar caso a caso, o que demanda muito tempo e esforço. Para contornar essa dificuldade, foram modelados os verbos irregulares mais prototípicos, contudo, a lista de verbos irregulares modelados neste trabalho não é exaustiva.

3.2.1.2 Experimento de desenvolvimento – aplicação para realização textual de verbos

O experimento dev - de aplicação - consiste em dois subexperimentos: **i)** realização textual de todo o paradigma de flexão verbal (todas as opções do sistema de ORIENTAÇÃO INTERPESSOAL (Presente, Imperativo_I, Pretérito_Perfectivo_I etc.) para os verbos quem compõem o conjunto de verbos do léxico do sistema de geração do robô-jornalista @DaMataReporter; **ii)** realização textual de verbos similares aos verbos do léxico do robô-jornalista @DaMataReporter, extraídos de um modelo de *embedding* de palavras, dentro dos parâmetros gramaticais para a flexão em exemplos de sentenças também extraídas do *corpus* do robô-jornalista @DaMataReporter. A Figura 41 a seguir apresenta um diagrama com o esboço geral desta etapa dos experimentos.

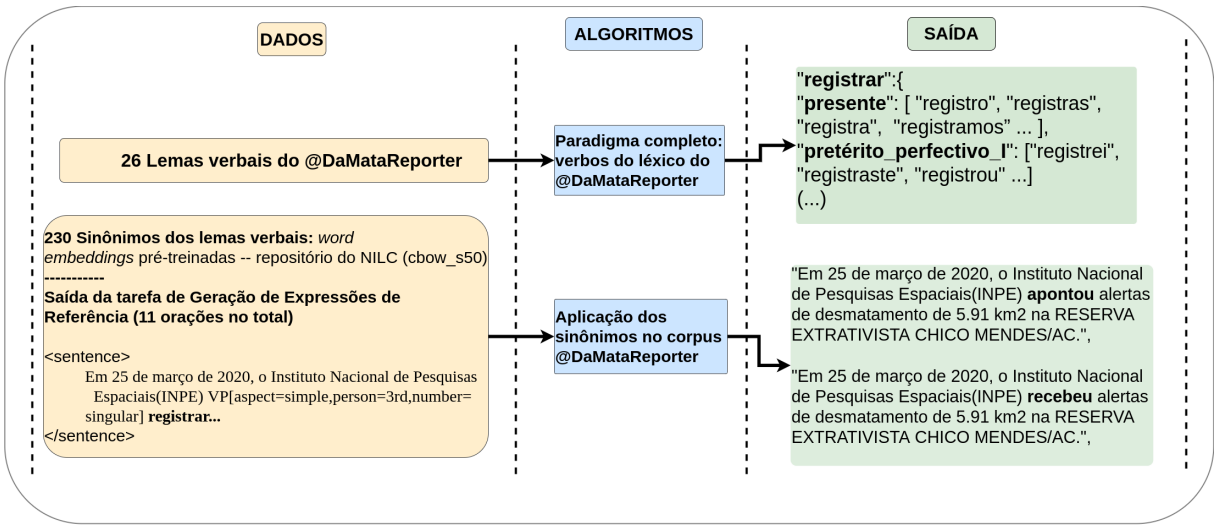
A subseção 3.2.1.2.1 a seguir apresenta mais detidamente os dados do experimento dev de aplicação do modelo baseado em regras na realização textual dos verbos no *corpus* do robô-jornalista @DaMataReporter.

3.2.1.2.1 Dados

Como apontamos na subseção 3.2.1.2, o experimento dev de aplicação do modelo para a realização textual do verbo é composto por dois subexperimentos: a realização de todo o paradigma de flexão dos verbos que compõem o léxico do @DaMataReporter; e a realização textual de verbos similares aos verbos do robô-jornalista, sendo aplicados em exemplos de sentenças extraídas do @DaMataReporter. Para a realização do paradigma

[9] Neste trabalho são denominados verbos irregulares os verbos que demandem uma mudança em seu radical no processo de flexão. Adotou-se, nesse sentido, uma perspectiva do processamento computacional necessário no radical do verbo no infinitivo.

Figura 41 – Fluxograma–teste aplicação (@DaMataReporter)



Fonte: elaborada para fins deste estudo.

de flexão verbal completo, foram utilizados 26 lemas verbais que compõem o banco de itens lexicais (verbos) do sistema de geração do robô-jornalista @DaMataReporter. A seguir verifica-se a lista completa de verbos extraídos do banco de elementos lexicais para realização textual de verbos na linha de produção do @DaMataReporter:

- 'desmatar', 'registrar', 'detectar', 'identificar', 'atingir', 'contabilizar', 'somar', 'totalizar', 'alcançar', 'chegar', 'reportar', 'observar', 'ser', 'representar', 'acontecer', 'ocorrer', 'existir', 'ter', 'apresentar', 'possuir', 'estar', 'acumular', 'relatar', 'sofrer','informar', 'haver';

Note-se que a lista não contém todos os verbos que podem ser verificados após a publicação. A lista se restringe aos verbos que são passíveis de variação, e efetivamente variam devido às flexões (principalmente de gênero e número) em seu contexto próximo, ou seja, verbos que não têm potencial de variação dentro da geração das sentenças não são incluídos no banco lexical para a geração. O banco lexical de verbos é composto por *tags*, salvas em arquivo *.json*, no formato de dicionários do tipo **"VP[aspect=simple,tense=past,voice=active,person=3rd,number=singular] registrar":["registrou]"**, onde pode-se verificar uma chave contendo os parâmetros para a flexão e o lema verbal; e um valor com o verbo flexionado. A lista de lemas é extraída das *tags* em uma das etapas do algoritmo desenvolvido para o experimento dev e são passadas para a função de flexão verbal do modelo baseado em regras. No âmbito da linha de produção do @DaMataReporter, essas *tags* são utilizadas na etapa de anotação do corpus.

No segundo subexperimento, foi utilizado o modelo de vetores de palavras (*word embeddings*) pré-treinados, no modelo de algoritmo *Word2vec*, o CBOW_50 dimensões

(*Continuous Bag of Words*)[10] disponibilizado pelo NILC – Núcleo Interinstitucional de Linguística Computacional – para a extração de verbos similares aos verbos do banco lexical do @DaMataReporter. Os *word embeddings* são modelos que resultam de técnicas de mineração de texto que convertem as palavras e as representam matematicamente como vetores. Os vetores são alocados num espaço matricial que possibilita que sejam estabelecidas relações de significado/similaridade entre as palavras, através de operações matemáticas. Do modelo de vetores de palavras, foram extraídas palavras similares, para cada um dos verbos da lista de lemas extraída do léxico do @DaMataReporter (aproximadamente 10 palavras similares para cada um dos 26 verbos extraídos do banco lexical do robô-jornalista - compondo aproximadamente 230 verbos similares)[11], para serem passados como lema para a função de realização verbal do realizador baseado em regras. A Figura 42, a seguir, dispõe dois dicionários contendo, cada um, o lema extraído do léxico do @DaMataReporter e suas respectivas palavras similares:

Figura 42 – Exemplos de palavras similares

```
"registrar":
["conter","apontar","configurar","encaminhar","usar",
"receber","ofertar","descartar", "liberar","trazer"],

"alcançar":
[ "atingir", "conquistar","representar","favorecer",
"refrear", "manter",  "superar", "suster","canalizar",
"alcançar"]
```

Fonte: elaborada para fins deste estudo

Após sua extração do modelo de vetores de palavras, os verbos similares aos verbos do banco lexical do robô-jornalista foram alocados em dicionários e salvos em formato *.json* para serem empregados como parâmetros para a função de flexão do verbo, na aplicação em 11 orações extraídas do robô-jornalista, resultando em uma lista de 90 orações, na fase seguinte do subexperimento. O Exemplo na Figura 43 apresenta uma sentença extraída do *corpus* do robô-jornalista. Esta fase da linha de produção de geração no robô-jornalista, a subtarefa de realização textual, recebe a sentença com *tags* de verbos a serem flexionados (VP[<parâmetros>] <lema>).

Figura 43 – Exemplo sentença @DaMataReporter

```
Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais
(INPE) VP[aspect=simple,person=3rd,number=singular] registrar alertas
de desmatamento de 5.91 km² na RESERVA EXTRATIVISTA CHICO MENDES/AC.
```

Fonte: elaborada para fins deste estudo com dados extraídos do corpus do @DaMataReporter

[10] Disponível em: <http://www.nilc.icmc.usp.br/embeddings>
[11] Ver lista em Apêndice B

A subseção 3.2.1.2.2 apresenta o desenvolvimento do experimento dev de aplicação das funções do modelo baseado em regras para a realização textual dos verbos flexionados.

3.2.1.2.2 Desenvolvimento

Como antecipado na introdução da subseção 3.2.1.2, este experimento de desenvolvimento é dividido em dois subexperimentos: a realização textual de todo o paradigma de flexão verbal dos verbos que compõem o banco lexical de verbos do robô-jornalista; e a realização textual de verbos similares em orações extraídas do robô-jornalista. A seguir, cada um desses subexperimentos será apresentado mais detidamente.

O primeiro subexperimento consiste em um algoritmo que foi desenvolvido seguindo as seguintes etapas

- **Extração dos lemas do robô-jornalista**: como apresentado na subseção 3.2.1.2.1 sobre os dados, neste subexperimento, foram extraídos 26 lemas do banco lexical de verbos do robô-jornalista para uma realização do paradigma completo de flexão verbal. Essa parte do algoritmo foi responsável por iterar sobre as *tags* que compõem o conjunto de léxico verbal (isto é, VP[<parâmetros>] <lema>), filtrar e extrair o lema da *tag*, e concatenar cada um deles em uma lista, que foi, por sua vez, alimentada à função para a flexão verbal.

- **Realização do paradigma completo para os verbos do conjunto léxico do robô-jornalista**: dados os verbos salvos em uma lista de lemas, foi desenvolvido um algoritmo que itera sobre a lista de lemas e sobre as listas contendo todos os potenciais parâmetros (i.e., gênero, número, termo no sistema de ORIENTAÇÃO INTERPESSOAL – Presente, Pretérito_Perfectivo_I etc., tipo de pessoa – 1, 2, 3pessoa etc.) para a realização da flexão verbal. A cada uma das iterações, os verbos flexionados são concatenados em uma lista, que por sua vez é concatenada em um dicionário que é atualizado ao fim de cada iteração da estrutura de repetição. Após o fim dos ciclos de repetição, o dicionário contendo todo o paradigma de flexão verbal foi salvo em formato *.json*.

O segundo subexperimento desta etapa da pesquisa consistiu em testar a realização textual de verbos similares dos lemas extraídos do banco lexical do robô-jornalista em exemplos de orações extraídas do @DaMataReporter. O algoritmo para a realização desta tarefa foi desenvolvido desta forma:

- **Importação do modelo de vetores de palavras**: o modelo já treinado, no algoritmo *Word2Vec–cbow* foi baixado em formato .txt e importado no ambiente de programação;

- **Elaboração do dicionário de palavras similares**: foi desenvolvido um algoritmo com estrutura de repetição que itera sobre a lista de lemas verbais do @DaMataReporter e, para cada um dos lemas, realiza uma busca no vocabulário do modelo de vetores de palavras pelas 10 entradas mais similares, através da função nativa do *Gensim*, a '*most_similar*'. Essa função retorna as entradas mais semelhantes, dada uma palavra de interesse. No fim de cada iteração da estrutura de repetição, cada lema e suas respectivas palavras similares foram concatenados em um dicionário. Na saída da estrutura de repetição, o dicionário completo, contendo cada lema e os 10 verbos similares (compondo cerca de 25 lemas e 230 verbos similares) foram salvos em formato *.json*.

- **realização textual de sentenças do @DaMataReporter**: foram selecionadas 11 orações que compõem o *corpus* do robô-jornalista para testar a realização textual dos lemas similares. Foi desenvolvido um pequeno algoritmo que itera sobre as listas de verbos similares e, para cada iteração, passa o lema e os parâmetros já determinados previamente (seguindo os parâmetros previstos na sentença, e.g, tipo_de_orientacao='pretérito_perfectivo_I',OI_numero='singular', genero=None, OI_tipo_de_pessoa='3pessoa') para a função de flexão verbal do modelo baseado em regras. Em cada iteração, a sentença é atualizada com o verbo similar já flexionado e concatenada em uma lista de sentenças. O pequeno algoritmo resultou em aproximadamente 90 orações com lemas similares flexionados pela função de flexão verbal do realizador superficial. A lista contendo todas as sentenças atualizadas foi salva em arquivo no formato *.json*.

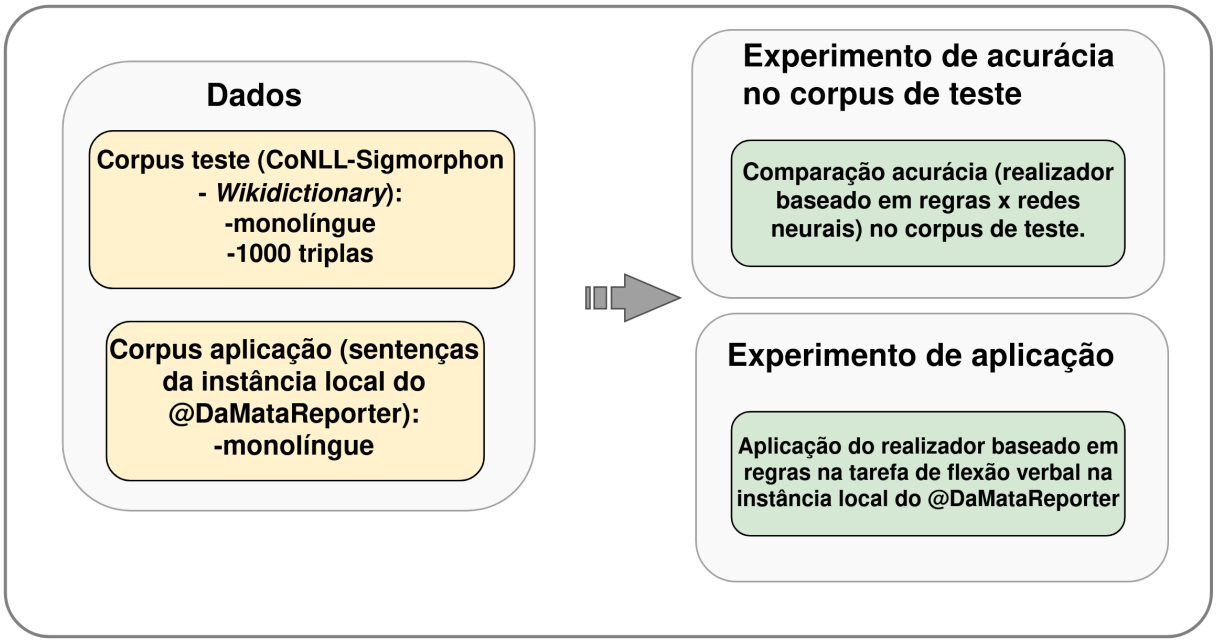
Os resultados do experimento de desenvolvimento - aplicação - serão apresentados no Capítulo 4.

