


**Universidade Estadual de Campinas (UNICAMP)**
**Instituto de Computação (IC)**
**Programa de Pós-Graduação em Ciência da Computação**

**Matheus Ferraroni Sanches**

# Métodos e Recursos de Dados para Modelos de Linguagem em Português Brasileiro: da Coleta e Adaptação à Privacidade em Aprendizado Federado

CAMPINAS
2026

**Matheus Ferraroni Sanches**

# Métodos e Recursos de Dados para Modelos de Linguagem em Português Brasileiro: da Coleta e Adaptação à Privacidade em Aprendizado Federado

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas (UNICAMP) como parte dos requisitos exigidos para a obtenção do título de Doutor em Ciência da Computação.

**Orientador: Prof. Dr. Leandro Aparecido Villas**

Este exemplar corresponde à versão final da tese defendida.

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Sa55m

Sanches, Matheus Ferraroni, 1996-
Métodos e recursos de dados para modelos de linguagem em português brasileiro : da coleta e adaptação à privacidade em aprendizado federado / Matheus Ferraroni Sanches. – Campinas, SP : [s.n.], 2026.

Orientador: Leandro Aparecido Villas.
Tese (doutorado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Computação.

1. Aprendizado federado (Aprendizado de máquina). 2. Privacidade de dados. 3. Aprendizado de máquina. 4. Processamento de linguagem natural. 5. Conjunto de dados. I. Villas, Leandro Aparecido, 1983-. II. Universidade Estadual de Campinas (UNICAMP). Instituto de Computação. III. Título.

Informações complementares

**Título em outro idioma:** Methods and data resources for Brazilian Portuguese language models : from data collection and adaptation to privacy in federated learning
**Palavras-chave em inglês:**
Federated learning (Machine learning)
Data privacy
Machine learning
Natural language processing
Datasets
**Titulação:** Doutor em Ciência da Computação
**Banca examinadora:**
Leandro Aparecido Villas [Orientador]
Flávia Coimbra Delicato
Elvis Fusco
Washington Luiz Miranda da Cunha
Helena de Almeida Maia
**Data de defesa:** 26-10-2026
**Programa de Pós-Graduação:** Ciência da Computação

**Objetivos de Desenvolvimento Sustentável (ODS)**
Não se aplica

**Identificação e informações acadêmicas do(a) aluno(a)**
- ORCID do autor: https://orcid.org/0000-0002-6904-7873
- Currículo Lattes do autor: http://lattes.cnpq.br/7066133969704063

# Folha de Aprovação

**Matheus Ferraroni Sanches**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação, defendida na data de 26/10/2026.

Composição da Comissão Examinadora:

Prof. Dr. Leandro Aparecido Villas
Universidade Estadual de Campinas

Profa. Dra. Flávia Coimbra Delicato
Universidade Federal Fluminense

Prof. Dr. Elvis Fusco
Fundação Shunji Nishimura de Tecnologia

Prof. Dr. Washington Luiz Miranda da Cunha
Universidade Estadual de Campinas

Profa. Dra. Helena de Almeida Maia
Universidade Estadual de Campinas

# Notas do Autor

Como esta pesquisa propõe técnicas, abordagens, modelos e estudos em português, a tese aqui apresentada mantém esse idioma, no qual as principais contribuições foram realizadas.

# Resumo

Os Large Language Models (LLMs) consolidaram-se como uma das tecnologias centrais da atualidade, empregados tanto diretamente por usuários quanto como infraestrutura para soluções mais complexas. A obtenção de níveis elevados de qualidade, contudo, depende, entre outros fatores, da disponibilidade de dados adequados ao domínio e ao idioma.

A privacidade dos participantes é um fator central no treinamento para mitigar o vazamento de informações sensíveis. Ainda assim, persistem desafios relacionados à escassez de dados adequados, sobretudo em domínios específicos ou em cenários com requisitos estruturais particulares, como diálogos multiturnos, nos quais a presença de informações pessoais pode aumentar o risco de vazamento. A literatura aponta que a insuficiência de dados apropriados limita o aprimoramento dos modelos atuais, ao passo que técnicas de proteção de privacidade podem comprometer diretamente seu desempenho.

Nesse contexto, este trabalho aborda a geração e a adaptação de dados em português brasileiro, sua utilização no pré-treinamento continuado de modelos de linguagem e a mitigação de vazamento de informações sensíveis em aprendizado federado. A pesquisa contempla dados conversacionais reais e sintéticos, adaptação de conjuntos existentes e coleta web em larga escala, além da avaliação controlada de modelos refinados, da modelagem de ataques de extração e da proposição de uma defesa baseada em substituição semântica.

Como resultado, foram desenvolvidas três metodologias de preparação de dados, implementadas em três ferramentas compartilhadas publicamente, além de um mecanismo experimental de defesa e dois pipelines experimentais. Também foram gerados ou adaptados seis conjuntos de dados para diferentes áreas. As metodologias propostas foram aplicadas à preparação de dados em grande volume para o treinamento e o refinamento de LLMs.

Também foram obtidas duas versões do Tucano2-0.6B-Base por pré-treinamento continuado, comparando uma composição geral de dados com outra que incorpora textos de fóruns, sob o mesmo volume de tokens e os mesmos hiperparâmetros. Em relação ao modelo inicial, ambas apresentaram ganhos na classificação de textos e melhorias na modelagem de linguagem. A substituição de parte dos dados gerais por textos de fóruns reduziu adicionalmente a perda na previsão de textos desse domínio.

No aprendizado federado, experimentos com dados pessoais sintéticos avaliaram ataques de extração e compararam a substituição semântica com uma implementação experimental de DP-AdamW. A substituição semântica reduziu a reprodução de informações sensíveis e apresentou menores loss, NLL e perplexidade no conjunto sintético held-out, nas condições avaliadas, demonstrando seu potencial para mitigar o vazamento de dados.

# Abstract

LLMs have become one of today's central technologies, used both directly by users and as infrastructure for more complex solutions. Achieving high levels of quality, however, depends, among other factors, on the availability of data appropriate to the domain and language.

Participant privacy is a central consideration in training to mitigate the leakage of sensitive information. Nevertheless, challenges persist regarding the scarcity of suitable data, particularly in specialized domains or settings with specific structural requirements, such as multi-turn dialogues, where personal information may increase the risk of leakage. The literature indicates that insufficient suitable data limits improvements to existing models, while privacy protection techniques may directly impair their performance.

In this context, this work addresses the generation and adaptation of Brazilian Portuguese data, their use in the continual pretraining of language models, and the mitigation of sensitive information leakage in federated learning. The research encompasses real and synthetic conversational data, the adaptation of existing datasets, and large-scale web data collection, alongside the controlled evaluation of adapted models, the modeling of extraction attacks, and the development of a defense based on semantic substitution.

As a result, three data preparation methodologies were developed and implemented in three publicly available tools, alongside one experimental defense mechanism and two experimental pipelines. Six datasets were also generated or adapted for different domains. The proposed methodologies were applied to large-scale data preparation for the training and refinement of LLMs.

Two versions of Tucano2-0.6B-Base were also obtained through continual pretraining, comparing a general data mixture with one incorporating forum texts, using the same token budget and hyperparameters. Compared with the initial model, both achieved gains in text classification and improvements in language modeling. Replacing part of the general data with forum texts further reduced the loss when predicting texts from that domain.

In federated learning, experiments using synthetic personal data evaluated extraction attacks and compared semantic substitution with an experimental implementation of DP-AdamW. Semantic substitution reduced the reproduction of sensitive information and achieved lower loss, NLL, and perplexity on the held-out synthetic dataset under the evaluated conditions, demonstrating its potential to mitigate data leakage.

# Lista de Figuras


1.1 Evolução anual dos registros recuperados pela busca no OpenAlex e do número médio de usuários ativos semanais (WAU) do ChatGPT (anos completos até 2025). . . . . 19

2.1 Mecanismo de atenção . . . . . 28
2.2 Fluxo conceitual simplificado de modelos baseados em Transformer. . . . . 29
2.3 Estrutura simplificada de uma conversa multiturno, multiparticipante e ramificada. . . . . 32
2.4 Funcionamento do aprendizado federado. . . . . 34
2.5 Envio de atualização local maliciosa durante uma rodada de aprendizado federado. . . . . 37
2.6 Redistribuição do modelo global afetado após agregação de uma atualização maliciosa. . . . . 38

4.1 Distribuição dos conjuntos de dados disponíveis em função do número de falantes nativos (L1) e não nativos (L2) de cada idioma. . . . . 59
4.2 Relação entre a metodologia MCCD e sua implementação para uma fonte de dados. . . . . 62
4.3 Estágios do Methodology for Creating Conversational Datasets (MCCD). . 64
4.4 Fluxos de conversa entre mensagens . . . . . 64
4.5 Aplicação da metodologia MCCD no Miner-XenForo para a construção dos conjuntos Adrenaline e OuterSpace. . . . . 65
4.6 Etapas do estágio de aquisição de dados. . . . . 65
4.7 Etapas do estágio de processamento. . . . . 67
4.8 Diagrama temporal com dependências entre eventos. . . . . 72
4.9 Pipeline de aquisição, anonimização e processamento do Miner-XenForo. . 74
4.10 Etapas do estágio de aquisição de dados do Miner-XenForo. . . . . 76
4.11 Exemplo das transformações aplicadas aos campos de identificação nos arquivos JSON. . . . . 78
4.12 Pipeline do estágio de processamento do Miner-XenForo. . . . . 79
4.13 Exemplo de referências fora da ordem cronológica. . . . . 81
4.14 Distribuição do número de mensagens por tópico nos conjuntos Adrenaline e OuterSpace. . . . . 84
4.15 Fluxo de avaliação . . . . . 85
4.16 Distribuição de categorias e subcategorias. . . . . 88
4.17 Distribuição por faixa do número de palavras. . . . . 89
4.18 Frequência dos termos mais comuns no IMDB PT-BR, com separação por polaridade e total sobreposto. . . . . 90
4.19 Frequência dos bigramas mais comuns. . . . . 90
4.20 Acurácia do MBE e do BERTimbau em diferentes tarefas de classificação. . 91

5.1 Fluxo de criação e edição de diálogos. . . . . 96
5.2 Diagrama de estados de uma história na metodologia Mechanical Dialogue Creator (MDC). . . . . 97
5.3 Passos para criar diálogos assíncronos. . . . . 97

5.4 Fluxo para selecionar uma história. . . . . 98
5.5 Fluxo principal simplificado para pedir pizza, com loop de alteração antes de endereço e pagamento. . . . . 101
5.6 Permutações possíveis das letras ABC, como ilustração do crescimento fatorial ($n!$). . . . . 102
5.7 Crescimento do número de permutações ($n!$) em função do tamanho do conjunto. . . . . 103
5.8 Exemplo de permutação. . . . . 103
5.9 Exemplo da interface da ferramenta. . . . . 105
5.10 Diagrama simplificado da comunicação entre o cliente e servidor MDC. . . 105
5.11 Multiplicação dos fluxos. . . . . 109
5.12 Multiplicação dos fluxos com aplicação de limite de uso por rótulo. . . . . 109

6.1 Exemplo de saída de *fill-mask* com top 20 probabilidades. . . . . 114
6.2 Número de citações do BrWaC por ano [Google Scholar, 2026]. . . . . 115
6.3 Fluxo de filtragem aplicado ao corpus BrWaC. . . . . 116
6.4 Processo de tradução do MultiWOZ e pós-processamento para realinhar *slot values* ao texto em PT-BR. . . . . 121
6.5 Verificação de inconsistências entre *slot values* e *utterances*. . . . . 123
6.6 Amostras com desalinhamento após tradução, detecção e correção manual (os rótulos do eixo horizontal reproduzem literalmente os valores de slot gerados pela tradução automática). . . . . 126

7.1 Fluxo da ferramenta BootCaT para criação iterativa de conjuntos de dados WaCky. . . . . 129
7.2 Exemplo da web representada como grafo. . . . . 130
7.3 Fluxo simplificado da metodologia de exploração e coleta recursiva proposta. 131
7.4 Diagrama de estados possíveis do sistema, desde a inicialização até a conclusão com sucesso ou erro. . . . . 133
7.5 Estágios de inicialização, aquisição e conclusão, com seus respectivos módulos. 134
7.6 Processo de inicialização da coleta a partir de consultas semânticas. . . . . 137
7.7 Processo de requisição, validação e persistência de URLs e conteúdos durante a coleta. . . . . 139
7.8 Processo de aquisição com destaque nas validações e filtros. . . . . 143
7.9 Múltiplas páginas de origem referenciando a mesma URL. . . . . 145
7.10 Modelagem mínima para atender aos requisitos de persistência da metodologia. . . . . 149
7.11 Banner de apresentação do sistema *WackyWacky*. . . . . 150
7.12 Principais componentes do WackyWacky e suas dependências funcionais. . 151
7.13 Modelo relacional do crawler com configuração dinâmica desacoplada. . . . 153
7.14 Demonstração da execução concorrente. . . . . 155
7.15 Pipeline principal de processamento. . . . . 156
7.16 Pipeline interno do componente `Requester`. . . . . 157
7.17 Distribuição tridimensional dos termos-semente, agrupados por K-Means. . 162
7.18 Distribuição dos tamanhos dos textos obtidos. . . . . 169
7.19 Cobertura acumulada das ocorrências de palavras pelos termos mais frequentes no conjunto final. . . . . 169
7.20 Palavras mais frequentes da coleção *WackyWacky*, excluindo stopwords portuguesas. . . . . 170

7.21 Bigramas mais frequentes da coleção *WackyWacky*, excluindo stopwords portuguesas. . . . . 171

8.1 Composição pareada dos dados de pré-treinamento dos modelos Geral e Fórum/Tec. . . . . 179

9.1 Visão geral do protocolo experimental federado. . . . . 191
9.2 Participantes e limites de acesso no protocolo federado. . . . . 192
9.3 Treinamento local e agregação em uma rodada federada. . . . . 193
9.4 Auditoria de extração realizada pelo avaliador confiável. . . . . 194
9.5 Substituição semântica rotativa com valores mantidos na rodada e renovados na seguinte. . . . . 194

# Lista de Tabelas


1.1 Registros bibliográficos recuperados pela consulta no OpenAlex, por ano. Os valores precedidos por ≈ são aproximados. Em 2026, contagem parcial de 1º de janeiro a 8 de setembro; * variação em relação ao total anual de 2025, sem anualização. . . . . 18

2.1 Representação das frases em formato *bag of words*. . . . . 27
2.2 Possíveis dimensões de organização de recursos e adequação a diferentes usos em LLMs. . . . . 31

3.1 Comparação entre metodologias relacionadas e propostas neste trabalho. . 42
3.2 Comparação entre as metodologias propostas para construção de conjuntos de diálogos Human-to-Human (H2H), multiturno e multiusuário. . . . . 43
3.3 Comparação entre ferramentas aplicáveis a diferentes idiomas. . . . . 44
3.4 Comparação entre conjuntos de dados multidomínio em português. . . . . 48
3.5 Comparação entre conjuntos de dados de diálogos relacionados. . . . . 49
3.6 Comparação entre conjuntos de dados de diálogos propostos. . . . . 50
3.7 Comparação do escopo experimental dos trabalhos de ataque com esta tese. 56
3.8 Comparação dos tipos de vazamento abordados nos trabalhos relacionados. 57
3.9 Comparação entre as defesas relacionadas e as defesas avaliadas nesta tese. 57

4.1 Conjuntos de Dados disponíveis por idioma em março de 2026. . . . . 59
4.2 Diálogo estruturado sobre impactos da inteligência artificial na educação. . 72
4.3 Comparação entre conjunto de dados Adrenaline e OuterSpace. . . . . 83
4.4 Exemplo com títulos, início das postagens e embeddings simulados (quebras manuais). . . . . 87

5.1 Exemplo de roteiro. . . . . 95
5.2 Exemplo de rótulo e diálogos. . . . . 100
5.3 Combinações teóricas para 14 posições com diferentes quantidades de alternativas por posição. . . . . 109

7.1 Responsabilidades da persistência no sistema. . . . . 152
7.2 Domínios bloqueados por categoria. . . . . 159
7.3 Configurações de Execução. . . . . 161
7.4 Distribuição de URLs por Status . . . . . 163
7.5 Distribuição dos estados selecionados da coleta. . . . . 164
7.6 Principais domínios por requisições, textos válidos e volume textual. . . . . 165
7.7 Dez domínios não Wikimedia com maior contribuição ao conteúdo textual deduplicado. . . . . 166
7.8 Domínios com maior número de filhos diretos na relação de descoberta e média de requisições desses filhos. . . . . 167
7.9 Distribuição dos domínios e das requisições por nível de recursão do domínio no snapshot analisado. . . . . 168
7.10 Síntese quantitativa da coleta e do conjunto textual analisado. As contagens de palavras e caracteres referem-se aos documentos após limpeza e deduplicação. . . . . 168

8.1 Distribuição das sequências de treinamento de 1.024 tokens por corpus e modelo. . . . 178
8.2 Hiperparâmetros do pré-treinamento continuado. . . . 180
8.3 Loss e perplexidade na avaliação intrínseca. Valores menores indicam melhor desempenho. . . . 184
8.4 Médias das métricas de classificação. . . . 184
8.5 NLL condicional e perplexidade por checkpoint. O passo zero corresponde ao Base. . . . 185
8.6 Contrastes pareados e IC95% nas três avaliações. . . . 186

9.1 Campos protegidos e propriedades preservadas pela substituição semântica rotativa. . . . 196
9.2 Modelo e configurações do treinamento federado não privado. . . . 198
9.3 Cenários avaliados no protocolo federado. . . . 199
9.4 Reprodução direcionada na rodada 20, média de duas execuções ($n = 2$). . . . 204
9.5 Reprodução de pares substitutos na rodada 20 ($n = 2$). . . . 205
9.6 Métricas médias de utilidade sintética ($n = 2$). . . . 206

# Lista de Abreviações e Siglas

| | |
|---|---|
| **DL** | Deep Learning |
| **DP** | Differential Privacy |
| **RDP** | Rényi Differential Privacy |
| **FL** | Federated Learning |
| **IA** | Inteligência Artificial |
| **MDC** | Mechanical Dialogue Creator |
| **NLL** | Log-Verossimilhança Negativa |
| **NLP** | Natural Language Processing |
| **LLM** | Large Language Model |
| **PII** | Personally Identifiable Information |
| **MLP** | Multi-Layer Perceptron |
| **H2H** | Human-to-Human |
| **LLMs** | Large Language Models |
| **MCCD** | Methodology for Creating Conversational Datasets |
| **SUMO** | Simulation of Urban MObility |
| **LAFt** | Latent Association Finetuning |
| **DP-SGD** | Differential Privacy Stochastic Gradient Descent |
| **MultiWOZ** | Multi-Domain Wizard-of-Oz |
| **FedAVG** | FederatedAveraging |

# Sumário


**1 Introdução** **17**
1.1 Contextualização e Motivação do Problema . . . . . 17
1.2 Objetivos . . . . . 21
1.3 Contribuições da Tese . . . . . 23
1.4 Estrutura da Tese . . . . . 25

**2 Referencial Teórico** **26**
2.1 Modelos de Linguagem . . . . . 26
2.2 Conjuntos de Dados para LLMs . . . . . 29
2.3 Aprendizado Federado . . . . . 33
2.4 Privacidade e Vazamento em Modelos Federados . . . . . 36

**3 Trabalhos Relacionados** **40**
3.1 Ecossistema de Conjuntos de Dados Textuais . . . . . 40
3.2 Privacidade e Vazamento de Dados em Aprendizado Federado . . . . . 51

**4 Geração de Conversas Multiturno e Multiparticipante** **58**
4.1 Visão Geral . . . . . 61
4.2 Criação de Corpus Conversacional Multiturno . . . . . 62
4.2.1 Estágio de Aquisição de Dados . . . . . 65
4.2.2 Estágio de Anonimização . . . . . 66
4.2.3 Estágio de Processamento . . . . . 67
4.2.4 Corpora Gerados . . . . . 72
4.3 Miner-XenForo . . . . . 73
4.3.1 Arquitetura . . . . . 74
4.3.2 Aquisição de Dados . . . . . 75
4.3.3 Anonimização . . . . . 77
4.3.4 Processamento . . . . . 79
4.4 Conjuntos de Dados Criados . . . . . 82
4.5 Avaliação Experimental . . . . . 84
4.5.1 Procedimentos . . . . . 85
4.5.2 Resultados Experimentais . . . . . 89

4.6 Conclusão do Capítulo . . . . . 91

**5 Geração Sintética de Conjuntos de Dados Conversacionais 92**
5.1 Visão Geral . . . . . 93
5.2 Metodologia . . . . . 94
5.3 Ferramenta . . . . . 104
5.4 Conjunto de Dados *Nutritious Food* . . . . . 107
5.5 Conclusão do Capítulo . . . . . 110

**6 Tratamento e Adaptação de Conjuntos de Dados Existentes 111**
6.1 Visão Geral . . . . . 112
6.2 BrWaC . . . . . 113
6.2.1 Metodologia . . . . . 116
6.2.2 Resultados . . . . . 117
6.3 MultiWOZ . . . . . 118
6.3.1 Metodologia . . . . . 120
6.3.2 Resultados . . . . . 124
6.4 Conclusão do Capítulo . . . . . 125

**7 Metodologia de Coleta Web Controlada e sua Instanciação no Sistema WackyWacky 128**
7.1 Visão Geral . . . . . 130
7.2 Metodologia de Coleta Web Controlada . . . . . 131
7.2.1 Fluxo da Metodologia . . . . . 133
7.2.2 Inicialização . . . . . 136
7.2.3 Exploração e Coleta de Páginas Web - Aquisição . . . . . 137
7.2.4 Controle de Execução . . . . . 139
7.2.5 Validação e Filtragem de Conteúdo . . . . . 142
7.2.6 Controle de Redundância . . . . . 144
7.2.7 Persistência e Representação dos Dados . . . . . 146
7.3 Implementação no Sistema WackyWacky . . . . . 149
7.3.1 Componentes Principais . . . . . 150
7.3.2 Modelo de Dados . . . . . 152
7.3.3 Execução e Concorrência . . . . . 154
7.3.4 Fluxo de Processamento . . . . . 156
7.3.5 Mecanismos de Controle . . . . . 158
7.4 Conjunto de Dados Gerado . . . . . 160
7.4.1 Características Gerais do Conjunto de Dados . . . . . 160
7.4.2 Estrutura e Distribuições da Coleta . . . . . 164
7.4.3 Qualidade Inicial dos Dados Textuais . . . . . 168
7.5 Limitações . . . . . 172
7.6 Conclusão do Capítulo . . . . . 173

**8 Pré-Treinamento Continuado e Avaliação em Português Brasileiro 175**
8.1 Desenho Experimental . . . . . 176
8.1.1 Modelos comparados . . . . . 176
8.1.2 Contrastes experimentais . . . . . 177
8.1.3 Variáveis controladas . . . . . 177
8.2 Dados Utilizados e Preparação . . . . . 178

8.3 Procedimento de Pré-Treinamento Continuado . . . 179
8.4 Protocolo de Avaliação . . . 180
8.4.1 Avaliação intrínseca de linguagem . . . 180
8.4.2 Classificação com representações congeladas . . . 181
8.4.3 Avaliações diagnósticas . . . 182
8.4.4 Análise estatística . . . 182
8.5 Resultados . . . 183
8.5.1 Avaliação intrínseca de linguagem . . . 183
8.5.2 Classificação de textos . . . 184
8.5.3 Resultados das avaliações diagnósticas . . . 185
8.5.4 Síntese dos contrastes . . . 186
8.6 Discussão e Limitações . . . 186

**9 Ataques de Extração e Defesa por Substituição Semântica em Aprendizado Federado 188**
9.1 Visão Geral . . . 189
9.2 Questões de Pesquisa e Hipóteses . . . 190
9.3 Modelo de Ameaça . . . 191
9.4 Mecanismo de Substituição Semântica . . . 194
9.5 Protocolo Experimental . . . 196
9.5.1 Modelo e Treinamento Federado . . . 197
9.5.2 Cliente Auxiliar e Amplificação do Padrão Canônico . . . 199
9.5.3 Métricas . . . 200
9.6 Resultados e Discussão . . . 204
9.7 Conclusão do Capítulo . . . 207

**10 Considerações Finais e Trabalhos Futuros 208**
10.1 Conclusões . . . 208
10.2 Produção Científica . . . 210
10.2.1 Journal Publicado . . . 210
10.2.2 Trabalhos em Conferências . . . 210
10.2.3 Cooperação em Outro Trabalho . . . 211
10.2.4 Premiação . . . 212
10.2.5 Metodologias Propostas . . . 212
10.2.6 Ferramentas Criadas . . . 212
10.2.7 Conjuntos de Dados Compartilhados . . . 213
10.2.8 Modelos Treinados . . . 213
10.3 Trabalhos Futuros . . . 213

**Referências Bibliográficas 215**

**Apêndice A WackyWacky PTBR Geral - Termos Semente 222**

**Anexo: Declaração de Uso de Inteligência Artificial Generativa 225**

# Capítulo 1

# Introdução

Nos últimos anos, LLMs têm ganhado cada vez mais espaço e se consolidado como uma das principais tecnologias de processamento de linguagem natural. Os resultados alcançados em tarefas complexas e multimodais, como tradução, classificação e geração de imagens, entre outros, estão diretamente relacionados à capacidade dos modelos e à disponibilidade de dados adequados aos problemas tratados [Bommasani et al., 2021, Min et al., 2023]. Ao mesmo tempo, a crescente preocupação com privacidade, governança de dados e exigências regulatórias tem alterado como pesquisadores e empresas lidam com a geração, o uso e o armazenamento desses dados. Nesse contexto, o Federated Learning (FL) é uma das técnicas que auxiliam na preservação da privacidade, uma vez que o treinamento ocorre de maneira descentralizada e os dados permanecem nos dispositivos dos usuários [McMahan et al., 2016]. A utilização desse tipo de técnica é particularmente útil em cenários nos quais a centralização é evitada, como em aplicações que lidam diretamente com dados sensíveis.

## 1.1 Contextualização e Motivação do Problema

Diversas áreas foram profundamente impactadas pelo advento da Deep Learning (DL), que serviu de base para avanços em diversas aplicações [LeCun et al., 2015]. Entre esses avanços, os LLMs têm ganhado cada vez mais destaque e alcançado resultados expressivos em tarefas como geração de texto, tradução e sumarização [Bommasani et al., 2021]. Também se destacam os modelos desenvolvidos em áreas correlatas que se beneficiam desses avanços, como os de imagem-para-texto, texto-para-vídeo, classificação e geração de texto. Em razão desses resultados e da versatilidade desses modelos, os LLMs têm se popularizado rapidamente tanto na indústria quanto na academia. Após a proposição da arquitetura Transformer [Vaswani et al., 2017] e a publicação dos modelos BERT [Devlin et al., 2019] e GPT-2 [Radford et al., 2019], as pesquisas relacionadas a LLMs passaram a ter cada vez mais relevância, resultando em um volume crescente de publicações [Xia et al., 2025, Min et al., 2023].

A ferramenta ChatGPT 5.5, da empresa OpenAI, foi utilizada para revisão gramatical e edição textual (inclusive da organização estrutural) e para suporte à geração ou otimização de códigos de programação. O conteúdo científico, as análises e as conclusões desta tese são de autoria e responsabilidade do pesquisador. A declaração de uso consta no anexo ao final do trabalho.

A Tabela 1.1 apresenta a contagem exploratória de registros bibliográficos do tipo *work* recuperados no OpenAlex em 8 de setembro de 2026. A busca por palavras-chave considerou, nos campos de título e resumo, os termos "large language model", "llm", "large language models" e "llms", com aplicação de um filtro de ano de publicação para cada contagem. Portanto, os valores correspondem ao número de registros retornados por essa estratégia, e não exclusivamente ao de artigos científicos revisados por pares. Trata-se de um levantamento exploratório, não de uma revisão sistemática da literatura; os resultados devem ser interpretados como indicador descritivo da evolução dos registros recuperados e estão sujeitos à cobertura da base e aos critérios da busca.

| **Ano** | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 |
|---|---|---|---|---|---|---|---|---|---|
| **Quantidade** | 656 | 693 | 1.212 | 1.012 | 1.055 | 1.242 | 1.396 | 1.438 | 1.638 |
| **Aumento** | – | 6% | 75% | -17% | 4% | 18% | 12% | 3% | 14% |
| | | | | | | | | | |
| **Ano** | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |
| **Quantidade** | 1.786 | 1.906 | 2.195 | 2.408 | 2.404 | 2.675 | 2.803 | 2.991 | 3.166 |
| **Aumento** | 9% | 7% | 15% | 10% | 0% | 11% | 5% | 7% | 6% |
| | | | | | | | | | |
| **Ano** | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 | 2026 |
| **Quantidade** | 3.699 | 4.547 | 5.582 | 6.927 | 8.494 | $\approx$ 24.050 | $\approx$ 60.640 | $\approx$ 121.700 | $\approx$ 221.600 |
| **Aumento** | 17% | 23% | 23% | 24% | 23% | 183% | 152% | 101% | 82%* |

Tabela 1.1: Registros bibliográficos recuperados pela consulta no OpenAlex, por ano. Os valores precedidos por $\approx$ são aproximados. Em 2026, contagem parcial de 1º de janeiro a 8 de setembro; * variação em relação ao total anual de 2025, sem anualização.

Os dados apresentados na Tabela 1.1 indicam uma tendência geral de crescimento das contagens anuais, com oscilações nos primeiros anos e aumentos mais acentuados a partir de 2023. Em relação ao ano anterior, os aumentos foram de aproximadamente 183% em 2023, 152% em 2024 e 101% em 2025. O valor de 2026 refere-se apenas aos registros desse ano recuperados até a data da coleta, cobrindo o período de 1º de janeiro a 8 de setembro de 2026. A variação de 82% compara essa contagem parcial com o total anual de registros de 2025, sem anualização; portanto, não representa uma taxa de crescimento entre dois anos completos.

A Figura 1.1 complementa essa análise ao comparar as contagens anuais de registros recuperados pela busca no OpenAlex com a quantidade de usuários do ChatGPT. Conforme discutido em [Chatterji et al., 2025], a adoção do ChatGPT ocorreu de forma acelerada desde seu lançamento em 2022, alcançando uma parcela significativa da população global em poucos anos. A simultaneidade entre o crescimento do uso do ChatGPT e o aumento da produção científica sobre LLMs indica uma clara e significativa convergência temporal entre difusão tecnológica e intensificação do interesse acadêmico. Esse reflexo também pode ser observado nos trabalhos científicos publicados, associações e conjuntos de palavras relacionadas com textos gerados por Inteligência Artificial (IA) tiveram um aumento de até 1500% [Kousha and Thelwall, 2026, Bick et al., 2024].

Os avanços obtidos estão diretamente relacionados à capacidade dos modelos e à disponibilidade de dados adequados para o treinamento [Zhu et al., 2025]. Nesse contexto, a utilização de dados alinhados ao domínio de aplicação no treinamento de LLMs favorece a obtenção de melhores desempenhos [Kaplan et al., 2020]. Apesar disso, diversos idiomas enfrentam problemas relacionados à escassez de dados adequados, sobretudo quando se

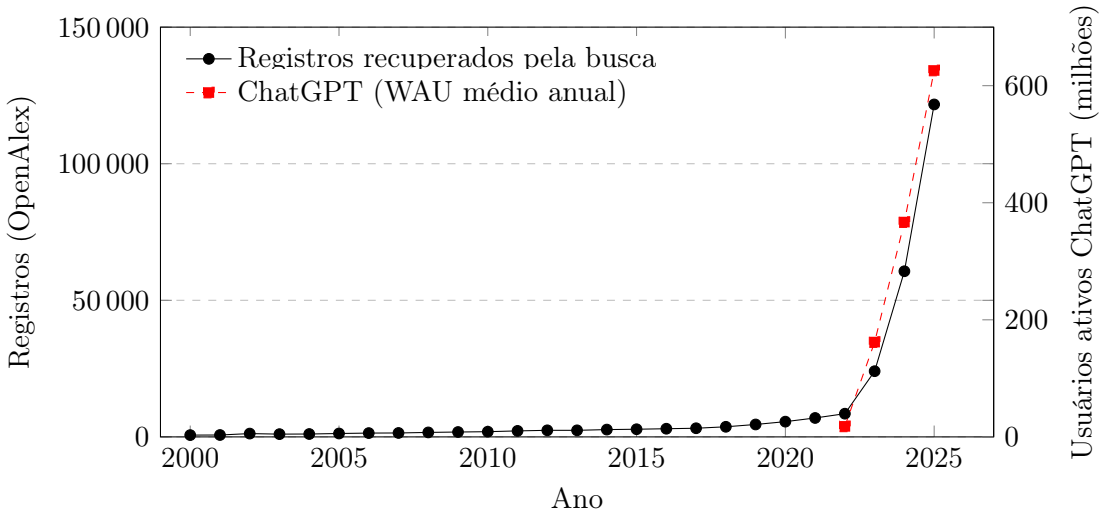



Figura 1.1: Evolução anual dos registros recuperados pela busca no OpenAlex e do número médio de usuários ativos semanais (WAU) do ChatGPT (anos completos até 2025).

considera o domínio específico da tarefa.

Além de disponibilizar e preparar conjuntos de dados, é necessário avaliar empiricamente seus efeitos sobre os modelos. O pré-treinamento continuado permite utilizar esses recursos para refinar um modelo previamente treinado, mas os ganhos observados podem decorrer tanto do treinamento adicional quanto da proximidade entre os dados utilizados e o domínio de avaliação.

Além de avaliar os efeitos dos dados sobre o desempenho dos modelos, é necessário considerar os riscos de exposição de informações sensíveis durante seu uso no treinamento. No treinamento federado, o servidor não tem acesso aos conjuntos de dados originais, o que contribui para mitigar o vazamento centralizado de informações [Kairouz and McMahan, 2021]. Apesar disso, os modelos locais ainda têm acesso aos dados originais em cada dispositivo participante do treinamento, o que pode levar ao vazamento de informações sensíveis em cenários específicos durante a geração das saídas do modelo [Zhu et al., 2019].

Mesmo com as melhorias em segurança proporcionadas pelo FL, ainda existem formas de comprometer o treinamento e obter dados sensíveis dos participantes. Dispositivos maliciosos que participam do treinamento podem gerar modelos locais comprometidos que, ao serem agregados, resultam no vazamento de dados sensíveis [Bagdasaryan et al., 2020, Xie et al., 2020].

Uma técnica que pode ser utilizada para esse tipo de comprometimento é a memorização baseada em overfitting [Carlini et al., 2019, Carlini et al., 2021]. Nessa técnica, o modelo é treinado para se ajustar excessivamente a padrões específicos, frequentemente de maneira pouco perceptível. O objetivo é induzir a geração de dados sensíveis na saída do modelo quando são utilizadas entradas que fizeram parte desse ajuste excessivo. Esse comportamento malicioso pode permanecer oculto na maioria do tempo, o que dificulta sua detecção.

Em aprendizado federado, a participação de mais clientes benignos pode diluir a influência de uma atualização maliciosa [Bagdasaryan et al., 2020, Xie et al., 2020]. Ainda

assim, ataques de *poisoning* com gatilhos textuais podem reforçar padrões no modelo global e favorecer sua reprodução posterior [Hu et al., 2025].

Nesse contexto, os principais desafios investigados nesta tese abrangem a disponibilidade de dados em português brasileiro, a avaliação de seus efeitos sobre os modelos e a mitigação de riscos à privacidade em treinamento federado, conforme detalhado a seguir:

- **Falta de dados para LLMs em português brasileiro:** a qualidade e os resultados obtidos com LLMs estão diretamente relacionados aos dados utilizados durante o treinamento. Apesar disso, a maioria dos dados necessários para esse tipo de modelagem está disponível apenas em inglês, o que limita a evolução de soluções em português brasileiro [Souza et al., 2020, Wagner Filho et al., 2018].

- **Necessidade de avaliação controlada dos efeitos da composição dos dados:** a disponibilidade de conjuntos de dados não demonstra, por si só, seus benefícios para o desempenho dos modelos. É necessário comparar o modelo original com versões refinadas, mantendo entre estas o mesmo volume de tokens, orçamento computacional e hiperparâmetros, para distinguir os efeitos do treinamento adicional daqueles associados à proximidade de domínio dos dados.

- **Equilíbrio entre privacidade e utilidade com Differential Privacy (DP):** em FL, mecanismos de DP podem ser aplicados durante o treinamento local dos clientes ou na agregação das atualizações, conforme o protocolo adotado. A limitação de contribuições e a adição de ruído podem reduzir o desempenho dos modelos, dependendo da tarefa, dos dados e do orçamento de privacidade. Assim, é necessário avaliar conjuntamente as garantias oferecidas, as hipóteses de confiança e o impacto na utilidade.

- **Poucos trabalhos sobre ataques de memorização baseados em overfitting:** a literatura tende a concentrar as pesquisas em inferência de pertencimento ou reconstrução estatística, deixando temas como a memorização baseada em overfitting, realizada por participantes maliciosos, com pouca exploração [Carlini et al., 2019, Mireshghallah et al., 2022, Zeng et al., 2024].

- **Poucas técnicas para melhorar a privacidade em LLMs com treinamento federado:** muitas das abordagens existentes para a proteção de dados sensíveis dependem de mecanismos gerais de defesa, deixando em aberto a mitigação do vazamento durante o treinamento local e a geração de saídas do modelo. Esse cenário evidencia a necessidade de técnicas específicas para reduzir a exposição de informações sensíveis em FL aplicado a LLMs [Piccialli et al., 2025].

- **Pouca exploração de ataques direcionados a tipos específicos de dados sensíveis:** a maioria dos trabalhos trata o vazamento de forma agregada, sem diferenciar tipos de informação sensível. No entanto, ataques podem ser direcionados a entidades específicas, como identificadores pessoais ou *tokens*, e a ausência dessa análise limita a compreensão do risco em aplicações práticas [Carlini et al., 2021, Hu et al., 2025, Szep et al., 2026].

- **Pouca literatura sobre transformações semânticas como defesa de privacidade:** estratégias como substituição de entidades, mascaramento ou geração sintética de dados ainda são pouco exploradas como mecanismos de defesa em FL. Embora possam reduzir a exposição de informações sensíveis, sua eficácia contra ataques adversariais permanece pouco investigada [Zhang et al., 2025, Sánchez and Batet, 2017].

Dessa forma, esta tese aborda a escassez de dados textuais adequados em português brasileiro para o treinamento de LLMs [Souza et al., 2020, Wagner Filho et al., 2018], investiga os efeitos de sua utilização no pré-treinamento continuado e examina riscos de vazamento de informações sensíveis em treinamento federado.

Para distinguir os efeitos do treinamento adicional daqueles associados à proximidade de domínio, este trabalho compara o modelo original com duas versões refinadas a partir do mesmo checkpoint, utilizando composições de dados distintas, mas mantendo iguais o volume de tokens e as condições de treinamento entre as versões refinadas. Essa comparação permite investigar a contribuição do refinamento e o efeito adicional da proximidade de domínio sobre o desempenho.

No eixo de privacidade, este trabalho considera o treinamento federado como cenário para investigar riscos de vazamento e mecanismos de mitigação [McMahan et al., 2016].

Para avaliar esses riscos, o protocolo experimental modela um cenário federado composto por clientes-vítima, que detêm conversas protegidas contendo registros pessoais sintéticos, e por um cliente auxiliar, que pode atuar de forma benigna ou adversária. Neste trabalho, o cliente auxiliar utiliza exclusivamente dados sintéticos próprios e não possui acesso aos nomes, aos registros ou às atualizações individuais das vítimas. Na variante adversária, a apresentação das conversas protegidas e o escopo da função de perda são modificados para investigar se essa intervenção aumenta a reprodução dos registros das vítimas pelo modelo global. A reprodução é mensurada por um avaliador confiável separado.

Esta tese articula a criação, a adaptação, a coleta e a curadoria de dados para LLMs em português brasileiro com a avaliação de seu uso no pré-treinamento continuado e a investigação de riscos de privacidade em aprendizado federado. A hipótese central é que a adequação dos dados ao domínio pode favorecer o desempenho dos modelos, enquanto a transformação local de informações sensíveis pode mitigar seu vazamento durante o treinamento federado. Essas dimensões são investigadas por meio de estudos complementares, com protocolos específicos de avaliação.

## 1.2 Objetivos

O objetivo geral desta tese é desenvolver e avaliar estratégias para ampliar a disponibilidade de dados adequados em português brasileiro, investigar sua contribuição para o refinamento de modelos de linguagem e mitigar riscos de vazamento em treinamento federado. Para isso, a pesquisa contempla a construção e a adaptação de conjuntos de dados, experimentos controlados de pré-treinamento continuado e a avaliação de ataques

e mecanismos de proteção à privacidade. Esses objetivos são desdobrados nas seguintes perguntas de pesquisa:

- **Pergunta 1:** Como bases de dados já existentes em português brasileiro, compostas por interações comunicativas entre múltiplos participantes, podem ser processadas, segmentadas e estruturadas de modo a constituírem diálogos multiturnos adequados ao treinamento de Large Language Models?

  Esta pergunta trata da criação de conjuntos de dados, especialmente conversacionais e compostos por diálogos multiturnos, para ampliar a disponibilidade de recursos em contextos com escassez de dados. Esses temas são abordados no capítulo 4.

- **Pergunta 2:** Como podem ser concebidos, gerados e estruturados dados conversacionais sintéticos em português brasileiro, compostos por múltiplos turnos e participantes, de forma controlada e metodologicamente consistente, para o treinamento de Large Language Models?

  Esta pergunta trata da geração sintética de dados como forma de mitigar, pontualmente, sua escassez e melhorar sua estrutura de geração e validação. Esses temas são abordados no capítulo 5.

- **Pergunta 3:** Como conjuntos de dados já consolidados na literatura podem ser adaptados para viabilizar seu uso eficaz no treinamento de Large Language Models?

  Esta pergunta trata da adaptação de conjuntos de dados existentes, por meio de tradução, reestruturação e adequação aos requisitos de treinamento de LLMs. Esses temas são abordados no Capítulo 6.

- **Pergunta 4:** Como coletar, filtrar, normalizar e estruturar sistematicamente dados textuais provenientes de websites em português brasileiro para gerar conjuntos de dados escaláveis e diversificados para o treinamento de Large Language Models?

  Esta pergunta trata da proposição e validação de uma metodologia para coletar, filtrar, normalizar e estruturar dados textuais provenientes de websites em português brasileiro, bem como de sua implementação em uma ferramenta. Esses temas são abordados no capítulo 7.

- **Pergunta 5:** Como o pré-treinamento continuado e a proximidade de domínio dos dados afetam o desempenho de um modelo de linguagem em português brasileiro quando o volume de dados, o orçamento computacional e os demais hiperparâmetros são mantidos constantes?

  Esta pergunta investiga os efeitos do pré-treinamento continuado e da proximidade de domínio dos dados por meio da comparação entre um modelo inicial e duas versões refinadas. As duas variantes de refinamento utilizam o mesmo volume de tokens, orçamento computacional equivalente e os mesmos hiperparâmetros, permitindo avaliar o efeito da composição dos dados. Esses temas são abordados no capítulo 8.

- **Pergunta 6:** Em que medida a substituição semântica rotativa reduz a reprodução de dados pessoais sintéticos durante o treinamento federado de um modelo de linguagem em português brasileiro, em comparação com a ausência de defesa, na presença de um cliente auxiliar benigno ou adversário?

  Esta pergunta trata da compreensão do problema de memorização induzida por overfitting em treinamento federado e do desenvolvimento de técnicas para mitigar esse problema em LLMs. Esses temas são abordados no capítulo 9.

As quatro primeiras perguntas concentram-se na produção e na preparação dos dados; a quinta examina os efeitos de sua utilização no refinamento de modelos; e a sexta investiga o vazamento de informações sensíveis e sua mitigação em treinamento federado. Em conjunto, esses estudos articulam a disponibilidade de recursos, a avaliação da utilidade dos modelos e a análise dos riscos à privacidade.

## 1.3 Contribuições da Tese

Ao longo desta pesquisa, foram produzidas contribuições metodológicas, técnicas, experimentais e de dados voltadas à ampliação dos recursos em português brasileiro, à avaliação do refinamento de modelos e à mitigação de vazamentos no treinamento federado de LLMs. Essas contribuições abrangem metodologias e ferramentas de preparação de dados, experimentos controlados de pré-treinamento continuado e cenários de ataque e defesa em aprendizado federado. As principais contribuições desta tese são:

- **Metodologia para Criação de Conjuntos de Dados Conversacionais:** Definimos um processo sistemático para a coleta, a organização e a validação de diálogos multiturnos e multiparticipantes, com foco na qualidade e na diversidade das interações (Seção 4.2).

- **Ferramenta que Implementa a Metodologia para Criação de Conjuntos de Dados Conversacionais:** Desenvolvemos uma ferramenta que automatiza a aplicação da metodologia proposta, permitindo a construção reprodutível de conjuntos de dados conversacionais a partir de fontes reais (Seção 4.3).

- **Conjunto de Dados Adrenaline:** Elaboramos um conjunto de dados conversacional extraído de fóruns temáticos com foco em tecnologia, estruturado em múltiplos turnos e adequado ao treinamento de LLMs em português brasileiro (Seção 4.4).

- **Conjunto de Dados OuterSpace:** Produzimos um conjunto de dados conversacional composto por diálogos multiparticipantes a partir de comunidades online com foco em tecnologia, jogos e assuntos gerais, ampliando a variedade de tópicos e estilos de interação (Seção 4.4).

- **Metodologia para Criação Sintética de Conjuntos de Dados Conversacionais:** Propusemos uma metodologia para a geração sintética de diálogos, orientada por roteiros, para suprir lacunas de dados em domínios específicos (Capítulo 5).

- **Conjunto de Dados *Nutritious Food*:** Disponibilizamos um conjunto de dados sintético focado em perguntas e respostas sobre informações nutricionais, útil para avaliação e treinamento em domínio restrito (Seção 5.4).

- **Filtragem de Ruídos do Conjunto de Dados BrWaC:** Realizamos um processo de filtragem e normalização para remover conteúdo ofensivo, duplicações e trechos de baixa qualidade, tornando o conjunto de dados adequado ao treinamento (Seção 6.2).

- **Tradução do Conjunto de Dados MultiWOZ:** Traduzimos e adaptamos o MultiWOZ para o português brasileiro, preservando suas anotações e sua estrutura semântica original (Seção 6.3).

- **Metodologia para Criação de Conjuntos de Dados a Partir da Web:** Propusemos uma metodologia de crawling controlado, com geração de URLs, exploração recursiva e filtragem progressiva, incorporando controle de profundidade, limitação por domínio, deduplicação e filtragem linguística (Seção 7.2).

- **Ferramenta que Implementa a Metodologia para Criação de Conjuntos de Dados a partir da Web:** Desenvolvemos uma ferramenta baseada em crawling distribuído, com múltiplos workers e estado global em banco de dados, permitindo coleta, filtragem e armazenamento estruturado de dados. A ferramenta, denominada WackyWacky, promove escalabilidade e reprodutibilidade (Seção 7.3).

- **Conjuntos de Dados Textuais Coletados a partir de Websites:** Geramos conjuntos de dados em larga escala a partir de websites reais, com filtragem, normalização e deduplicação, organizados com metadados estruturais para uso em modelos de linguagem (Seção 7.4).

- **Pré-Treinamento Continuado e Avaliação Controlada em Português Brasileiro:** Investigamos os efeitos do pré-treinamento continuado e da proximidade de domínio comparando os modelos Base, Geral e Fórum/Tec, com orçamento de tokens e condições de treinamento equivalentes entre os dois modelos refinados e pareamento das sequências compartilhadas. O estudo produziu dois modelos refinados e um protocolo reprodutível de avaliação de linguagem, classificação com representações congeladas e análises diagnósticas (Capítulo 8).

- **Substituição Semântica Rotativa de Dados Pessoais em FL:** Implementamos um mecanismo local que substitui os nove campos protegidos, inclusive o nome, por valores sintéticos do mesmo tipo e formato antes da tokenização. As substituições permanecem consistentes durante o treinamento local de cada rodada e são renovadas na rodada seguinte (Seção 9.4).

- **Avaliação Comparativa de Defesas contra Reprodução de Dados em FL:** Avaliamos a substituição semântica rotativa em comparação com a ausência de defesa e com DP-AdamW no treinamento federado do Tucano 2 refinado. O protocolo utiliza dados pessoais sintéticos e clientes auxiliares benignos ou adversários, com

auditoria por um avaliador confiável separado, permitindo analisar conjuntamente reprodução dos valores originais e utilidade sintética (Seção 9.5).

- **Consolidação de Resultados Científicos:** Parte das contribuições desta tese resultou em trabalhos publicados ou submetidos a um periódico e a três conferências, incluindo um trabalho premiado com *best paper*.

## 1.4 Estrutura da Tese

O restante desta tese está organizado da seguinte forma: O Capítulo 2 apresenta a fundamentação teórica utilizada no desenvolvimento das propostas, ferramentas e soluções desta tese. O Capítulo 3 apresenta os trabalhos relacionados, bem como a discussão das abordagens existentes na literatura. Complementarmente, são discutidas as limitações e as vantagens de cada trabalho. O Capítulo 4 discute o problema da falta de dados adequados em português brasileiro e como ele foi abordado neste trabalho, apresentando uma metodologia, uma ferramenta e dois conjuntos de dados desenvolvidos para auxiliar na mitigação dessa limitação. O Capítulo 5 apresenta uma metodologia e uma ferramenta orientadas ao *crowdsourcing* para a criação controlada de conjuntos de dados conversacionais baseados em roteiros. A validação é conduzida com o conjunto *Nutritious Food*, criado pela instanciação direta de cinco roteiros, com um diálogo por roteiro, e inclui a análise do aumento combinatório de dados por permutação de turnos associados a rótulos equivalentes. O Capítulo 6 descreve estratégias para mitigar parcialmente o problema da escassez de dados específicos por meio da adaptação de conjuntos de dados existentes. Nesse contexto, um conjunto de dados foi traduzido para o português brasileiro, enquanto outro passou por um processo de filtragem de dados de baixa qualidade. O Capítulo 7 apresenta uma metodologia e uma ferramenta para a criação recursiva de conjuntos de dados a partir de páginas da web, com diversos filtros e regras voltados à melhoria da qualidade final dos dados gerados. O Capítulo 8 apresenta o experimento controlado de pré-treinamento continuado em português brasileiro, descrevendo a composição dos dados de cada variante, o procedimento de treinamento e as avaliações empregadas para investigar os efeitos do refinamento e da proximidade de domínio dos dados. O Capítulo 9 investiga a reprodução de dados pessoais em um cenário controlado de treinamento federado de LLMs em português brasileiro, considerando a participação de um cliente auxiliar benigno ou adversário. O capítulo avalia a substituição semântica rotativa em comparação com a ausência de defesa e com DP-AdamW, analisando a redução da exposição dos dados originais e a preservação da utilidade sintética do modelo. O Capítulo 10 apresenta a conclusão e discute os desenvolvimentos futuros dos trabalhos apresentados nesta tese.

# Capítulo 2

# Referencial Teórico

Este capítulo apresenta os conceitos fundamentais que embasam o desenvolvimento das propostas desta tese. Seu objetivo é contextualizar os principais elementos teóricos relacionados ao processamento de linguagem natural, aos modelos de linguagem modernos e ao aprendizado federado, bem como os desafios de privacidade associados a esses sistemas. Esses conceitos são essenciais para a compreensão das abordagens, metodologias, ferramentas e experimentos apresentados nos capítulos seguintes.

O restante deste capítulo está organizado da seguinte forma. A Seção 2.1, *Modelos de Linguagem*, apresenta os principais conceitos relacionados a modelos de linguagem. Em seguida, a Seção 2.2, *Conjuntos de Dados para LLMs*, discute a estrutura e a geração de dados conversacionais. A Seção 2.3, *Aprendizado Federado*, aborda os fundamentos do aprendizado federado e suas implicações para privacidade. Por fim, a Seção 2.4, *Privacidade e Vazamento em Modelos Federados*, descreve ataques de vazamento de informação e estratégias de mitigação.

## 2.1 Modelos de Linguagem

No contexto de Natural Language Processing (NLP), modelos de linguagem constituem uma classe de modelos voltada à estimação de distribuições de probabilidade sobre sequências textuais. Essa capacidade pode apoiar tarefas como geração de texto, tradução, sumarização e classificação, embora cada aplicação envolva formulações ou adaptações específicas. Em suas abordagens iniciais, muitos sistemas dependiam de regras manuais, engenharia de atributos e modelos estatísticos com capacidade limitada de representar contexto, limitando a generalização semântica e o desempenho em diversas tarefas.

Como contraponto às abordagens modernas de modelagem contextual, a Tabela 2.1 apresenta uma representação textual baseada em *Bag of Words* [Galke and Scherp, 2022]. Nessa abordagem, o texto é convertido em um vetor de frequências de palavras, frequentemente utilizado como entrada para modelos tradicionais, como regressão logística, *support vector machines* e classificadores Naive Bayes. Apesar de sua simplicidade, a abordagem não preserva a ordem das palavras nem representa adequadamente relações semânticas e dependências contextuais.

| Frase | o | gato | observa | outro | cachorro | perseguiu | peixe |
|---|---|---|---|---|---|---|---|
| **O gato observa outro gato** | 1 | 2 | 1 | 1 | 0 | 0 | 0 |
| **O cachorro perseguiu o gato** | 2 | 1 | 0 | 0 | 1 | 1 | 0 |
| **O gato perseguiu o cachorro** | 2 | 1 | 0 | 0 | 1 | 1 | 0 |

Tabela 2.1: Representação das frases em formato *bag of words*.

A tabela ilustra duas limitações centrais de representações *Bag of Words*. Primeiro, as frases "O cachorro perseguiu o gato" e "O gato perseguiu o cachorro" recebem o mesmo vetor, embora invertam as relações entre agente e paciente da ação. Segundo, a coluna referente ao termo "peixe" permanece zerada em todos os exemplos, indicando a esparsidade produzida por vocabulários com termos ausentes nas instâncias analisadas.

As técnicas para NLP evoluíram rapidamente nos últimos anos, principalmente a partir de 2017, com a proposta da arquitetura Transformer [Vaswani et al., 2017], que pavimentou o caminho para modelos cada vez mais eficientes. Nesse contexto, os modelos de linguagem de grande porte (LLMs) se tornaram centrais para o avanço da área, pois aprendem regularidades linguísticas em larga escala e podem ser aplicados, com pouca ou nenhuma adaptação adicional, a tarefas com diferentes níveis de complexidade [Bommasani et al., 2021, Radford et al., 2019].

Em sua formulação estatística clássica, um modelo de linguagem pode ser descrito como a probabilidade condicional da próxima palavra, dado o histórico anterior. Para uma sequência $w_1, \ldots, w_T$, a probabilidade conjunta pode ser fatorada como

$$\hat{P}(w_1^T) = \prod_{t=1}^{T} \hat{P}(w_t \mid w_1^{t-1}) \tag{2.1}$$

em que $w_t$ denota a palavra na posição $t$ e $w_i^j = (w_i, \ldots, w_j)$ representa uma subsequência [Bengio et al., 2003]. Esse tipo de modelagem é amplamente utilizado em aplicações como reconhecimento de fala, tradução automática e recuperação de informação, pois permite estimar quão provável é uma sequência em língua natural [Min et al., 2023].

Para tornar esse problema tratável, abordagens clássicas como os modelos de $n$-gramas assumem que a próxima palavra depende principalmente das $n-1$ palavras mais recentes. Assim, em vez de considerar todo o histórico, a aproximação utilizada é

$$\hat{P}(w_t \mid w_1^{t-1}) \approx \hat{P}(w_t \mid w_{t-n+1}^{t-1}) \tag{2.2}$$

o que reduz a complexidade do modelo, mas também limita sua capacidade de capturar dependências de longo alcance e introduz problemas de esparsidade quando novas combinações de palavras não aparecem no corpus de treinamento. Modelos gerativos baseados em decodificadores, como a família GPT, preservam essa lógica autorregressiva, mas a aplicam sobre *tokens* representados em espaços vetoriais contínuos. Outros modelos, como BERT, adotam objetivos bidirecionais de reconstrução.

Nesse contexto, o texto é segmentado em tokens, e cada unidade recebe um *embedding*, isto é, um vetor que busca representar propriedades sintáticas e semânticas em um espaço contínuo hiperdimensional. Diferentemente das codificações esparsas, essas representa-

ções permitem que tokens semanticamente semelhantes ocupem regiões próximas nesse hiperespaço, favorecendo a generalização do modelo.

A arquitetura Transformer organiza esse processamento em camadas compostas principalmente por mecanismos de autoatenção e redes neurais *feed-forward*. Essa estrutura favorece a paralelização do treinamento e amplia a capacidade de modelar dependências de longo alcance de forma mais eficiente do que arquiteturas estritamente sequenciais. A partir dessas camadas, o modelo produz, em cada posição, uma representação que pode ser usada para estimar a distribuição de probabilidade do próximo token. Em modelos de linguagem autorregressivos, o treinamento costuma otimizar a perda de previsão do próximo token:

$$\mathcal{L}_{\mathrm{LM}} = -\sum_{t=1}^{T} \log P_\theta(x_t \mid x_{<t}) \tag{2.3}$$

em que $x_{<t}$ denota os tokens anteriores à posição $t$ [Radford et al., 2018]. Desse modo, modelos de linguagem autorregressivos de grande porte podem ser entendidos como uma extensão em larga escala da modelagem clássica de linguagem, em que o objetivo permanece estimar a distribuição do próximo token a partir do contexto anterior. Em outras famílias de modelos, como arquiteturas bidirecionais, o pré-treinamento pode envolver objetivos distintos, como a reconstrução de tokens mascarados [Vaswani et al., 2017, Radford et al., 2019, Bommasani et al., 2021, Bengio et al., 2003].

Para detalhar o papel da autoatenção, a Figura 2.1 ilustra, de forma esquemática, como um token pode atribuir pesos distintos a outros tokens da sequência durante a construção de sua representação contextual. Isso favorece a paralelização do treinamento e melhora o tratamento de dependências de longo alcance, dois fatores decisivos para o desempenho observado em tarefas complexas de linguagem [Vaswani et al., 2017, Min et al., 2023].

A Figura 2.1 apresenta um exemplo esquemático do mecanismo de atenção usando a frase “O cachorro assustou o gato porque ele latiu alto”. Nesse caso, a interpretação associa “ele” a “cachorro”, e pesos maiores nesse token indicam maior contribuição para a representação contextual do pronome. Esses pesos, contudo, têm função ilustrativa e não constituem, por si só, uma explicação causal da decisão do modelo.

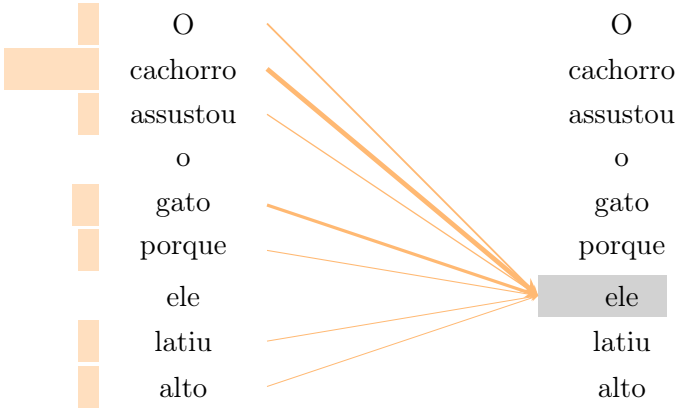



Figura 2.1: Mecanismo de atenção

A Figura 2.2 resume o fluxo de processamento de um modelo baseado em Transfor-

mer. Primeiro, os tokens de entrada representam a sequência textual após a segmentação do texto original. Em seguida, cada token é convertido em um *embedding*, isto é, uma representação vetorial que serve de base para o processamento subsequente. Essas representações são então refinadas por camadas Transformer com o mecanismo de atenção, responsáveis por integrar informações de diferentes partes da sequência e produzir estados contextuais. Por fim, esses estados contextualizados são utilizados para gerar a predição ou a saída do modelo, como o próximo token, uma classe ou uma resposta em linguagem natural.