3.2.2 Experimentos de validação – *teste*

Após a realização do experimento dev - acurácia - no *corpus* de desenvolvimento e do experimento dev - aplicação, com a flexão de palavras similares aos verbos do @DaMataReporter, extraídos do modelo de vetores de palavras e a realização textual de exemplos de sentenças do robô-jornalista, foram realizados experimentos de validação de acurácia, no *corpus* de teste e de aplicação, na linha de produção do robô-jornalista, análogos ao desenho dos experimentos de desenvolvimento. O experimento de validação de acurácia foi realizado em um dos corpora de teste, extraído da tarefa compartilhada no CoNLL_SIGMORPHON–2017. O experimento de validação de acurácia objetivou testar a acurácia das funções de flexão verbal após a etapa pré-experimental no *corpus* de desenvolvimento e o aprimoramento do modelo baseado em regras. O experimento de validação de aplicação visou a aplicar as funções do modelo baseado em regras em uma replica do robô-jornalista @DaMataReporter, armazenada localmente, para checar

a viabilidade de aplicação na linha de produção no perfil oficial do robô-jornalista. A Figura 44 apresenta o desenho geral dos experimentos de validação.

Figura 44 – Fluxograma com *setup* dos experimentos de validação



Fonte: elaborada para fins deste estudo.

A subseção 3.2.2.1 a seguir apresenta o desenho do experimento de validação de acurácia detalhadamente.

3.2.2.1 Experimento de validação – acurácia no *corpus* de teste

3.2.2.1.1 Dados

Como antecipado no parágrafo introdutório da subseção 3.2.2, o desenho do experimento de validação de acurácia segue o mesmo desenho experimental do experimento de desenvolvimento, mas é aplicado em um *corpus* de teste. O *corpus* utilizado nesta etapa do experimento é um dos *corpora* de teste disponibilizados pelo CoNLL_SIGMORPHON–2017, o 'portuguese-uncovered-test'[12], que segue o mesmo formato e dimensões do *corpus* de desenvolvimento (utilizado para o experimento dev - de acurácia): é composto de 1000 triplas contendo lema, verbo flexionado, e os parâmetros para a flexão (e.g., relatar/relatou/V;3;PL;IND;PST;PRF). A triplas são importadas no mesmo formato apresentado no Quadro 3.

3.2.2.1.2 Desenvolvimento

Seguindo o desenho do experimento de desenvolvimento - de acurácia - realizado no *corpus* de desenvolvimento, o experimento de validação no *corpus* de teste consistiu no

[12] Disponível em:<https://github.com/sigmorphon/conll2017/blob/master/answers/task1/portuguese-uncovered-test>

uso do mesmo algoritmo. Em suma, o algoritmo é composto pelas seguintes fases:

- Conversão dos rótulos dos parâmetros para a flexão verbal do *corpus* de teste para alimentá-los adequadamente à função de flexão verbal do modelo baseado em regras; em seguida, é realizada a importação do *corpus* de teste ('portuguese-uncovered-test') do CoNLL_SIGMORPHON–2017(tarefa 1).
- Filtragem dos lemas e parâmetros do *corpus* de teste; aplicação dos lemas e parâmetros na função **flexionarVerbo(*<args>*)**.
- Comparação dos resultados obtidos com a conjugação do modelo baseado em regras com o padrão ouro estabelecido pelo *corpus* de teste.
- Filtragem e armazenamento das conjugações corretas e incorretas após a comparação.
- Comparação do percentual de acurácia do modelo baseado em regras com o percentual de acurácia dos modelos estado da arte apresentados no CoNLL_SIGMORPHON–2018 (tarefa 1) para a flexão verbal no português brasileiro.

Os resultados dos testes de acurácia, a partir da comparação de performance entre o modelo baseado em regras e o padrão ouro estabelecido pelo *corpus* de desenvolvimento; e a partir da comparação de performance entre o modelo baseado em regras desenvolvido no âmbito deste trabalho e os modelos desenvolvidos no âmbito do CoNLL_SIGMORPHON–2018, no *corpus* de teste, serão apresentados e discutidos no Capítulo 4.

A subseção 3.2.2.2 apresenta o desenho do experimento de validação - aplicação do modelo baseado em regras na linha de produção de uma instância local do robô-jornalista @DaMataReporter.

3.2.2.2 Experimento de validação – aplicação em instância local da linha de produção do robô-jornalista

Visando a verificar a viabilidade de aplicação das funções de flexão verbal do modelo baseado em regras diretamente na linha de produção do robô-jornalista, uma instância local do robô foi implementada. Por ser uma instância local, não se aplica para esse cenário de teste a postagem automática das notícias na rede social. Essa instância roda com dados de períodos especificados convenientemente (setembro 2020 a março 2021), apenas para verificação das funções de flexão. Os resultados dessa verificação de viabilidade, ou seja, as sentenças geradas com verbos flexionados pelas funções de flexão do modelo baseado em regras, serão apresentados no Capítulo 4.

A subseção 3.2.2.2.1 a seguir apresenta os dados utilizados no experimento de validação - aplicação das funções do modelo na linha de produção do @DaMataReporter.

3.2.2.2.1 Dados

No experimento dev de aplicação, como apresentado na subseção 3.2.1.2, foram utilizados exemplos de sentenças, extraídas do *corpus* do robô-jornalista @DaMataReporter. Essas sentenças emulavam o que seria a saída da subtarefa de geração de expressões de referência, que seriam, por sua vez, a entrada para a subtarefa de realização textual. Contudo, essas sentenças foram extraídas convenientemente para testes isolados da função de flexão verbal, ou seja, fora do ambiente de geração do robô-jornalista, e, portanto, foram pressupostas e abstraídas todas as etapas anteriores na linha de produção do robô-jornalista. Para esta etapa do experimento de validação, como antecipado na introdução do subseção 3.2.2.2, uma instância local da linha de produção do robô-jornalista foi implementada para testar as funções de flexão verbal desenvolvidas no modelo baseado em regras. Assim, todas as etapas da geração estão localmente em produção, excetuando-se a de publicação no *Twitter*. Note-se, então, que diferente do teste de realização textual desenvolvido no experimento de desenvolvimento (a seleção conveniente de alguns exemplos de sentenças), no experimento de validação, mais sentenças do *corpus* estão disponíveis para a realização superficial (seguindo os critérios da linha de produção oficial do robô-jornalista).

A primeira etapa da linha de produção é a seleção de conteúdo para a geração. Reitera-se que os dados para esta etapa são extraídos através da API do DETER, que disponibiliza os dados publicamente[13].

Para o treinamento das 'gramáticas' e extração dos *templates* para a geração, foram anotadas 5 entradas (*entry*) de cada um dos corpora (o de postagens mensais e o de postagens diárias) do @DaMataReporter. Cada entrada do *corpus* contém as *tags* que estabelecem os *templates*, baseando-se nas etapas do processo de geração: seleção de conteúdo (<*meaning*>), ordenação do discurso (<*ordering*>), estruturação(<*structuring*>), lexicalização(<*lexicalization*>), expressões de referência (<*reference*>), realização superficial (<*text*>). As anotações mais críticas, no que tange ao objeto desta tese, são realizadas no ambiente da *tag* <*lexicalization*>, pois é nele que se encontram os parâmetros que serão alimentados às funções de flexão verbal na linha de produção. Os verbos são representados no corpus, na *tag* <*lexicalization*> no formato 'VP[experience=<valor>,functionInGroup=<valor>, lemma=<valor>, person=<valor>, number=<valor>, mood=<valor>, tense=<valor>, aspect=...]'. A Figura 45 apresenta a estrutura básica das entradas nos *corpora* do robô (apresenta apenas uma entrada e uma sentença da entrada), extraídas do *corpus* para postagens mensais e com as anotações já convertidas para o padrão necessário para a passagem de parâmetros para a função de flexão verbal do modelo baseado em regras.

Os *corpora* empregados para o treinamento do robô-jornalista efetivamente em

[13] Disponível em: <http://terrabrasilis.dpi.inpe.br/app/dashboard/alerts/legal/amazon/daily>

produção são compostos por 14 entradas, o mensal, e 23 entradas, o diário. A subseção 3.2.2.2.2 a seguir apresenta o desenvolvimento do teste de viabilidade do uso das funções de flexão verbal do modelo baseado em regras na instância local do robô-jornalista.

3.2.2.2.2 Desenvolvimento

O teste de verificação de viabilidade de aplicação segue os mesmos passos da linha de produção efetiva do robô-jornalista. Como demonstrado na subseção 2.3.1.1.1, a ingestão dos dados é realizada automaticamente e extrai dados do DETER (INPE); dadas as regras de negócio, fazem a interpretação e selecionam o conteúdo relevante a ser gerado textualmente, considerando convenientemente o período de setembro de 2020 a março de 2021 para a geração das postagens mensais, e um período de 30 dias anteriores à data em que o experimento é executado. As unidades de conteúdo selecionadas passam então por uma função de geração, que as submetem aos parâmetros de ordenação, estruturação, lexicalização, e geração de referências estabelecidos na etapa de treinamento, para cada um dos modelos (mensal e diário).

Como antecipado na subseção 3.2.2.2.1, a etapa de anotação que é crítica para a aplicação das funções de flexão verbal do modelo baseado em regras, na etapa de realização textual, é a *<lexicalization>*, nos *corpora*, na qual operam-se as anotações de *tags* para verbos, substantivos etc, que carregarão os parâmetros gramaticais para sua flexão na etapa de realização textual. Atualmente, na linha de produção efetiva do robô-jornalista, a flexão verbal ocorre desta forma: na etapa de realização textual, o sistema rastreia o *corpus* por *tags* do tipo **'VP[aspect=simple, tense=past, voice=passive, person=3rd, number=singular, gender=male]'** que foram anotadas na *tag: lexicalization*, anterior ao treinamento do modelo; ao encontrar, o sistema então busca um arquivo de léxicos,

Figura 45 – Exemplo de anotação do corpus

```
<entry date="2019-6-01">
        <meaning>
            <unit>TOTAL_DEFORESTATION(area="914",location="deter-amz",
              month="6",year="2019")</unit>
(...)
        </meaning>
        <ordering>
            <unit>TOTAL_DEFORESTATION(area="914",location="deter-amz",
              month="6",year="2019")</unit>
(...)
        </ordering>
        <structuring>
            <paragraph>
                <sentence>
                    <unit>TOTAL_DEFORESTATION(area="914",location=
                    "deter-amz",month="6",year="2019")</unit>
                </sentence>
(...)
        </structuring>
        <lexicalization>
            <paragraph>
                <sentence>
                    No mês de MONTH_1 de YEAR_1 , VP[experience=Ser,
                    functionInGroup=Auxiliar,lemma=ser,person=3,gender=none
                    number=Plur,mood=Ind,tense=Past,aspect=Perf] VP[experience
                    =Fazer,functionInGroup=Evento,lemma=desmatar,person=none,
                    gender=Masc,number=Plur,mood=none,tense=Past,aspect=Perf]
                    AREA_1 de LOCATION_1 , de acordo com o monitoramento de
                    INSTITUTE_1 .
                </sentence>
            </paragraph>
(...)
        </lexicalization>
        <references>
            <paragraph>
                <sentence>
                   <reference tag="LOCATION_1" entity="deter-amz"
                    gender="female" number="singular">a Amazônia Legal</reference>
                </sentence>
            </paragraph>
(...)
        </references>
        <text>
            <paragraph>
                <sentence>
                    No mês de junho de 2019, foram desmatados 914 km²
                    da Amazônia Legal, de acordo com o monitoramento
                    do Instituto Nacional de Pesquisas Espaciais (INPE).
                </sentence>
            </paragraph>
(...)
        </text>
</entry>
```

Fonte: *Corpus* do @DaMataReporter (grifo meu).

anotados e salvos como dicionários em formato *.json*, que armazenam dados do tipo **chave:valor**(as **chaves** correspondem às *tags* encontradas no corpus, com os parâmetros para a flexão dos verbos, e o **valor** corresponde ao verbo em sua forma flexionada).

Esse processo busca a contrapartida **chave** (contendo os parâmetros) da *tag* encontrada no *corpus* e retorna então o **valor** (verbo flexionado), concatenando-o à sentença objetivo da geração na posição conveniente.

Por sua vez, na instância local, elaborada para teste das funções de flexão verbal

do modelo baseado em regras, operaram-se mudanças tanto na maneira como são feitas as anotações dos corpora (preparando-os para a etapa de treinamento das 'gramáticas'), quanto na função *Realization.generate* do sistema do robô-jornalista, que é responsável por executar as etapas de geração. A anotação das *tags* 'VP[<parâmetros>]' sofreu adequações que permitem a passagem dos parâmetros necessários para o funcionamento das funções de flexão do realizador baseado em regras, seguindo o padrão **'VP[experience=Ser, functionInGroup=Evento, lemma=ser, person=3,gender=none number=Sing, mood=Ind, tense=Past, aspect=Perf]'**, que sendo passada à função de flexão retorna o verbo flexionado **'foi'**. Os pares **chave:valor** dos parâmetros (e.g., lemma=ser, person=3, number=Sing, mood=Ind, tense=Past, aspect=Perf) seguem o padrão de anotação estabelecido pelo *Universal Dependencies*[14].