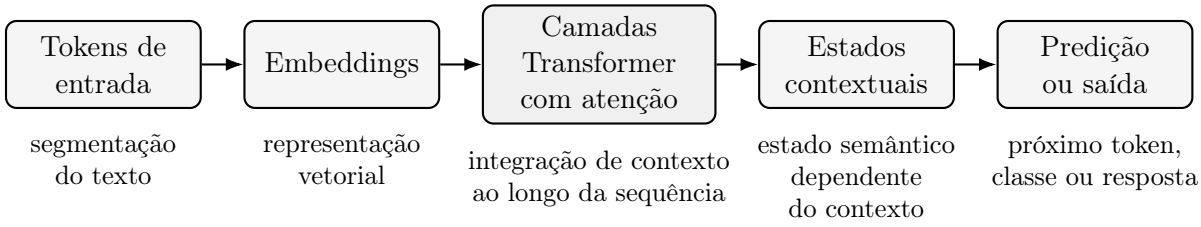



Figura 2.2: Fluxo conceitual simplificado de modelos baseados em Transformer.

Sobre essa base, os LLMs se consolidaram e, em geral, passaram a ser caracterizados pela combinação entre arquitetura Transformer, grande volume de dados, elevado número de parâmetros e treinamento em larga escala. Mais do que o tamanho isoladamente, ganha destaque a capacidade desses modelos de reutilizar padrões linguísticos, estatísticos e contextuais aprendidos durante o pré-treinamento em diferentes tarefas. Modelos como BERT e GPT ilustram esse movimento ao adotar, respectivamente, objetivos bidirecionais e autorregressivos em regimes de pré-treinamento de larga escala [Devlin et al., 2019, Radford et al., 2019, Bommasani et al., 2021, Kaplan et al., 2020].

Essa discussão é particularmente relevante quando o idioma e o domínio de aplicação diferem daqueles predominantes no treinamento original do modelo. Nesses cenários, modelos treinados majoritariamente em inglês ou em conjuntos de dados generalistas podem apresentar limitações em português brasileiro, seja por cobertura lexical insuficiente, diferenças sintáticas ou menor familiaridade com terminologias especializadas. A existência de iniciativas voltadas ao português, como o BERTimbau, reforça que a adequação linguística e temática dos dados influencia diretamente o comportamento e o desempenho desses modelos [Souza et al., 2020, Min et al., 2023].

## 2.2 Conjuntos de Dados para LLMs

O desempenho de LLMs depende de diversos fatores, como o regime de treinamento, a estrutura e a adequação dos dados utilizados em sua construção e adaptação. Em contextos de escassez de recursos em determinado idioma, domínio ou tipo de tarefa, a obtenção de conjuntos de dados adequados constitui uma condição crítica para o treinamento desse tipo de modelo, pois pode limitar seu desempenho [Bommasani et al., 2021, Min et al., 2023]. Nesse cenário, compreender a natureza dos recursos disponíveis e o modo como esses dados são preparados é fundamental para analisar como diferentes estratégias de criação e adaptação de conjuntos de dados mitigam essas limitações.

Entre as principais estão o nível de processamento dos dados, o grau de estruturação explícita e o tipo de uso previsto. Essas dimensões não são excludentes, com um mesmo recurso sendo processado, anotado e organizado em formato conversacional ou instrucional. Essa distinção é relevante porque cada dimensão impõe exigências próprias de preparação, validação e adequação à tarefa. A Tabela 2.2 sintetiza essas diferenças.

Na distinção por processamento, o corpus pode ser classificado como bruto ou processado. O corpus bruto corresponde a uma coleção de textos ainda não preparada para uso direto, enquanto o corpus processado já passou por etapas como limpeza, remoção de duplicatas, normalização e reorganização do conteúdo. Corpora derivados de coleta web em larga escala ilustram essa diferença, pois costumam reunir materiais heterogêneos cuja utilidade prática depende de preparação adicional antes do uso consistente em aplicações de linguagem [Baroni et al., 2009, Wagner Filho et al., 2018].

Conjuntos de dados anotados associam rótulos, marcações ou metadados ao conteúdo textual. Por isso, são particularmente adequados a tarefas supervisionadas, nas quais o modelo aprende a partir de exemplos explicitamente estruturados.

Conjuntos instrucionais organizam pares de instrução e resposta, com ênfase no seguimento de comandos e em restrições de formato. Essa dimensão é especialmente útil em etapas de ajuste fino voltadas ao comportamento responsivo e ao alinhamento com as instruções do usuário.

Conjuntos conversacionais preservam a dinâmica de interação entre participantes, incluindo dependências entre turnos e o contexto acumulado ao longo das iterações de um diálogo [Min et al., 2023, Budzianowski et al., 2020]. Sua principal particularidade está na necessidade de representar não apenas o conteúdo de cada mensagem, mas também as relações estruturais que sustentam a interpretação da conversa.

Além das dimensões apresentadas, a utilidade de um conjunto de dados depende de requisitos adicionais de qualidade. Mesmo quando apresenta estrutura compatível com a tarefa, o conjunto pode conter ruído textual, duplicação de conteúdo, linguagem ofensiva, metadados incompletos, desalinhamento entre texto e anotação, baixa diversidade linguística, cobertura insuficiente de domínio e inconsistências estruturais. Em LLMs, esses problemas podem introduzir vieses, reduzir a robustez da aprendizagem e comprometer a validade da avaliação, sobretudo quando o desempenho observado passa a refletir limitações, ruídos ou lacunas do próprio conjunto utilizado [Bommasani et al., 2021, Min et al., 2023].

Por essa razão, a preparação de dados pode envolver etapas de curadoria, filtragem, validação e documentação. A utilidade de um conjunto de dados não decorre apenas de seu volume, mas também de sua representatividade, cobertura linguística e aderência ao domínio específico de uso do modelo. Esse ponto é particularmente relevante em português brasileiro e em domínios especializados, nos quais a disponibilidade de recursos costuma exigir esforços adicionais de construção, limpeza e reorganização [Sanches et al., 2022a, Souza et al., 2020]. Desse modo, qualidade dos dados abrange ausência de ruído, adequação ao cenário de uso e rastreabilidade das transformações aplicadas ao longo do processo.

No caso de dados conversacionais, a complexidade aumenta porque a unidade de interpretação deixa de ser o texto isolado e passa a incluir relações entre mensagens. Conceitos

| Dimensão e Categoria | Estrutura típica dos textos | Uso em LLMs | Principal limitação |
|---|---|---|---|
| Processamento Corpus bruto | Heterogêneos Sem padronização | Pré-treinamento em larga escala após preparação adicional | Elevado ruído e baixa adequação direta à tarefa |
| Processamento Corpus processado | Limpos, normalizados e organizados | Pré-treinamento, adaptação de domínio e análise linguística | Pode não conter supervisão ou estrutura interacional |
| Estrutura de uso Dados anotados | Rótulos, marcações ou metadados | Treinamento e avaliação supervisionados | Alto custo de anotação e cobertura limitada |
| Estrutura de uso Dados instrucionais | Pares de instrução e resposta | Ajuste fino voltado ao seguimento de comandos | Sensível à qualidade do alinhamento entre instrução e saída |
| Estrutura de uso Dados conversacionais | Turnos, participantes e histórico de interação | Modelagem de diálogo, resposta contextual e interação multiturno | Maior complexidade estrutural e menor disponibilidade |

Tabela 2.2: Possíveis dimensões de organização de recursos e adequação a diferentes usos em LLMs.

como turno, diálogo, participante, histórico conversacional, contexto e resposta direta se tornam centrais, uma vez que o significado de uma intervenção depende das mensagens anteriores e do papel que ela ocupa na interação. Em uma conversa, a mesma sequência lexical pode assumir funções distintas conforme o encadeamento discursivo, o estado do diálogo e os participantes envolvidos [Budzianowski et al., 2020].

Essa dependência estrutural se intensifica em conversas multiturno, multiparticipantes e ramificadas. Em interações reais, uma mensagem pode responder diretamente a outra, retomar um tópico anterior ou coexistir com subdiscussões paralelas, o que torna insuficiente representar o diálogo apenas como uma lista linear de sentenças. Nesses casos, preservar a relação entre mensagens e a posição de cada turno é essencial para reconstruir o fluxo interacional, o contexto local e o contexto global.

Dessa forma, a interpretação de uma mensagem no turno $t$ pode ser representada por $I(m_t) = f\big(m_t, (m_1, \ldots, m_{t-1}), P, R_t, C_t\big)$, em que $I(m_t)$ corresponde à interpretação da mensagem atual $m_t$, $(m_1, \ldots, m_{t-1})$ representa o histórico conversacional anterior, $P$ denota os participantes envolvidos, $R_t$ indica o papel discursivo desempenhado pela mensagem no diálogo e $C_t$ expressa o estado contextual da interação naquele momento. Essa

formulação explicita que o sentido de uma intervenção não decorre apenas de seu conteúdo isolado, mas da combinação entre o enunciado corrente, o histórico, os participantes, o papel discursivo e o estado contextual da interação, com a função $f$ representando conceitualmente essa dependência.

A Figura 2.3 ilustra essa organização de forma simplificada e evidencia por que dados conversacionais exigem metodologias específicas de extração, modelagem e validação.

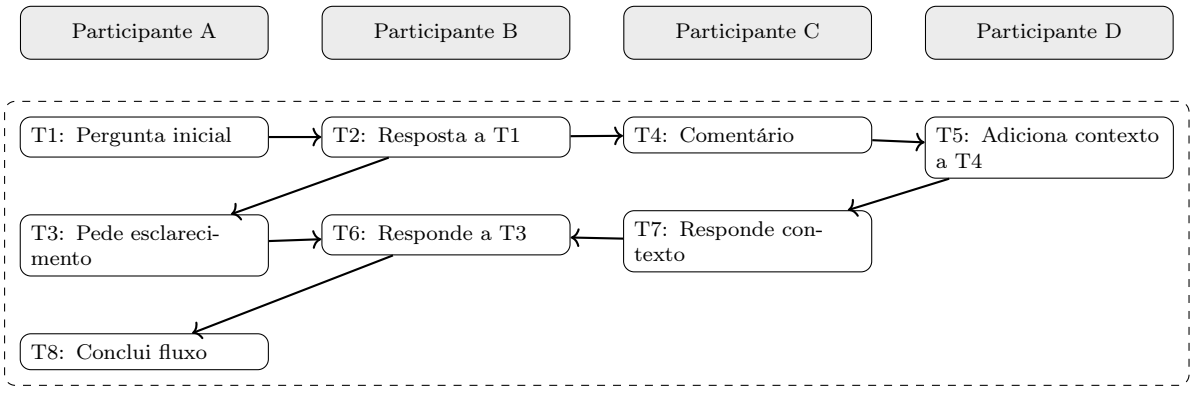



Figura 2.3: Estrutura simplificada de uma conversa multiturno, multiparticipante e ramificada.

A partir dessas características estruturais, os conjuntos de dados para LLMs podem ser obtidos de múltiplas fontes e por diferentes estratégias. Entre as fontes mais relevantes estão interações humanas reais, como fóruns, chats, atendimentos e outras plataformas digitais. Esses dados tendem a apresentar maior naturalidade linguística, diversidade pragmática e aderência a contextos concretos de uso, o que é importante para modelos que precisam lidar com linguagem em condições menos controladas. Em contrapartida, também podem trazer ruído, informalidade, dados sensíveis e estrutura irregular, exigindo curadoria para seu uso adequado [Sanches et al., 2022a].

Além do uso de interações reais, a produção controlada de dados pode ampliar a disponibilidade de recursos em domínios e idiomas com baixa cobertura. Uma possibilidade é a elicitação por *crowdsourcing*, na qual participantes humanos produzem exemplos a partir de instruções definidas. Outra possibilidade é a geração sintética automática ou semiautomática, baseada em regras, *templates*, simulação de cenários, reformulação de exemplos existentes ou uso de modelos de linguagem.

Também é possível recorrer à ampliação de dados por aumento de dados. Nesse caso, novas instâncias são derivadas de exemplos existentes por meio de operações como paráfrase, substituição lexical e variação de entidades ou rótulos. Enquanto a geração sintética pode produzir exemplos a partir de regras, cenários, modelos ou sementes externas ao conjunto original, o aumento de dados tende a ampliar a diversidade mantendo dependência semântica e estrutural mais direta em relação aos dados de partida. Assim, as duas estratégias respondem a problemas diferentes e exigem validações distintas.

Uma alternativa é a adaptação de recursos existentes, que permite seu uso em contextos nos quais, sem tratamento prévio, eles seriam pouco adequados ou mesmo inviáveis.

Esse processo pode envolver tradução, limpeza, normalização, reorganização estrutural, revisão de metadados e outros ajustes voltados à preservação do sentido, da consistência e da adequação ao novo cenário de uso [Sanches et al., 2022a, Budzianowski et al., 2020, Wagner Filho et al., 2018].

A coleta web controlada constitui outra estratégia relevante para a obtenção de dados. Nesse cenário, técnicas de *crawling* partem de sementes iniciais e exploram recursivamente documentos, domínios e links, combinando etapas de filtragem, normalização, deduplicação e rastreabilidade. Abordagens desse tipo constituem antecedentes metodológicos importantes para a construção de corpora web em larga escala, com apoio de ferramentas como o BootCaT e de recursos como a família WaCky [Baroni and Bernardini, 2004, Baroni et al., 2009, Wagner Filho et al., 2018]. Embora anteriores ao uso contemporâneo de corpora massivos em LLMs, esses trabalhos consolidaram práticas relevantes de coleta, filtragem e organização de dados textuais provenientes da web.

Os conjuntos da família WaCky são resultado da coleta web em larga escala apoiada em requisições iterativas a mecanismos de busca e no uso de *crawlers*. Como consequência, produzem corpora amplos, heterogêneos e com alto grau de ruído. Por isso, embora esses recursos sejam valiosos como base para pré-treinamento ou para a extração de subconjuntos específicos, sua utilidade prática depende de etapas posteriores de pós-processamento e curadoria sistemática.

A utilidade de um conjunto de dados para LLMs depende simultaneamente de sua estrutura, qualidade e adequação à tarefa e ao contexto linguístico em que será empregado. A avaliação de um recurso não pode se limitar ao volume disponível ou à estrutura, mas deve considerar também sua origem, nível de processamento, consistência interna e capacidade de sustentar o comportamento esperado do modelo. Assim, compreender como diferentes tipos de dados são obtidos, transformados e validados é condição central para analisar seu papel na construção e adaptação de sistemas de linguagem.

## 2.3 Aprendizado Federado

Em muitos contextos atuais, os dados usados para treinar modelos estão distribuídos entre diferentes clientes federados, que podem corresponder a servidores locais, veículos, celulares ou outras entidades computacionais. Nesse cenário, centralizar todos os registros em um único repositório pode ser caro ou indesejável por razões de privacidade, governança e conformidade regulatória. O aprendizado federado surge como uma alternativa a esse problema ao permitir que múltiplos participantes treinem colaborativamente um mesmo modelo sem exigir a transferência dos dados brutos para um servidor central [Kairouz and McMahan, 2021].

Nesse paradigma de treinamento, um servidor de agregação coordena o processo global, enquanto os clientes mantêm seus próprios dados locais e executam etapas de treinamento em seus ambientes. O modelo global corresponde à versão compartilhada dos parâmetros que o servidor distribui aos participantes no início de cada rodada. A unidade operacional desse processo é a rodada de treinamento, composta pela distribuição do modelo global, ajuste local nos clientes, envio das atualizações e agregação no servidor. A lista a seguir

detalha esse fluxo.

1. **Distribuição do modelo global:** o servidor envia a versão corrente do modelo aos clientes selecionados.
2. **Treinamento local:** cada cliente ajusta o modelo com seus próprios dados, sem compartilhar esses dados diretamente.
3. **Envio das contribuições locais:** os clientes retornam ao servidor os pesos locais atualizados ou, conforme a implementação, as diferenças entre os pesos locais e o modelo global recebido.
4. **Agregação global:** o servidor combina as contribuições recebidas para gerar uma nova versão do modelo global.
5. **Redistribuição:** a nova versão atualizada do modelo é compartilhada com os clientes em uma rodada posterior.

Os passos executados durante o treinamento federado estão representados na Figura 2.4. A figura destaca a separação entre dados locais e coordenação global com os dispositivos mantendo seus registros localmente, enquanto o servidor recebe apenas atualizações associadas ao modelo em cada rodada de treinamento.

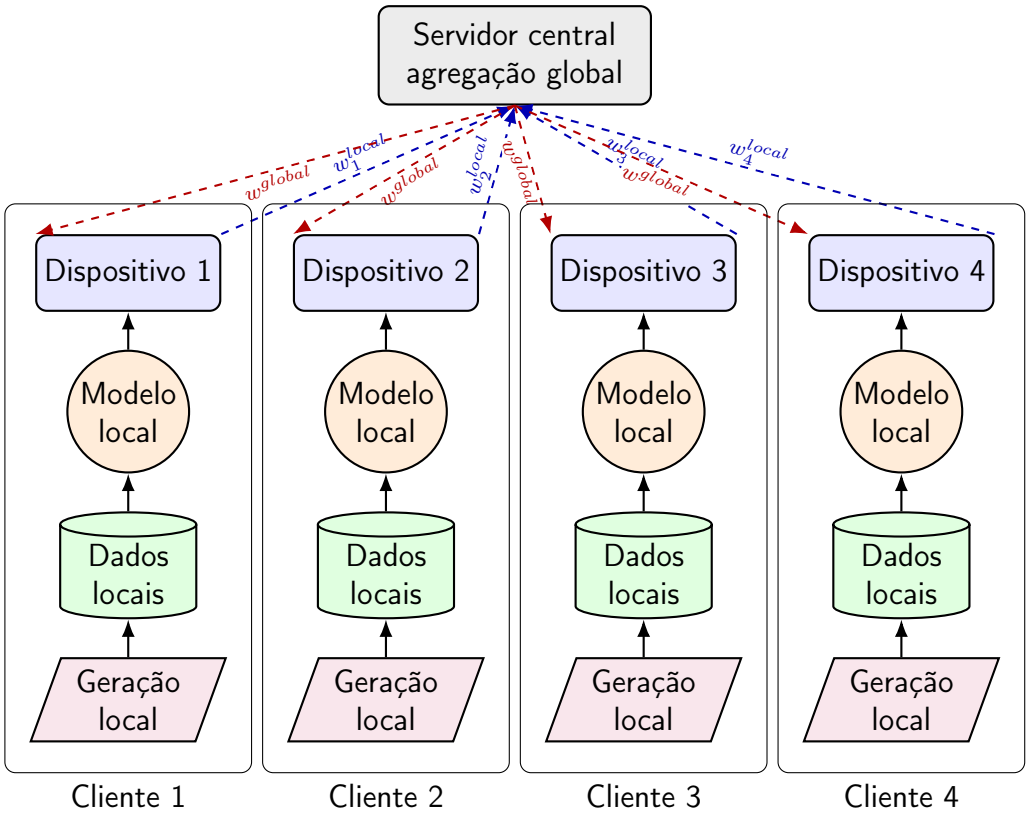



Figura 2.4: Funcionamento do aprendizado federado.

A natureza iterativa do treinamento federado permite que o modelo global incorpore informações extraídas de dados distribuídos sem exigir a centralização dos registros locais.

Em cada rodada, a atualização global depende das contribuições dos clientes selecionados, o que torna o desempenho do processo sensível à disponibilidade dos participantes, à qualidade das atualizações locais e ao custo de comunicação entre clientes e servidor [Kairouz and McMahan, 2021].

Após receber os modelos locais atualizados dos clientes participantes, o servidor os agrega para produzir uma nova versão do modelo global. Quando a implementação transmite deltas em vez de modelos completos, a atualização pode ser expressa de forma equivalente pela agregação ponderada das variações locais em relação ao modelo global recebido pelos clientes. Uma estratégia clássica para isso é o FederatedAveraging (FedAVG) [McMahan et al., 2017]. Nessa abordagem, o servidor atualiza o modelo global por meio de uma média ponderada dos modelos locais recebidos dos clientes participantes.

A Equação 2.4 descreve a agregação no servidor do algoritmo FedAVG para o conjunto de clientes participantes da rodada $t$, denotado por $S_t$. Nessa formulação, o novo modelo global ($w^{(t+1)}$) é calculado por meio de uma média ponderada dos modelos locais atualizados ($w_k^{(t+1)}$). O coeficiente de ponderação atribuído a cada cliente é proporcional ao tamanho de seu conjunto de dados local ($n_k$) em relação ao volume total de dados dos clientes selecionados na rodada ($\sum_{j \in S_t} n_j$). A participação total corresponde ao caso particular em que $S_t = \{1, \ldots, K\}$.

$$w^{(t+1)} = \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} w_k^{(t+1)}. \tag{2.4}$$

Durante sua utilização, diferentes características podem afetar diretamente a execução do aprendizado federado. Os dados distribuídos entre os clientes frequentemente apresentam distribuições locais distintas, caracterizando heterogeneidade estatística em cenários *non-IID*. Também é comum que os participantes apresentem forte heterogeneidade de hardware, memória, energia disponível e qualidade de conexão, o que afeta a velocidade do treinamento e a estabilidade das rodadas. O custo de comunicação também pode ser relevante devido ao envio repetido dos parâmetros do modelo, especialmente quando são utilizados modelos de grande porte. Esses fatores fazem com que a disponibilidade dos clientes influencie quais dispositivos podem participar de cada rodada de treinamento.

Outra característica relevante é que, embora o treinamento federado reduza a necessidade de centralização dos dados brutos, ele não elimina riscos de privacidade associados às atualizações de modelo nem impede a participação de clientes maliciosos, que podem introduzir ataques de envenenamento ou backdoors. Portanto, privacidade e segurança devem ser tratadas como dimensões complementares do treinamento federado [Kairouz and McMahan, 2021, Bagdasaryan et al., 2020].

Essas características tornam o aprendizado federado particularmente atraente para o treinamento e a adaptação de LLMs, pois dados textuais frequentemente contêm informações pessoais, institucionais ou sensíveis que, preferencialmente, não devem ser centralizadas nem transmitidas. Além disso, modelos de linguagem apresentam alta capacidade de memorizar padrões raros, e sua escala amplia os desafios de comunicação, disponibilidade e heterogeneidade computacional em ambientes distribuídos [Piccialli et al., 2025, Hu et al., 2025]. Assim, a utilização de FL oferece uma estratégia relevante para adaptação distribuída de LLMs, ainda exigindo mecanismos complementares de proteção contra

vazamento de informações.

## 2.4 Privacidade e Vazamento em Modelos Federados

Uma das principais vantagens do FL é reduzir os riscos associados à centralização dos dados. Essa descentralização, contudo, não elimina o risco de privacidade, pois informações sensíveis presentes no treinamento local podem ser incorporadas aos parâmetros do modelo e reaparecer em inferências futuras. Esse problema pode ocorrer com maior frequência em contextos com modelos de grande porte, dados heterogêneos e múltiplas rodadas de ajuste fino [Kairouz and McMahan, 2021, Zhu et al., 2019, Piccialli et al., 2025].

Em modelos de linguagem treinados ou ajustados localmente, dados sensíveis presentes nos exemplos de treinamento podem ser aprendidos como associações textuais específicas, sobretudo quando aparecem em sequências raras, repetidas ou altamente identificáveis. Nesta tese, consideramos sensíveis as instâncias de Personally Identifiable Information (PII) capazes de identificar, direta ou indiretamente, pessoas naturais, incluindo nomes, CPFs e outros documentos, e-mails, telefones e endereços. Em dados conversacionais, esses elementos podem surgir em entradas do usuário, respostas do sistema e instruções [Sousa and Kern, 2023, Szep et al., 2026].

A análise desse risco envolve conceitos próximos, mas não equivalentes. A **memorização** ocorre quando sequências, entidades ou associações específicas passam a influenciar o comportamento do modelo, ainda que não apareçam de imediato nas saídas. Esse risco pode aumentar com **overfitting**, embora modelos com bom desempenho médio também possam reter padrões raros, repetidos ou incomuns.

O problema se torna mais grave quando a informação memorizada deixa de ser apenas latente. A **exposição** ocorre quando fragmentos ou valores sensíveis aparecem durante a inferência e ficam acessíveis a usuários ou atacantes. Já a **extração** corresponde ao esforço ativo de provocar essa revelação por meio de consultas direcionadas e estratégias de consulta e *prompting*, convertendo a memorização em vazamento efetivo.

Essa distinção é relevante porque a memorização não depende necessariamente de queda evidente no desempenho agregado do modelo. Quando esse conteúdo memorizado aparece durante a inferência ou é revelado por consultas direcionadas, a retenção latente passa a configurar vazamento observável [Carlini et al., 2019, Mireshghallah et al., 2022, Zeng et al., 2024, Carlini et al., 2021].

Em FL, a superfície de ataque apresenta características específicas quando o adversário participa do processo federado como um cliente. Nesse modelo de ameaça, o atacante controla os dados usados no treinamento local, manipula a atualização enviada ao servidor e observa o modelo global redistribuído, enquanto o servidor e os demais clientes seguem o protocolo federado.

Em cenários de aprendizado federado, ataques de *poisoning* e indução à memorização são relevantes, pois clientes maliciosos podem inserir, repetir ou reforçar associações sensíveis durante o treinamento local para que essas associações sejam incorporadas ao modelo global após a agregação [Kairouz and McMahan, 2021, Bagdasaryan et al., 2020, Xie et al., 2020].

Mesmo quando representam pequena parcela dos clientes, participantes maliciosos podem induzir LLMs federados a preservar associações entre contextos textuais e valores sensíveis. Essas associações podem corresponder a dados sensíveis já presentes em conjuntos locais vulneráveis ou a padrões artificialmente inseridos para testar e explorar a propensão do modelo à memorização.

Após a agregação, essas associações podem ser exploradas por *prompts* direcionados para provocar exposição ou extração [Hu et al., 2025].

A Figura 2.5 ilustra a etapa em que um participante malicioso atua como cliente regular no processo federado. Enquanto os clientes honestos treinam modelos locais com seus próprios dados e enviam atualizações convencionais ao servidor, o cliente comprometido utiliza dados manipulados para produzir uma atualização local enviesada.

Essa atualização, representada por $w_3^{mal}$, pode carregar associações induzidas por *poisoning* e reforçadas por regimes locais propensos à memorização, incluindo repetição de exemplos, baixa diversidade dos dados ou ajuste excessivo a padrões específicos. Com isso, a contribuição maliciosa segue o mesmo fluxo de agregação das contribuições legítimas.

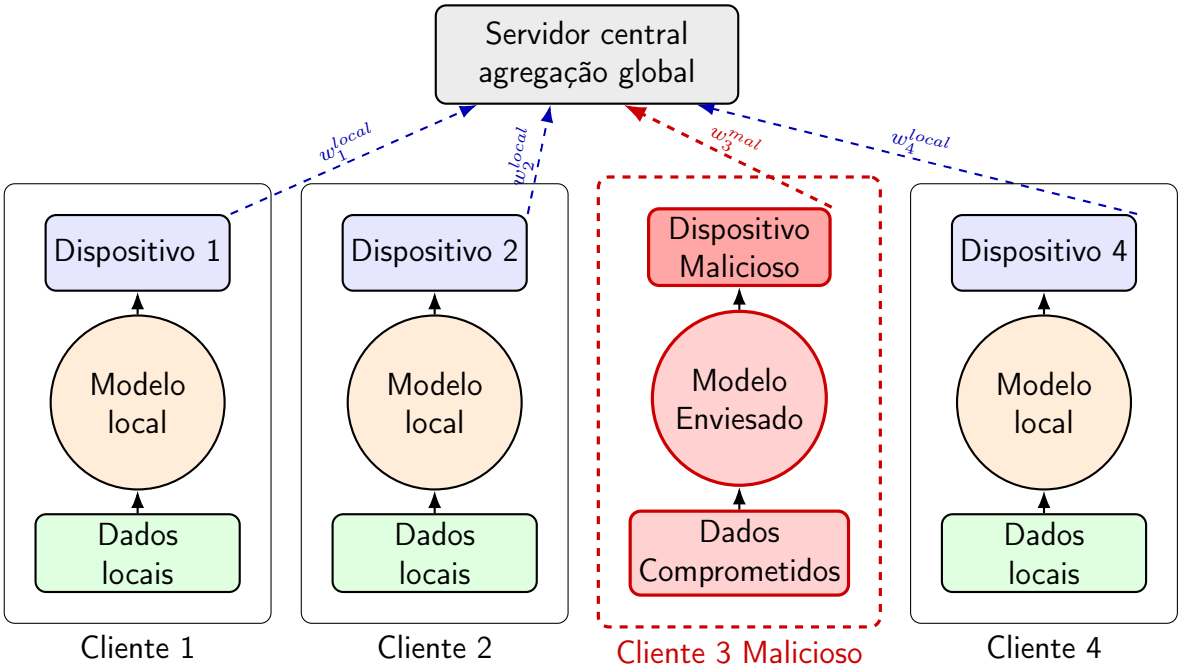



Figura 2.5: Envio de atualização local maliciosa durante uma rodada de aprendizado federado.

A Figura 2.6 mostra a etapa posterior à agregação, quando a influência da atualização maliciosa já foi incorporada ao modelo global. Nesse caso, o servidor passa a redistribuir o modelo comprometido para todos os participantes, inclusive os clientes honestos, ampliando o alcance das associações indesejadas aprendidas durante a rodada federada. Assim, a ameaça não se limita ao dispositivo atacante: ela pode se propagar pelo ciclo de treinamento e criar condições para a exposição ou extração de informações sensíveis em interações futuras.

O cliente malicioso também recebe o modelo comprometido, o que lhe permite avaliar se as associações induzidas foram preservadas após a agregação e ajustar sua estratégia nas

rodadas seguintes. Esse retorno cria um ciclo de realimentação com o atacante testando a persistência dos padrões induzidos, ajustando gatilhos textuais e reforçando entidades sensíveis nas rodadas seguintes, o que aumenta a probabilidade de exposição e extração.