As subfunções que compõem a função para flexão verbal do realizador textual baseado em regras, por terem sido desenvolvidas sob o arcabouço teórico da Linguística Sistêmico-Funcional, e mais especificamente amparando-se em descrições de base Sistêmico-Funcional do português brasileiro, adotam, em grande medida, a nomenclatura apresentada pelos autores e em conformidade com as regras de formatação propostas pela teoria Sistêmico-Funcional[15]. Contudo, entende-se a necessidade, para uma facilidade de anotação dos corpora, da adoção de um padrão de anotação já amplamente difundido e aceito pela comunidade científica. Nesse sentido, a função final de flexão verbal desenvolvida nesta tese, e que aninha todas as subfunções necessárias para a execução da tarefa de realização textual do verbo no português brasileiro, faz as conversões necessárias dos rótulos de anotação baseados no modelo do *Universal Dependencies* que são pertinentes e encontram contrapartida análoga na nomenclatura dos elementos gramaticais que compõem a função de flexão verbal desenvolvida nesta tese.

Outra mudança necessária foi implementada na função que faz a realização de todo o processo de realização textual do robô-jornalista. Como mencionado, na linha de produção do robô efetiva atualmente, a função de geração (*generate*) da classe *Realization* busca *tags* do tipo 'VP[...]' no *corpus* e em seguida realiza uma busca pela **chave** correspondente no arquivo de léxicos. Ao encontrar essa chave, o sistema retorna o **valor** que corresponde ao verbo flexionado, seguindo os parâmetros da chave. Na instância local do robô-jornalista, elaborada para teste, a função de flexão do realizador baseado em regras foi incorporada à função de geração(*Realization.generate*). As alterações no padrão de anotação, e a incorporação da função de flexão diretamente na função geral de realização, permitem que

[14] Disponível em: <https://universaldependencies.org/u/feat/index.html>

[15] Tentei, na medida do possível, conciliar normas adotadas pela linguística Sistêmico-Funcional com normas específicas da linguagem de programação adotada para desenvolvimento, especialmente no que tange à nomeação de parâmetros e seus valores. Contudo, em alguns casos as diferenças se mostravam irreconciliáveis. Nesses casos, tomei certa liberdade na nomeação, o que resulta em casos de ruptura com ambas as perspectivas de formatação, mas que não interferem diretamente no funcionamento das funções.

o sistema busque a *tag* 'VP[...]' no *corpus* e automaticamente passe os parâmetros como entrada para a função de flexão verbal do realizador baseado em regras, que retorna o verbo flexionado de acordo com os parâmetros. Esta alterações eliminam a necessidade do acesso intermediário ao léxico para efetuar a busca das chaves correspondentes às *tags* com os parâmetros para flexão e dos valores (verbos flexionados) correspondentes.

Dadas essas alterações necessárias no padrão de anotação e a anotação dos *corpora* seguindo os padrões necessários para a passagem dos parâmetros para as funções de flexão verbal, foram treinados os modelos, com base nos *corpora* anotados, formando-se os *templates* que ficam disponíveis para a geração. Após o treinamento, foram executadas as etapas da linha de produção seleção de conteúdo – restrito ao período de setembro de 2020 e março de 2021, ordenação, estruturação, lexicalização, geração de expressões de referência e realização superficial – com as funções de flexão em funcionamento para a realização textual dos verbos. A saída do sistema foi salva em formato .txt. O Capítulo 4 apresenta os resultados dos experimentos de validação de acurácia e de aplicação do realizador textual superficial baseado em regras na linha de produção do robô jornalista @DaMataReporter.

4 RESULTADOS E DISCUSSÃO

Como apresentado no Capítulo 3, o experimento de desenvolvimento foi segmentado em dois subexperimentos. Um experimento de desenvolvimento - envolvendo dois módulos: um de acurácia, no *corpus* de desenvolvimento do CoNLL-SIGMORPHON de 2017, visando ao teste e aprimoramento do modelo baseado em regras; e um de aplicação, em verbos que compõem o banco lexical do robô-jornalista @DaMataReporter e em conjuntos de verbos similares, minerados no modelo de vetores de palavras, *cbow_50*, do NILC, também aplicados em exemplos de orações extraídas do @DaMataReporter. Um experimento de validação - também subdividido em dois módulos: um de acurácia, aplicado no *corpus* de teste do CoNLL-SIGMORPHON de 2017, realizando-se o contraste entre os resultados do realizador baseado em regras e resultados alcançados por redes neurais desenvolvidas no âmbito do CoNLL-SIGMORPHON; e de aplicação na linha de produção, em uma instância local do robô-jornalista @DaMataReporter. As subseções a seguir apresentam os resultados para cada um dos experimentos: experimento dev de acurácia (subseção 4.1.1) e de aplicação (subseção 4.1.2) e experimento de validação de acurácia(subseção 4.2.1) e de aplicação (subseção 4.2.2).

4.1 RESULTADOS – EXPERIMENTO DE DESENVOLVIMENTO

4.1.1 Resultados de acurácia – experimento de desenvolvimento

Nesta seção, são apresentados os resultados de acurácia do realizador baseado em regras, em sua aplicação no corpus de desenvolvimento do CoNLL-SIGMORPHON 2017, em comparação com os resultados obtidos pelos sistemas mais bem colocados para o português no âmbito do projeto do ConNLL-SIGMORPHON, nas edições de 2017 e 2018. Em primeiro lugar, é delineado o desenho experimental do CoNLL-SIGMORPHON (COTTERELL et al., 2017; COTTERELL et al., 2018), em particular as métricas aplicadas para o teste de acurácia dos sistemas no âmbito do projeto, para que sejam estabelecidos os parâmetros de comparação. Então, são apresentados os testes de acurácia do realizador no corpus de desenvolvimento (*corpus-dev*). Por fim, serão discutidos os resultados de acurácia e os tipos de erros apresentados pelo realizador baseado em regras.

No âmbito do CoNLL-SIGMORPHON (2017 e 2018), subtarefa 1, os sistemas (supervisionados) desenvolvidos devem retornar uma forma flexionada, dado um lema e parâmetros pertinentes para a flexão, nas diferentes línguas sob análise. Acesso ao conjunto de dados para treinamento dos sistemas foi dividido em baixo (100 *tokens*), médio (1000 *tokens*) ou alto (1000 *tokens*). Os testes dos sistemas foram executados no *corpus* de teste, composto por 1000 triplas (lema e parâmetros). A acurácia dos sistemas foi computada

através da porcentagem de acerto dos paradigmas esperados, ou seja, o percentual de acerto dos sistemas quanto à previsão dos paradigmas de flexão dados os parâmetros (como já foi mencionado no Capítulo 3, os corpora de teste contêm células com os verbos flexionados corretamente, de acordo com os parâmetros, formando o padrão ouro). O CoNLL-SIGMORPHON 2018 (52 línguas sob análise) segue o mesmo desenho experimental de (COTTERELL et al., 2017), para subtarefa 1, de flexão, com a diferença que introduz novas línguas (103 ao todo).

Para computar a acurácia do modelo baseado em regras, foi utilizada a mesma métrica aplicada nesses experimentos: com base nas 1000 entradas estabelecidas pelo padrão ouro, foram comparadas as porcentagens de acerto entre o realizador baseado em regras e os resultados dos sistemas desenvolvidos no âmbito no CoNLL-SIGMORPHON. Note-se que, por se tratar ainda do experimento de desenvolvimento, fase de aprimoramento do sistema, são comparados o resultado obtido no *corpus-dev*, ou seja, o corpus de desenvolvimento, com os resultados finais dos sistemas no âmbito do CoNLL. O teste no *corpus-test* foi aplicado após esta fase e será apresentado na subseção 4.2.1. A Tabela 1 apresenta os resultados dos sistemas desenvolvidos no âmbito da iniciativa do CoNLL-SIGMORPHON, para os três moldes de acesso aos recursos para treinamento (baixo, médio e alto) e o resultado do teste de acurácia alcançado pelo realizador baseado em regras desenvolvido nesta tese. Entre parênteses, verificam-se os identificadores dos grupos que desenvolveram os sistemas.

Tabela 1 – Resultados de acurácia – experimento de desenvolvimento

	high	*medium*	*low*
SIGMORPHON 2017	99,30%(LMU-2)	95,00%(LMU-2)	73,30%(CLUZH-7)
SIGMORPHON 2018	98,60%(uzh-2)	94,80%(uzh-2)	75,80%(uzh-2)
Realizador baseado em regras		98,4%	

Fonte: elaborada para fins deste estudo

Nota: Dados sobre acurácia dos sistemas no âmbito do projeto CoNLL-SIGMORPHON extraídos de (COTTERELL et al., 2017; COTTERELL et al., 2018).

A Tabela 1 mostra, em primeiro lugar, os resultados mais altos de acurácia alcançados pelos sistemas no CoNLL-SIGMORPHON. Da esquerda para a direita na tabela, verifica-se que, em 2017, o sistema do time *LMU-2*[1] obteve os melhores resultados, alcançando uma acurácia de 99,30% (com acesso a 10000 *tokens* de exemplo para treinamento – *high*), e 95,00% (com acesso a 1000 *tokens* de exemplo para treinamento (*medium*); na condição *low* (acesso a 100 tokens para treinamento), o sistema que obteve o melhor resultado de acurácia foi o time *CLUZH-7*[2], com 73,30% de acurácia. Na edição de 2018

[1] *Ludwig-Maximilian University of Munich-Sistema 2*
[2] University of Zurich

do CoNLL-SIGMORPHON, o time *uzh-2*[3] obteve o resultado de acurácia mais alto nas três condições de treinamento: 98,60% (*high*), 94,80% (*medium*), 75,80% (*low resource*).

Por sua vez, o realizador baseado em regras obteve uma acurácia de 98.4% no corpus de desenvolvimento (*corpus-dev*) do CoNLL-SIGMORPHON–2017. Considerando-se, então, os melhores resultados dos modelos de ambas as edições do CoNLL-SIGMORPHON, o realizador baseado em regras alcançou resultados competitivos já no experimento aplicado no corpus de desenvolvimento, ficando apenas 0,2 pontos percentuais abaixo do segundo melhor resultado. Esse resultado foi obtido na última iteração da função para os parâmetros apresentados pelo *corpus*. Como descrito na subseção 3.2.1.1.2, a cada iteração do experimento de desenvolvimento, foram realizadas mudanças nas funções de flexão verbal, visando a obter melhores resultados de acurácia. Após diversas iterações, as mudanças necessárias para uma melhoria significativa do realizador envolviam a modelagem particular de regras para cada lema, por se tratarem de verbos irregulares [4]. Como já apontado na subseção 3.2.1.1.2, foram mitigados alguns erros do realizador baseado em regras que resultam da irregularidade dos paradigmas de flexão verbal no português, mas não foi possível contemplar todo o potencial de verbos irregulares. Após a última iteração de melhorias nas funções de flexão verbal do realizador baseado em regras, o experimento dev retornou a seguinte lista de verbos com flexões que diferiam do padrão ouro, dados os parâmetros para a flexão (ver Figura 46).

Figura 46 – Erros experimento de desenvolvimento

Lema, padrão_ouro, parâmetros:
inerir iniras V;2;SG;SBJV;PRS / preferir prefiram V;3;PL;IMP;POS
genufletir genuflita;V;3;SG;IMP;POS / recair recaiamos V;1;PL;IMP;NEG
boxear boxeiem V;3;PL;SBJV;PRS / arrecear;arreceiam;V;3;PL;IND;PRS
reler relede V;2;PL;IMP;POS / decair decaídos;V.PTCP;MASC;PL;PST
perseguir persigam V;3;PL;IMP;POS / caber coubéreis V;2;PL;IND;PST;PRF
extrair extraía V;1;SG;IND;PST;IPFV / presentear presenteia V;2;SG;IMP;POS
presentear presenteia V;2;SG;IMP;POS / injungir injunja V;3;SG;IMP;POS
tanger tanjam V;3;PL;IMP;NEG / encobrir encubrais V;2;PL;IMP;NEG

⇓

Lista de erros, saída do realizador baseado em regras dados os parâmetros
[ineras, preferam, genufleta, recaamos, boxeem, arreceam, relei, decaidos,
perseguam, cabêreis, extraia, presentea, injunga, tangam, encobrais]

Fonte: elaborada para fins deste estudo

Como verifica-se na Figura 46, os verbos que constam na lista de erros da saída do algoritmo utilizado no experimento dev tratam-se necessariamente de verbos irregulares (por exemplo, inerir, preferir, perseguir, injungir, tanger, encobrir, caber). A modelagem de funções para a flexão dos verbos irregulares, como mencionado, demanda a modelagem

[3] University of Zurich
[4] Reitera-se que adotou-se irregularidade como necessidade de processamento no radical do verbo para realização da flexão.

particular de funções responsáveis pela flexão de cada lema, ou no máximo, a modelagem de conjuntos de verbos irregulares com características de flexão análogas. Na etapa de aprimoramento do realizador baseado em regras, foram implementadas funções específicas para alguns verbos irregulares. Contudo, essa modelagem para verbos específicos restringiu-se a verbos mais frequente/comuns, especialmente verbos se seriam produtivos na linha de produção do robô-jornalista. Note-se que a lista de erros da Figura 46 retrata, em sua maioria, verbos pouco frequentes, raros. Nesses casos, o realizador baseado em regras demandará uma constante reciclagem com a implementação de funções que realizem especificamente cada um dos verbos irregulares, de acordo com a demanda, em seu processo evolutivo.

Na subseção 4.1.2 serão apresentados os resultados do experimento dev de aplicação.

4.1.2 Resultados de aplicação – experimento de desenvolvimento

Nesta seção são apresentados os resultados do experimento de desenvolvimento - aplicação. Em primeiro lugar, retoma-se o delineamento do experimento de desenvolvimento. São, então, apresentados exemplos de realização para cada um das etapas do experimento dev. Por fim, apresenta-se uma discussão dos resultados do experimento dev de aplicação.