Após a agregação, o modelo global também pode preservar associações induzidas pelo cliente malicioso. A extração, nesse caso, incide principalmente sobre padrões introduzidos ou reforçados no treinamento local malicioso, salvo quando o cenário de ameaça também inclui mecanismos específicos para inferir dados dos clientes honestos.

Esse ciclo de realimentação e a extração de padrões ou *canários* inseridos pelo próprio atacante caracterizam uma classe de ameaça, não um requisito geral de extração em FL. Em outra classe, de amplificação estrutural entre clientes, o participante reforça prefixos ou um *template* canônico com dados próprios, sem acesso aos nomes ou aos registros pessoais das vítimas, para investigar se o modelo global completa esses padrões com PII memorizada nos dados dos clientes honestos. Ataques por prefixos contextuais e Latent Association Finetuning (LAFt) ilustram essa possibilidade [Hu et al., 2025]. A configuração desta tese se aproxima dessa segunda classe: o cliente auxiliar usa perfis sintéticos renovados a cada rodada, sem adaptá-los às respostas do modelo, e o mascaramento do prefixo reforça a continuação canônica. Um avaliador confiável separado consulta o modelo global e mede a reprodução dos registros das vítimas.

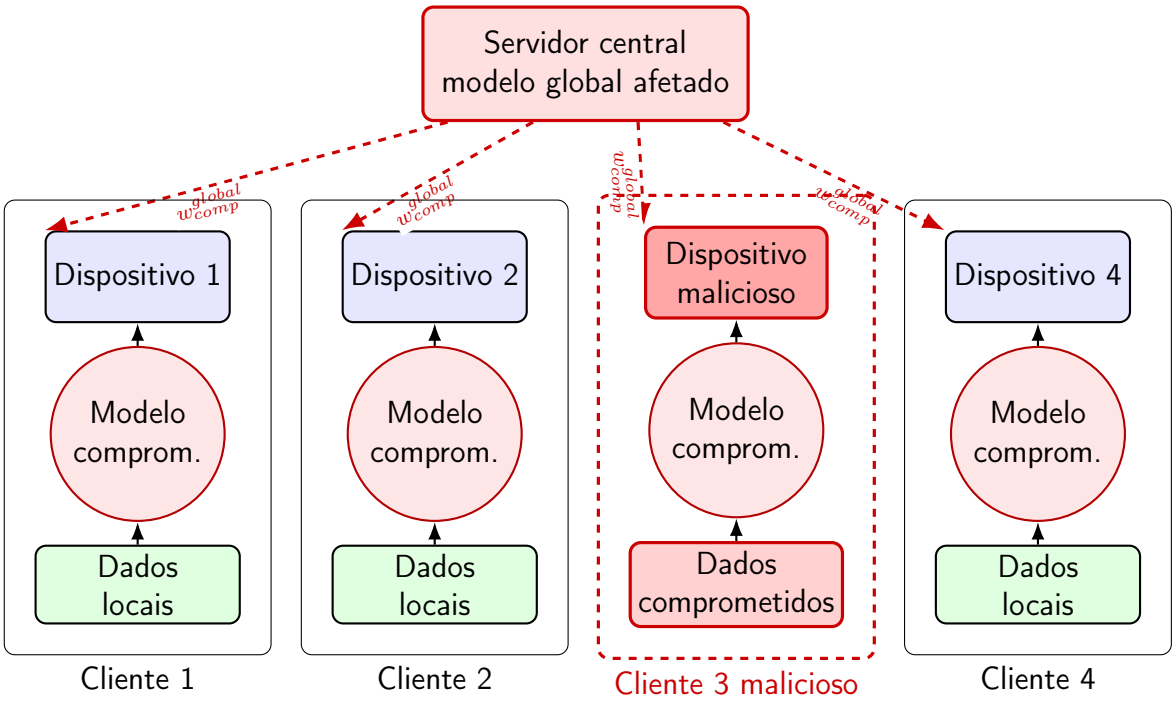



Figura 2.6: Redistribuição do modelo global afetado após agregação de uma atualização maliciosa.

Para mitigar esse problema, estratégias baseadas em DP, como *DP-SGD*, limitam a influência de exemplos, usuários ou clientes por meio de *clipping* e adição de ruído durante a otimização. Em FL, essa proteção precisa explicitar o nível da garantia adotada, pois a privacidade pode incidir sobre registros locais, usuários ou atualizações de clientes antes da agregação.

Outra frente de proteção envolve abordagens baseadas em anonimização, mascaramento, remoção de entidades e filtragem, que reduzem o conteúdo identificável antes do treinamento. Apesar de apresentarem bons resultados, essas técnicas possuem limites. Transformações mais agressivas nos conjuntos de dados tendem a alterar a semântica e reduzir a naturalidade linguística, resultando na degradação da convergência e da utilidade dos modelos. Em ambientes federados, restrições de memória, comunicação e processamento tornam esse custo ainda mais crítico [Sousa and Kern, 2023, Piccialli et al., 2025, Szep et al., 2026].

Assim, a proteção contra vazamentos em LLMs federados pode atuar antes, durante e após o treinamento para reduzir a presença de PII nos dados locais, limitar a influência de exemplos ou clientes na atualização agregada e avaliar a exposição por consultas direcionadas, sem comprometer a utilidade do modelo nem a estabilidade da convergência federada.

# Capítulo 3

# Trabalhos Relacionados

Este capítulo apresenta os trabalhos relacionados às contribuições desta tese. Os principais estudos identificados estão organizados em duas seções. A Seção 3.1 discute conjuntos de dados textuais e conversacionais, com foco na coleta, curadoria, tradução e geração de dados para o português brasileiro. A Seção 3.2 aborda privacidade e vazamento de dados em aprendizado federado, posicionando a proposta desta tese em relação a ataques, memorização e defesas em modelos de linguagem.

## 3.1 Ecossistema de Conjuntos de Dados Textuais

A literatura oferece metodologias, ferramentas e conjuntos de dados para a construção, adaptação e avaliação de dados textuais utilizados no treinamento, ajuste fino e avaliação de LLMs. Entretanto, esses recursos raramente combinam, de forma simultânea, português brasileiro, escala, rastreabilidade, controle de qualidade, estrutura conversacional, diálogos H2H, múltiplos participantes, múltiplos turnos e anotação semântica. Esta seção examina os trabalhos relacionados a partir dessas dimensões, posicionando as contribuições desta tese no ecossistema de dados textuais e conversacionais.

Os trabalhos relacionados estão organizados em quatro categorias: metodologias de coleta e preparação, ferramentas de construção de dados, corpora multidomínio em português e conjuntos de dados conversacionais. A comparação considera forma de geração, idioma, fonte dos dados, estruturação, tipo de anotação e adequação a cenários conversacionais.

Nos trabalhos desta seção, o termo anotação é restringido à anotação semântica ou conversacional, isto é, rótulos associados a exemplos ou turnos, como intenções, atos de diálogo, estados, slots, ações ou categorias equivalentes. Outros tipos de informações, como anotações linguísticas automáticas, marcações estruturais e metadados de proveniência não classificam o conjunto como anotado, pois caracterizam o corpus, mas não representam comportamento conversacional.

**Metodologias de coleta e preparação de dados**

Como alternativa para aumentar a disponibilidade de conjuntos de dados textuais, surgem iniciativas que utilizam a web como fonte primária. Nesse contexto, Chakrabarti, van

den Berg e Dom propõem o *Focused Crawler* [Chakrabarti et al., 1999], uma abordagem voltada à descoberta de recursos relevantes para um conjunto predefinido de tópicos.

Nessa abordagem, o mecanismo de classificação utiliza uma taxonomia de tópicos e exemplos de páginas para estimar a probabilidade de um documento pertencer ao assunto desejado, em vez de depender apenas de palavras-chave. A partir dessa pontuação, o sistema direciona a exploração dos hiperlinks descobertos, reduzindo a área explorada e evitando páginas consideradas irrelevantes. A exploração pode seguir estratégias de foco rígido, que interrompem a coleta em um caminho quando a página avaliada não se enquadra no tópico, ou de foco suave, que utilizam a pontuação de relevância para definir a prioridade dos próximos links na fila de visitas.

Com isso, a coleta se torna mais seletiva que buscas exaustivas ou consultas baseadas apenas em palavras-chave, favorecendo a construção de coleções temáticas mais relevantes. O trabalho é anterior ao cenário de corpora para LLMs e não aborda deduplicação, rastreabilidade persistente, filtros de toxicidade, português brasileiro ou dados conversacionais.

Seguindo esse estilo de criação de conjuntos de dados, Baroni et al. apresentam a iniciativa WaCky [Baroni et al., 2009], composta por corpora web de grande escala para inglês, alemão e italiano. O trabalho também compara os recursos gerados com corpora tradicionais, buscando avaliar cobertura lexical, composição textual e qualidade dos dados coletados da web.

O WaCky demonstra que corpora web podem ser construídos em larga escala por meio de sementes lexicais, coleta automatizada, filtragem estrutural e deduplicação, criando os conjuntos ukWaC, deWaC e itWaC. Sua principal contribuição para esta tese é metodológica: evidencia a viabilidade de corpora web massivos. Sua principal limitação, para o escopo deste trabalho, é a ausência de controle voltado ao português brasileiro, rastreabilidade persistente e estrutura conversacional.

A avaliação qualitativa mostra que corpora web, como o ukWaC, tendem a ser dominados por tópicos como educação, computadores, universidades, governo, organizações e outros textos da esfera pública. Em contraste, corpora tradicionais apresentam maior presença de ficção, narrativas e transcrições de linguagem falada. Assim, a iniciativa WaCky mostra que corpora web podem capturar usos mais recentes e dinâmicos da língua em relação aos corpora tradicionais usados como comparação.

Apesar dessas contribuições, os corpora produzidos dependem de filtros restritivos e permanecem sem estrutura conversacional. Além disso, o trabalho ainda enfrenta limitações típicas de corpora web, como ruído textual, dificuldades na remoção de *boilerplate* e desafios de deduplicação.

A Tabela 3.1 sintetiza essa comparação e posiciona, em relação aos trabalhos externos, duas contribuições da tese: a Metodologia de Coleta Web Controlada e o processo de limpeza e preparação de conjuntos existentes.

As abordagens comparadas nessa tabela têm em comum o objetivo de gerar ou adaptar conjuntos de dados textuais. O *Focused Crawler*, o WaCky e a Metodologia de Coleta Web Controlada partem da web como fonte primária, mas diferem no mecanismo de exploração. O primeiro prioriza páginas por relevância temática, enquanto o segundo utiliza buscadores e combinações de bigramas como sementes. A proposta desta tese

organiza a coleta por critérios explícitos de exploração.

Já a Limpeza de Dados não constitui uma estratégia de coleta, mas um processo posterior de transformação, filtragem e adequação de dados já existentes para permitir seu uso em domínios antes inviáveis.

A comparação também evidencia diferenças quanto ao idioma, à estruturação e à anotação. Enquanto WaCky não é direcionado ao português, as propostas desta tese consideram esse idioma como requisito. Em relação à estrutura dos dados, as metodologias de coleta web produzem predominantemente texto não estruturado, ao passo que a etapa de limpeza é adaptável a diferentes formatos de entrada. Dessa forma, a proposta não se limita a obter mais dados, mas também busca controlar sua origem, seu processamento e sua adequação para uso posterior.

| **Categoria** | **Relacionado** | **Relacionado** | **Proposta** | **Proposta** |
|---|---|---|---|---|
| **Trabalho** | Focused Crawler | WaCky | Metodologia de Coleta Web Controlada | Limpeza de Dados |
| **Tipo de Geração** | Coleta web | Coleta web | Coleta web | Processamento |
| **Português** | Suporta | Não | Suporta | Suporta |
| **Mecanismo** | Busca guiada por relevância temática | Buscadores com bigramas identificados | Metodologia de Coleta web Controlada | N/A |
| **Estruturado** | Não | Não | Não | Suporta |
| **Anotado** | Não | Não | Não | N/A |

Tabela 3.1: Comparação entre metodologias relacionadas e propostas neste trabalho.

Enquanto a Tabela 3.1 compara abordagens gerais de coleta e preparação de textos, a Tabela 3.2 se concentra nas metodologias propostas para geração de diálogos H2H multiturno e multiusuário. Nesse recorte, MCCD e MDC compartilham o objetivo de produzir dados conversacionais em português, mas seguem estratégias distintas. O MCCD utiliza dados coletados da web para identificar, extrair e estruturar interações já existentes, combinando coleta controlada com organização posterior dos diálogos. O MDC, por sua vez, se baseia em *crowdsourcing* para orientar a produção dos diálogos desde sua origem.

Essa diferença metodológica afeta diretamente o tipo de dado produzido. No MCCD, a estruturação ocorre após a coleta e pode combinar diálogos estruturados com texto originalmente não estruturado. No MDC, os dados já são produzidos em formato estruturado e anotado, o que facilita o controle sobre turnos, participantes, intenções e demais informações necessárias para uso em tarefas conversacionais. Apenas o MDC prevê explicitamente o uso de aumento de dados, pois sua organização permite aplicar técnicas de interpolação para ampliar o volume de exemplos gerados.

Em conjunto, as duas tabelas mostram que as contribuições desta tese atuam em níveis complementares. A Metodologia de Coleta Web Controlada e os processos de Limpeza de Dados tratam da obtenção e preparação de corpora textuais de forma mais ampla, enquanto MCCD e MDC especializam esse esforço para o domínio conversacional.

| Categoria | Proposta | Proposta |
|---|---|---|
| **Trabalho** | MCCD | MDC |
| **Tipo de Geração** | Coleta web | Crowdsourcing |
| **Português** | Suporta | Suporta |
| **Mecanismo** | Coleta web Controlada | Interface própria |
| **Estruturado** | Diálogos Estruturados<br>Texto Não Estruturado | Estruturado |
| **Anotado** | Não | Sim |
| **Aumento de Dados** | Não | Sim |

Tabela 3.2: Comparação entre as metodologias propostas para construção de conjuntos de diálogos H2H, multiturno e multiusuário.

**Ferramentas para construção e anotação de dados**

Nesse contexto, Urbanek e Ringshia propõem um framework de código aberto chamado Mephisto [Urbanek and Ringshia, 2024] para conduzir tarefas de coleta e anotação. A proposta organiza o ciclo de trabalho por meio de abstrações para tarefas, execuções, unidades, agentes, interfaces, provedores de multidão e bases de dados. Além dos conjuntos descritos, ferramentas de *crowdsourcing* também orientam a criação e a anotação de dados em cenários nos quais a participação humana precisa de controle, registro e revisão.

O Mephisto também incorpora recursos de controle de qualidade, revisão e qualificação de trabalhadores. Esses recursos são relevantes para conjuntos conversacionais porque a coleta humana costuma exigir instruções claras, validação de respostas, acompanhamento de histórico e filtragem de contribuições inadequadas.

Apesar dessas contribuições, a principal limitação está em sua complexidade operacional, exigindo configuração de componentes, definição de fluxos, integração com provedores, preparação de interfaces e gestão dos dados coletados. Esse nível de engenharia pode dificultar a adoção por pesquisadores que desejam executar coletas menores ou iterar rapidamente sobre tarefas de anotação. Por essa razão, conjuntos criados dessa maneira tendem a apresentar tamanho limitado, e a falta de integração nativa com técnicas de aumento de dados restringe ainda mais sua ampliação.

O *Amazon Mechanical Turk* [Crowston, 2012] é uma plataforma de *crowdsourcing* para distribuir pequenas tarefas a trabalhadores humanos. Em dados textuais, apoia produção, anotação, classificação e validação de exemplos.

Dessa forma, essa ferramenta é tratada como infraestrutura de execução de *crowdsourcing*, não como metodologia de geração de diálogos. Sua escala, custo e qualidade dependem do desenho da tarefa, da remuneração e dos critérios de controle adotados. Sua relevância para esta tese está em evidenciar limitações operacionais de plataformas genéricas, como coordenação de participantes, controle de qualidade e ausência de mecanismos específicos para diálogos assíncronos, multiusuário e anotados.

Outra referência importante para a construção de corpora a partir da web é o BootCaT, proposto por Baroni e Bernardini [Baroni and Bernardini, 2004]. Esta ferramenta implementa um procedimento de *bootstrapping* para criar corpora especializados e extrair termos técnicos a partir de uma pequena lista inicial de palavras sementes, geralmente

composta por poucos termos representativos do domínio de interesse. Essas sementes são combinadas em consultas submetidas a mecanismos de busca, e as páginas recuperadas formam um corpus inicial. Em seguida, novos termos são extraídos desse material e podem alimentar novas rodadas de consulta, permitindo expandir iterativamente o corpus.

O BootCaT implementa um fluxo de mineração web para duas tarefas complementares, a criação de corpora especializados e a extração terminológica. Ao final do processo, o corpus coletado pode ser utilizado para identificar termos simples e expressões de múltiplas palavras, considerando critérios como frequência e padrões de conexão entre palavras. Desse modo, o BootCaT é relevante como referência de construção rápida de corpora *ad hoc*.

Outro aspecto importante é a modularidade da ferramenta. O BootCaT foi concebido como um conjunto de programas que podem ser executados separadamente, permitindo inspecionar arquivos intermediários, ajustar parâmetros e reutilizar partes específicas do fluxo de coleta e extração. Apesar da modularidade e da utilidade para construção rápida de corpora *ad hoc*, o BootCaT depende de mecanismos de busca externos e da qualidade das páginas retornadas. Essa dependência reduz o controle sobre amostragem, trajetória de coleta e reprodutibilidade, além de não produzir dados conversacionais estruturados.

A Tabela 3.3 compara ferramentas com fluxo próprio de construção, estruturação ou extração de dados. O Amazon Mechanical Turk é discutido separadamente por atuar como infraestrutura genérica de execução de tarefas humanas, e não como ferramenta específica de modelagem ou estruturação de corpora conversacionais. A comparação reúne Mephisto e BootCaT como ferramentas relacionadas, enquanto WackyWacky, MDC e Miner XenForo representam ferramentas propostas ou diretamente associadas às contribuições deste trabalho.

| Categoria | Relacionado | Relacionado | Proposta | Proposta | Proposta |
|---|---|---|---|---|---|
| **Trabalho** | Mephisto | BootCaT | WackyWacky | MDC | Miner XenForo |
| **Tipo de Geração** | Crowdsourcing | Coleta web | Coleta web | Crowdsourcing | Coleta web |
| **Domínio** | Geral | Geral Filtrável | Geral | Geral | Geral |
| **Fonte dos Dados** | Crowdsourcing | Web | Web | Interface própria | Fóruns XenForo |
| **Interação** | Customizável | N/A | N/A | H2H | H2H |
| **Diálogos** | Suporta | Não | Não | Sim | Sim |
| **Multiusuário** | Suporta | N/A | N/A | Sim | Sim |
| **Multiturno** | Suporta | N/A | N/A | Sim | Sim |
| **Estratégia de Busca Automática** | N/A | Termos semente | Termos semente | N/A | URLs semente |
| **Diálogos Estruturados** | Suporta | Não | Não | Sim | Sim |
| **Anotado** | Suporta | Não | Não | Sim | Não |

Tabela 3.3: Comparação entre ferramentas aplicáveis a diferentes idiomas.

A comparação evidencia dois eixos de diferenciação: a origem dos dados e o grau de estruturação produzido. Mephisto e MDC dependem de participação humana e oferecem maior controle sobre a anotação enquanto BootCaT, WackyWacky e Miner XenForo priorizam escala por coleta web, mas diferem no tipo de estrutura resultante.

No grupo de coleta web, BootCaT e WackyWacky operam a partir de termos semente submetidos a mecanismos de busca. A diferença está no uso posterior desses resultados: BootCaT permanece dependente de novas consultas aos buscadores para ampliar a coleção, enquanto WackyWacky utiliza as URLs encontradas na busca inicial como pontos de partida para uma exploração recursiva. Miner XenForo, por outro lado, parte diretamente de URLs semente e é direcionado à extração de dados em fóruns baseados na plataforma XenForo.

A tabela também evidencia diferenças no tratamento de diálogos. Mephisto oferece suporte customizável a tarefas interativas e pode ser adaptado a diferentes cenários de coleta, incluindo interações com múltiplos usuários e múltiplos turnos. Já BootCaT e WackyWacky são ferramentas voltadas à coleta textual ampla e não suportam geração de diálogos estruturados.

MDC e Miner XenForo são mais diretamente alinhados ao objetivo desta tese, pois lidam com diálogos H2H, multiusuário e multiturno. O MDC apoia a produção controlada desses diálogos por meio de uma interface própria e roteiros pré-definidos, enquanto o Miner XenForo permite extrair interações já existentes em fóruns online.

Quanto à anotação, Mephisto e MDC oferecem suporte explícito a dados anotados. BootCaT e WackyWacky não são apresentados como ferramentas anotadas na comparação, e Miner XenForo estrutura diálogos extraídos de fóruns, mas não realiza anotações.

**Conjuntos de Dados Multidomínio em Português**

Para o português, Wagner Filho et al. apresentam o *Brazilian Portuguese Web as Corpus* (BrWaC) [Wagner Filho et al., 2018], um corpus aberto construído a partir de páginas web para disponibilizar um recurso de grande escala para pesquisas em processamento de linguagem natural.

O trabalho adapta a metodologia WaCky ao português, organizando a construção do conjunto de dados em etapas de identificação de URLs sementes, rastreamento, limpeza e deduplicação. A restrição inicial a domínios associados ao Brasil busca reduzir a presença de outras variantes de idioma.

O processo de coleta partiu de mais de 60 milhões de páginas rastreadas e resultou em 3,53 milhões de documentos, 145 milhões de sentenças e aproximadamente 2,68 bilhões de tokens. Essa diferença entre o volume bruto e o volume final decorre de uma filtragem onde cerca de 95% dos documentos inicialmente coletados foram descartados, principalmente por duplicação exata ou por tamanho insuficiente após a remoção de elementos estruturais das páginas.

Uma contribuição metodológica relevante do BrWaC é a deduplicação em nível de sentença. Em vez de comparar pares de documentos completos, o método verifica a ocorrência de sentenças longas já observadas e remove documentos com alta proporção de conteúdo não original. Essa estratégia reduz o custo de processamento e contribui para a baixa repetição de sentenças longas no corpus final.

O conjunto final reúne textos de mais de 121 mil sites distintos, o que amplia a diversidade das fontes. Apesar de não incluir anotações de intenções, o corpus recebe anotação sintática com o *parser* Palavras [Pacheco and Guaranha, 2022].

Mesmo com essas contribuições, o BrWaC também evidencia desafios recorrentes de

corpora derivados da web. A filtragem por domínios brasileiros não garante completamente a exclusão de outras variantes do português, e a coleta automatizada pode preservar ruído estrutural, páginas de baixa qualidade, conteúdo potencialmente ofensivo e distribuição temática pouco controlada.

O *Open Super-large Crawled Aggregated coRpus*, ou OSCAR [Abadji et al., 2022], é uma referência importante entre corpora web multilíngues derivados do Common Crawl. Seu objetivo é disponibilizar grandes volumes de texto não anotado, organizados por idioma e preparados para processamento de linguagem natural, aprendizado de máquina e pré-treinamento de modelos de linguagem.

Outro aspecto relevante é a preocupação com deduplicação e identificação de documentos semelhantes em escala. O projeto utiliza *Locality Sensitive Hashing* para apoiar a detecção de documentos duplicados ou quase duplicados. O projeto também inclui marcadores de qualidade, como indicações de textos ou sentenças curtas, cabeçalhos, rodapés e segmentos ruidosos. Essas marcações são relevantes porque permitem que usuários apliquem filtros adicionais conforme os requisitos de cada tarefa.

Com foco em escala e cobertura multilíngue, este projeto oferece recursos para 166 idiomas, incluindo línguas de médio e baixo recurso. Na versão 23.01, o subconjunto no idioma inglês concentra cerca de 3,4 TB, aproximadamente 1,23 bilhão de documentos e 523 bilhões de palavras. Outros subconjuntos também apresentam escala elevada, como o chinês, com cerca de 1,4 TB e 138 milhões de documentos. Para o português, o OSCAR registra aproximadamente 36,1 milhões de documentos, 15,2 bilhões de palavras e 105 GB de conteúdo.

Ainda assim, a deduplicação não elimina todos os desafios de corpora web de grande escala, pois a remoção agressiva de repetições pode afetar conteúdos legítimos, como conversas, listas, letras de música ou documentos com estrutura recorrente. Por isso, em alguns casos, parte do custo de deduplicação e filtragem permanece sob responsabilidade do pesquisador que utiliza o recurso. Apesar de sua escala, o OSCAR também evidencia limitações típicas de corpora extraídos da web, como ruído de navegação, vieses de coleta, documentos multilíngues mal segmentados e problemas da classificação automática. Assim, o OSCAR demonstra a importância de corpora web multilíngues para o treinamento de modelos, mas também reforça que escala não substitui avaliação de qualidade, documentação, filtragem e adequação ao uso pretendido.

O Projeto Carolina [Crespo et al., 2023] é um corpus geral de português desenvolvido para oferecer um recurso aberto e documentado para pesquisas linguísticas e aplicações em processamento de linguagem natural. O projeto realiza a curadoria das fontes, registra a proveniência e caracteriza a tipologia dos textos, tornando o conjunto relevante para estudos que exigem controle sobre origem, natureza e composição dos dados utilizados.

Os textos deste projeto acompanham metadados estruturados, permitindo a identificação da fonte, tipo textual e outras características relevantes para seleção, filtragem e análise. Essa organização facilita estudos comparativos e reduz parte da opacidade comum em coleções derivadas da web, nas quais documentos de origens distintas são frequentemente agregados sem rastreabilidade suficiente.

Este conjunto contém centenas de milhões de tokens e milhões de textos, incluindo materiais provenientes de diferentes domínios, como documentos institucionais, textos

legislativos e judiciários, literatura em domínio público, jornais, redes sociais e wikis. A metodologia utilizada reforça a necessidade de que recursos derivados da web sejam acompanhados por informações que permitam auditoria, filtragem e uso responsável.

Corrêa et al. apresentam o GigaVerbo, uma coleção agregada e deduplicada de textos em português usada no pré-treinamento da família Tucano. O trabalho Tucano 2 amplia essa linha com o GigaVerbo-v2 e descreve receitas abertas de pré-treinamento continuado. Esses recursos se relacionam diretamente com os experimentos desta tese, que incluem o GigaVerbo na mistura de dados e usam o Tucano2-0.6B-Base como modelo inicial [Corrêa et al., 2025, Corrêa et al., 2026].

A Tabela 3.4 apresenta uma comparação entre conjuntos de dados multidomínio relacionados e conjuntos propostos nesta tese para o português. Essa análise considera idioma, escala, forma de geração, fonte dos dados, estratégia de construção e presença de anotação.

Os conjuntos comparados compartilham a característica de terem grande volume, mas diferem quanto ao escopo linguístico e à origem. BrWaC, GigaVerbo, Carolina, BrWaC-CLEAN e WackyWacky PTBR Geral são voltados ao português, enquanto OSCAR é multilíngue e deriva do Common Crawl. Em relação à fonte, BrWaC, Carolina e WackyWacky PTBR Geral utilizam a web como origem principal. O GigaVerbo agrega corpora de fontes públicas diversas. Já o BrWaC-CLEAN é produzido a partir do próprio BrWaC, por meio de um processo de limpeza, enquanto OSCAR depende de uma coleção ampla previamente disponibilizada pelo Common Crawl.

Os conjuntos BrWaC e WackyWacky PTBR Geral adotam uma estratégia de geração automática baseada em termos semente, enquanto Carolina é organizado sob a perspectiva de *web-as-corpus*, com maior atenção à procedência e à tipologia textual. O GigaVerbo combina corpora públicos com deduplicação e filtragem. Já OSCAR e BrWaC-CLEAN não aparecem associados a uma estratégia de coleta própria na comparação, pois o primeiro reutiliza dados do Common Crawl e o segundo atua sobre um corpus existente.

Quanto à anotação e às informações associadas, a comparação distingue ausência de anotação, metadados de qualidade, proveniência, anotação linguística e metadados de coleta. Carolina possui metadados de proveniência e tipologia, enquanto BrWaC possui anotação linguística. GigaVerbo registra metadados de qualidade e WackyWacky PTBR Geral registra metadados de coleta e exploração. Conforme a definição adotada nesta seção, esses metadados e as anotações linguísticas não contam como anotação semântica ou conversacional.

Assim, cada conjunto proposto cobre uma lacuna distinta. BrWaC-CLEAN atua sobre curadoria e filtragem de um corpus consolidado. WackyWacky PTBR Geral amplia a coleta web controlada em português. Adrenaline e OuterSpace exploram diálogos reais de fóruns. MultiWOZ-PTBR preserva anotações conversacionais em português. Por fim, Nutritious Food avalia geração controlada e aumentada de diálogos anotados.

**Conjuntos de Dados de Diálogos**

Entre os trabalhos que exploram conversas reais em larga escala, o *Ubuntu Dialogue Corpus*, proposto por Lowe et al. [Lowe et al., 2015], se destaca. Por reunir interações humanas multiturno extraídas de logs reais, o corpus se tornou uma referência para tarefas

| Categoria | Relacionado | Relacionado | Relacionado | Relacionado | Proposta | Proposta |
|---|---|---|---|---|---|---|
| **Trabalho** | BrWaC | GigaVerbo | OSCAR | Carolina | BrWaC-CLEAN | WackyWacky PTBR Geral |
| **Português** | Sim | Sim | Sim | Sim | Sim | Sim |
| **Tamanho** | Grande | Grande | Grande | Grande | Grande | Grande |
| **Tipo de Geração** | Coleta web | Agregação de corpora | Common Crawl | Coleta web | Limpeza | Coleta web |
| **Fonte dos Dados** | Web | Múltiplas fontes públicas | Common Crawl | Web | BrWaC | Web |
| **Estratégia** | Automático com termos semente | Agregação e deduplicação | N/A | Web-as-Corpus | N/A | Automático com termos semente |
| **Anotação** | Linguística (Palavras) | Metadados de qualidade | Ausente | Metadados de proveniência | Ausente | Metadados de coleta |

Tabela 3.4: Comparação entre conjuntos de dados multidomínio em português.

de seleção de respostas em sistemas de diálogo.