Como antecipado na subseção 3.2.1.2, o experimento dev de aplicação consiste em 2 subexperimentos: realização de todo o paradigma de flexão dos verbos que compõem o conjunto de verbos do léxico do sistema de geração do robô-jornalista @DaMataReporter; e a realização textual de orações extraídas do robô-jornalista, sendo utilizada a função do realizador superficial baseado em regras para a flexão dos verbos do banco lexical do robô e potenciais verbos similares, extraídos de um modelo de linguagem, por meio de vetores de palavras, do NILC.

No primeiro subexperimento, foi elaborado um algoritmo para a realização do paradigma completo de flexão dos verbos do banco lexical do @DaMataReporter, como delineado na subseção 3.2.1.2. Esse algoritmo tem com saída arquivos *.json* com dicionários de dicionários com listas contendo as respectivas flexões do paradigma no formato '"chave"(lema): {"chave"(Orientação Interpessoal): [valores(verbos flexionados)]'}. A Figura 47 apresenta o paradigma completo de flexão de dois verbos – 'desmatar' e 'identificar'.

Como pode-se verificar na Figura 47, são contemplados todos termos do sistema de ORIENTAÇÃO INTERPESSOAL, de acordo com a descrição do verbo no português brasileiro (ver Sá (2016)), ou seja, Presente, Pretérito_perfectivo, Subjuntivo_optativo, Gerúndio etc., demonstrando o potencial do sistema para o aumento de variabilidade do discurso entregue pelo sistema de geração no contexto do robô-jornalismo. Note-se que os exemplos tratam-se de um verbo regular e um irregular forte. No caso do verbo regular ('desmatar'), o sistema retorna, sem qualquer problema, o paradigma completo de flexão, pois a função geral de flexão, sem alterações, garante a acurácia. No caso do verbo irregular forte

('identificar'), uma subfunção específica para verbos terminados em '-**car**' foi elaborada para que fossem realizadas as adequações necessárias tanto no radical quanto no sufixo de flexão. Esse é um exemplo de casos que permitem a abstração de características comuns em um conjunto de verbos irregulares, permitindo a implementação de funções que não sejam restritas a apenas um lema. Esse tipo de implementação é impossível em casos de verbos anômalos, como o verbo 'ser', que necessita de uma função particular, e que não permite extrapolar características de flexão para outros verbos. Outros casos que permitem a abstração de características para um subgrupo dos verbos irregulares são os verbos terminados em '-dizer, -cer, -çar, -gar, -fazer, -ruir'.

Figura 47 – Exemplos paradigma de flexão

"**desmatar**":

{"***presente***": ["desmato","desmatas","desmata","desmatamos","desmatais","desmatam"],

"***pretérito__perfectivo__I***": ["desmatei", "desmataste","desmatou","desmatamos","desmatastes", "desmataram"],

"***pretérito__perfectivo__II***": ["desmatara", "desmataras", "desmatara", "desmatáramos","desmatáreis", "desmataram"],

"***pretérito__imperfectivo***": ["desmatava", "desmatavas", "desmatava", "desmatávamos", "desmatáveis", "desmatavam"],

"***passado__volitivo***": [desmataria", "desmatarias", "desmataria", "desmataríamos", "desmataríeis", "desmatariam"],

"***futuro***": ["desmatarei", "desmatarás", "desmatará", "desmataremos", "desmatareis", "desmatarão"],

"***subjuntivo__conjuntivo***": ["desmate", "desmates", "desmate", "desmatemos", "desmateis", "desmatem"],

"***subjuntivo__condicional***": ["desmatasse", "desmatasses", "desmatasse", "desmatássemos", "desmatásseis", "desmatassem"],

"***subjuntivo__optativo***": ["desmatar", "desmatares", "desmatar", "desmatarmos", "desmatardes", "desmatarem"],

"***não__finito__concretizado***": ["desmatar", "desmatares", "desmatar", "desmatarmos", "desmatardes", "desmatarem"],

"***imperativo__I***": [, "desmata", "desmate", "desmatemos", "desmatai", "desmatem"],

"***imperativo__II***": [, "desmates", "desmate", "desmatemos", "desmateis", "desmatem"],

"***gerúndio***": ["desmatando"]

"***particípio***": ["desmatado", "desmatada", "desmatados", "desmatadas"]},

"**identificar**":

{"***presente***": ["identifico", "identificas", "identifica", "identificamos", "identificais", "identificam"],

"***pretérito__perfectivo__I***": ["identifiquei", "identificaste", "identificou", "identificamos", "identificastes", "identificaram"],

"***pretérito__perfectivo__II***": ["identificara", "identificaras", "identificara", "identificáramos", "identificáreis", "identificaram"],

"***pretérito__imperfectivo***": ["identificava", "identificavas", "identificava", "identificávamos", "identificáveis", "identificavam"],

"***passado__volitivo***": ["identificaria", "identificarias", "identificaria", "identificaríamos", "identificaríeis", "identificariam"],

"***futuro***": ["identificarei", "identificarás", "identificará", "identificaremos", "identificareis", "identificarão"],

"***subjuntivo__conjuntivo***": ["identifique", "identifiques", "identifique", "identifiquemos", "identifiqueis", "identifiquem"],

"***subjuntivo__condicional***": ["identificasse", "identificasses", "identificasse", "identificássemos", "identificásseis", "identificassem"],

"***subjuntivo__optativo***": ["identificar", "identificares", "identificar", "identificarmos", "identificardes", "identificarem"],

"***não__finito__concretizado***": ["identificar", "identificares", "identificar", "identificarmos", "identificardes", "identificarem"],

"***imperativo__I***": [, "identifica", "identifique", "identifiquemos", "identificai", "identifiquem"],

"***imperativo__II***": [, "identifiques", "identifique", "identifiquemos", "identifiqueis", "identifiquem"],

"***gerúndio***": ["identificando"]

"***particípio***": ["identificado", "identificada", "identificados", "identificadas"]}

Fonte: Elaborado para fins deste estudo

No segundo subexperimento, foi extraído um dicionário contendo 10 verbos similares aos 26 verbos do banco lexical do robô-jornalista. Os aproximadamente 230 verbos similares foram extraídos do vocabulário dos vetores de palavras e foram salvos em *.json*, no formato chave (lema principal):[valores(verbos similares)]. A Figura 48 apresenta três entradas do dicionário, respectivamente as palavras similares a 'registrar, totalizar, reportar'.

Figura 48 – Exemplos dicionário de verbos similares

{
"***registrar***": ["conter", "apontar", "configurar", "encaminhar", "usar", "receber", "ofertar", "descartar", "liberar", "trazer"],

---

"***totalizar***": ["embolsar", "contabilizar", "atingir", "estipular", "acrescentar", "valorizar", "elevar", "reduzir", "atingir", "cobrar"],

---

"***reportar***": ["expor", "associar", "submeter", "entregar", "confirmar", "apresentar", "encaminhar", "revelar", "apontar", "justificar"]}

Fonte: Elaborado para fins deste estudo

Note-se que os verbos são considerados similares dentro da lógica de modelagem do modelo de vetores de palavras. Dessa forma, nem todos os elementos do dicionário são produtivos no contexto realizado pelas orações do robô-jornalista. Em outras palavras, é necessária uma filtragem para seleção das palavras similares que efetivamente tenham potencial de aplicação na linha de produção do robô-jornalista. Por exemplo, na oração 'o Instituto Nacional de Pesquisas Espaciais (INPE) **registrou** alertas de desmatamento de 5.91 km2 (...)', os verbos similares que seriam considerados produtivos no contexto poderiam ser: 'apontar, receber, encaminhar, liberar, trazer'. Nesse sentido, utilizar quatro palavras similares aleatoriamente na linha de produção aumenta consideravelmente o potencial de variabilidade na geração de textos do robô.

Esse potencial de variabilidade verbal pode ser aplicado diretamente no robô-jornalista, através da anotação manual, na etapa de anotação dos corpora para treinamento dos *templates*, dos parâmetros pertinentes (e em certa medida cristalizados quanto às informações de ancoragem no tempo, dadas as necessidades de um dado contexto local) para uma dada flexão verbal nas *tags*, e a randomização dos valores que serão disponibilizados como entrada para o parâmetro 'lemma', com base nas listas de palavras similares previamente filtradas como pertinentes para o contexto. Como mostrado anteriormente, foram selecionados aproximadamente 10 verbos similares para cada verbo dentro do banco léxico do robô-jornalista (algumas entradas não retornaram 10 verbos similares do modelo de vetores de palavras), e nem todos podem ser considerados produtivos no contexto pertinente, isto é, apesar de serem sinonímias aproximadas, não poderiam ser consideradas verbos similares naquele contexto. Esse mesmo critério pode ser empregado diretamente na linha de produção, ou seja, pode-se expandir o número de verbos que são passíveis de variação, para que sejam randomizados verbos similares e para serem passados como lemas para as funções de flexão. Note-se que nesses casos, os verbos não necessariamente sofreriam variações de número e pessoa, por exemplo, por não estarem em contextos que sofrem mudanças de dados do 'conteúdo selecionado' para geração. Em outras palavras, é

possível aumentar a variabilidade lexical mesmo dos verbos que não estão em contextos 'críticos' para a a linha de produção da geração, ou seja, que não são afetados em número e pessoa por mudança nos dados numéricos, por exemplo. Para cada uma das 11 orações extraídas do @DaMataReporter, foram geradas orações com a flexão dos verbos similares, resultando em 91 orações [5]. A Figura 49 apresenta uma oração, cujo Processo é realizado pelo verbo 'registrar', e as 10 sentenças com as respectivas palavras similares extraídas como verbos similares do modelo de vetores de palavras como alternativas para sua realização. A sentenças cujos verbos são considerados 'produtivos' com relação ao potencial de sinonímia no dado contexto estão em negrito.

[5] Ver lista completa de orações com verbos similares no Apêndice C

Figura 49 – Verbos similares em sentenças do @DaMataReporter

Sentença principal:

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) **registrou** alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC."

Sentenças com verbos similares:

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *conteve* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *apontou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *configurou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *encaminhou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

Ëm 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *usou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *recebeu* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *ofertou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *descartou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *liberou* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC.",

---

"Em 25 de março de 2020, o Instituto Nacional de Pesquisas Espaciais(INPE) *trouxe* alertas de desmatamento de 5.91 km2 na RESERVA EXTRATIVISTA CHICO MENDES/AC."

Fonte: elaborada para fins deste estudo

As sentenças dispostas na Figura 49 mostram como as funções responsáveis pela flexão verbal funcionam em um cenário que já se aproxima do cenário de produção do robô-jornalista. Dados os mesmos parâmetros para a flexão, tanto para o verbo que realiza o Processo na sentença principal, quanto para os verbos similares, as sentenças retornam uma saída satisfatória em todos os casos. Uma análise mais detida sobre sua natureza quanto ao tipo mostra que, em sua grande maioria, nesse contexto específico, tratam-se de verbos regulares. Tem-se, contando com o verbo 'registrar', que realiza o Processo na sentença principal, oito verbos regulares: "apontar", "configurar", "encaminhar", "usar", "receber", "ofertar", "descartar", "liberar". Como já reiterado, nos casos de verbos regulares

a função de flexão verbal se comporta com baixas chances de erro. Em contrapartida, fazem parte da lista de verbos similares, dois verbos irregulares: "conter"e "trazer". No caso desses dois exemplos de verbos irregulares, o realizador baseado em regras se comporta satisfatoriamente, pois foram desenvolvidas funções específicas para o verbo 'trazer', e para o verbo 'ter', do qual deriva-se o verbo 'conter'.

O experimento de desenvolvimento - aplicação - visava a verificar como as funções se comportariam em sentenças já extraídas do robô-jornalista, tentando antecipar como se comportariam na eventual aplicação na instância local deste. Os resultados efetivos de aplicação na linha de produção de uma instância local do @DaMataReporter são apresentados na subseção 4.2.2.

A seção 4.2, a seguir, apresenta os resultados para os experimentos de acurácia, que permite o teste de acurácia efetivo do realizador baseado em regras no corpus de teste extraído do CoNLL-SIGMORPHON–2017; e de aplicação, que permite o teste das funções de flexão verbal em uma instância local do robô-jornalista @DaMataReporter'.

4.2 RESULTADOS DO EXPERIMENTO DE VALIDAÇÃO

4.2.1 Resultados de acurácia – experimento de validação

Seguindo o mesmo desenho já delineado na subseção 3.2.1.1, cujos resultados foram apresentados na subseção 4.1.1, nesta seção são apresentados os resultados de acurácia do realizador baseado em regras no *corpus* de teste do CoNLL-SIGMORPHON 2017. Os resultados são, reitera-se, apresentados em contraste com os melhores resultados para o português obtidos pelos sistemas no âmbito do projeto do ConNLL-SIGMORPHON, nas edições de 2017 e 2018. O resultado do experimento de validação realizado no *corpus* de teste também é comparado com o resultado obtido no experimento dev, no *corpus* de desenvolvimento, descrito na subseção 3.2.1.1. São discutidos os resultados de acurácia e os tipos de erros apresentados pelo realizador baseado em regras em sua aplicação no *corpus* de teste.

Como apresentado na subseção 4.1.1, a acurácia do modelo baseado em regras, seguindo o desenho do projeto CoNLL-SIGMORPHON, foi computada através da comparação da porcentagem de acerto, em contraste com os paradigmas esperados (padrão ouro) no *corpus* de desenvolvimento. O teste de acurácia no *corpus* de teste seguiu o mesmo desenho. O experimento de desenvolvimento - acurácia (no *corpus dev*) tinha o objetivo de testar como o sistema se comportava, buscando a exposição de erros e falhas, e ensejando o aprimoramento (adequação das funções, modelagem de funções específicas para verbos irregulares ainda não contemplados) do modelo baseado em regras a cada iteração. Por outro lado, o experimento de validação no *corpus* de teste foi aplicado apenas uma vez, após um número satisfatório de iterações e melhorias do modelo baseado em regras na

fase de desenvolvimento.