O corpus foi construído a partir de registros de salas de bate-papo da rede IRC utilizadas pela comunidade Ubuntu, com participantes interagindo para resolver problemas técnicos sobre o sistema operacional. Dessa forma, os diálogos apresentam o domínio de suporte técnico específico no idioma inglês.

O *Ubuntu Dialogue Corpus* foi organizado para conter conversas com pelo menos três turnos, alcançando média aproximada de 7,71 turnos por diálogo com cerca de 930 mil diálogos e mais de 7,1 milhões de respostas contendo aproximadamente 100 milhões de palavras.

As interações são textuais, H2H e segmentadas em diálogos e turnos, mas não possuem anotação semântica comparável à de conjuntos orientados a tarefas, como o Multi-Domain Wizard-of-Oz (MultiWOZ).

O trabalho de Lowe et al. demonstra o potencial de comunidades online como fonte de dados conversacionais, se aproximando do MCCD, que também parte de interações reais para construir conjuntos de dados multiturno. Apesar disso, o *Ubuntu Dialogue Corpus* utiliza logs de IRC em inglês e voltados a suporte técnico enquanto a metodologia proposta nesta tese é capaz de utilizar fontes de dados que contenham algum tipo de referência entre as mensagens. Já a instanciação da metodologia no Miner XenForo apresenta um caso de uso em que fóruns online em português brasileiro são utilizados como base para reconstruir fluxos conversacionais a partir de tópicos, postagens e relações entre mensagens.

Outro conjunto de referência para diálogos orientados a tarefas é o MultiWOZ, proposto por Budzianowski et al. [Budzianowski et al., 2020] e construído pelo método *Wizard-of-Oz* [Kelley, 1984]. Nesse protocolo, participantes humanos simulam a interação entre usuário e assistente. O usuário recebe um objetivo a cumprir e interage com um operador humano, o "wizard", que responde como se fosse o sistema conversacional. Essa estratégia gera diálogos naturais e mantém controle sobre metas, domínios, estados de diálogo e anotações semânticas. A coleta ocorreu por meio de *crowdsourcing*, com 1.249 trabalhadores.

O conjunto de dados abrange sete domínios relacionados a serviços urbanos e turísticos, incluindo atrações, hospitais, polícia, hotéis, restaurantes, táxis e trens. As conversas podem envolver um único domínio ou múltiplos domínios conectados em uma mesma sessão, como ocorre quando o usuário solicita hospedagem, busca atrações e reserva transporte de maneira sequencial. Ao todo, o corpus reúne 10.438 diálogos, dos quais 3.406 são de domínio único e 7.032 são multidomínio, totalizando 115.434 turnos. A extensão das inte-

rações se evidencia pelo fato de cerca de 70% dos diálogos possuírem mais de dez turnos, o que indica maior necessidade de manutenção de contexto.

O MultiWOZ também inclui atos de diálogo, estados, domínios, slots e valores associados às metas dos usuários, permitindo seu uso em tarefas como rastreamento de estado do diálogo, seleção de ações e geração de respostas. Embora seja um conjunto de tamanho médio e originalmente em inglês, suas anotações agregam valor para sistemas conversacionais e permitem adaptação ao português brasileiro. Ao mesmo tempo, o uso do MultiWOZ em outro idioma evidencia desafios de tradução e adaptação, uma vez que as falas devem preservar fluência textual e alinhamento entre enunciados, slots, valores, atos e estados de diálogo.

A comparação é sintetizada na Tabela 3.5, que compara os conjuntos de dados relacionados que servem como referência para a discussão sobre diálogos. A comparação considera idioma, tamanho, forma de geração, domínio, fonte dos dados, tipo de interação, presença de diálogos, estruturação, anotação e suporte a aumento de dados.

O *Ubuntu Dialogue Corpus* e o MultiWOZ são em inglês, possuem interações H2H e são multiturno, mas diferem em escala, domínio, anotações e estrutura. O primeiro possui grande volume, específico do domínio de suporte ao Ubuntu e derivado de logs de IRC, enquanto o segundo é multidomínio, de tamanho médio, estruturado e anotado. As anotações do MultiWOZ cobrem atos de diálogo, estados, domínios, slots e valores, favorecendo tarefas supervisionadas como rastreamento de estado do diálogo, seleção de ações e geração de respostas.

| **Categoria** | **Relacionado** | **Relacionado** |
|---|---|---|
| **Trabalho** | Ubuntu Dialogue Corpus | MultiWOZ |
| **Idioma** | Inglês | Inglês |
| **Tamanho** | Grande | Médio |
| **Tipo de Geração** | Coleta | Crowdsourcing |
| **Domínio** | SO Ubuntu | Multidomínio |
| **Fonte de Dados** | Logs IRC Ubuntu | Crowdsourcing |
| **Tipo de Interação** | H2H | H2H |
| **Diálogos** | Sim | Sim |
| **Multiusuário** | Parcial | Não |
| **Multiturno** | Sim | Sim |
| **Estratégia** | N/A | N/A |
| **Estruturado** | Sim | Sim |
| **Anotado** | Não | Sim |
| **Aumento de Dados** | Não | Não |

Tabela 3.5: Comparação entre conjuntos de dados de diálogos relacionados.

A Tabela 3.6 apresenta os conjuntos de dados propostos nesta tese. Em contraste com

os trabalhos relacionados, todos os conjuntos dessa tabela estão disponíveis em português e contêm diálogos H2H multiturno.

Os conjuntos Adrenaline Dataset e OuterSpace Dataset são de grande volume, específicos de domínio e construídos por coleta web a partir de fóruns XenForo. Ambos são multiusuário, multiturno, estruturados e obtidos por estratégia automática baseada em URLs semente, mas não são anotados. Esses conjuntos se aproximam do *Ubuntu Dialogue Corpus* por partirem de interações reais, mas diferem por serem em português e por utilizarem fóruns online em vez de logs de IRC.

O MultiWOZ-PTBR é uma tradução do MultiWOZ, preservando a característica de conjunto estruturado e anotado. Por isso, mantém a natureza multidomínio e multiturno do recurso original, mas passa a atender ao requisito de idioma considerado nesta tese.

O Nutritious Food representa uma estratégia diferente das anteriores, pois é um conjunto pequeno, específico de domínio e gerado por *crowdsourcing*. Ele é estruturado, anotado, multiusuário e multiturno, além de ser o único entre os conjuntos propostos que apresenta aumento de dados. Dessa forma, complementa os conjuntos de grande volume obtidos por coleta web e o conjunto traduzido a partir do MultiWOZ.

| **Categoria** | **Proposta** | **Proposta** | **Proposta** | **Proposta** |
|---|---|---|---|---|
| **Trabalho** | Adrenaline Dataset | OuterSpace Dataset | MultiWOZ PTBR | Nutritious Food |
| **Idioma** | Português | Português | Português | Português |
| **Tamanho** | Grande | Grande | Médio | Pequeno |
| **Tipo Geração** | Coleta web | Coleta web | Tradução | Crowdsourcing |
| **Domínio** | Específico | Específico | Multidomínio | Específico |
| **Fonte de Dados** | XenForo Adrenaline | XenForo OuterSpace | MultiWOZ | Crowdsourcing |
| **Tipo Iteração** | H2H | H2H | H2H | H2H |
| **Diálogos** | Sim | Sim | Sim | Sim |
| **Multiusuário** | Sim | Sim | Não | Sim |
| **Multiturno** | Sim | Sim | Sim | Sim |
| **Estratégia** | Automática com URLs semente | Automática com URLs semente | N/A | Interface própria |
| **Estruturado** | Sim | Sim | Sim | Sim |
| **Anotado** | Não | Não | Sim | Sim |
| **Aumento de Dados** | Não | Não | Não | Sim |

Tabela 3.6: Comparação entre conjuntos de dados de diálogos propostos.

Em conjunto, as duas tabelas mostram que os recursos relacionados oferecem referências importantes, mas não cobrem simultaneamente os requisitos de português, diálogos H2H, múltiplos turnos, múltiplos usuários, estruturação e anotação. As propostas desta tese atuam sobre essa lacuna por meio de estratégias complementares: mineração de fóruns reais, adaptação de um conjunto anotado consolidado e geração controlada.

## 3.2 Privacidade e Vazamento de Dados em Aprendizado Federado

Embora o aprendizado federado reduza a exposição direta de dados sensíveis ao evitar sua centralização, essa propriedade não garante a privacidade do modelo treinado. Gradientes, atualizações locais e comportamentos memorizados podem revelar informações sobre exemplos de treinamento, sobretudo quando atualizações federadas, memorização em modelos de linguagem e participantes maliciosos interagem no mesmo protocolo.

Esta seção organiza os riscos de privacidade em FL a partir de quatro eixos principais: inferência de pertencimento, reconstrução por gradientes e atualizações compartilhadas, ataques conduzidos por participantes maliciosos e extração de conteúdo memorizado em modelos de linguagem. Em seguida, são discutidas defesas baseadas em privacidade diferencial e sanitização textual.

**Inferência de pertencimento**

A inferência de pertencimento constitui uma primeira classe de vazamento em modelos treinados, pois permite ao adversário inferir se um registro específico participou do treinamento, ainda que não recupere diretamente o conteúdo desse registro. Shokri et al. [Shokri et al., 2017] formalizam esse risco em cenários de caixa-preta, mostrando que diferenças de confiança e generalização podem revelar a presença de exemplos individuais no conjunto de treino.

Para isso, os autores treinam modelos de sombra e usam as diferenças entre as saídas produzidas para exemplos vistos e não vistos para construir um classificador de ataque, explorando sinais de confiança, generalização imperfeita e sobreajuste. A inferência de pertencimento representa uma classe específica de risco de privacidade, distinta da reconstrução de dados a partir de gradientes e da extração gerativa de conteúdo memorizado. Nesse tipo de ataque, o vazamento não consiste necessariamente em recuperar o conteúdo do exemplo, mas em revelar sua presença no conjunto de treinamento.

Os resultados indicam que o ataque pode ser efetivo em cenários práticos de caixa-preta. Em modelos multiclasse treinados sobre registros de compras, a precisão mediana chegou a 94% no Google Prediction API e a 74% na Amazon Machine Learning. Em dados hospitalares do Texas, a precisão ficou acima de 70%, evidenciando risco em domínios sensíveis.

Esse resultado evidencia uma exposição inferencial sem reconstrução do conteúdo, servindo como contraste para a tese, que se concentra em vazamento textual gerativo de entidades sensíveis.

**Reconstrução por gradientes e atualizações compartilhadas**

Zhu, Liu e Han [Zhu et al., 2019], em *Deep Leakage from Gradients*, avaliam se o compartilhamento de gradientes preserva a privacidade em treinamento distribuído ou colaborativo. Os autores concluem que esses gradientes podem conter sinal suficiente para revelar dados privados usados localmente por um cliente. O ataque assume que o adversário tem acesso aos gradientes compartilhados, ao modelo e aos pesos correntes. A

partir disso, inicializa entradas e rótulos artificiais e os otimiza para minimizar a distância entre os gradientes produzidos por esses dados artificiais e os gradientes reais observados.

Embora o ataque seja validado também em visão computacional, o ponto mais relevante para esta tese está nos experimentos com texto. Em uma tarefa de modelo de linguagem mascarado com BERT, os autores mostram que o DLG consegue recuperar sequências textuais a partir dos gradientes compartilhados. Como palavras são unidades discretas, o ataque é adaptado para otimizar *embeddings* artificiais.

Após a otimização, os vetores recuperados são convertidos de volta para palavras por meio da entrada mais próxima na matriz de *embeddings* do BERT. A avaliação é feita token a token, comparando a sentença reconstruída com o texto original usado no treinamento. Os tokens inicialmente aparecem como ruído, mas trechos corretos surgem por volta das iterações 20 e 30. Ao final, o conteúdo principal das frases é reconstruído com alta fidelidade, com poucas falhas ligadas à tokenização.

A análise de custo e defesas indica que lotes maiores tendem a exigir mais iterações para reconstrução. O trabalho também identifica que defesas simples, como ruído moderado ou meia precisão, não impedem o vazamento, enquanto a poda de gradientes passa a ser efetiva quando remove mais de 20% dos valores.

Geiping et al. [Geiping et al., 2020] aprofundam a discussão sobre inversão de gradientes em aprendizado federado, mostrando que atualizações compartilhadas podem revelar exemplos de treinamento mesmo em arquiteturas complexas. Os autores usam uma perda baseada em similaridade de cosseno, combinada com otimização por Adam e regularização por variação total, o que torna a reconstrução mais estável em redes profundas.

Os resultados indicam que a privacidade pode ser violada mesmo quando o modelo já foi treinado e mesmo em arquiteturas profundas, como redes neurais residuais. O estudo também mostra que aumentar largura ou profundidade da rede não elimina o vazamento. Em alguns casos, redes mais largas produzem reconstruções de maior qualidade, embora com maior custo computacional para o atacante. Estratégias como usar lotes maiores ou múltiplos passos locais antes de compartilhar a atualização também reduzem a qualidade média da reconstrução, mas não oferecem garantia de proteção.

Em conjunto, DLG e os ataques posteriores de inversão de gradientes mostram que a ameaça se desloca dos dados brutos para as atualizações compartilhadas, que podem preservar sinal suficiente para reconstrução mesmo sem centralização dos exemplos.

O estudo de Bagdasaryan et al. [Bagdasaryan et al., 2020] analisa ataques de *backdoor* em aprendizado federado. Os autores mostram que participantes maliciosos podem manipular atualizações locais para inserir comportamentos escolhidos no modelo global. O ataque proposto é baseado em *model replacement*. Nesse caso, o cliente comprometido treina localmente um modelo com o comportamento malicioso e escala sua atualização antes da agregação, de modo que a média global incorpore o modelo do atacante.

Os autores avaliam o ataque em classificação de imagens e previsão de palavras. Nos experimentos, um único participante malicioso em uma rodada já pode fazer a tarefa de *backdoor* atingir 100% de acerto no modelo global. Em previsão de texto com 80.000 participantes, 8 usuários comprometidos foram suficientes para atingir 50% de acerto no *backdoor*, enquanto o envenenamento tradicional de dados exigiria cerca de 400 usuários maliciosos para resultado semelhante.

As estratégias *train-and-scale* e *constrain-and-scale* são avaliadas como formas de contornar detectores de anomalias no servidor. A primeira treina o modelo malicioso livremente e depois escala a atualização até o limite aceito pela defesa. Nesse cenário, o ataque pode manter atualizações aparentemente normais e ainda atingir 50% de acerto no *backdoor* em previsão de palavras.

A estratégia *constrain-and-scale* é mais efetiva contra detectores sofisticados, como os baseados em similaridade de cosseno. Nesse caso, a restrição de evasão entra na própria função de perda durante o treinamento local, em vez de aparecer apenas como um ajuste posterior de escala. Com isso, o cliente malicioso preserva maior similaridade com o modelo global anterior e mantém alta precisão na tarefa de *backdoor*.

Embora o objetivo do trabalho seja a inserção de *backdoors*, o resultado demonstra que um cliente isolado pode deslocar o comportamento do modelo global por meio da agregação. Essa capacidade é o elo conceitual entre ataques de *model replacement* e cenários em que um participante malicioso induz padrões destinados a aumentar a memorização e a posterior extração de entidades sensíveis.

**Memorização e extração em modelos de linguagem**

O trabalho de Carlini et al. [Carlini et al., 2019] é uma referência central para a avaliação de memorização não intencional em modelos generativos de sequência. Os autores investigam em que condições redes neurais de linguagem podem memorizar informações raras ou únicas presentes nos dados de treinamento, como senhas, números de cartão de crédito, identificadores pessoais ou mensagens privadas. Neste trabalho, o foco do estudo está em medir quando um segredo específico passa a ser suficientemente favorecido pelo modelo a ponto de poder ser recuperado.

A metodologia proposta introduz sequências raras artificiais no conjunto de treinamento e mede o grau de memorização por meio da métrica de exposição, estimando a redução da incerteza do atacante ao tentar adivinhar um segredo depois de observar o comportamento do modelo, usando a perplexidade como sinal de quão provável a sequência se tornou para a rede. Os experimentos mostram que a memorização pode ocorrer antes mesmo de sinais clássicos de *overfitting*. Em modelos de linguagem, sequências raras repetidas poucas vezes já podem atingir níveis elevados de exposição, e valores altos dessa métrica se traduzem em extração prática.

Para demonstrar essa relação, os autores propõem um algoritmo de busca que reduz substancialmente o espaço de tentativas necessário para recuperar segredos. Em um estudo com e-mails reais do conjunto Enron [Klimt and Yang, 2004], o método permitiu recuperar informações sensíveis, como números de cartão de crédito e de seguro social, presentes nos dados de treinamento.

Defesas tradicionalmente associadas à redução de sobreajuste, como *dropout*, *weight decay* e quantização, não são suficientes para impedir a memorização de segredos. Entre as defesas avaliadas, o treinamento com privacidade diferencial reduz de forma mais efetiva a exposição, embora afete a utilidade.

Hu et al. [Hu et al., 2025] investigam extração de dados privados entre clientes durante refinamento federado de LLMs em cenários *cross-silo*. A ameaça envolve um participante semi-honesto que tenta recuperar PII de outros clientes a partir do modelo global, usando

apenas informações disponíveis localmente.

O ataque explora prefixos contextuais associados a entidades sensíveis. Neste cenário, o atacante consulta o modelo com trechos que antecedem PII em seus próprios dados, prioriza prefixos frequentes e, em uma variante mais forte, aplica LAFt para reforçar localmente associações entre prefixos e entidades, deslocando o risco de privacidade para a etapa de geração. Assim, o modelo global pode completar padrões textuais com informações memorizadas de participantes que nunca compartilharam seus exemplos brutos.

Os resultados mostram que a extração pode atingir níveis expressivos em um corpus jurídico anotado com categorias granulares de PII. No Qwen1-8B, a cobertura foi de 22,93% com prefixos contextuais padrão e chegou a 56,57% com prefixos expandidos, embora com queda forte de eficiência devido ao aumento no número de consultas. A técnica LAFt elevou a cobertura para 28,30%, e categorias como nomes, endereços e datas de nascimento apareceram entre as mais vulneráveis. O estudo também relata vazamento em modelos como Llama-2-7B e OLMo2-1124-7B após uma rodada de ajuste federado, sugerindo que o fenômeno não depende apenas de contaminação prévia do pré-treinamento.

Para esta tese, esse trabalho é um comparador direto por tratar de vazamento textual em LLMs federados e por adotar um atacante com acesso limitado aos próprios dados. A principal diferença está no foco da contribuição. O estudo avalia extração entre clientes e usa correspondência exata como critério de vazamento. Esta tese, por sua vez, analisa entidades sensíveis em português brasileiro e investiga se a intervenção de um cliente auxiliar adversário aumenta a reprodução desses dados.

Nesta tese, as consultas direcionadas são realizadas por um avaliador confiável separado, com acesso aos nomes e aos registros corretos das vítimas, e não pelo cliente auxiliar.

**Defesas formais e transformação textual pré-treinamento**

Os ataques discutidos mostram que, em FL, manter os dados nos clientes não elimina o risco de vazamento, pois gradientes, atualizações e comportamentos memorizados ainda podem expor informações sensíveis. Por isso, a discussão passa das superfícies de ataque para o problema de mitigação desses vazamentos.

Nesse contexto, a substituição semântica investigada nesta tese é tratada como uma forma específica de transformação pré-treinamento, preservando plausibilidade linguística e estrutura de entidade. Isso permite avaliar a troca entre redução de correspondência exata e manutenção de padrões textuais sensíveis.

Sousa e Kern [Sousa and Kern, 2023] apresentam uma revisão sistemática sobre métodos de aprendizado profundo para preservação de privacidade em processamento de linguagem natural de trabalhos publicados entre 2016 e 2020. O estudo demonstra que a proteção de dados textuais envolve tanto mecanismos aplicados antes do treinamento quanto estratégias usadas durante a modelagem, a inferência e a verificação de riscos.

Entre as abordagens discutidas, técnicas de anonimização, desidentificação, mascaramento e transformação textual se destacam, pois procuram reduzir a presença de informações identificáveis antes do treinamento ou do compartilhamento dos dados. O trabalho também considera mecanismos baseados em DP, criptografia e computação segura, além

de estratégias de verificação voltadas a detectar conteúdo sensível, medir memorização e avaliar riscos de exposição. Em conjunto, essas abordagens mostram que a preservação de privacidade em PLN pode atuar tanto no nível dos dados quanto durante o treinamento, a inferência e a auditoria dos modelos.

Essa revisão também destaca que a preservação de privacidade em linguagem natural tende a envolver uma troca entre proteção, utilidade e custo computacional. Transformações textuais muito agressivas podem remover informações identificáveis, mas também degradar a semântica, a fluidez e o desempenho dos modelos. Por outro lado, técnicas com garantias formais, como DP, podem reduzir riscos de rastreabilidade e memorização, mas frequentemente introduzem ruído, sobrecarga e perda de utilidade.

Abadi et al. [Abadi et al., 2016] propõem o treinamento de redes neurais com garantias de privacidade diferencial, introduzindo uma versão privada da descida de gradiente estocástico. O método limita a contribuição de cada exemplo por meio de *clipping* das normas dos gradientes individuais e adiciona ruído calibrado antes da atualização dos parâmetros. Com isso, o treinamento passa a controlar quanto a presença ou ausência de um registro específico pode influenciar o modelo final.

Além do mecanismo de treinamento, o trabalho apresenta o *moments accountant*, uma técnica de contabilidade de privacidade que acompanha a perda acumulada ao longo das iterações. Essa contribuição permite estimar o orçamento de privacidade consumido durante o treinamento, evitando análises excessivamente conservadoras. Nos experimentos, os autores mostram que é possível treinar modelos úteis sob privacidade diferencial, mas também evidenciam que a proteção introduz perdas de desempenho, especialmente em tarefas mais complexas.

Este trabalho fundamenta o uso de Differential Privacy Stochastic Gradient Descent (DP-SGD) como defesa formal baseada em *clipping*, ruído calibrado e contabilidade de privacidade. Essa abordagem funciona como linha de base com garantias formais no protocolo experimental e contrasta com a substituição semântica proposta por atuar durante a otimização, e não antes do treinamento local. Essa diferença explicita o principal trade-off avaliado, pois a proteção diferencialmente privada pode reduzir a utilidade do modelo e aumentar o custo de treinamento.

Yousefpour et al. [Yousefpour et al., 2021] operacionalizam, em PyTorch, o treinamento com DP usado como linha de base prática. O trabalho mostra que aplicar DP-SGD em modelos reais exige cálculo de gradientes por amostra, contabilidade de privacidade e restrições sobre camadas incompatíveis, tornando explícito o custo de transformar a garantia formal em um protocolo treinável. Embora esta abordagem reduza parte da sobrecarga em tarefas convencionais e com lotes maiores, os resultados indicam custos elevados em arquiteturas sequenciais e em camadas de *embedding*, ponto especialmente importante para aplicações em linguagem.

**Comparação**

A comparação a seguir sintetiza como os trabalhos discutidos se relacionam com o escopo experimental desta tese. A Tabela 3.7 destaca quatro dimensões: uso de FL, presença de LLMs ou dados textuais, existência de um cliente atacante e consideração explícita de dados em português brasileiro. Essa organização evidencia que os traba-

lhos anteriores cobrem partes importantes do problema, mas não combinam todos esses elementos no mesmo cenário experimental. Estudos de inferência de pertencimento e reconstrução por gradientes ajudam a caracterizar o risco de privacidade, enquanto trabalhos sobre *backdoor*, memorização e extração textual se aproximam mais do modelo de ameaça considerado nesta tese. Ainda assim, a combinação entre FL, modelos de linguagem, cliente malicioso e entidades sensíveis em PT-BR permanece como o principal diferencial do estudo proposto.

| Trabalho | FL | LLMs/texto | Cliente atacante | PT-BR |
|---|---|---|---|---|
| Shokri et al. [Shokri et al., 2017] | | | | |
| Zhu, Liu e Han [Zhu et al., 2019] | ✓ | ✓ | | |
| Geiping et al. [Geiping et al., 2020] | ✓ | | | |
| Bagdasaryan et al. [Bagdasaryan et al., 2020] | ✓ | ✓ | ✓ | |
| Carlini et al. [Carlini et al., 2019] | | ✓ | | |
| Hu et al. [Hu et al., 2025] | ✓ | ✓ | ✓ | |
| **Esta tese** | ✓ | ✓ | ✓ | ✓ |

Tabela 3.7: Comparação do escopo experimental dos trabalhos de ataque com esta tese.

A Tabela 3.8 complementa essa comparação ao deslocar o foco do escopo experimental para o tipo de vazamento estudado, comparando os mesmos trabalhos. Nesse caso, a distinção central está entre ataques que apenas indicam a presença de um exemplo no treinamento, ataques que buscam reconstruir dados a partir de atualizações e ataques que exploram memorização para extrair conteúdo sensível. Essa leitura mostra que a inferência de pertencimento e a reconstrução por gradientes representam riscos relevantes, mas distintos da exposição textual de entidades sensíveis.

A tese se posiciona no eixo de memorização e extração de PII, pois investiga a reprodução dos registros das vítimas pelo modelo global e o possível efeito adicional da intervenção adversária do cliente auxiliar.

A última comparação, apresentada na Tabela 3.9, sintetiza as defesas relacionadas ao problema investigado e as avaliadas nesta tese. A tabela separa abordagens textuais aplicadas antes do treinamento, mecanismos formais baseados em DP, substituição semântica e avaliação explícita de extração em FL. Essa distinção é importante porque as defesas discutidas na literatura atuam em níveis diferentes. Revisões sobre privacidade em PLN organizam técnicas de anonimização, mascaramento e transformação textual. Métodos baseados em DP oferecem garantias formais durante o treinamento. Bibliotecas como Opacus tornam essas garantias operacionalmente viáveis em PyTorch. Nesta tese, são avaliadas duas defesas: a substituição semântica antes do treinamento local e DP-AdamW, implementado com Opacus durante o treinamento. A primeira preserva a plausibilidade textual das entidades substituídas, enquanto a segunda permite comparar

| Trabalho | Pertenc | Reconstr | Memoriz | Extração de PII |
|---|---|---|---|---|
| Shokri et al. [Shokri et al., 2017] | ✓ | | | |
| Zhu, Liu e Han [Zhu et al., 2019] | | ✓ | | |
| Geiping et al. [Geiping et al., 2020] | | ✓ | | |
| Bagdasaryan et al. [Bagdasaryan et al., 2020] | | | | |
| Carlini et al. [Carlini et al., 2019] | | | ✓ | ✓ |
| Hu et al. [Hu et al., 2025] | | | ✓ | ✓ |
| **Esta tese** | | | ✓ | ✓ |

Tabela 3.8: Comparação dos tipos de vazamento abordados nos trabalhos relacionados.

a proteção textual com uma abordagem baseada em DP.

| Trabalho | Defesa textual | Antes do treino | Subst. semântica | Extração em FL | DP |
|---|---|---|---|---|---|
| Sousa e Kern [Sousa and Kern, 2023] | ✓ | ✓ | | | |
| Abadi et al. [Abadi et al., 2016] | | | | | ✓ |
| Yousefpour et al. [Yousefpour et al., 2021] | | | | | ✓ |
| **Esta tese** | ✓ | ✓ | ✓ | ✓ | ✓ |

Tabela 3.9: Comparação entre as defesas relacionadas e as defesas avaliadas nesta tese.

Em conjunto, as três tabelas mostram que os trabalhos relacionados cobrem componentes importantes do problema, mas deixam uma lacuna na interseção entre FL, LLMs, cliente malicioso, dados sensíveis em português brasileiro e defesa textual antes do treinamento.

Essa interseção define a lacuna explorada pela tese: avaliar, no mesmo protocolo federado, o efeito da intervenção adversária do cliente auxiliar sobre a reprodução, a extração textual de PII em PT-BR sob consultas do avaliador confiável e a mitigação local por substituição semântica.

# Capítulo 4

# Geração de Conversas Multiturno e Multiparticipante

**Pergunta de Pesquisa 1:** Como bases de dados já existentes em português brasileiro, compostas por interações comunicativas entre múltiplos participantes, podem ser processadas, segmentadas e estruturadas de modo a constituírem diálogos multiturnos adequados ao treinamento de Large Language Models?

A utilização de conjuntos de dados adequados é um fator decisivo para o desenvolvimento e avaliação de sistemas baseados em linguagem natural, influenciando diretamente o desempenho e a capacidade de generalização dos modelos. Neste capítulo, propomos uma metodologia para a construção de conjuntos de dados conversacionais que preservam propriedades estruturais da interação, especialmente múltiplos turnos, múltiplos participantes e relações estruturais e semânticas entre interações, resultando na geração de conjuntos de alta qualidade para viabilizar o treinamento de LLMs com maior robustez. Como parte dessa proposta, instanciamos a metodologia em uma ferramenta e a aplicamos na geração de novos conjuntos.

Apesar da relevância desse tipo de conjunto de dados, sua disponibilidade ainda impõe restrições importantes em muitas aplicações. A Tabela 4.1 apresenta a quantidade de conjuntos de dados disponíveis nos vinte principais idiomas no site Hugging Face em março de 2026 [Hugging Face, 2026], considerando todos os tipos disponíveis para cada idioma. A coluna "Proporção" representa a razão entre a quantidade de conjuntos de dados de um idioma e a quantidade correspondente ao idioma imediatamente subsequente na tabela. Como a tabela exibe apenas os vinte primeiros idiomas, a proporção do Tcheco (1,08) é calculada em relação ao 21º idioma, que possui 696 conjuntos de dados. É possível observar uma forte dominância do inglês, com 3,35 vezes mais conjuntos de dados do que o francês, que ocupa a segunda posição em disponibilidade. Também é possível observar uma tendência acentuada de redução nessa quantidade nos demais idiomas.

A Figura 4.1 apresenta, em um gráfico de dispersão, uma visão complementar à tabela apresentada. É possível observar que a oferta de conjuntos de dados não segue o crescimento encontrado na base de falantes, mas, sim, se concentra desproporcionalmente em alguns poucos idiomas. A quantidade disponível nos dois primeiros idiomas, inglês e francês, é maior do que a soma de todos os demais idiomas apresentados.

A diferença encontrada é particularmente crítica em sistemas conversacionais, nos

| Posição | Idioma | Conjuntos de Dados Disponíveis | Falantes (milhão) | Proporção |
|---|---|---|---|---|
| 1 | Inglês | 42687 | 1528 | 3,35 |
| 2 | Francês | 12751 | 312 | 3,03 |
| 3 | Chinês | 4202 | 1184 | 1,38 |
| 4 | Espanhol | 3040 | 558 | 1,14 |
| 5 | Alemão | 2662 | 134 | 1,18 |
| 6 | Japonês | 2248 | 126 | 1,15 |
| 7 | Português | 1956 | 267 | 1,07 |
| 8 | Italiano | 1823 | 66 | 1,09 |
| 9 | Árabe | 1675 | 454 | 1,01 |
| 10 | Coreano | 1664 | 82 | 1,02 |
| 11 | Vietnamita | 1625 | 97 | 1,17 |
| 12 | Indonésio | 1393 | 252 | 1,11 |
| 13 | Turco | 1254 | 91 | 1,06 |
| 14 | Hindi | 1186 | 609 | 1,03 |
| 15 | Holandês | 1153 | 25 | 1,04 |
| 16 | Polonês | 1106 | 43 | 1,34 |
| 17 | Persa | 827 | 83 | 1,01 |
| 18 | Tailandês | 819 | 71 | 1,05 |
| 19 | Bengali | 779 | 284 | 1,04 |
| 20 | Tcheco | 752 | 12 | 1,08 |

Tabela 4.1: Conjuntos de Dados disponíveis por idioma em março de 2026.

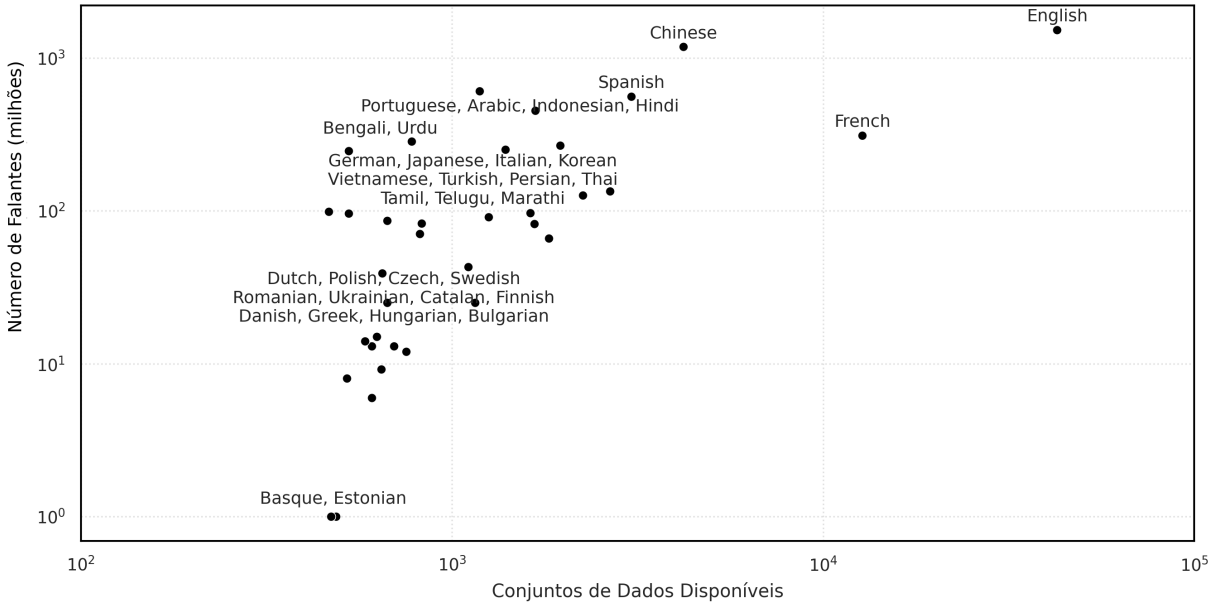



Figura 4.1: Distribuição dos conjuntos de dados disponíveis em função do número de falantes nativos (L1) e não nativos (L2) de cada idioma.

quais o desempenho não depende apenas da disponibilidade de conjuntos de dados no idioma-alvo, mas também da preservação da estrutura interacional observada no cenário

de uso. Em aplicações baseadas em pergunta e resposta, modelos treinados com conjuntos que mantêm encadeamento entre mensagens, múltiplos turnos e múltiplos participantes tendem a apresentar maior robustez, por refletirem com mais fidelidade a dinâmica das interações reais.

Apesar de sua grande relevância, a geração de conjuntos de dados conversacionais que atendam a tais requisitos envolve desafios que limitam sua disponibilidade. Os principais desafios são:

1. **Dinâmica multiturno:** Conjuntos de dados que contenham múltiplos turnos de mensagens geralmente são limitados em tamanho, dada a complexidade de sua geração, que muitas vezes é manual ou sintética. Por sua vez, a identificação automática pode ser desafiadora devido à presença de referências implícitas, o que compromete a coerência ao longo da conversa.

2. **Dinâmica multiparticipante:** Conversas complexas criadas por humanos, tanto no mundo real quanto na internet, geralmente contam com a participação de diversas pessoas interagindo entre si e compartilhando respostas e opiniões distintas. Essa característica é extremamente escassa nos conjuntos de dados disponíveis na literatura.

3. **Escala:** LLMs se beneficiam diretamente do tamanho dos conjuntos de dados utilizados. Apesar disso, idioma, estrutura e domínio específico são fatores que rapidamente limitam a disponibilidade de conjuntos adequados ao modelo em treinamento.

Mesmo com essas limitações, a extração sistemática, em larga escala, ainda é um problema em aberto, especialmente no caso de conjuntos de dados conversacionais que preservam a estrutura multiturno e multiparticipante.

Embora seja difícil atender simultaneamente a esses requisitos, essa dinâmica é comum em conversas reais entre humanos, com participantes que se alternam em turnos e cada nova resposta tende a endereçar ou acrescentar informação à anterior. Como esse padrão se mantém em diferentes meios de comunicação, bases disponíveis na web constituem uma fonte potencial para a extração de conjuntos de dados conversacionais estruturados, desde que o processo de extração mantenha as relações e estruturas discutidas.

Ao longo deste capítulo, são discutidas soluções para os seguintes problemas: (*i*) como construir uma metodologia específica para a criação de conjuntos de dados, utilizando como fonte principal qualquer base que contenha algum tipo de referência entre perguntas e respostas, com identificação de múltiplos participantes e múltiplos fluxos conversacionais; (*ii*) como utilizar fontes não estruturadas para criar conjuntos de dados conversacionais com diálogos multiturnos e multiparticipantes; (*iii*) como desenvolver ferramentas adequadas para gerar tais conjuntos de dados; (*iv*) impacto do uso de conjuntos de dados adequados ao domínio em tarefas de modelagem linguística e classificação, como etapa inicial para avaliar sua utilidade em cenários de treinamento e adaptação de LLMs.

O restante deste capítulo está organizado da seguinte forma: a Seção 4.1 apresenta uma visão geral das abordagens propostas neste trabalho, com destaque para o MCCD como método de extração de diálogos reais. Em seguida, a Seção 4.2 detalha a metodologia

MCCD, seus requisitos, suas etapas de aquisição, anonimização e processamento, além das recomendações de armazenamento para os corpora gerados.

A Seção 4.3 descreve a ferramenta Miner-XenForo, implementação do MCCD para fóruns baseados em XenForo, incluindo sua arquitetura, seu pipeline e suas decisões de implementação. Na sequência, a Seção 4.4 apresenta os conjuntos de dados criados com a metodologia proposta, suas fontes, seus volumes e suas características quantitativas.

Por fim, a Seção 4.5 discute a avaliação experimental, os procedimentos adotados e os resultados obtidos na comparação entre modelos em tarefas de classificação. A Seção 4.6 sintetiza as principais contribuições do capítulo.

## 4.1 Visão Geral

Ao longo deste trabalho, a geração de conjuntos de dados é tratada como um problema metodológico central para o treinamento e o uso eficiente de LLMs, especialmente em cenários conversacionais com múltiplos turnos. Em contraste com abordagens que tomam a disponibilidade de dados adequados como premissa, este trabalho assume que a construção desses conjuntos integra o próprio processo científico e pode influenciar diretamente os resultados obtidos pelos modelos treinados.

As soluções propostas neste capítulo se organizam em abordagens sistemáticas para a criação de conjuntos de dados capazes de atender a características específicas, como interações com múltiplos turnos e múltiplos participantes.

A abordagem proposta é o MCCD, concebido como uma metodologia para extrair conjuntos de dados conversacionais a partir de dados semiestruturados que contenham algum tipo de relação entre perguntas e respostas. Esse processo é viabilizado pela identificação automática de fluxos conversacionais, múltiplos turnos e múltiplos participantes, mesmo quando tais relações não estão anotadas diretamente na fonte original. O MCCD fundamenta tanto o desenvolvimento da ferramenta Miner-XenForo, elaborada com base nas diretrizes propostas, quanto a criação de dois conjuntos de dados extraídos de fóruns online.

O MCCD foi desenvolvido de forma modular e é dividido em três estágios principais: aquisição, anonimização e processamento. A aquisição é responsável pelo acesso e pela obtenção dos dados necessários. O estágio de anonimização orienta o tratamento de informações identificáveis, conforme o escopo de cada implementação. O estágio de processamento identifica as conversas e realiza o tratamento dos dados recebidos dos estágios anteriores.

Essa organização modular permite adaptações em cenários específicos nos quais a anonimização ou algum outro estágio não seja necessário, sem comprometer o funcionamento geral da proposta.

Como saída final da metodologia, ou das ferramentas que a implementam, é obtido um conjunto de dados com todas as mensagens identificadas e organizadas cronologicamente. O conjunto resultante inclui as conversas identificadas e a indicação dos participantes que integram cada conversa e geraram cada mensagem.

## 4.2 Criação de Corpus Conversacional Multiturno

Esta seção define a metodologia MCCD, proposta para gerar conjuntos de dados por meio da identificação de múltiplos fluxos de conversas, de forma independente da fonte de dados. Também discute como adaptar cada estágio e etapa a diferentes cenários.

O principal objetivo desta metodologia é produzir conjuntos de dados compostos não apenas por dados brutos, mas também por diálogos limpos e diferentes fluxos de conversa. Nesse contexto, a identificação de conversas permite maior cobertura do espaço de respostas, aumento da diversidade estrutural e melhor representação de interações com múltiplos participantes. Os diálogos identificados também devem ser multiturnos, com múltiplos usuários, e permitir, de forma simples, a identificação de diferentes maneiras de responder às mesmas mensagens.

A metodologia MCCD define etapas e regras gerais para a construção de conjuntos de dados conversacionais. A implementação dessas etapas considera os requisitos do conjunto de dados e as características da fonte utilizada. A Figura 4.2 apresenta essa relação.

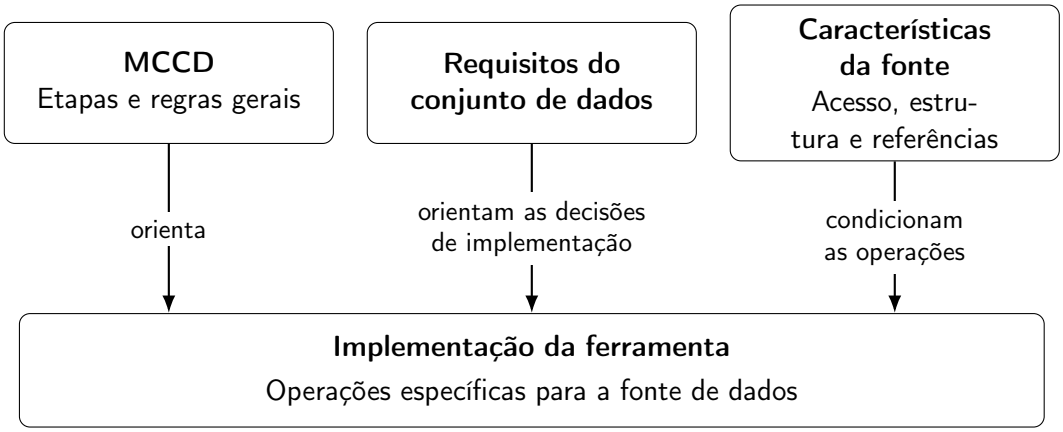



Figura 4.2: Relação entre a metodologia MCCD e sua implementação para uma fonte de dados.

A adaptação ocorre na implementação das operações previstas pela metodologia, como o acesso aos dados, a limpeza das mensagens e a identificação das referências. Essas operações consideram a estrutura da fonte e os requisitos do conjunto de dados, preservando as etapas gerais e as regras de reconstrução dos fluxos conversacionais da MCCD. Dessa forma, a metodologia pode orientar diferentes implementações.

Ferramentas que implementem esta metodologia podem optar por implementar essa metodologia parcial ou integralmente. Implementações que utilizem apenas a etapa inicial, isto é, os estágios relacionados à aquisição, podem se beneficiar de regras maleáveis, por se tratar do único estágio dependente do domínio e da fonte de dados. Isso auxilia na geração de conjuntos de dados estruturados e organizados. Ainda assim, tais implementações não se beneficiarão do principal diferencial da metodologia: a identificação e a estruturação de diálogos multiturnos e multiparticipantes.

Caso a segunda metade seja utilizada como referência, a identificação das conversas será possível em bases de dados que atendam aos requisitos mínimos para essa tarefa. Essa etapa concentra os principais diferenciais e vantagens. Conjuntos de dados gerados

a partir da identificação de conversas resultam em maior cobertura do espaço de respostas, aumento da diversidade estrutural dos diálogos e melhor representação de interações com múltiplos participantes.

Para identificar os fluxos de conversa, é necessário empregar técnicas específicas, discutidas na Seção 4.2.3, capazes de modelar o conjunto de mensagens como um grafo direcionado, no qual cada nó representa uma mensagem e cada aresta representa uma relação explícita de referência, permitindo a identificação de todos os caminhos válidos entre mensagens iniciais e finais. Diferentes fluxos de conversa podem começar e terminar nas mesmas mensagens, mas percorrer caminhos distintos. Isso é essencial para viabilizar a geração de um conjunto de dados de perguntas e respostas com diferentes respostas para as mesmas perguntas ou mensagens.

MCCD foi concebida para oferecer grande versatilidade, com redução de requisitos e dependências para sua implementação. Ainda assim, existem requisitos mínimos para que os conjuntos de dados gerados com base na metodologia atendam aos objetivos propostos e a um padrão mínimo de qualidade. Esses requisitos são descritos a seguir.

- **Requisito Humano:** As mensagens que comporão o conjunto de dados devem ter sido produzidas por humanos.
- **Mecanismo de Referência:** A fonte de dados deve fornecer um mecanismo explícito de referência entre mensagens. Essa referência pode apontar para uma mensagem inteira ou para um trecho específico dela. Esse mecanismo deve ser capaz de referenciar pelo menos uma mensagem.

O requisito de dados humanos, isto é, de texto produzido em interações humano-humano, possibilita que os conjuntos de dados gerados incluam características linguísticas distintas. Em cenários nos quais a fonte original dos dados é baseada em sistemas web, esses dados podem incluir emojis, gírias e marcas de digitação em smartphones. Isso permite que LLMs representem de maneira mais adequada determinadas palavras e sentenças [Sanches et al., 2022b].

Embora mecanismos de referência sejam simples de utilizar e estejam presentes em muitas fontes de dados, ainda pode haver cenários em que essa referência não esteja disponível. Nesses casos, a metodologia pode ser executada parcialmente, com a geração e a limpeza dos dados, mas sem a identificação e a estruturação das conversas com base nessas referências.

O conjunto final deve ser acompanhado de documentação formal que descreva os processos de transformação, anonimização e estruturação, possibilitando a reprodutibilidade da construção e rastreabilidade dos dados. Apesar disso, fontes de dados ativas podem evoluir naturalmente e impossibilitar a reconstrução idêntica em momentos distintos, uma vez que os próprios dados evoluíram. A Figura 4.3 apresenta os três estágios do MCCD.

O Estágio de Aquisição de Dados é responsável por obter os dados nas fontes originais e organizar esses dados de maneira estruturada para dar continuidade ao processo. Este estágio está documentado na Seção 4.2.1.

O Estágio de Anonimização é responsável por anonimizar, mascarar ou remover dados sensíveis, utilizando técnicas que reduzam o impacto semântico e estatístico dos dados

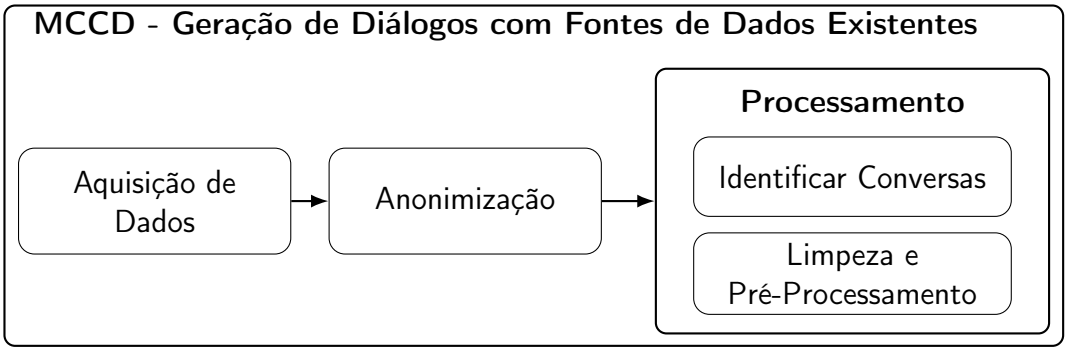



Figura 4.3: Estágios do MCCD.

originais. Esse estágio é essencial para o compartilhamento de novos conjuntos de dados gerados com base nesta metodologia. Sua descrição está na Seção 4.2.2.

Por fim, o Estágio de Processamento é responsável por limpar e organizar os dados brutos, de modo a viabilizar a identificação dos múltiplos fluxos de conversa. Os processos realizados e as técnicas sugeridas são apresentados na Seção 4.2.3.

No contexto do MCCD, uma conversa é definida como um caminho válido em um grafo direcionado de mensagens, contendo pelo menos dois nós conectados por relações de referência. A Figura 4.4 ilustra como diferentes referências podem gerar múltiplas conversas. Cada caminho entre um nó 'Final' e o nó 'Inicial' constitui uma conversa, mesmo que algumas mensagens se repitam.

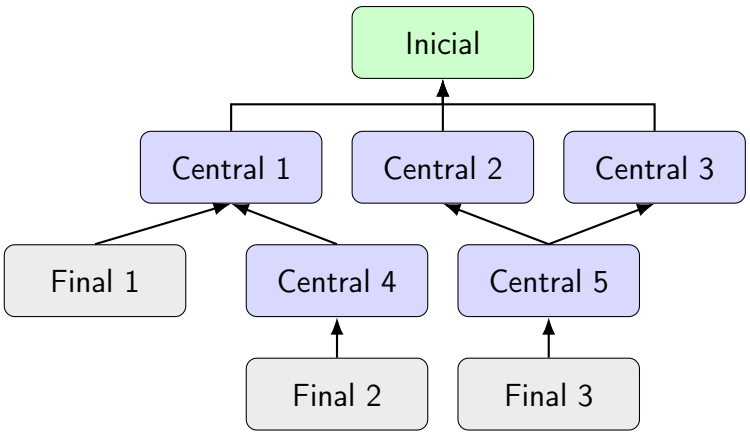



Figura 4.4: Fluxos de conversa entre mensagens

A Figura 4.5 apresenta a implementação da metodologia MCCD na ferramenta Miner-XenForo e sua aplicação aos fóruns Adrenaline e OuterSpace.

O Miner-XenForo implementa as etapas de aquisição, anonimização e processamento da MCCD para fóruns baseados na plataforma XenForo. As características dessa plataforma orientam as operações de acesso, extração das mensagens e identificação das referências, preservando as etapas gerais da metodologia. Sua aplicação aos fóruns Adrenaline e OuterSpace originou os respectivos conjuntos de dados, disponibilizados em versões bruta e processada.

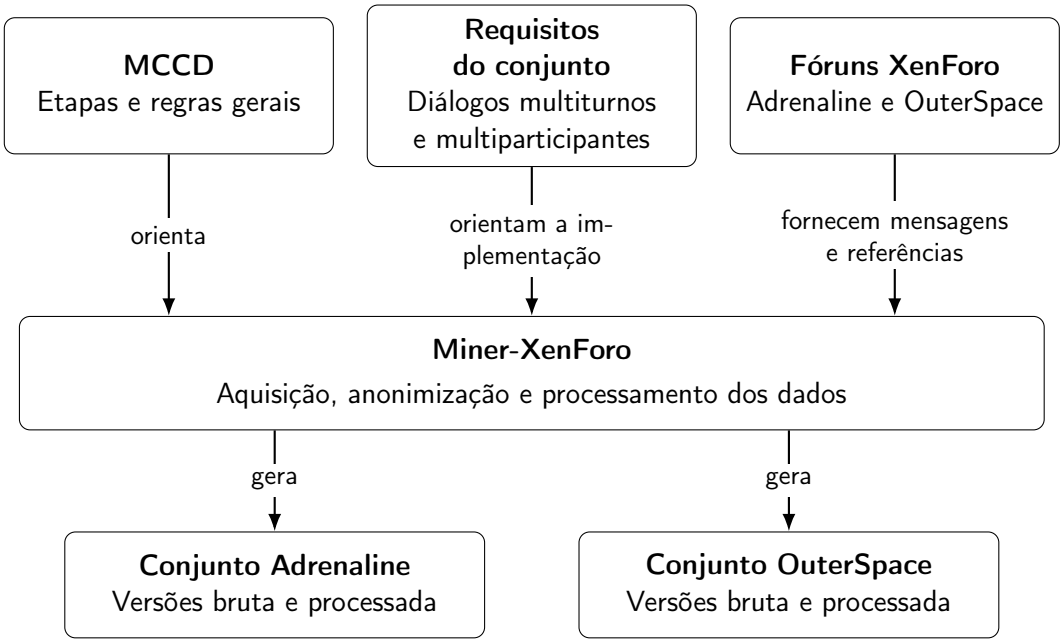



Figura 4.5: Aplicação da metodologia MCCD no Miner-XenForo para a construção dos conjuntos Adrenaline e OuterSpace.

### 4.2.1 Estágio de Aquisição de Dados

O MCCD foi concebido para funcionar de maneira agnóstica à fonte dos dados. Em outras palavras, as definições desta seção foram formuladas para se adaptar a diferentes formas de acesso, estruturas e organizações dos dados nas bases originais. Também devem operar com qualquer idioma e codificação, o que permite sua aplicação em diferentes domínios e formatos de dados.

Os requisitos para a execução deste estágio são a estrutura dos dados e a forma de acesso ao local em que estão armazenados. Ainda assim, algumas aplicações específicas podem demandar parâmetros adicionais, como credenciais de acesso ou mecanismos de paralelização para acelerar a aquisição, uma vez que este estágio costuma estar entre os mais lentos devido às limitações de transferência nas bases de dados. A Figura 4.6 apresenta as etapas executadas neste estágio.

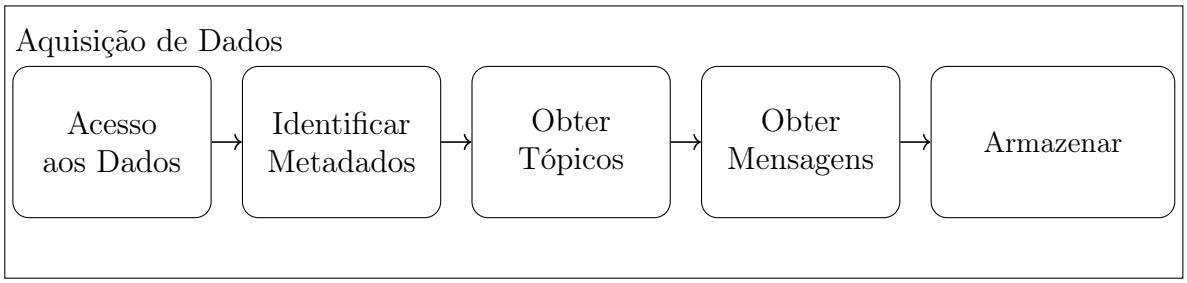



Figura 4.6: Etapas do estágio de aquisição de dados.

Nas definições, cada etapa é delimitada por responsabilidades específicas que permitem a construção iterativa do conjunto de dados sendo obtido.

**Etapa 1 - Acesso aos Dados:** Consiste em realizar o acesso aos dados. A meto-

dologia não define de forma explícita como esse acesso deve ser realizado, deixando essa decisão a cargo da implementação que segue o MCCD. Com isso, sua integração com diferentes bases é preservada e a imposição de restrições à sua utilização é evitada.

**Etapa 2 - Identificar Metadados:** Corresponde à identificação de metadados. Nela, já podem ser identificadas informações relevantes para o conjunto de dados final, como elementos da estrutura original de armazenamento, tópicos, categorias e *timestamps*.

**Etapa 3 - Obter Tópicos:** Consiste na obtenção de tópicos, explorando a organização hierárquica da fonte para estruturar semanticamente os dados coletados. A obtenção de tópicos passa a explorar esse tipo de organização para adicionar semântica aos dados e organizar o material de acordo com a estrutura original. Novamente, essa é uma etapa sensível ao domínio e à fonte de dados, ficando fortemente atrelada à implementação. Ainda assim, esse mapeamento foi incorporado à metodologia para cobrir o maior número possível de etapas observadas em cenários reais e fornecer base teórica para implementações mais adaptativas no futuro.

**Etapa 4 - Obter Mensagens:** É responsável pela obtenção das mensagens. Nesta etapa, são coletadas mensagens individuais ainda não organizadas em fluxos conversacionais no conjunto de dados em construção, mas apenas estruturas de armazenamento e mensagens isoladas que podem fazer referência entre si. A identificação das conversas é realizada posteriormente, em uma etapa específica do estágio de processamento.

Durante a aquisição das mensagens, a estrutura de armazenamento dos dados é percorrida em busca das mensagens relacionadas a um determinado tópico ou categoria. Nesse processo, todos os metadados, participantes, *timestamps*, texto das mensagens e demais informações relevantes devem ser salvos. Em razão do requisito de manter a metodologia agnóstica à fonte dos dados, o MCCD não determina explicitamente como esses dados devem ser percorridos.

**Etapa 5 - Armazenamento:** É responsável exclusivamente pelo armazenamento, sem transformação semântica ou estrutural dos dados coletados. O armazenamento pode ser incremental, permitindo a atualização de conjuntos já existentes. Nesses casos, basta adicionar a diferença correspondente aos novos dados.

Uma vez armazenados de maneira organizada, os dados podem seguir para o estágio de processamento, no qual serão efetivamente transformados em um conjunto de dados estruturado com os fluxos de conversa identificados.

### 4.2.2 Estágio de Anonimização

O estágio de anonimização do MCCD tem como objetivo mitigar riscos de identificação e exposição de informações pessoais nos conjuntos de dados gerados. Essa preocupação é especialmente relevante quando os dados serão compartilhados ou utilizados no treinamento de modelos de linguagem. Entretanto, a execução de procedimentos de remoção, mascaramento ou substituição não permite presumir, por si só, que o conjunto resultante esteja livre de PII. O alcance da proteção depende dos campos examinados, dos mecanismos empregados e da avaliação das informações que permanecem nos dados.

Esse estágio envolve dois processos: a identificação das informações a tratar e a alteração dos campos ou trechos identificados. A análise deve considerar tanto os metadados,

como nomes de usuários e endereços de e-mail, quanto possíveis informações pessoais presentes no conteúdo das mensagens. A metodologia não determina um mecanismo único de identificação e alteração, pois essas decisões dependem da estrutura, do conteúdo e das particularidades da fonte utilizada. Cada implementação deve, portanto, explicitar quais campos são tratados e quais permanecem fora de seu escopo.

Para reduzir alterações indesejadas na semântica das sentenças, recomenda-se que, durante a anonimização automatizada, as informações identificadas sejam substituídas por valores fictícios compatíveis com sua função no texto, como um nome sintético no lugar do nome original. Essa recomendação busca preservar a utilidade linguística dos dados, sem pressupor que a substituição elimine todos os riscos de identificação ou produza, necessariamente, melhoria de desempenho dos modelos.

Quando a identificação consistente dos participantes for necessária à representação das conversas, cada valor original deve manter a mesma correspondência com seu substituto no escopo definido pela implementação. Substituir o mesmo participante por identificadores diferentes a cada ocorrência pode introduzir artificialmente novos participantes e comprometer a interpretação das interações. Quando essa correspondência é mantida por uma tabela que associa valores originais e substitutos, o procedimento deve ser descrito como pseudonimização, sem assumir que os dados se tornaram integralmente anônimos.

Ao concluir o estágio, o formato e a organização dos arquivos devem permanecer compatíveis com os módulos seguintes, limitando-se as alterações aos campos ou trechos selecionados para tratamento. A documentação do conjunto deve distinguir as transformações efetivamente realizadas das recomendações gerais da metodologia.

### 4.2.3 Estágio de Processamento

O Processamento constitui o principal estágio do MCCD, dada a abrangência de suas responsabilidades na identificação e no armazenamento dos principais dados gerados pela metodologia. Quando executado integralmente, cada tópico tende a resultar em diversos arquivos contendo todas as conversas identificadas com duas ou mais mensagens (e seus respectivos participantes).

A Figura 4.7 apresenta as etapas que compõem o estágio de Processamento e como elas interagem entre si. O pipeline segue um fluxo sequencial de transformação de dados entre etapas. As quatro primeiras etapas transformam e geram novos dados que passarão a compor o conjunto final. A última etapa é responsável apenas por armazenar esses dados de maneira estruturada e organizada.