A Tabela 2 apresenta, mais uma vez, os resultados dos sistemas desenvolvidos no âmbito do CoNLL-SIGMORPHON, para os três moldes de acesso aos recursos para treinamento (baixo, médio e alto) em contraste com os resultados para ambos os testes de acurácia aplicados no realizador baseado em regras desenvolvido nesta tese.

Tabela 2 – Resultados de acurácia – Experimento de validação

	high	*medium*	*low*
SIGMORPHON 2017	99,30%(LMU-2)	95,00%(LMU-2)	73,30%(CLUZH-7)
SIGMORPHON 2018	98,60%(uzh-2)	94,80%(uzh-2)	75,80%(uzh-2)
Realizador baseado em regras (Pré-exp.)		98,4%	
Realizador baseado em regras (Exp.)		99,0%	

Fonte: elaborada para fins deste estudo

Nota: Dados sobre acurácia dos sistemas no âmbito do projeto CoNLL-SIGMORPHON extraídos de Cotterell et al. (2017), Cotterell et al. (2018).

A Tabela 2 reitera os resultados mais bem colocados no âmbito do projeto CoNLL-SIGMORPHON, edições de 2017 e 2018, e a acurácia alcançada pelo modelo baseado em regras no experimento dev realizado no *corpus* de desenvolvimento, paralelamente ao resultado do exame de acurácia aplicado no *corpus* de teste, permitindo assim o contraste entre eles. Como pode-se verificar na Tabela 2, no experimento de validação realizado diretamente no *corpus* de teste, o modelo baseado em regras alcançou um *score* de 99,0%, ultrapassando em 0,6% o resultado obtido no *corpus* de desenvolvimento e em 0,4% o melhor resultado obtido na edição de 2018 do projeto do CoNLL. Além disso, fica atrás do melhor resultado entre as edições por apenas 0,3%.

Esses resultados demonstram que o modelo baseado em regras desenvolvido nesta tese se mostrou competitivo em contraste com os modelos de redes neurais desenvolvidos no âmbito do CoNLL-SIGMORPHON - 2017 e 2018, na medida em que retornou resultados consistentes, aproximando-se da acurácia apontada para os melhores resultados em ambas as edições do projeto.

Figura 50 – Erros experimento de validação

Lema, padrão_ouro, parâmetros:
idear ideiem V;3;PL;IMP;POS/ marquetear marqueteiem V;3;PL;IMP;POS
atribuir atribuístes V;2;PL;IND;PST;PFV / progredir progridem V;3;PL;IND;PRS
manusear manuseia V;3;SG;IND;PRS / reger rejas V;2;SG;SBJV;PRS
atingir atinjam V;3;PL;IMP;POS / inerir inira V;3;SG;SBJV;PRS
influir influísse V;1;SG;SBJV;PST;IPFV

⇓

Lista de erros, saída do realizador baseado em regras dados os parâmetros:

["ideem", "marqueteem", "atribuistes", "progredem", "manusea","regas","atingam", "inera", "infera", "influisse"]

Fonte: elaborada para fins deste estudo

Como a Figura 50 apresenta, e seguindo o mesmo padrão encontrado no resultado da última iteração das funções de flexão verbal no *corpus* de desenvolvimento, os erros que resultam do experimento no *corpus* de teste se restringem a verbos irregulares, mais especificamente. Como discutido sobre os tipos de irregulares e as necessidades na modelagem das funções, em alguns casos não é possível abstrair padrões (ou os padrões são muito restritos, chegando a ser restrito a um único lema.

Alguns casos interessantes de irregularidade podem ser pinçados da lista de erros para que sejam discutidos. Os lemas "atribuir"e "influir", por exemplo, tratam-se de verbos em terminação "-IR". O padrão apresentado pela descrição do português de base sistêmico-funcional (ver Sá (2016)) para o Pretérito_perfectivo_I - 2 pessoa - plural é o sufixo 'iste', e para Subjuntivo_condicional - 1 pessoa-singular é o sufixo 'isse'. Contudo, em alguns casos, como é o caso dos verbos "atribuir"e "influir", é necessário a inserção do acento gráfico agudo, mudando os padrões para 'íste' e 'ísse' nessas seleções de Orientação interpessoal. Nestes casos, uma maneira de conseguir generalizar seria realizar o parseamento dos morfemas do radical, buscando por padrões que possam determinar os padrões de acentuação a serem modelados, buscando generalizações dentro dos verbos irregulares.

Na subseção 4.2.2 serão apresentados os resultados do experimento de validação - aplicação.

4.2.2 Resultados de aplicação – experimento de validação

Nesta seção são apresentados os resultados do experimento de validação de aplicação das funções de flexão verbal na linha de produção da instância local do robô-jornalista @DaMataReporter. Em primeiro lugar, o delineamento do experimento de validação é reiterado. São, então, apresentados exemplos de anotação e as sentenças resultantes correspondentes, com a realização textual dos verbos através das funções de flexão verbal do realizador baseado em regras, dados os parâmetros anotados. São apresentadas sentenças correspondentes à execução do sistema de geração, instância local do robô-jornalista, para os dados do período convenientemente selecionado (setembro de 2020 a março de 2021) para a geração das postagens mensais e para o período de 30 dias anteriores à execução do experimento de validação. Por fim, são discutidos os resultados de aplicação das funções do realizador baseado em regras na linha de produção da instância local do @DaMataReporter.

Como apresentado na subseção 3.2.2.2, o experimento de validação - aplicação, para teste de viabilidade do uso das funções de flexão na linha de produção do robô-

jornalista, é executado em uma instancia local do robô-jornalista. Essa instância local do sistema foi implementada com todas as funções da linha de produção efetiva do @DaMataReporter, excetuando-se a função de postagem nas redes sociais. Os dados utilizados para o treinamento fazem parte dos *corpora* anotados para a linha de produção efetiva do robô: foram selecionadas 5 entradas para cada um dos *corpora*, o de postagens diárias e mensais, que foram anotadas seguindo o padrão necessário para a passagem de parâmetros adequada às funções de flexão verbal do realizador baseado em regras. Essas entradas foram a fonte de treinamento do modelo e extração dos *templates* para a geração. Dado o treinamento e a extração dos *templates*, as etapas da linha de produção (seleção de conteúdo, ordenação etc.) são executadas pela função de geração do sistema. As saídas do sistema de geração foram salvas em formato .txt.

A Figura 51 apresenta a anotação de duas sentenças no padrão adequado à passagem dos parâmetros para as funções de flexão verbal do realizador baseado em regras e a realização textual correspondente a cada uma delas. Note-se que a figura não contempla, mas subentende os dados das outras *tags* correspondentes às outras etapas da linha de produção.

Figura 51 – Exemplo de anotação do *corpus* e realização textual de sentença

```
<entry date="2020-9-01">
(...)

        <lexicalization>
            <paragraph>
                <sentence>
                    No mês de MONTH_1 de YEAR_1 , VP[experience=Ser,
                    functionInGroup=Auxiliar,lemma=ser,person=3,gender=none
                    number=Plur,mood=Ind,tense=Past,aspect=Perf] VP[experience
                    =Fazer,functionInGroup=Evento,lemma=desmatar,person=none,
                    gender=Masc,number=Plur,mood=none,tense=Past,aspect=Perf]
                    AREA_1 de LOCATION_1 , de acordo com o monitoramento de
                    INSTITUTE_1 .
                </sentence>
           <sentence>
                    Com AREA_1 , a principal causa de devastação VP[experience=Ser,
                    functionInGroup=Evento,lemma=ser,person=3,gender=none,
                    number=Sing,mood=Ind,tense=Past,aspect=Perf] CAUSE_1 .
                </sentence>
           </paragraph>
        </lexicalization>
```

```
        <text>
            <paragraph>
                <sentence>
                    No mês de setembro de 2020, foram desmatados  965 km²
                    da Amazônia Legal, de acordo com o monitoramento
```

```
                    do Instituto Nacional de Pesquisas Espaciais (INPE).
                </sentence>
                 <sentence>
                    Com 954 km², a principal causa de devastação foi o
                    desmatamento com solo exposto, que deixa a terra sem
                    nenhuma cobertura florestal.
                 </sentence>
            </paragraph>
(...)
        </text>
</entry>
```

Fonte: *corpus* do @DaMataReporter (grifo meu).

A localização das anotações em padrão 'VP[<>]' correspondem à localização dos verbos na realização textual das sentenças. Na primeira sentença, as funções de flexão verbal foram responsáveis pela flexão de dois verbos: na posição de Auxiliar, realizado pelo verbo auxiliar 'ser'; e na posição de Evento, realizado pelo verbo lexical 'desmatar', ambos realizando do grupo verbal na voz passiva. Por sua vez, na segunda sentença a função de flexão verbal foi responsável pela flexão do verbo 'ser', encerrando a função de Evento (verbo lexical). Dentre os verbos realizados superficialmente nas sentenças, temos o verbo irregular-anômalo 'ser', nas funções de Evento e Auxiliar no grupo verbal, e o verbo regular 'desmatar' na função de Evento. Esses lemas estão completamente contemplados pelas funções de flexão do realizador, isto é, todas as opções do sistema de ORIENTAÇÃO INTERPESSOAL, são cobertas. No caso do verbo anômalo 'ser', por ser um verbo comum muito frequente no português, tanto na função de Evento quanto na função de Auxiliar em grupos verbais em voz passiva, demandaram uma modelagem particular. As flexões do verbo regular 'desmatar', por sua vez, são completamente contempladas pelas funções genéricas do realizador baseado em regras. As orações que realizam o contexto de situação em ambas as sentenças são ancoradas no Pretérito_perfectivo_I, e têm a necessidade de flexão para esses parâmetros completamente satisfeita pela realização textual dos verbos flexionados pelo realizador baseado em regras.

A Figura 52 dispõe as sentenças resultantes da execução do sistema de geração na instância local do robô-jornalista, dada a seleção conveniente do período de setembro de 2020 a março de 2021 (7 meses), dados os *templates* resultantes da anotação de 5 entradas para treinamento para dados mensais. Com base nesses parâmetros, esperava-se que fossem geradas sete potenciais postagens, isto é, parágrafos compostos por um número variável de sentenças (4 ou 5, a depender da escolha dos *templates* dada a seleção de conteúdo). Os verbos que foram flexionados pelas funções de flexão verbal do realizador baseado em regras estão em negrito.

Figura 52 – Sentenças – saída experimento de validação – dados mensais

No mês de setembro de 2020, **foram desmatados** 965 km² da Amazônia Legal, de acordo com o monitoramento do Instituto Nacional de Pesquisas Espaciais (INPE). Com 954 km², a principal causa de devastação **foi** o desmatamento com solo exposto, que deixa a terra sem nenhuma cobertura florestal. O Pará **foi** o estado que mais **teve** desmatamento (426 km²). O município que **teve** mais desmatamento no mês **foi** São Félix do Xingu (Pará), com 70 km² de área desmatada. Com 14 km², a FLORESTA NACIONAL DO JAMANXIM / Pará **foi** a Unidade de Conservação mais devastada no mês.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), **foram desmatados** 836 km² da Amazônia legal, em outubro de 2020. O principal tipo **foi** o desmatamento com solo exposto, deixando a terra sem vegetação, com 833 km² de área **desmatada**. O estado com mais desmatamento no mês **foi** Pará (398 km²), enquanto o município mais **devastado foi** Porto Velho / Rondônia, com 53 km² de área **desmatada**. Com 9 km², a RESERVA EXTRATIVISTA CHICO MENDES / Acre **foi** a Unidade de Conservação mais **devastada** no mês.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), **foram desmatados** 310 km² da Amazônia Legal, em novembro de 2020. A principal causa **foi** o desmatamento com solo exposto, que significa a retirada de toda a cobertura da floresta, aquele que deixa a terra sem nenhuma vegetação, **somando** 306 km². O estado com mais desmatamento no mês **foi** Pará (121 km²), enquanto o município mais **devastado foi** Pacaja / Pará, com 15 km² de área **desmatada**. Com 1 km², a FLORESTA NACIONAL DE ALTAMIRA / Pará **foi** a Unidade de Conservação mais **devastada** no mês.

No mês de dezembro de 2020, **foram desmatados** 216 km² da Amazônia Legal, de acordo com o monitoramento do Instituto Nacional de Pesquisas Espaciais (INPE). Com um total de 213 km², a principal causa de destruição da Amazônia Legal no mês **foi** o desmatamento com solo exposto, aquele que deixa a terra sem nenhuma vegetação. O estado com mais desmatamento no mês **foi** Pará (100 km²), enquanto o município mais **devastado foi** São Félix do Xingu / Pará, com 16 km² de área desmatada. A Unidade de Conservação onde **teve** mais desmatamento **foi** a ÁREA DE PROTEÇÃO AMBIENTAL DO TAPAJÓS / Pará, com 1 km² de área **desmatada**.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), **foram desmatados** 86 km² da Amazônia Legal, em janeiro de 2021. O principal tipo **foi** o desmatamento com solo exposto, deixando a terra sem vegetação, com 85 km² de área **desmatada**. Mato Grosso foi o estado que mais **teve** desmatamento (45 km²). O município que **teve** mais desmatamento no mês **foi** Altamira (Pará), com 8 km² de área **desmatada**. Com 0 km², a RESERVA EXTRATIVISTA CHICO MENDES / Acre **foi** a Unidade de Conservação mais **devastada** no mês.

O Instituto Nacional de Pesquisas Espaciais (INPE) **identificou** um aumento de 43% de área desmatada da Amazônia Legal no último mês, **atingindo** 123 km² devastados em fevereiro de 2021. O principal tipo foi o desmatamento com solo exposto, que significa a retirada de toda a cobertura da floresta, com 121 km² de área **desmatada**. O município mais **desmatado foi** Apui / Amazonas (9 km²), enquanto o estado mais **afetado** no geral **foi** Pará (36 km²). A Unidade de Conservação onde **teve** mais desmatamento **foi** a FLORESTA NACIONAL DE ITAITUBA II / Pará, com 1 km² de área **desmatada**.