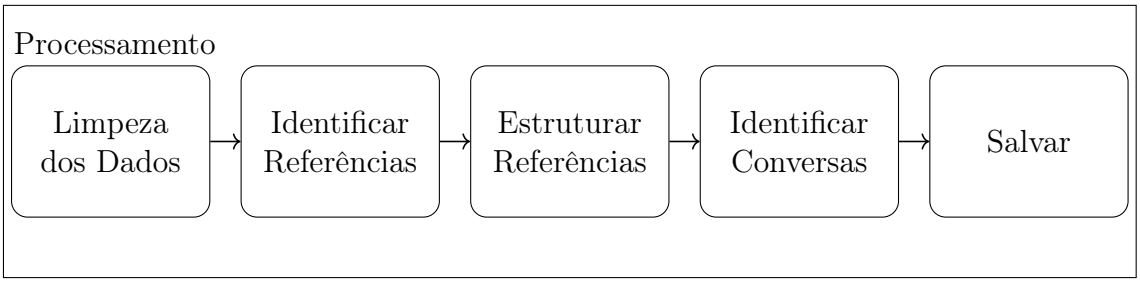



Figura 4.7: Etapas do estágio de processamento.

Os processos deste estágio determinam diretamente a qualidade e estrutura do conjunto de dados final, pois definem quais dados são transformados, preservados ou descartados ao longo deste fluxo.

Uma característica importante desse estágio é a possibilidade de utilizar, como entrada inicial, tanto a saída gerada pelo estágio de Aquisição quanto a do estágio de Anonimização, já que a anonimização não altera a estrutura nem a organização dos dados. Isso amplia a versatilidade da metodologia e permite que, em cenários aplicáveis, o conjunto final de dados seja gerado sem anonimização.

Todas as etapas descritas neste estágio podem ser executadas de maneira sequencial, com conclusão de uma etapa após a outra, ou de maneira paralela, à medida que as mensagens são processadas. Apesar dessa flexibilidade organizacional, introduzida para simplificar implementações futuras, a execução paralela exige mecanismos explícitos de controle de estado, como marcação de etapas por mensagem e armazenamento intermediário, para evitar recomputações e inconsistências na identificação de fluxos. Esse controle é necessário porque a identificação das conversas pode percorrer a mesma mensagem diversas vezes para obter os dados necessários.

Uma vez iniciado o estágio de Processamento, a primeira etapa consiste na reestruturação semântica dos dados. Esta é uma etapa crítica, pois transforma os dados em representações textuais estruturadas, preservando informações semânticas relevantes, permitindo seu uso no treinamento de LLMs. Nesse contexto, ela é responsável por lidar com diversos elementos não textuais, como imagens, ícones, emojis, elementos HTML e afins. Embora esses elementos não constituam o objetivo final de muitos LLMs, sua remoção direta pode comprometer a qualidade final do conjunto de dados gerado, uma vez que eles carregam significado semântico relevante para os textos processados.

A seguir, é apresentada uma lista de exemplos destinada a auxiliar na definição de elementos textuais que possam substituir elementos não textuais, com foco na preservação da semântica do elemento original. Essa lista deve servir como referência e precisa ser refinada de acordo com o domínio, o tipo e a origem dos dados utilizados.

- **<visual>:** Representa conteúdos visuais não textuais, como imagens, figuras, diagramas, capturas de tela ou ilustrações. Metadados opcionais podem indicar o tipo, a posição ou a função semântica do elemento.

- **<symbolic> {} </symbolic>:** Representa símbolos gráficos ou pictográficos que carregam significado contextual, como emojis, ícones ou reações visuais. O tipo ou a categoria semântica do elemento pode ser incluído como metadado.

- **<embedded> {} </embedded>:** Representa conteúdos externos ou multimodais incorporados ao texto, como vídeos, áudios, mapas, visualizações interativas ou outros artefatos não textuais.

- **<reference> {} {} </reference>:** Representa referências externas, internas ou cruzadas; o primeiro campo identifica o destino, como uma URL ou um identificador lógico, e o segundo registra o rótulo textual ou texto-âncora associado.

- **<placeholder>:** Representa a presença de um elemento cujo conteúdo específico é irrelevante ou foi intencionalmente omitido.
- **<quotation> {} </quotation>:** Representa trechos provenientes de outras fontes ou reutilizações explícitas de conteúdo, permitindo identificar repetições e reduzir vieses decorrentes de duplicação textual.
- **<relation> {} </relation>:** Representa relações estruturais ou discursivas entre unidades textuais, como respostas, citações encadeadas ou dependências lógicas entre mensagens.
- **<executable> {} </executable>:** Representa trechos executáveis ou formais, como código, expressões matemáticas, comandos ou pseudocódigo, cujo conteúdo pode ser abstraído conforme o objetivo do processamento.
- **<container> {} </container>:** Representa conteúdos encapsulados integralmente, como documentos anexos, blocos estruturados ou fontes externas incorporadas como uma unidade única.

Para todo conjunto de dados gerado a partir desta metodologia, é necessário documentar integralmente todas as transformações realizadas, sobretudo na etapa de limpeza. Isso permite que pesquisadores que utilizem o conjunto de dados consigam identificar detalhadamente como essa etapa foi conduzida e quais pontos de atenção devem ser considerados.

Uma vez concluída a etapa de limpeza, é possível iniciar a identificação das referências entre as mensagens. Essa etapa tem como objetivo documentar, de maneira estruturada, quais mensagens respondem a outras mensagens. A origem de cada dado pode oferecer diferentes formas de realizar essa identificação. Em fóruns online, por exemplo, é comum o uso de mecanismos de resposta direta entre mensagens, preservando explicitamente as referências entre elas.

A estruturação dessas referências é essencial para a identificação das conversas, pois é a partir delas que se tornam identificáveis os participantes e os diferentes fluxos presentes em cada conversa. No âmbito do MCCD, não se determina uma estrutura específica para essas referências, desde que elas atendam ao requisito de indicar qual mensagem responde a outra.

Com as referências estruturadas, é possível iniciar a etapa de identificação de conversas. Esta é a principal etapa deste estágio, pois é responsável por percorrer todas as mensagens obtidas e aplicar técnicas destinadas à identificação dos fluxos de conversa e de seus participantes.

Conforme definido anteriormente, as mensagens e suas referências podem ser modeladas como um grafo direcionado, no qual cada nó representa uma mensagem e cada aresta representa uma relação de resposta. Nesta etapa, a identificação de conversas consiste em enumerar os caminhos válidos nesse grafo, respeitando as restrições cronológicas e estruturais adotadas pela implementação.

Para realizar a identificação dos fluxos de conversa, as mensagens são percorridas de trás para frente. Esse comportamento foi adotado porque a navegação a partir das referências explicitamente registradas em cada mensagem reduz a ambiguidade e o custo

de busca, evitando indexar mensagens por referências inversas. Para cada mensagem, as referências válidas são identificadas e percorridas de maneira recursiva, reconstruindo o caminho da mensagem atual até uma mensagem sem referências válidas a percorrer.

Fluxo de conversa é definido como um caminho simples maximal no subgrafo de $G$ formado pelas referências válidas, orientado no sentido das relações de resposta, que conecta mensagens sem respostas posteriores válidas a uma mensagem inicial sem referências anteriores válidas. Todos os participantes que produziram as mensagens desse caminho passam a compor a conversa correspondente. Esse processo é semelhante à identificação de todos os caminhos entre folhas e nós-raiz em estruturas de árvore. O funcionamento do procedimento descrito é apresentado no Algoritmo 1.

**Algorithm 1** Identificação de fluxos de conversa

```
1: procedure ConstruirFluxos(mensagem, indice, caminho)
2:     caminhoAtual ← caminho || [mensagem]
3:     referencias ← ExtrairReferencias(mensagem)
4:     temReferenciaValida ← falso
5:     for all r ∈ referencias do
6:         if Existe(indice, r) e NaoFormaCiclo(caminhoAtual, r) then
7:             anterior ← ObterMensagem(indice, r)
8:             if anterior.data < mensagem.data then
9:                 temReferenciaValida ← verdadeiro
10:                ConstruirFluxos(anterior, indice, caminhoAtual)
11:            end if
12:        end if
13:    end for
14:    if temReferenciaValida = falso then
15:        if Tamanho(caminhoAtual) > 1 then
16:            ArmazenarFluxo(caminhoAtual)
17:        end if
18:    end if
19: end procedure
```

No Algoritmo 1, o parâmetro *caminho* registra a sequência de mensagens percorrida na chamada recursiva atual. A operação $caminho \,||\, [mensagem]$ cria uma nova lista, sem alterar a lista recebida, de modo que cada ramificação mantém seu próprio caminho e não modifica atributos compartilhados das mensagens. Adicionalmente, apenas referências que respeitam a ordem temporal entre mensagens são consideradas válidas, garantindo consistência cronológica na construção dos fluxos.

A variável local *temReferenciaValida* indica se ao menos uma referência passou pelas verificações de ausência de ciclo, existência no índice e ordem temporal estrita. A função *NaoFormaCiclo* verifica se a mensagem indicada pela referência já aparece em *caminhoAtual*. Quando nenhuma referência satisfaz esses critérios, o caminho é encerrado na mensagem atual, mesmo que a lista original de referências não esteja vazia. Assim, referências inexistentes, cíclicas ou dirigidas a mensagens com data igual ou posterior são descartadas, mas não invalidam o trecho já percorrido. Esse trecho é armazenado somente se contiver mais de uma mensagem. Se houver referências válidas, apenas elas

são percorridas, sem emitir o caminho parcial na mensagem atual.

É possível observar que o processo de identificação de fluxos de conversa se baseia no percurso recursivo das mensagens, garantindo que todas as mensagens envolvidas em determinada conversa sejam examinadas. Para isso, as referências de cada mensagem são obtidas no início do algoritmo. O processo de identificação dessas referências não é determinado pelo MCCD, uma vez que depende da base de dados e da implementação adotada. A recursão é então aplicada a cada referência válida até o ponto em que não existam mais referências válidas a serem percorridas. Ao final, os fluxos de conversa identificados são retornados para armazenamento.

Durante o processamento, ciclos são evitados verificando se a mensagem indicada por uma referência já aparece no caminho da chamada recursiva atual, impedindo a formação de caminhos cíclicos. Cada chamada inicial deve fornecer uma lista vazia. Como a lista é estendida sem alterar a recebida, buscas independentes não herdam mensagens de chamadas anteriores.

A Tabela 4.2 apresenta um exemplo de diálogo e é composta por quatro colunas. A coluna "ID" contém o identificador da mensagem. A coluna "Conteúdo" apresenta a mensagem escrita por um usuário. A coluna "Respondendo" indica o ID da mensagem à qual aquela mensagem responde. Por fim, a coluna "Respondido por" indica quais mensagens responderam a determinada mensagem.

Neste exemplo, é possível notar que uma mensagem pode responder a mais de uma mensagem ou ser respondida por mais de uma mensagem simultaneamente. Essa é uma característica importante para permitir a identificação de múltiplos fluxos de conversa durante a geração do conjunto de dados. A mensagem "1", por exemplo, foi respondida pelas mensagens "2", "3" e "4". Além disso, a mensagem "5" responde, ao mesmo tempo, às mensagens "3" e "4".

A mensagem "6", que não responde a nenhuma outra e também não é respondida por qualquer outra, compõe o conjunto de dados como um todo, mas não faz parte de nenhum fluxo de conversa, uma vez que não há referências a ela. Os fluxos identificáveis nesse exemplo decorrem das ligações "10 → 9", "9 → 5", "9 → 8", "8 → 7", "8 → 4", "7 → 2", "7 → 3", "5 → 3", "5 → 4", "4 → 1", "3 → 1" e "2 → 1".

A Figura 4.8 apresenta a conversa da Tabela 4.2, destacando as referências entre as mensagens dos usuários. É possível perceber como essas relações podem se tornar rapidamente complexas. Ainda assim, a organização dessas estruturas agrega valor significativo ao conjunto de dados final, pois permite obter diferentes respostas para as mesmas mensagens ou sequências de mensagens. Cada caminho possível entre as setas corresponde a um fluxo de conversa distinto, ainda que alguns trechos sejam compartilhados.

Seguindo os passos propostos para a identificação dos fluxos de conversa, a primeira mensagem a ser analisada seria a mensagem "10". Ao seguir sua única referência, a mensagem "9" é alcançada. A partir dela, o fluxo da conversa se divide em dois novos caminhos, correspondentes aos fluxos "$10-9-8$" e "$10-9-5$". Cada um desses caminhos continua sendo explorado por meio das referências até alcançar uma mensagem que não faz referência a nenhuma outra, completando, assim, um fluxo de conversa. Ao todo, existem cinco fluxos de conversa distintos neste exemplo: "$10-9-5-3-1$", "$10-9-5-4-1$", "$10-9-8-7-2-1$", "$10-9-8-7-3-1$" e "$10-9-8-4-1$".

| ID | Conteúdo | Respondendo | Respondido por |
|---|---|---|---|
| **1** | Como a inteligência artificial pode transformar a educação nos próximos anos? | | 2, 3, 4 |
| **2** | Vejo impacto direto na personalização do aprendizado e no ritmo individual dos alunos. | 1 | 7 |
| **3** | Concordo, e acrescento a automação de tarefas administrativas para professores. | 1 | 5, 7 |
| **4** | Também há riscos, como dependência excessiva de tecnologia e desigualdade de acesso. | 1 | 5, 8 |
| **5** | Esses pontos mostram que o avanço precisa ser técnico e também social e ético. | 3, 4 | 9 |
| **6** | A inovação tecnológica sempre avança mais rápido que a adaptação cultural. | | |
| **7** | Personalização e automação são ótimas, mas exigem formação adequada dos educadores. | 2, 3 | 8 |
| **8** | Sem formação e políticas públicas, os riscos citados superam os benefícios. | 7, 4 | 9 |
| **9** | No equilíbrio entre inovação, ética e inclusão está o verdadeiro potencial da IA educacional. | 5, 8 | 10 |
| **10** | Portanto, a transformação depende mais de governança e estratégia do que da tecnologia em si. | 9 | |

Tabela 4.2: Diálogo estruturado sobre impactos da inteligência artificial na educação.

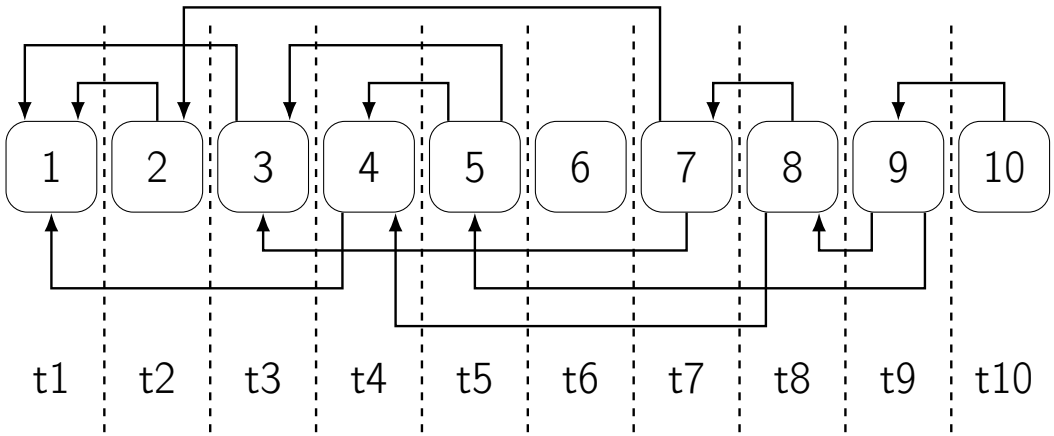



Figura 4.8: Diagrama temporal com dependências entre eventos.

### 4.2.4 Corpora Gerados

Como metodologia, o MCCD apresenta apenas sugestões de dados que podem ser armazenados para compor o conjunto de dados final, sem definir de forma estrita tipos de dados, estruturas ou outras características específicas. Essa abordagem não prescritiva da metodologia busca permitir sua adaptação a diferentes domínios e formatos de origem.

A relação a seguir apresenta, de forma sistemática, as informações cujo armazenamento é recomendado:

- **Metadados:** Documentar metadados referentes às bases de dados, ao ferramental empregado e aos dados relativos ao próprio processo de geração é importante para que futuros pesquisadores possam determinar as condições e o funcionamento do

processo de geração do conjunto de dados.

- **Categorias e Tópicos:** Categorias e tópicos devem ser diferenciados explicitamente: categorias representam níveis hierárquicos superiores de organização temática, enquanto tópicos correspondem a unidades específicas de discussão associadas a uma categoria. Ambos devem ser armazenados com suas relações hierárquicas preservadas.

- **Mensagens e Textos Limpos:** Devem ser mantidas duas representações distintas das mensagens: a versão original após o tratamento de anonimização, preservando a estrutura e conteúdo textual sem pré-processamento adicional, e a versão processada, que contém dados limpos e normalizados conforme as etapas de pré-processamento.

- **Relações:** Este é um dos dados mais importantes gerados durante o processamento, pois representa explicitamente as relações entre mensagens, modeladas como arestas em um grafo direcionado, no qual cada nó corresponde a uma mensagem e cada aresta representa uma referência a outra mensagem. A partir desse dado, é possível reconstruir conversas e identificar fluxos de conversa paralelos, o que o torna especialmente valioso em conjuntos de dados conversacionais.

- **Estruturas:** Corresponde à forma original de organização dos dados, incluindo organização lógica e metadados. Esse dado auxilia na identificação de tendências e pode ser utilizado para refinar os dados empregados em modelos com requisitos específicos de contexto.

- **Conversas:** Todos os fluxos de conversa identificados devem ser armazenados em ordem cronológica. Esses fluxos podem ser utilizados durante o treinamento de LLMs para a identificação de diferentes formas de responder a uma mesma mensagem.

## 4.3 Miner-XenForo

Esta Seção apresenta a ferramenta Miner-XenForo, desenvolvida com base no MCCD e com foco em fóruns baseados no software XenForo [XenForo Ltd., 2026]. A ferramenta automatiza a coleta, transformação e estruturação de dados de fóruns web, com foco na identificação de fluxos conversacionais.

A escolha de fóruns construídos utilizando XenForo foi em razão da ampla adoção desse tipo de plataforma online, da padronização das estruturas de dados e da compatibilidade de sua organização, em categorias e tópicos, com os requisitos estruturais do MCCD. Essa estrutura das mensagens, juntamente com a presença de referências estruturadas entre elas, permite a geração de conjuntos de dados estruturados, com diálogos multiturnos e multiparticipantes, organizados em diferentes fluxos conversacionais, como estabelecido pela metodologia.

Nesta Seção, são descritos: (a) a arquitetura e o fluxo de execução da ferramenta; (b) as decisões de implementação necessárias para adaptar o MCCD ao domínio de fóruns web;

(c) os principais formatos de armazenamento e os artefatos intermediários gerados; e (d) as limitações e considerações éticas associadas ao processo de aquisição e anonimização.

### 4.3.1 Arquitetura

A arquitetura do Miner-XenForo foi implementada seguindo os padrões definidos pelo MCCD. Ela constitui um caso de estudo alinhado às propostas da metodologia, funcionando como base para validação dos resultados e dos conjuntos de dados gerados. Sua organização modular define responsabilidades bem separadas entre os estágios de aquisição, anonimização e processamento. Cada módulo opera utilizando JSON intermediários persistidos, permitindo desacoplamento entre etapas e permitindo execução independente ou incremental.

Os módulos "Obter Tópicos", "Obter Mensagens" e "Armazenamento" executam internamente operações paralelas e assíncronas. No entanto, a comunicação entre módulos e estágios ocorre de forma sequencial, definindo transições entre módulos através da sincronização entre aquisição, anonimização e processamento.

A Figura 4.9 apresenta o fluxo completo da ferramenta, evidenciando a separação em três estágios e o fluxo de dados entre eles, incluindo pontos de paralelização "OS Threads" e módulos com requisições externas "R", que concentram os principais gargalos de tempo de execução associados a operações de rede.

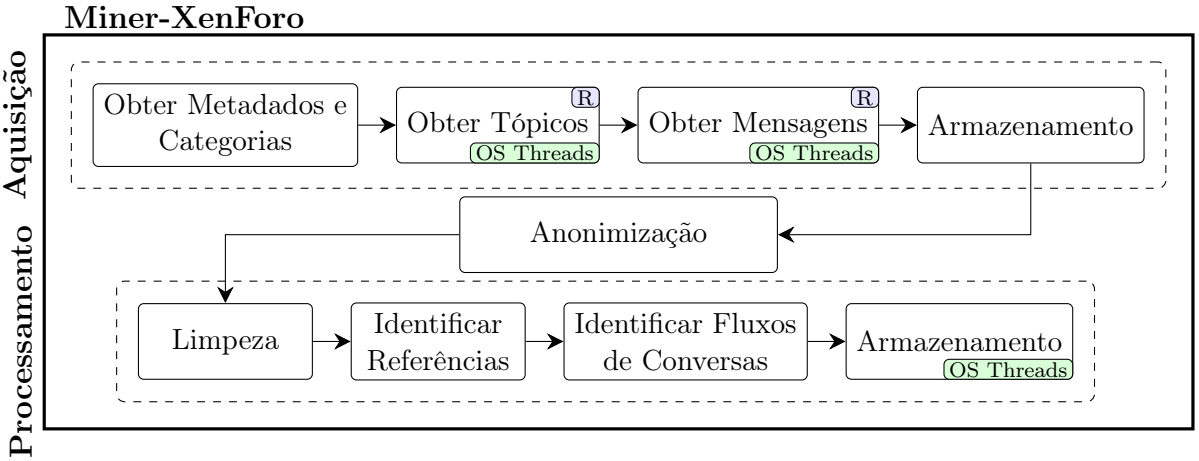



Figura 4.9: Pipeline de aquisição, anonimização e processamento do Miner-XenForo.

É possível observar que, na parte superior da Figura, estão os módulos relacionados ao estágio de aquisição. Todos esses módulos participam diretamente da obtenção e da estruturação dos dados que serão utilizados no estágio de processamento. O funcionamento desses módulos é detalhado na Seção 4.3.2.

A anonimização é composta por um único módulo, responsável por realizar transformações nos campos de identificação dos metadados para reduzir o risco de vazamento e identificação de usuários no conjunto de dados final gerado. Essas transformações são descritas na Seção 4.3.3.

Na parte inferior da Figura estão o estágio e os módulos referentes ao processamento dos dados, onde estão localizados os principais módulos de identificação de referências e

de fluxos de conversa. A descrição completa dos módulos de processamento é apresentada na Seção 4.3.4.

### 4.3.2 Aquisição de Dados

A característica de ser agnóstico à fonte é um dos principais aspectos do MCCD. Nesse contexto, a ferramenta Miner-XenForo constitui uma especialização adaptada, capaz de se conectar a fóruns online, que constituem a fonte de dados padrão desta implementação.

Como fonte de dados, a ferramenta Miner-XenForo foi desenvolvida para se conectar a fóruns online construídos com XenForo [XenForo Ltd., 2026] e obter os dados neles disponibilizados. Essa decisão foi tomada porque o XenForo já é uma plataforma consolidada e amplamente utilizada por diversos fóruns online. Com isso, pesquisadores podem utilizar o Miner-XenForo para criar novos conjuntos de dados.

Dessa maneira, todo o estágio de aquisição foi especializado, a partir do MCCD, para obter adequadamente os dados com base na estrutura utilizada por esses fóruns. A lista a seguir apresenta essa estrutura e os elementos para os quais o Miner-XenForo está preparado para obter os dados necessários.

1. Página Principal
    (a) Lista com categorias. (Referência ao item 2)
    (b) Lista com subcategorias. (Referência ao item 3)
2. Categoria
    (a) Lista com subcategorias. (Referência ao item 3)
3. Subcategoria
    (a) Lista com tópicos. (Referência ao item 4)
4. Tópico (também chamado de “thread” em alguns fóruns).
    (a) Lista com mensagens.

Nesta ferramenta, o estágio de aquisição especializa as definições do MCCD para o domínio do XenForo, com ênfase em: (i) acessar a fonte de dados, ou seja, o fórum online; (ii) definir a organização hierárquica específica e os metadados do fórum; (iii) obter os dados de forma iterativa, por meio de múltiplas requisições HTTP; e (iv) padronizar e armazenar os dados obtidos em uma estrutura lógica. A Figura 4.10 apresenta as etapas executadas para completar esse estágio.

A lista a seguir apresenta uma descrição das responsabilidades de cada etapa:

1. **Acessar a fonte dos dados:** estabelecer a conexão com a fonte e realizar as requisições necessárias para obter o conteúdo. Como o MCCD é agnóstico à fonte, ele não define uma tecnologia específica. Portanto, a implementação deve seguir os requisitos da plataforma-alvo.

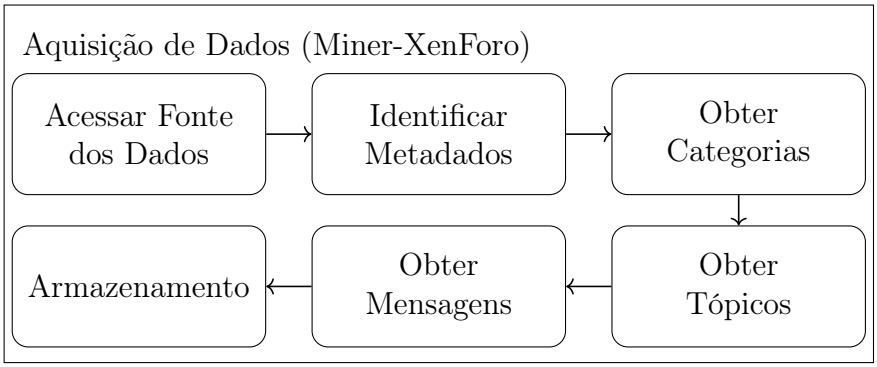



Figura 4.10: Etapas do estágio de aquisição de dados do Miner-XenForo.

2. **Identificar metadados:** reconhecer e extrair informações de contexto, como *timestamps*, nomes de usuários e detalhes estruturais, quando disponíveis.

3. **Obter categorias:** recuperar a organização do conteúdo em categorias e subcategorias, permitindo navegar até os elementos de discussão.

4. **Obter tópicos:** identificar os tópicos (ou *threads*) pertencentes a cada subcategoria. Em fóruns com paginação, essa etapa pode exigir múltiplas requisições para navegar através de paginações, resultando em novas páginas.

5. **Obter mensagens e armazenar:** iterar sobre os tópicos, coletar as mensagens e persistir os dados, preservando as relações entre metadados, tópicos e mensagens. Como essa etapa pode ser demorada, o armazenamento pode ser realizado de forma incremental para reduzir o risco de perda de dados.

Além da URL, a implementação oferece parâmetros opcionais para controlar o comportamento do processo de coleta:

- **Comandos de recarregamento:**
  - **reload-threads**: percorre as páginas de cada subcategoria em busca de novos tópicos.
  - **reload-posts**: percorre as páginas de cada tópico em busca de novas mensagens.
- **max-request**: define quantas requisições simultâneas podem ser realizadas. É recomendável ter cautela, pois o site pode ativar mecanismos de limite de requisições e bloquear novas tentativas de conexão.
- **cache-pages**: habilita um cache de páginas para depuração e adaptação do processo de aquisição. Esse recurso não deve ser utilizado em produção, pois o volume de dados armazenados cresce rapidamente.

Quando o estágio de aquisição é iniciado, primeiramente é feita uma requisição à página inicial, a partir da qual se obtém a maior parte dos metadados do site, das categorias e, quando disponíveis, das subcategorias. Em seguida, o Miner-XenForo verifica os arquivos

locais para determinar se a execução atual corresponde à atualização de um conjunto de dados existente ou à criação de um novo conjunto desde o início, pois isso altera como os dados são persistidos localmente.

Todas as requisições de tópicos e mensagens precisam ser paginadas, uma vez que fóruns construídos com XenForo normalmente possuem diversos tópicos e mensagens em cada tópico, apresentados em conjuntos menores que precisam ser percorridos. Isso ocorre devido à estrutura utilizada para organizar as conversas e os temas nos fóruns.

Esse tipo de estrutura faz com que as etapas de obtenção de tópicos e mensagens sejam as mais lentas, uma vez que são as que envolvem o maior volume de dados a ser obtido. Ainda assim, a ferramenta Miner-XenForo foi construída para tirar proveito de mecanismos de *threads*, realizando a aquisição de forma paralela. Embora isso reduza consideravelmente o tempo de aquisição, também aumenta a quantidade de requisições feitas aos sites.

Uma vez obtidos todos os dados, é possível dar início ao próximo estágio da ferramenta, executando a anonimização dos dados.

### 4.3.3 Anonimização

O estágio de anonimização foi implementado seguindo os requisitos do MCCD, mascarando dados que possam ser utilizados para identificar usuários dos fóruns. Embora não seja um estágio obrigatório, sua execução ocorre imediatamente após a conclusão do estágio de aquisição. Nesta implementação, o estágio de anonimização realiza a pseudonimização dos identificadores presentes nos metadados dos fóruns.

A decisão de executar esse processo após a aquisição se deve ao fato de que isso simplifica o processo de mascaramento dos dados. Os identificadores tratados não são substituídos por valores independentes a cada ocorrência, o que poderia comprometer avaliações estatísticas. Alternativamente, são substituídos por identificadores únicos gerados de maneira determinística a partir dos dados originais, garantindo, assim, que um mesmo dado seja substituído de forma consistente em todas as suas ocorrências.

Este módulo sobrescreve os arquivos gerados durante a aquisição, mantendo sua estrutura e organização originais. Por isso, os resultados da aquisição e da anonimização podem ser utilizados diretamente pelo estágio de processamento, pois seguem os mesmos formatos.

Na estrutura produzida pela aquisição, há três grupos de arquivos que podem conter dados capazes de identificar usuários e que devem ser anonimizados para a disponibilização pública do conjunto de dados ou para o treinamento de LLMs:

- **Metadados** de tópicos e subcategorias (arquivos JSON), que armazenam campos relacionados ao autor do tópico e/ou ao usuário associado à subcategoria.
- **Mensagens** (arquivos JSON) associadas aos tópicos, que incluem informações sobre o usuário responsável por cada mensagem.
- **Conteúdo das mensagens** criadas pelos participantes do fórum, que pode incluir dados sensíveis no texto ou em outros conteúdos publicados.