No mês de março de 2021, **foram desmatados** 21 km² da Amazônia legal, de acordo com o monitoramento do Instituto Nacional de Pesquisas Espaciais (INPE). Com 21 km², a principal causa de devastação **foi** o desmatamento com solo exposto, que significa a retirada de toda a cobertura da floresta. Pará **foi** o estado que mais **teve** desmatamento (14 km²). O município que **teve** mais desmatamento no mês **foi** Dom Eliseu (Pará), com 5 km² de área **desmatada**. A Unidade de Conservação que **foi** mais **devastada foi** a FLORESTA NACIONAL DO IQUIRI / Amazonas (0 km²).

Fonte: elaborada para fins deste estudo

Como mostra a Figura 52, os verbos em negrito, que são realizados superficialmente pelas funções de flexão do realizador baseado em regras, apresentaram flexão adequada aos parâmetros demandados pelo contexto. Em sua grande maioria, os lemas verbais que

foram objeto da flexão verbal são verbos regulares (e.g., 'desmatar, devastar, atingir, afetar, somar'). Contudo, alguns irregulares também foram contemplados (e.g., 'ser, identificar'). Em ambos os casos, a flexão foi satisfatória, reiterando que verbos irregulares necessitaram de modelagem particular na etapa de desenvolvimento e aprimoramento do sistema.

Como mencionado anteriormente, foram anotadas 5 entradas (*<entry>*) para treinamento do modelo e extração dos *templates*, e a geração contemplou o período de setembro de 2020 a março 2021 para o *corpus* mensal. Esse período foi selecionado convenientemente, somente para que houvesse dados diferentes, na fase de seleção de conteúdo, para a geração de algumas sentenças, dadas seleções aleatórias dos *templates*, visando ao teste das funções de flexão verbal para alguns verbos no *corpus*, ou seja, um período maior poderia ter sido selecionado. Contudo, a única mudança observada seria nos dados selecionados na fase de seleção de conteúdo para a geração. Em outras palavras, o sistema realizaria mais iterações, selecionando *templates* aleatoriamente, realizando superficialmente as mesmas sentenças (com dados diferentes), em combinações diferentes, mas análogas às apresentadas na Figura 52. Isso justifica a seleção conveniente do período de 6 meses para a execução do experimento de validação.

As mesmas etapas foram aplicadas para o experimento de validação com as postagens diárias. O *corpus* é composto por 5 entradas anotadas no padrão adequado à passagem dos parâmetros para as funções de flexão do realizador baseado em regras; os *templates* foram treinados; foi selecionado um período de 30 dias anteriores à data de execução do experimento de validação para a etapa de seleção de conteúdo e geração das mensagens. Após a execução das funções de geração para esses parâmetros, um conjunto de sentenças foi satisfatoriamente gerado, como mostram os exemplo na Figura 53. Os verbos que são realizados superficialmente pelas funções do realizador baseado em regras estão em negrito.

Figura 53 – Sentenças – saída experimento de validação – dados diários

Em 4 de março de 2021, o sistema de monitoramento do Instituto Nacional de Pesquisas Espaciais (INPE) **alertou** o desmatamento de 5,46 km² na cidade de Dom Eliseu / Pará. O alerta diário de desmatamento **gerado** pelo Instituto **tem** como principal motivo o desmatamento com solo exposto, deixando a terra sem vegetação.

Em 4 de março de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **rastreou** alertas de desmatamento em São Félix do Xingu / Pará, **somando** 3,96 km². A principal causa para o desmatamento na região **foi** o desmatamento com solo exposto, que deixa a terra sem vegetação.

Em 24 de fevereiro de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **reportou** avisos de desmatamento **somando** 1,91 km² em Colniza / Mato Grosso, que **acumulou** 2 dias de alertas. Colniza **acumula** 3,73 km² em fevereiro. O desmatamento com solo exposto, deixando a terra sem vegetação, **foi** a principal causa de desmatamento.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), Caroebe / Roraima **teve** alertas de desmatamento no dia 24 de fevereiro de 2021 que **somaram** 1,41 km². Caroebe **soma** 1,56 km² de área **desmatada** em fevereiro. O desmatamento com solo exposto, que deixa a terra sem vegetação, **foi** a principal causa de desmatamento.

Em 24 de fevereiro de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **reportou** avisos de desmatamento **somando** 1,33 km² em Novo Aripuana / Amazonas, que **acumulou** 3 dias de alertas. Novo Aripuana **soma** 6,13 km² de área desmatada em fevereiro. A principal causa para o desmatamento na região **foi** o desmatamento com solo exposto, deixando a terra sem vegetação.

Em 25 de fevereiro de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **reportou** avisos de desmatamento **somando** 1,19 km² em Comodoro / Mato Grosso, que **acumulou** 3 dias de alertas. Comodoro **acumula** 2 km² em fevereiro. O alerta diário de desmatamento **gerado** pelo Instituto **tem** como principal motivo o desmatamento com solo exposto, que deixa a terra sem vegetação.

Em 4 de março de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **rastreou** alertas de desmatamento em Paragominas / Pará, somando 1,18 km². O desmatamento com solo exposto, que deixa a terra sem vegetação, **foi** a principal causa de desmatamento.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), Apui / Amazonas **teve** alertas de desmatamento no dia 24 de fevereiro de 2021 que **somaram** 1,15 km². Apui **soma** 9,02 km² de área **desmatada** em fevereiro. O desmatamento com solo exposto, deixando a terra sem vegetação, **foi** a principal causa de desmatamento.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), Caracarai / Roraima **teve** alertas de desmatamento no dia 24 de fevereiro de 2021 que **somaram** 1,05 km². Caracarai soma 4,67 km² de área **desmatada** em fevereiro. O alerta diário de desmatamento gerado pelo Instituto **tem** como principal motivo o desmatamento com solo exposto, deixando a terra sem vegetação.

Em 24 de fevereiro de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **reportou** avisos de desmatamento **somando** 0,88 km² em Sao Joao da Baliza / Roraima, que **acumulou** 2 dias de alertas. Sao Joao da Baliza **acumula** 1,30 km² em fevereiro. A principal causa para o desmatamento na região **foi** o desmatamento com solo exposto, que deixa a terra sem vegetação.

Em 4 de março de 2021, o Instituto Nacional de Pesquisas Espaciais (INPE) **rastreou** alertas de desmatamento em Tome-Açu / Pará, **somando** 0,83 km². A principal causa de desmatamento **foi** o desmatamento com solo exposto, que deixa a terra sem vegetação.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), Rorainópolis / Roraima **teve** alertas de desmatamento no dia 24 de fevereiro de 2021 que **somaram** 0,75 km². Rorainópolis soma 7,74 km² de área **desmatada** em fevereiro. O desmatamento com solo exposto, que deixa a terra sem vegetação, **foi** a principal causa de desmatamento.

Fonte: elaborada para fins deste estudo

Devido à modelagem do algoritmo de seleção de conteúdo específico para as postagens diárias, não é possível precisar quantas postagens seriam realizadas superficialmente de antemão. Os exemplos apresentados na Figura 53 apresentam várias combinações (*templates*) que contemplam todas as possibilidades da aplicação das funções de flexão verbal do realizador baseado em regras. Contudo, a execução do sistema de geração nesta instância do experimento de validação resultou em 45 postagens, com 2/3 sentença cada (salvas em .txt). Como no experimento executado no *corpus* do modelo mensal, todos os verbos foram gerados satisfatoriamente segundo os parâmetros anotados, de acordo com a necessidade dos contextos. Seguindo o padrão do *corpus* mensal, o *corpus* diário é composto, em sua maioria, por verbos regulares ('gerar, alertar, rastrear, somar, reportar, acumular, desmatar'), mas também apresenta alguns irregulares ('ter, ser').

Os resultados apresentados para a execução da instância local do robô-jornalista, tanto para as potenciais postagens mensais quanto para as diárias, demonstram que uma aplicação efetiva das funções de flexão verbal do realizador baseado em regras são produtivas: permite a flexão verbal satisfatória em ambos os casos em todos os contextos; elimina a necessidade de um arquivo de dicionários no padrão <parâmetros:verbo flexionado> para a chamada dos verbos flexionados, na medida em que executa diretamente as funções de flexão, alimentando-as com os parâmetros adequados à realização textual do verbo pertinente ao contexto; tem potencial de facilitar a anotação dos corpora.

5 CONSIDERAÇÕES FINAIS

Esta tese adotou uma perspectiva dos Estudos Multilíngues, desenvolvidos no escopo da Linguística Sistêmico-Funcional, que examinam o potencial de interação e aproximação entre campos correlatos, modelando o campo fenomênico em termos de *reflexão* e **ação**, adotando uma perspectiva **ativa**, que promove a aplicação dos subsídios resultantes das reflexões teóricas no desenvolvimento e implementação de sistemas em campos variados, por exemplo, a Geração de Língua Natural. Nesse sentido, esta tese adotou uma perspectiva da Linguística Computacional no escopo da Linguística Sistêmico-Funcional, abordando o Processamento de Língua Natural, especificamente a Geração de Língua Natural. Para consecução de seus objetivos, esta tese tomou descrições do português brasileiro, de base Sistêmico-Funcional, disponíveis, para subsidiar o desenvolvimento de um módulo de funções baseadas em regras, voltado à realização textual, independente de domínio, que contempla a escala de ordens, para aplicação em linhas de produção em sistemas de Geração de Língua Natural.

Esse tipo de integração entre as perspectivas teórica e descritiva e a de implementação também é prevista na modelagem apresentada por Teich (1999a) sobre o ambiente metalinguístico, através do princípio de **realização**. De acordo com o modelo metalinguístico proposto, a teoria (Sistêmico-Funcional) e as descrições baseadas nela são **realizadas**, pela **representação computacional**, que, por sua vez, é **realizada** pela **implementação**. Assim, a implementação computacional promovida nesta tese, voltada à Geração de Língua Natural, fecha o ciclo metalinguístico (e metateórico) do modelo proposto por Teich (1999a), e validando as descrições e a teoria as subjaz. Neste sentido, a relevância desta tese resulta do seu potencial de contribuição no âmbito teórico e descritivo, no escopo da Linguística Sistêmico-Funcional, bem como no âmbito de potencial de aplicação.

Partindo dessa perspectiva de integração entre os modos reflexivo e ativo, a pesquisa desenvolvida nesta tese teve como principal objetivo explorar os recursos de Geração de Língua Natural visando a elucidar processos que dizem respeito à produção de significados, por meio da implementação de um módulo realizador textual (baseado em regras e independente de domínio, contemplando a escala de ordens da lexicogramática do português brasileiro), que realiza, computacionalmente, recursos lexicogramaticais do português brasileiro.

Para além do objetivo principal, esta tese teve como objetivos secundários, a realização de experimentos comparativos de acurácia, entre o realizador baseado em regras e os resultados obtidos por modelos baseados em Redes Neurais, na tarefa de flexão verbal nos *corpora* de desenvolvimento e teste compilados no âmbito do projeto de tarefas compartilhadas do CoNLL-SIGMORPHON (COTTERELL et al., 2017; COTTERELL et

al., 2018); realização experimentos de aplicação das funções (de flexão verbal) do realizador textual baseado em regras em verbos (e verbos similares) extraídos do banco lexical do robô-jornalista e na subtarefa de realização textual na linha de produção de uma instância local do robô-jornalista @DaMataReporter.

A pesquisa relatada nesta tese organizou-se como segue. No Capítulo 2 foram apresentados os fundamentos teóricos do modelo linguístico Sistêmico-Funcional, e dos Estudos Multilíngues, que balizaram o desenvolvimento das funções computacionais que compõem o realizador textual baseado em regras; também foram apresentados os fundamentos do Processamento de Língua Natural, em particular a Geração de Língua Natural, tanto de maneira geral, quanto dentro do escopo da Linguística Sistêmico-Funcional.

No Capítulo 3, foram apresentados os critérios de desenvolvimento das funções que realização as unidades da escala de ordens do português brasileiro. É importante reiterar, retomando a relação de realização estabelecida entre a implementação computacional, as descrições linguísticas e a teoria que as subjaz (a LSF), que a modelagem da organização da linguagem como apresentada pela Linguística Sistêmico-Funcional tem impacto direto na metodologia de desenvolvimento das funções do realizador baseado em regras. Neste sentido, foram apresentados os principais preceitos que balizaram o desenvolvimento das funções, em especial, a organização dos eixos paradigmático e sintagmático, do espectro metafuncional, as dimensões de estratificação e instanciação, que motivaram a modelagem dos recursos de **sistema** e de **estrutura**, bem como as descrições que orientaram as implementações.Além disso, foram apresentados os desenhos experimentais, tanto do experimento de desenvolvimento (acurácia e aplicação), quanto do experimento de validação (acurácia e aplicação).

No Capítulo 4, foram apresentados os resultados dos experimentos, destacando-se a acurácia do realizador textual baseado em regras, na tarefa de flexão verbal, em comparação com os resultados dos modelos baseados em Redes Neurais no âmbito do CoNLL-SIGMORPHON (2017/2018). Além disso, destacou-se a aplicação das funções na instância local do robô-jornalista @DaMataReporter, com resultados satisfatório para a tarefa de flexão verbal, possibilitando a diversificação e variabilidade da realização de verbos.

Em relação ao objetivo principal desta tese, a saber, a implementação de um módulo com funções de realização textual, baseado em regras e independente de domínio, contemplando a escala de ordens do português brasileiro, pode-se considerar que ele foi parcialmente cumprido. Foram implementadas funções que realizam desde o morfema, até a oração. Contudo, como foi mencionado, na ordem da palavra verbal:verbo, não foram desenvolvidas funções que contemplem todo o potencial de realização de verbos irregulares, devido ao seu potencial de variabilidade. Além disso, na ordem da oração, como mencionado no Capítulo 3, não foram desenvolvidas funções que contemplem todo o

potencial de realização, como, por exemplo, de tipos de oração com potencial de projeção (e,g. mental e verbal), devido ao grande número de parâmetros previstos e a recursão necessária para sua aplicação. Nesse sentido, esta tese contribuiu nos âmbitos teórico e descritivo, na medida em que permitiu a implementação de recursos computacionais para a realização textual das unidades da escala de ordens, fomentada por descrições do português brasileiro disponíveis, que são, por sua vez, ancoradas nos conceitos teóricos da Linguística Sistêmico-Funcional. Através do princípio da realização – [teoria[representação linguística[representação computacional[implementação]]]] (ver Teich (1999a)) –, as implementações computacionais fecham o ciclo dos modos reflexivo e ativo, apresentados no âmbito dos Estudos Multilíngues (ver Matthiessen et al. (2008)).

Com relação aos objetivos secundários, pode-se considerar que foram alcançados, pois foram aplicados experimentos de acurácia, retornando resultados satisfatórios em comparação a resultados alcançados por Redes Neurais, na tarefa de flexão verbal a partir de parâmetros de flexão; foram realizados testes de aplicação, também com resultado satisfatório, na flexão verbal: de verbos extraídos dos *corpora* do @DaMataReporter e seus similares, extraídos de vetores de palavra; de sentenças extraídas dos *corpora* do@DaMataReporter; de verbos na linha de produção de uma instância local do robô-jornalista @DaMataReporter.

Considerando-se que os objetivos principal e secundários foram atingidos, com algumas ressalvas, é importante ressaltar que trabalhos futuros ainda são necessários para uma refatoração e melhoramento das funções que compõem o módulo de recursos lexicogramaticais. A implementação de recursos lexicogramaticais para a realização textual do português brasileiro, independente de domínio, sob uma perspectiva Sistêmico-Funcional, e baseada em regras, pode ser uma alternativa produtiva a longo prazo, pois, devido ao potencial de variabilidade necessário, possibilita um maior controle da tarefa. Contudo, alguns desafios são inerentes ao desenvolvimento manual de funções com base em descrições linguísticas. Em primeiro lugar, o módulo de recursos é ancorado em descrições do português brasileiro de base sistêmica, e dependem, assim, de uma ampla disponibilidade de descrições, se visa a modelar todo o potencial de recursos da língua. As descrições de base sistêmica do português brasileiro estão em um estágio avançado de desenvolvimento, mas ainda não contemplaram o potencial em sua totalidade. Assim, foi necessário a utilização da descrição de outras línguas (e.g. inglês), quando a congruência dos sistemas possibilitava (e em um baixo grau de delicadeza), para a implementação das funções de realização textual. Além disso, características particulares do português resultaram em limitações, destacando-se, por exemplo, no caso das palavras verbais:verbos, a necessidade de implementação, em alguns casos individual, de funções que realizam verbos irregulares. Por sua vez, no caso das funções de realização das orações, há a necessidade de muitos parâmetros de entrada, por ser necessária a entrada de todos os parâmetros para realização de todos os elementos que realizam cada uma das unidades da escala de ordens (oração grupo palavra morfema),

o que torna o uso dessas funções uma tarefa difícil. Nesse respeito, é necessário que haja uma refatoração, na tentativa de simplificar a passagem de parâmetros. Com relação às orações que apresentam potencial de projeção, é necessário que haja uma refatoração e que sejam desenvolvidos mecanismos de passagem de parâmetros para funções recursivas de realização da oração (projetada), pois a previsão de necessidade de entrada de mais de 800 parâmetros faz com que o uso dessas funções seja inviável. Para além disso, é necessário que sejam desenvolvidas as outras opções dos sistemas que organizam, de maneira geral, a oração, a saber, as opções proeminentes do sistema de TEMA, as opções menos prototípicas do sistema de MODO. Ademais, mediante as refatorações necessárias no realizador baseado em regras, futuramente, é possível a tentativa de implementação de um modelo híbrido, baseando em regra e em redes neurais, que possa se beneficiar tanto da fluência e dos modelos neurais quanto da segurança e controle proporcionadas pelo modelo baseado em regras.

Os experimentos de acurácia, que envolveram a flexão verbal com base em parâmetros, comparando-se os resultados do realizador textual baseado em regras e o resultado obtido por Redes Neurais no âmbito do CoNLL-SIGMORPHON, retornaram resultados satisfatórios. O percentual de acurácia se mostrou competitivo, ficando apenas alguns pontos percentuais abaixo dos sistemas ponta-a ponta. Por ser baseado em regras, o realizador textual desenvolvido nesta tese possibilita um melhor controle das tarefas de realização superficial. No que diz respeito à aplicação das funções do realizador textual, é necessário o uso das funções que realizam o verbo e o grupo verbal na linha de produção oficial do robô-jornalista, por exemplo, para verificação da necessidade de refatoração das funções para ampla aplicação em sistemas de geração. São necessárias ainda, mediante a refatoração e adequação das funções de realização da oração, experimentos de aplicação destas em sistemas de geração que demandem uma maior variabilidade na realização nessa unidade.

REFERÊNCIAS

BATEMAN, J. et al. *Specification of a Discourse Grammar Development Tool. Deliverable R2. 2.1 (Preliminary Version) of WP 2Grammar Integration', esprit Basic Research Project 6665 dandelion*. [S.l.]: GMD-IPSI Darmstadt, Darmstadt, 1994. Citado na página 32.

BATEMAN, J.; ZOCK, M. *Natural Language Generation*. Oxford University Press, 2012. v. 1. Disponível em: <http://oxfordhandbooks.com/view/10.1093/oxfordhb/9780199276349.001.0001/oxfordhb-9780199276349-e-15>. Citado 4 vezes nas páginas 21, 51, 52 e 56.

BATEMAN, J. A. KPML development environment: multilingual linguistic resource development and sentence generation. 1996. Disponível em: <https://www.researchgate.net/publication/37929155_KPML_Development_Environment_multilingual_linguistic_resource_development_and_sentence_generation>. Citado 2 vezes nas páginas 21 e 51.

BATEMAN, J. A. Sentence generation and systemic grammar: an introduction. *Iwanami lecture series: language sciences*, v. 8, p. 1–45, 1997. Citado 2 vezes nas páginas 21 e 51.

BATEMAN, J. A. et al. The rapid prototyping of natural language generation components: an application of functional typology. In: *Proceedings of the 12th international conference on artificial intelligence*. [S.l.: s.n.], 1991. p. 966–971. Citado 2 vezes nas páginas 21 e 51.

BATEMAN, J. A.; MATTHIESSEN, C. M. I. M.; ZENG, L. Multilingual natural language generation for multilingual software: a functional linguistic approach. v. 13, n. 6, p. 607–639, 1999. Citado 2 vezes nas páginas 21 e 51.

BECHARA, E. *Moderna gramática portuguesa–Atualizada pelo novo acordo ortográfico*. [S.l.: s.n.], 2012. Citado na página 100.

BRAGA, A. B. C. *O sistema de transitividade no inglês e no português brasileiro: caracterização da função circunstância com base em textos originais e traduzidos*. Dissertação de mestrado — Universidade Federal de Minas Gerais, 2016. Citado 2 vezes nas páginas 24 e 37.

CAFFAREL, A.; MARTIN, J. R.; MATTHIESSEN, C. M. *Language typology: A functional perspective*. [S.l.]: John Benjamins Publishing, 2004. v. 253. Citado na página 32.

CAMPOS, J. et al. Towards fully automated news reporting in brazilian portuguese. p. 543–554, 2020. ISSN 0000-0000. Conference Name: Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional Publisher: SBC. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/12158>. Citado 8 vezes nas páginas 52, 54, 55, 56, 57, 59, 61 e 63.

COTTERELL, R. et al. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. *arXiv preprint arXiv:1706.09031*, 2017. Citado 17 vezes nas páginas 6, 8, 23, 25, 96, 97, 98, 105, 106, 107, 113, 114, 115, 122, 123, 131 e 132.

COTTERELL, R. et al. The conll–sigmorphon 2018 shared task: Universal morphological reinflection. *arXiv preprint arXiv:1810.07125*, 2018. Citado 12 vezes nas páginas 6, 8, 23, 25, 96, 107, 113, 114, 122, 123, 131 e 132.

FERREGUETTI, K. *As orações existenciais em inglês e português brasileiro: um estudo baseado em corpus*. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2014. Citado na página 37.

FERREGUETTI, K. *A frase preposicional com função de qualificador no grupo nominal: um estudo de equivalentes textuais no par linguístico inglês e português brasileiro*. phdthesis — Universidade Federal de Minas Gerais, 2018. Citado 3 vezes nas páginas 24, 37 e 86.

FERREIRA, T. C. et al. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*, 2019. Citado 2 vezes nas páginas 59 e 63.

FIGUEREDO, G. P. *Uma descrição sistêmico-funcional da estrutura do grupo nominal em português orientada para os estudos lingüísticos da tradução*. Tese (Dissertação (Mestrado)) — Universidade Federal de Minas Gerais, Belo Horizonte: Faculdade de Letras da UFMG / PosLin, jul. 2007. Disponível em: <https://repositorio.ufmg.br/handle/1843/MGSS-77ZJ7W>. Citado 4 vezes nas páginas 24, 37, 80 e 81.

FIGUEREDO, G. P. *Introdução ao perfil metafuncional do português brasileiro: contribuições para os estudos multilíngues*. Tese (Doutorado) — UFMG, Faculdade de Letras, Belo Horizonte, 2011. Citado 6 vezes nas páginas 24, 31, 37, 38, 46 e 90.

FIRTH, J. R. Personality and language in society. v. 42, n. 1, p. 37–52, 1957. Citado na página 31.

HALLIDAY, M. A. K. *Language as social semiotic*. London: Hodder Arnold, 1978. ISBN 978-0-7131-6259-2. Citado 2 vezes nas páginas 44 e 65.

HALLIDAY, M. A. K. *On grammar*. [S.l.]: Bloomsbury Publishing, 2002. v. 1. ISBN 0-8264-4944-1. Citado 5 vezes nas páginas 24, 33, 34, 35 e 36.

HALLIDAY, M. A. K. *On language and linguistics*. [S.l.]: Continuum, 2003. (Collected works of M. A. K. Halliday, ed. by Jonathan J. Webster ; Vol. 3). OCLC: 162251207. ISBN 978-0-8264-8824-4 978-0-8264-5869-8. Citado 2 vezes nas páginas 24 e 65.

HALLIDAY, M. A. K.; MATTHIESSEN, C. Construing experience through meaning: A language-based approach to cognition. *Continuum, London*, 1999. Citado 3 vezes nas páginas 30, 31 e 32.

HALLIDAY, M. A. K.; MATTHIESSEN, C. M. I. M. *Halliday's introduction to functional grammar*. Fourth edition. Milton Park, Abingdon, Oxon: Routledge, 2014. ISBN 978-0-415-82628-0 978-1-4441-4660-8. Citado 14 vezes nas páginas 34, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 65, 86 e 88.

HALLIDAY, M. A. K.; WEBSTER, J. *Continuum companion to systemic functional linguistics*. [S.l.]: Continuum, 2009. OCLC: 276648367. ISBN 978-0-8264-9447-4 978-0-8264-9448-1. Citado 4 vezes nas páginas 19, 20, 50 e 51.

JURAFSKY, D. *Speech & language processing*. [S.l.]: Pearson Education India, 2000. Citado na página 49.

MATTHIESSEN, C. *Lexicogrammatical cartography: English systems*. [S.l.]: Internat. Language Sciences Publ., 1995. Citado na página 36.

MATTHIESSEN, C.; TERUYA, K.; LAM, M. *Key terms in systemic functional linguistics*. [S.l.]: A&C Black, 2010. Citado 9 vezes nas páginas 34, 35, 38, 40, 41, 45, 46, 47 e 48.

MATTHIESSEN, C.; TERUYA, K.; WU, C. Multilingual studies as a multi-dimensional space of interconnected language studies. *Meaning in context: Implementing intelligent applications of language studies*, London: Continuum, p. 146–220, 2008. Citado 8 vezes nas páginas 6, 8, 17, 19, 28, 29, 30 e 133.

MATTHIESSEN, C. et al. The multex generator and its environment: application and development. In: *Natural Language Generation*. [S.l.: s.n.], 1998. Citado 2 vezes nas páginas 21 e 51.

MATTHIESSEN, C. M. The environments of translation. *Exploring translation and multilingual text production: Beyond content*, Mouton de Gruyter Berlin & New York, p. 41–124, 2001. Citado na página 48.

MATTHIESSEN, C. M. Applying systemic functional linguistics in healthcare contexts. *Text & Talk*, De Gruyter Mouton, v. 33, n. 4-5, p. 437–466, 2013. Citado na página 43.

MATTHIESSEN, C. M. I. M.; BATEMAN, J. A. *Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese*. [S.l.]: Pinter Pub Ltd, 1992. ISBN 978-0-86187-711-9. Citado na página 51.

MATTHIESSEN, C. M. I. M.; MANN, W. C. Demonstration of the nigel text generation computer program. In: BENSON, J. D.; GREAVES, W. S. (Ed.). *Systemic Perspectives on Discourse*. [S.l.]: Ablex Publishing Company, 1985. v. 1, p. 50–83. ISBN 978-0-89391-193-5. Google-Books-ID: gfe2AAAAIAAJ. Citado 2 vezes nas páginas 21 e 51.

OLIVEIRA, R. G. de. Applying the general upper model for automatic generation of spatial language in brazilian portuguese. 2013. Citado 8 vezes nas páginas 6, 8, 21, 22, 23, 63, 64 e 68.

PAULA, A. A. d. *Orações verbais–uma descrição sistêmico funcional dos processos de representação do dizer do português brasileiro*. Dissertação de mestrado — Universidade Federal de Ouro Preto, 2018. Citado 2 vezes nas páginas 24 e 37.

REITER, E.; DALE, R. *Building natural language generation systems*. [S.l.]: Cambridge university press, 2000. Citado 9 vezes nas páginas 17, 50, 52, 53, 56, 58, 59, 62 e 64.

ROSA, A. L. et al. DaMata: A robot-journalist covering the brazilian amazon deforestation. In: *Proceedings of the 13th International Conference on Natural Language Generation*. Association for Computational Linguistics, 2020. p. 103–106. Disponível em: <https://www.aclweb.org/anthology/2020.inlg-1.15>. Citado 8 vezes nas páginas 52, 53, 55, 57, 58, 61, 62 e 64.

SANTOS, R. L. d. *Processamento de Linguagem Natural*. 2020. Disponível em: <https://www.sbc.org.br/14-comissoes/394-processamento-de-linguagem-natural>. Citado na página 50.

SYLAK-GLASSMAN, J. et al. A universal feature schema for rich morphological annotation and fine-grained cross-lingual part-of-speech tagging. In: SPRINGER. *International Workshop on Systems and Frameworks for Computational Morphology*. [S.l.], 2015. p. 72–93. Citado na página 97.

Sá, A. d. M. *Uma descrição sistêmico-funcional do grupo verbal do português brasileiro orientada para os estudos da tradução*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte: Faculdade de Letras da UFMG / PosLin, 2016. Disponível em: <https://repositorio.ufmg.br/handle/1843/MGSS-ACXPQ8>. Citado 7 vezes nas páginas 24, 37, 73, 82, 83, 116 e 124.

TEICH, E. *Systemic functional grammar in natural language generation: linguistic description and computational representation*. [S.l.]: Cassell, 1999a. (Communication in artificial intelligence). ISBN 978-0-304-70168-1. Citado 7 vezes nas páginas 17, 20, 21, 32, 51, 131 e 133.

APÊNDICE A – PARADIGMA DE FLEXÃO VERBAL DO @DAMATAREPORTER

"**desmatar**":

{"**presente**": ["desmato","desmatas","desmata","desmatamos","desmatais","desmatam"],

"**pretérito perfectivo I**": ["desmatei", "desmataste","desmatou","desmatamos","desmatastes", "desmataram"],

"**pretérito perfectivo II**": ["desmatara", "desmataras", "desmatara", "desmatáramos","desmatáreis", "desmataram"],

"**pretérito imperfectivo**": ["desmatava", "desmatavas", "desmatava", "desmatávamos", "desmatáveis", "desmatavam"],

"**passado volitivo**": [desmataria", "desmatarias", "desmataria", "desmataríamos", "desmataríeis", "desmatariam"],

"**futuro**": ["desmatarei", "desmatarás", "desmatará", "desmataremos", "desmatareis", "desmatarão"],

"**subjuntivo conjuntivo**": ["desmate", "desmates", "desmate", "desmatemos", "desmateis", "desmatem"],

"**subjuntivo condicional**": ["desmatasse", "desmatasses", "desmatasse", "desmatássemos", "desmatásseis", "desmatassem"],

"**subjuntivo optativo**": ["desmatar", "desmatares", "desmatar", "desmatarmos", "desmatardes", "desmatarem"],

"**não finito concretizado**": ["desmatar", "desmatares", "desmatar", "desmatarmos", "desmatardes", "desmatarem"],

"**imperativo I**": [, "desmata", "desmate", "desmatemos", "desmatai", "desmatem"],

"**imperativo II**": [, "desmates", "desmate", "desmatemos", "desmateis", "desmatem"],

"**gerúndio**": ["desmatando"]

"**particípio**": ["desmatado", "desmatada", "desmatados", "desmatadas"]},

"**registrar**":

{ "**presente**": ["registro","registras","registra","registramos","registrais", "registram"],

"**pretérito perfectivo I**": ["registrei","registraste","registrou","registramos", "registrastes","registraram"],

"**pretérito perfectivo II**": ["registrara", "registraras", "registrara", "registráramos", "registráreis", "registraram"],

"**pretérito imperfectivo**": ["registrava", "registravas", "registrava", "registrávamos", "registráveis", "registravam"],

"**passado volitivo**": ["registraria", "registrarias", "registraria", "registraríamos", "registraríeis", "registrariam"],

"**futuro**": ["registrarei", "registrarás", "registrará", "registraremos", "registrareis", "registrarão"],

"**subjuntivo conjuntivo**": ["registre", "registres", "registre", "registremos", "registreis", "registrem"],

"**subjuntivo condicional**": ["registrasse", "registrasses", "registrasse", "registrássemos", "registrásseis", "registrassem"],

"**subjuntivo optativo**": ["registrar", "registrares", "registrar", "registrarmos", "registrardes", "registrarem"],

"**não finito concretizado**": ["registrar", "registrares", "registrar", "registrarmos", "registrardes", "registrarem"],

"**imperativo I**": [-", "registra", "registre", "registremos", "registrai", "registrem"],

"**imperativo II**": [-", "registres", "registre", "registremos", "registreis", "registrem"],

"**gerúndio**": ["registrando"]

"**particípio**":}

"**identificar**":

{"**presente**": ["identifico", "identificas", "identifica", "identificamos", "identificais", "identificam"],

"**pretérito perfectivo I**": ["identifiquei", "identificaste", "identificou", "identificamos", "identificastes", "identificaram"],

"**pretérito perfectivo II**": ["identificara", "identificaras", "identificara", "identificáramos", "identificáreis", "identificaram"],

"**pretérito imperfectivo**": ["identificava", "identificavas", "identificava", "identificávamos", "identificáveis", "identificavam"],

"**passado volitivo**": ["identificaria", "identificarias", "identificaria", "identificaríamos", "identificaríeis", "identificariam"],

"**futuro**": ["identificarei", "identificarás", "identificará", "identificaremos", "identificareis", "identificarão"],

"**subjuntivo conjuntivo**": ["identifique", "identifiques", "identifique", "identifiquemos", "identifiqueis", "identifiquem"],

"**subjuntivo condicional**": ["identificasse", "identificasses", "identificasse", "identificássemos", "identificásseis", "identificassem"],

"**subjuntivo optativo**": ["identificar", "identificares", "identificar", "identificarmos", "identificardes", "identificarem"],

"**não finito concretizado**": ["identificar", "identificares", "identificar", "identificarmos", "identificardes", "identificarem"],

"**imperativo I**": [-", "identifica", "identifique", "identifiquemos", "identificai", "identifiquem"],

"**imperativo II**": [-", "identifiques", "identifique", "identifiquemos", "identifiqueis", "identifiquem"],

"**gerúndio**": ["identificando"]

"**particípio**": ["identificado", "identificada", "identificados", "identificadas"]}

"**detectar**":

{ "**presente**": ["detecto", "detectas", "detecta", "detectamos", "detectais", "detectam"],

"**pretérito perfectivo I**": ["detectei", "detectaste", "detectou", "detectamos", "detectastes", "detectaram"],

"**pretérito perfectivo II**": ["detectara", "detectaras", "detectara", "detectáramos", "detectáreis", "detectaram"],

"**pretérito imperfectivo**": ["detectava", "detectavas", "detectava", "detectávamos", "detectáveis", "detectavam"],

"**passado volitivo**": ["detectaria", "detectarias", "detectaria", "detectaríamos", "detectaríeis", "detectariam"],

"**futuro**": ["detectarei", "detectarás", "detectará", "detectaremos", "detectareis", "detectarão"],

"**subjuntivo conjuntivo**": ["detecte", "detectes", "detecte", "detectemos", "detecteis", "detectem"],

"**subjuntivo condicional**": ["detectasse", "detectasses", "detectasse", "detectássemos", "detectásseis", "detectassem"],

"**subjuntivo optativo**": ["detectar", "detectares", "detectar", "detectarmos", "detectardes", "detectarem"],

"**não finito concretizado**": ["detectar", "detectares", "detectar", "detectarmos", "detectardes", "detectarem"],

"**imperativo I**": [-", "detecta", "detecte", "detectemos", "detectai", "detectem"],

"**imperativo II**": [-", "detectes", "detecte", "detectemos", "detecteis", "detectem"]

"**detectar**": ["detectando"],

"**detectar**": ["detectado", "detectada", "detectados", "detectadas"] }

APÊNDICE B – LISTA DE LEMAS E SINÔNIMOS

"**registrar**":
["conter","apontar","configurar","encaminhar","usar",
"receber","ofertar","descartar", "liberar","trazer"],

"**contabilizar**":
["atingir","deduzir","suportar","ultrapassar","duplicar",
"aumentar","reduzir","ressarcir","acumular","embolsar"]

"**somar**":
["amealhar","subir","acrescentar","agregar","conquistar",
"marcar","anotar","ganhar","chegar","embolsar"]

"**totalizar**":
["embolsar","contabilizar","atingir","estipular","acrescentar",
"valorizar","elevar","reduzir","atingir","cobrar"]

"**alcançar**":
["atingir","conquistar","representar","favorecer","refrear",
"manter","superar","suster","canalizar","alcançar"]

"**chegar**":
["retornar","dirigir","rumar","voltar","assistir",
"regressar","aceder","antecipar","juntar","agarrar"]

"**reportar**":
["expor","associar","submeter","entregar","confirmar",
"apresentar","encaminhar","revelar","apontar","justificar"]

"**observar**":
["resumir","vislumbrar","mostrar","apreciar","apontar",
"concluir","definir","enunciar","descortinar","destacar"]

"**ser**":
["revelar","afigurar","revelar","ser","ver"]

"**representar**":
["substituir","simbolizar","indicar","invocar","atingir",
"levar","alcançar","utilizar","reintroduzir","reivindicar"]

"**acontecer**":
["ocorrer","realizar","repetir","alterar","iniciar","seguir",
"esperar","infeccionar","durar","chover"]

"**desmatar**":
["tragar","ventilar","regenerar","financiar","desmontar",
"urbanizar","amainar","escavar","verticalizar","exorcisar"]

"**ocorrer**":
["realizar","acontecer","resultar","acarretar","observar",
"efectuar","produzir","ocasionar","implicar","traduzir"]

"**existir**":
["haver","subsistir","solucionar","justificar","implicar",
"ocorrer","exercer","realizar","julgar","surgir"]

```
"ter":
["ter","ter","haver"]
"apresentar":
["apreciar","expor","resumir","realizar","examinar",
"confirmar","revelar","lançar","reportar"]
"possuir":
["obter","constituir","existir","ostentar","englobar",
"haver","considerar","adquirir"]
"estar":
["ficar","sentir","permanecer","permancer",
"continuar","mostrar","manter"]
"acumular":
["conter","contabilizar","gerar","registrar","diminuir",
"descartar","embolsar","atingir","reter","absorver"]
"relatar":
["explicar","detalhar","apontar","rebater","encaminhar",
"conduzir","narrar","atestar","transmitir","denunciar"]
"sofrer":
["provocar","causar","originar","ocasionar","acarretar",
"desencadear","produzir","causar","implicar","registar"]
```

Fonte: Elaborada para fins deste estudo

APÊNDICE C – ORAÇÕES @DAMATAREPORTER COM SINÔNIMOS

["No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) consultou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) detalhou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) contatou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) informou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) adiantou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) examinou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) contactou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) expôs", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) indicou", "No dia 31 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) reveu", "que existiu alertas de desmatamento", "que implicou alertas de desmatamento", "que justificou alertas de desmatamento", "que tomou alertas de desmatamento", "que acarretou alertas de desmatamento", "que subsistiu alertas de desmatamento", "que incluiu alertas de desmatamento", "que exerceu alertas de desmatamento", "que aceitou alertas de desmatamento", "que indiciou alertas de desmatamento", "que amealham 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que subem 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que acrescentam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que agregam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que conquistam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que marcam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que anotam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que ganham 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que chegam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "que embolsam 1.315173359776571 km² no município de Alta Floresta D'oeste/RO", "contendo 7 dias com alertas no mês na região.", "contabilizando 7 dias com alertas no mês na região.", "gerando 7 dias com alertas no mês na região.", "registrando 7 dias com alertas no mês na região.", "diminuindo 7 dias com alertas no mês na região.", "descartando 7 dias com alertas no mês na região.", "embolsando 7 dias com alertas no mês na região.", "atingindo 7 dias com alertas no mês na região.", "retendo 7 dias com alertas no mês na região.", "absorvendo 7 dias com alertas no mês na região.", "Alta Floresta D'oeste tem 12.685352175179451 km² de desmatamento em julho.", "Alta Floresta D'oeste há 12.685352175179451 km² de desmatamento em julho.", "A principal causa dos alertas de desmatamento foi o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento foi o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento revelou o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento afigurou o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento revelou o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento foi o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento veu o desmatamento com solo exposto,", "A principal causa dos alertas de desmatamento foi o desmatamento com solo exposto,", "que deixa a terra sem vegetação.", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) expôs avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) associou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) submeteu avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) entregou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) confirmou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) apresentou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) encaminhou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) revelou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) apontou avisos de desmatamento", "Em 8 de julho de 2019, o Instituto Nacional de Pesquisas Espaciais (INPE) justificou avisos de desmatamento", "amealhando 12.007300834420757 km² na cidade de Apuí/AM", "subindo 12.007300834420757 km² na cidade de Apuí/AM", "acrescentando 12.007300834420757 km² na cidade de Apuí/AM", "agregando 12.007300834420757 km² na cidade de Apuí/AM", "conquistando 12.007300834420757 km² na cidade de Apuí/AM", "marcando 12.007300834420757 km² na cidade de Apuí/AM", "anotando 12.007300834420757 km² na cidade de Apuí/AM", "ganhando 12.007300834420757 km² na cidade de Apuí/AM", "chegando 12.007300834420757 km² na cidade de Apuí/AM", "embolsando 12.007300834420757 km² na cidade de Apuí/AM", "que conteve 11 dias com alertas.", "que contabilizou 11 dias com alertas.", "que gerou 11 dias com alertas.", "que registrou 11 dias com alertas.", "que diminuiu 11 dias com alertas.", "que descartou 11 dias com alertas.", "que embolsou 11 dias com alertas.", "que atingiu 11 dias com alertas.", "que reteve 11 dias com alertas.", "que absorveu 11 dias com alertas.", "Apuí amealha 75.16054508934322 km² de área

desmatada no mês de julho.", "Apuí sube 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí acrescenta 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí agrega 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí conquista 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí marca 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí anota 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí ganha 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí chega 75.16054508934322 km² de área desmatada no mês de julho.", "Apuí embolsa 75.16054508934322 km² de área desmatada no mês de julho."]

Fonte: Elaborada para fins deste estudo