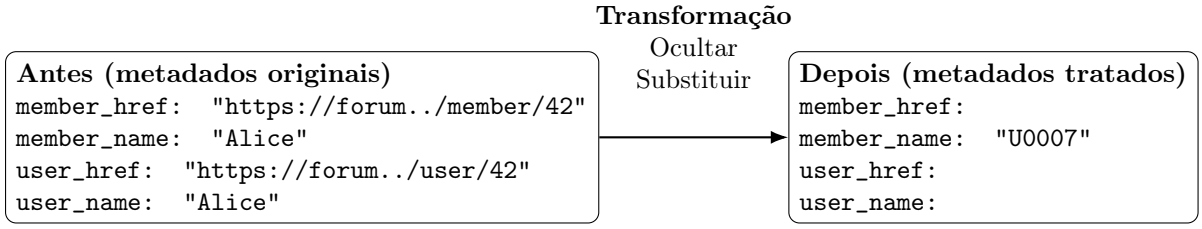



Figura 4.11: Exemplo das transformações aplicadas aos campos de identificação nos arquivos JSON.

Nesta implementação do Miner-XenForo, a anonimização se concentra nos metadados associados a tópicos e mensagens, realizando a alteração de dados sensíveis nesses campos. A identificação e o mascaramento de possíveis dados sensíveis no conteúdo das mensagens não estão implementados por padrão, devido à alta variabilidade linguística e à necessidade de técnicas avançadas de processamento de linguagem natural. Portanto, a execução desse módulo não garante a ausência de PII no corpus.

Na disponibilização e reutilização de dados de fóruns, o conteúdo das mensagens merece atenção específica, pois pode incluir informações de contato ou pagamento inseridas pelos participantes. Quando os requisitos de uso demandarem proteção adicional, esse conteúdo pode ser tratado em uma etapa complementar, combinando expressões regulares para identificar formatos estruturados, como telefones, e-mails, CEPs e CPFs, com análise contextual de possíveis chaves PIX.

A estratégia de anonimização consiste em substituir o campo `member_name` por um identificador único (*ID*). Sempre que o mesmo `member_name` é identificado, ele é substituído pelo mesmo ID. A ferramenta, por sua vez, remove completamente os metadados relacionados ao perfil e ao nome dos usuários, uma vez que essas informações podem ser utilizadas para obter dados sensíveis por meio de referência cruzada.

O uso de um identificador único por usuário permite manter a consistência entre arquivos e análises, por exemplo, ao contabilizar quantas mensagens um mesmo autor publicou, sem exibir os nomes originais nos campos substituídos. Na prática, esse mapeamento é implementado por meio de um contador sequencial, associado a uma tabela de correspondência que garante que cada valor original receba sempre o mesmo identificador (ex.: `U0001` e `U0002`). A Figura 4.11 mostra como esses dados são alterados e quais são os valores antes e depois desse processo.

Como ferramenta, a anonimização está disponível para uso e permanece ativa por padrão. Ainda assim, pode ser desativada, mantendo os identificadores originais nos metadados. No entanto, é recomendável que ela seja sempre executada, principalmente se o conjunto de dados vier a ser publicado em algum momento. A possibilidade de não obrigar a anonimização foi mantida para permitir cenários específicos de uso estritamente interno, sem distribuição dos dados.

Com a execução deste estágio, são gerados arquivos com os campos de identificação tratados de modo a preservar a estrutura organizacional original dos arquivos de entrada.

### 4.3.4 Processamento

O estágio de processamento do Miner-XenForo recebe como entrada os dados provenientes da aquisição ou da anonimização, que compartilham a mesma estrutura de armazenamento. O objetivo deste estágio é transformar os dados brutos contendo elementos HTML, organizados conforme a estrutura original da fonte e ainda não segmentados em fluxos de conversas, em um conjunto de dados pronto para ser utilizado por modelos LLMs. Para isso, o estágio de processamento executa os seguintes módulos: (i) limpeza dos dados, (ii) identificação de referências, (iii) identificação de fluxos de conversas e (iv) salvamento estruturado dos dados. Esses módulos e esse fluxo podem ser vistos na Figura 4.12.




Figura 4.12: Pipeline do estágio de processamento do Miner-XenForo.

Embora, conceitualmente, o estágio de processamento constitua um único componente, sua separação em módulos menores, com responsabilidades bem definidas, possibilita a alteração e a ativação de cada parte de forma mais simples.

Ao iniciar o estágio de processamento, o primeiro módulo a ser executado é o módulo de limpeza. Este módulo refina as definições da metodologia para o cenário específico em que é aplicado, isto é, sites de fóruns on-line. Para isso, elementos específicos encontrados em páginas HTML precisam ser tratados e transformados, de maneira determinística, em elementos úteis aos LLMs. Dessa forma, os principais mecanismos de limpeza utilizados são:

- **Remoção/normalização de HTML**: conversão do corpo da mensagem em texto, preservando quebras de linha, listas, códigos e blocos.
- **Remoção de elementos de interface**: eliminação de trechos não conversacionais associados a componentes da plataforma, como botões, assinaturas, rodapés e contadores.
- **Tratamento de citações**: identificação e separação de blocos de citação (*quote*) do texto novo escrito pelo autor, permitindo (a) inferir referências e (b) evitar que o texto citado seja duplicado como conteúdo original.
- **Normalização textual**: padronização de espaços, quebras de linha, caracteres invisíveis e sequências repetidas, a fim de estabilizar as etapas posteriores.

Para executar o mecanismo de remoção/normalização de HTML, foi utilizada a lista de conversão de elementos HTML apresentada a seguir. Essa lista transforma elementos não textuais em representações textuais, preservando, tanto quanto possível, sua semântica original.

Essas marcações são uma instanciação específica, para o tratamento de HTML em fóruns XenForo, das categorias semânticas abstratas definidas na Seção 4.2.3; por isso, a

correspondência entre elas não é necessariamente um para um. Por exemplo, <image> e <emoji> instanciam <visual> e <symbolic>; <mediaembed> instancia <embedded>, enquanto <iframe> representa a incorporação integral de conteúdo externo e se aproxima de <container>. O marcador <url> instancia <reference>: em <url> {destino} {texto-âncora} </url>, o primeiro campo informa o destino (URL), e o segundo, o texto-âncora. Já <link> representa uma URL quando destino ou texto não precisam ser preservados. As demais correspondências são <quote> para <quotation>, <answering> para <relation> e <code> para <executable>; esta lista não define um marcador específico para <placeholder>.

- **<image>:** Usado para substituir elementos de imagem. Variações dessa tag podem incluir a localização ou o tipo da imagem.
- **<emoji> {} </emoji>:** Usado para substituir emojis representados como caracteres ou imagens. O tipo de emoji deve ser incluído como metadado na tag.
- **<mediaembed> {} </mediaembed>:** Usado para substituir tipos genéricos de mídia, como vídeos, mapas, reprodutores de música, entre outros. Quando possível, outros dados relevantes devem ser incluídos como metadados.
- **<url> {} {} </url>:** Usado para substituir hiperlinks. O destino e o texto devem ser preservados como metadados.
- **<link>:** Usado para mascarar URLs quando o destino ou o texto não forem importantes.
- **<quote> {} </quote>:** Usado para representar informações já contidas em outras fontes, permitindo identificar dados repetidos e evitar viés por repetição nos modelos.
- **<answering> {} </answering>:** Usado para indicar qual trecho representa uma referência a outra mensagem.
- **<code> {} </code>:** Usado para substituir código de programação por uma tag. O código real pode ser omitido, dependendo do objetivo e dos requisitos.
- **<iframe> {} </iframe>:** Usado para representar conteúdos de outras fontes incorporados integralmente.

Uma vez que os textos estejam limpos, é possível dar continuidade ao estágio de processamento e iniciar o módulo de identificação de referências entre as mensagens. Para isso, todos os tópicos e mensagens são percorridos em busca de referências a outras mensagens e a trechos de mensagens, uma vez que fóruns construídos com XenForo permitem referências explícitas, isto é, à mensagem inteira, ou implícitas, que se referem apenas a um trecho de outra mensagem. Como os fluxos de conversas ainda não foram gerados, a ferramenta Miner-XenForo prioriza as referências explícitas quando estão disponíveis.

Outro ponto relevante é a possibilidade de editar mensagens a qualquer momento, o que pode gerar referências que não respeitam a ordem cronológica. Isso ocorre, principalmente,

quando uma mensagem é editada para responder a outra criada após sua publicação. Esse tipo de situação também pode gerar referências circulares, o que influencia diretamente a identificação dos fluxos de conversas. Por esse motivo, referências que não sigam a ordem das mensagens são descartadas, garantindo aciclicidade e consistência temporal.

A Figura 4.13 mostra um exemplo em que a edição de mensagens gera uma referência fora da ordem cronológica, resultando em um caminho circular. A primeira linha apresenta as mensagens originais e uma referência simples. Na segunda linha, a mensagem cinco é criada e faz referência a uma mensagem já existente. Na terceira linha, a mensagem dois é editada e passa a referenciar a mensagem cinco. Dessa maneira, surge um caminho circular entre as mensagens dois e cinco.

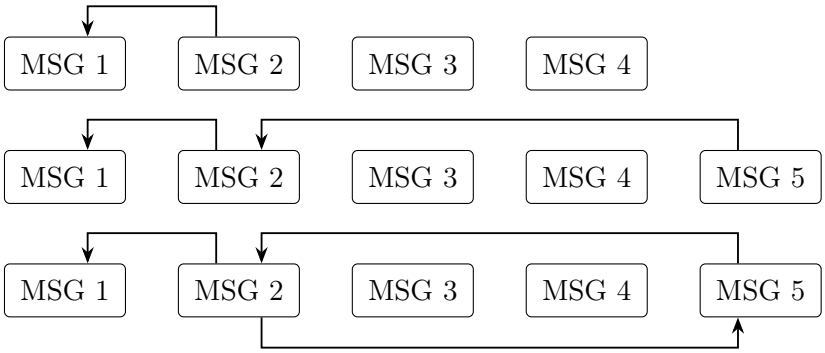



Figura 4.13: Exemplo de referências fora da ordem cronológica.

Uma vez identificadas as referências, é possível iniciar o módulo responsável pela identificação dos fluxos de conversa, construindo um conjunto de dados com as conversas já separadas e estruturadas. Em alto nível, esse módulo agrupa, com base nas referências, as mensagens que fazem parte de uma mesma conversa.

Seguindo a proposta de algoritmo do MCCD, as mensagens são consideradas em ordem cronológica inversa, com a detecção e o descarte de edições e ciclos já percorridos. A busca por um caminho maximal é iniciada apenas a partir de mensagens sem respostas posteriores válidas (nós folha) que referenciem pelo menos uma outra mensagem. Para cada referência, inicia-se uma nova busca pela mensagem referenciada, repetindo esse processo recursivamente até que seja encontrada uma mensagem que não referencie nenhuma outra.

Todos os caminhos percorridos, desde a primeira mensagem que faz alguma referência em uma conversa até a mensagem que não referencia nenhuma outra, são identificados como fluxos de conversas.

Por padrão, todos os participantes que tenham ao menos uma mensagem no fluxo identificado passam a fazer parte daquele diálogo específico. Com isso, conversas com a participação de mais de duas pessoas se tornam, automaticamente, multiparticipantes.

Com a conclusão deste módulo, o conjunto final de dados está completo e pode ser utilizado no treinamento de LLMs. Os arquivos finais são salvos de acordo com a seguinte estrutura:

- Um arquivo contendo dados gerais sobre o conjunto de dados e o processo de geração, como quantidades, tamanhos, data de criação e fórum utilizado.

- Uma pasta para cada categoria identificada, contendo os arquivos dos tópicos correspondentes.
- Um arquivo para cada tópico, contendo as mensagens a ele pertencentes.
- Uma pasta contendo todos os fluxos de conversas, com um arquivo para cada fluxo identificado.
  - Cada arquivo de fluxo contém as mensagens na ordem em que foram trocadas.
  - Metadados sobre o tópico também são incluídos para fins de estatística e refinamento.

## 4.4 Conjuntos de Dados Criados

Esta seção apresenta os conjuntos de dados gerados a partir do MCCD e da ferramenta Miner-XenForo. O objetivo é descrever as bases utilizadas, o processo de geração e as principais características dos conjuntos de dados resultantes, destacando como a metodologia permite identificar diálogos multiturnos e multiparticipantes, mesmo quando essas relações não estão explicitamente anotadas.

Os conjuntos de dados foram construídos com base em fóruns públicos, o que permite explorar interações reais e variadas e manter a identificação das conversas por meio das referências entre mensagens. Para reduzir riscos à privacidade, o estágio de anonimização é aplicado antes do compartilhamento dos dados, removendo ou substituindo identificadores nos metadados.

Embora ambos os conjuntos de dados apresentados tenham sido criados com a mesma metodologia e ferramenta, a partir de fóruns on-line de estilo semelhante, o conteúdo de cada fórum é distinto, resultando em conjuntos finais com características bastante diferentes.

Essa característica não constitui uma garantia, mas o uso de fóruns on-line permite usar a moderação e as diretrizes dessas plataformas como apoio para a remoção de conteúdos problemáticos. Assim, a qualidade dos dados tende a ser superior à de conjuntos de dados sem qualquer moderação de conteúdo.

Os conjuntos de dados são compartilhados em duas versões: uma com os dados brutos, contendo mensagens, tópicos e categorias, e outra que inclui, além desses elementos, os arquivos processados e os diálogos identificados. A separação permite disponibilizar versões com menor volume, evitando o custo de armazenamento associado aos dados processados. Em ambas as versões, os campos de identificação dos metadados foram tratados.

A lista a seguir descreve brevemente as duas bases de dados utilizadas na geração dos conjuntos apresentados nesta seção.

- **Adrenaline:** Um dos maiores e mais ativos fóruns brasileiros voltados a tecnologia e jogos. Seus principais temas estão relacionados a notícias e análises de tecnologia, hardware, jogos, periféricos e ofertas. Por se tratar de um fórum de tecnologia, são

comuns discussões e resoluções de dúvidas técnicas sobre componentes de computadores, como monitores e hardware em geral.

- **OuterSpace:** Fórum brasileiro focado em eletrônicos, jogos e temas mais abertos. Seus tópicos abrangem desde hardware e veículos até cultura, religião e cinema. A plataforma ainda oferece espaço dedicado a classificados de produtos de tecnologia e subfóruns específicos para o desenvolvimento de jogos.

A Tabela 4.3 apresenta uma comparação dos dois conjuntos de dados obtidos. A coluna "Adrenaline/OuterSpace" apresenta a razão entre os valores exibidos para cada conjunto, arredondada para duas casas decimais.

| **Métrica** | **Adrenaline** | **OuterSpace** | **Adrenaline/ OuterSpace** |
|---|---|---|---|
| **Tamanho Bruto (GB)** | 2,0 | 4,6 | 0,43 |
| **Tamanho Processado (GB)** | 71 | 5,4 | 13,15 |
| **Total de Tópicos** | 356K | 570K | 0,62 |
| **Total de Mensagens** | 9,5M | 24M | 0,40 |
| **Total de Tokens** | 477M | 1B | 0,48 |
| **Tamanho Médio por Tópico (mensagens)** | 26 | 42 | 0,62 |
| **Tokens Médios por Mensagem** | 50 | 43 | 1,16 |
| **Tamanho Médio das Conversas (mensagens)** | 34 | 3 | 11,33 |
| **Total de Conversas** | 4,5M | 3M | 1,50 |
| **Tokens Únicos** | 1,7M | 2,7M | 0,63 |
| **Tokens Médios por Tópicos** | 1,3K | 1,8K | 0,72 |
| **Fonte** | https://forum.adrenaline.com.br/ | https://forum.outerspace.com.br/ | - |

Tabela 4.3: Comparação entre conjunto de dados Adrenaline e OuterSpace.

É possível observar que, embora ambos os conjuntos de dados tenham tamanho bruto semelhante, o tamanho dos arquivos processados varia drasticamente. Isso se deve a características específicas de cada fonte de dados, como a maior densidade de referências por mensagem no Adrenaline, o que aumenta o número de fluxos conversacionais identificados durante o processamento, resultando em maior expansão combinatória dos dados processados.

Ambos os conjuntos de dados possuem uma grande quantidade de tópicos: 356K e 570K, respectivamente. Cada tópico pode abrigar uma ou mais mensagens sobre um tema específico. Ao todo, foram identificadas 9.5M e 24M de mensagens nas bases escolhidas. Novamente, essas diferenças são explicadas pelas características de cada base, nas quais cada tópico tende a ter mais mensagens no OuterSpace do que no Adrenaline. As mensagens, por sua vez, apresentam diferenças quantitativas relevantes entre os conjuntos. No Adrenaline, há uma tendência de menos mensagens por tópico, mas com mensagens 16% maiores do que no OuterSpace.

A maior diferença entre os conjuntos de dados está no tamanho médio das conversas, que é de 34 mensagens no Adrenaline e de 3 no OuterSpace. Em conversas maiores,

mais pessoas tendem a interagir e a referenciar ainda mais mensagens. Isso se relaciona diretamente com a quantidade de conversas identificadas, que aumenta em 50%.

A quantidade de tokens únicos também apresenta uma diferença relevante, o que pode estar relacionado aos temas abordados pelos fóruns escolhidos. Como o Adrenaline trata de temas mais específicos, menos tokens únicos foram identificados. Já no OuterSpace, onde há temas mais abrangentes e sem categoria específica, mais tokens únicos podem ser identificados na base de dados gerada.

A Figura 4.14 mostra a quantidade de mensagens por tópico em ambos os conjuntos de dados. É possível observar que a distribuição das mensagens é quase exponencial, com muitos tópicos contendo poucas mensagens e poucos tópicos contendo muitas mensagens. Embora ambos apresentem uma grande quantidade de tópicos com apenas uma mensagem, esse número é significativamente mais elevado no OuterSpace, enquanto a queda no número de respostas tende a ser mais suave no Adrenaline. Esses fatores, em conjunto com as características apresentadas anteriormente, sugerem menor continuidade de interação por tópico no OuterSpace, medida pela maior proporção de tópicos com uma única mensagem.

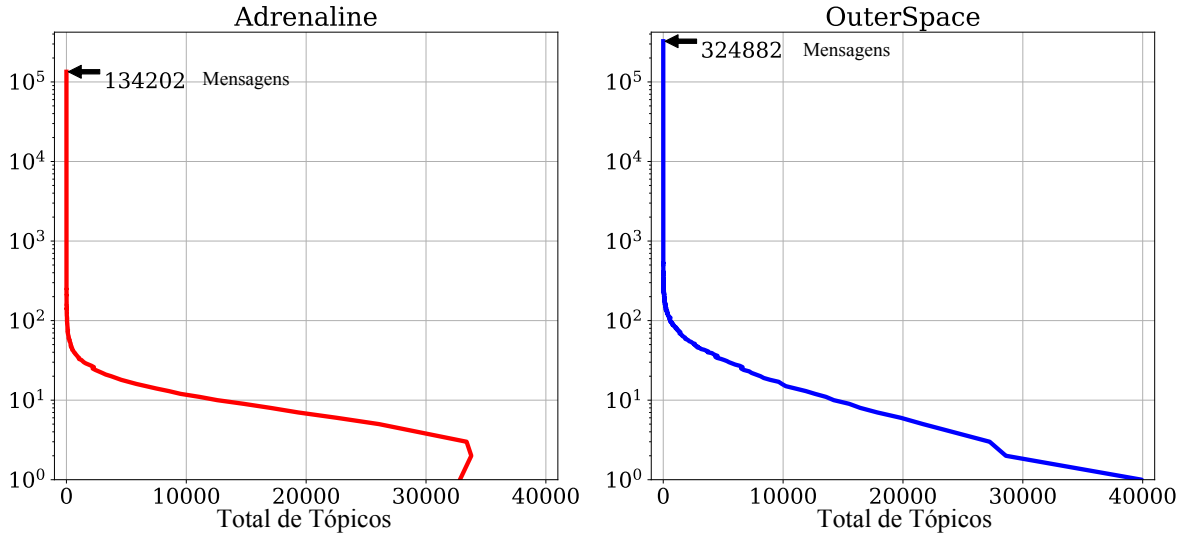



Figura 4.14: Distribuição do número de mensagens por tópico nos conjuntos Adrenaline e OuterSpace.

Todos os conjuntos de dados descritos nesta seção podem ser acessados publicamente[1].

## 4.5 Avaliação Experimental

Esta seção apresenta uma avaliação comparativa entre dois modelos BERT treinados para tarefas de classificação. Essas tarefas foram selecionadas para evidenciar o impacto sobre os resultados quando os modelos precisam lidar com cenários específicos e com vocabulário restrito a um domínio particular. O primeiro modelo escolhido foi o BERTimbau [Souza et al., 2020], um modelo consolidado na literatura para o português brasileiro,

[1] `https://github.com/MatheusFerraroni/datasets_from_minerxenforo`

construído com base na arquitetura BERT [Devlin et al., 2019]. Já o segundo modelo é um modelo de referência BERT treinado do zero, utilizando parcialmente os dados do conjunto Adrenaline.

O principal objetivo desta avaliação é analisar o impacto da aplicação da metodologia proposta, de sua implementação em ferramenta e dos conjuntos de dados gerados no desempenho dos modelos nas tarefas de classificação, medido pela acurácia. Nesse contexto, avaliamos se o uso de dados próximos ao domínio da tarefa contribui para a melhoria dos resultados obtidos. Dessa maneira, algumas das limitações relacionadas à disponibilidade de dados para o treinamento de modelos em domínios específicos podem ser mitigadas a longo prazo.

### 4.5.1 Procedimentos

Para realizar a avaliação experimental e definir o impacto dos dados em cenários distintos, foram escolhidas duas tarefas de classificação: (*i*) classificação em domínio específico, voltada à determinação da categoria e da subcategoria de tópicos de tecnologia; e (*ii*) classificação em domínio geral, voltada à identificação do sentimento de críticas de filmes.

**Tarefa 1: Classificação de Categoria e Subcategoria**

Para avaliar os conjuntos de dados propostos, focando no impacto do treinamento com uma gramática mais próxima à do cenário de aplicação, foi definido um fluxo de utilização dos dados que parte do treinamento de um modelo BERT do zero e segue até dois modelos auxiliares responsáveis pela classificação da categoria e da subcategoria de tópicos do conjunto de dados Adrenaline. A Figura 4.15 apresenta esse fluxo.

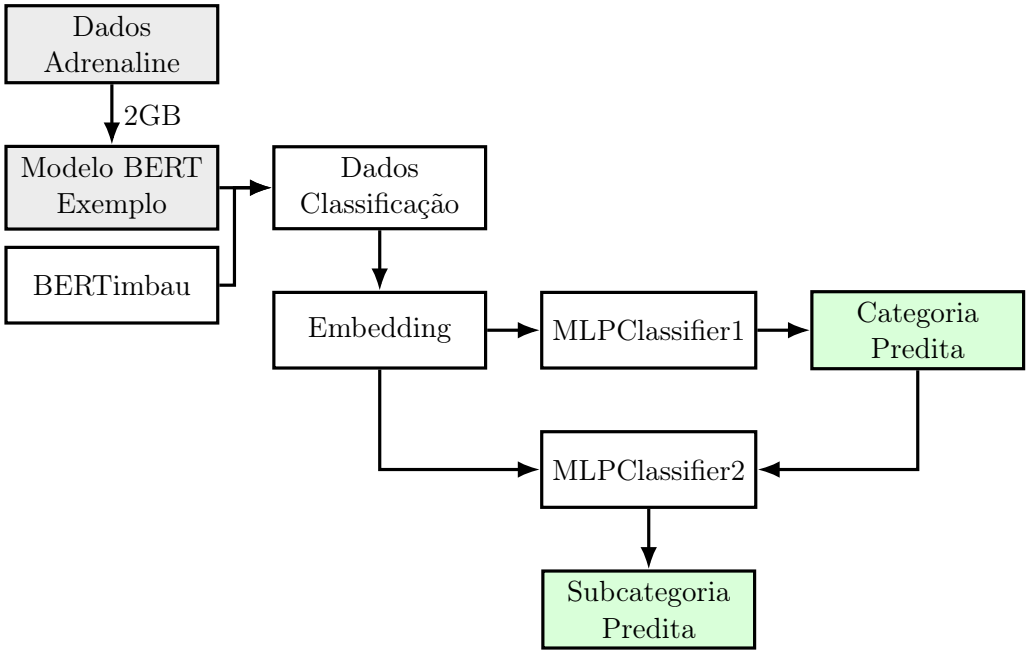



Figura 4.15: Fluxo de avaliação

As caixas cinzas do fluxo estão relacionadas ao treinamento, do zero, de um modelo de linguagem baseado na arquitetura BERT, utilizando os dados brutos do conjunto Adrenaline. Esse corpus possui cerca de 2 GB e 477 milhões de tokens, volume comumente

insuficiente para treinar modelos de linguagem de grande escala. O modelo de referência voltado ao domínio específico do conjunto Adrenaline será denominado Modelo BERT Exemplo (MBE).

Apesar do treinamento limitado, o objetivo não é desenvolver um modelo competitivo de uso geral, mas avaliar o impacto de representações no domínio-alvo. Nesse contexto, o modelo funciona como um controle experimental, isolando os efeitos da adaptação de domínio. A hipótese avaliada é que representações aprendidas exclusivamente no domínio alvo, mesmo com volume limitado de dados, podem ter melhores resultados do que modelos generalistas.

O MBE foi treinado por uma época sobre esse corpus, com taxa de aprendizado de $5 \times 10^{-5}$, utilizando o otimizador $AdamW$ e decaimento de peso de $5 \times 10^{-7}$. O comprimento máximo das sequências durante o treinamento foi definido em 512 tokens. Além da dimensão de saída de 768 valores, a configuração registrada não informa o número de camadas, o número de cabeças de atenção, o total de parâmetros, o tamanho do lote nem se o tokenizador foi treinado com dados do Adrenaline ou herdado do BERTimbau.

A divisão de 75% para treinamento e 25% para teste, descrita adiante, foi aplicada somente aos classificadores Multi-Layer Perceptron (MLP). A metodologia não registra a exclusão prévia dos tópicos de teste do corpus usado para o pré-treinamento do MBE. Portanto, não é possível afirmar que os textos de teste ficaram fora dos 2 GB de dados do Adrenaline. Esses textos podem ter sido apresentados ao MBE sem seus rótulos durante o pré-treinamento, uma possível exposição que deve ser considerada na interpretação dos resultados.

Após o treinamento completo do modelo, o MBE e o BERTimbau, um modelo pré-treinado para o português brasileiro, são utilizados para processar os tópicos classificados e gerar seus *embeddings*. Para isso, os modelos recebem como entrada os títulos dos tópicos e o conteúdo da primeira postagem. Em seguida, o token "[CLS]" gerado na saída do modelo é utilizado, cujos valores da última camada são extraídos e utilizados como *embedding*. O resultado é um vetor unidimensional de 768 valores reais, que carrega a representação semântica da sentença fornecida como entrada ao modelo. Durante o treinamento dos classificadores, os pesos dos modelos de linguagem permaneceram congelados, e os *embeddings* foram usados como atributos de entrada.

O uso simultâneo do MBE e do BERTimbau permite avaliar o impacto do treinamento em domínio específico em comparação com um modelo pré-treinado de uso geral.

A Tabela 4.4 apresenta um exemplo de como os tópicos se configuram após a adição dos *embeddings*. Além das informações exibidas na tabela, cada item também está associado a uma categoria e a uma subcategoria.

Após a preparação do conjunto de dados, são treinados os modelos de classificação. O conjunto balanceado contém 23.400 instâncias distribuídas em 26 subcategorias, com 900 instâncias cada. Para isso, foi utilizado um classificador do tipo MLP. Um dos modelos foi treinado especificamente para a classificação da categoria, utilizando apenas os *embeddings*.

Para a subcategoria, outro MLP foi treinado com os *embeddings* e com a classificação da categoria obtida pelo primeiro MLP como entrada adicional. Essa abordagem favorece modelos capazes de prever corretamente a categoria durante a classificação da

| | | Embedding | |
|---|---|---|---|
| **Título** | **Início da Postagem** | **MBE** | **BERTimbau** |
| Mini PC vs NAS para homelab | Pessoal, estou montando um homelab pequeno e queria opiniões sobre consumo... | [0.233, 0.3134, 0.700, 0.441, 0.992, 0.128...] | [0.112, 0.9041, 0.520, 0.301, 0.665, 0.047...] |
| LLM local: custos e privacidade | Alguém aqui já está rodando LLM local em produção? Queria entender custos... | [0.889, 0.1203, 0.440, 0.776, 0.218, 0.503...] | [0.532, 0.7712, 0.090, 0.654, 0.331, 0.708...] |
| Melhor SSD NVMe para workstation | Preciso trocar meu NVMe para workloads pesados de build de dados... | [0.451, 0.6621, 0.180, 0.540, 0.127, 0.864...] | [0.994, 0.2104, 0.630, 0.488, 0.905, 0.116...] |
| Podman em produção: armadilhas comuns | Depois de migrar do Docker para Podman encontrei alguns problemas... | [0.102, 0.4455, 0.930, 0.266, 0.714, 0.391...] | [0.287, 0.5561, 0.740, 0.119, 0.842, 0.053...] |
| Backup 3-2-1 sem dor de cabeça | Quero montar um backup 3-2-1 simples usando NAS e cloud barata... | [0.671, 0.3342, 0.270, 0.905, 0.452, 0.219...] | [0.415, 0.9823, 0.310, 0.773, 0.228, 0.601...] |

Tabela 4.4: Exemplo com títulos, início das postagens e embeddings simulados (quebras manuais).

subcategoria.

O registro experimental disponível não informa se, nas instâncias de treinamento do segundo classificador, essa categoria correspondia ao rótulo verdadeiro ou à predição do primeiro classificador. Também não especifica como a categoria foi codificada antes da concatenação com o *embedding* de 768 dimensões, por exemplo, como valor inteiro, vetor binário de sete posições (*one-hot*) ou vetor de probabilidades da saída (*softmax*). Por isso, esses detalhes não podem ser determinados com base nas informações disponíveis, o que limita a reprodução exata do classificador em cascata.

Ambos os classificadores MLP utilizaram a mesma arquitetura, composta por duas camadas ocultas com 200 e 100 neurônios, respectivamente, função de ativação ReLU, otimizada com Adam e treinada por 10 iterações. Em ambos os treinamentos, 75% dos dados foram utilizados para treinamento e 25% para teste, com divisão estratificada e sem validação cruzada.

Os modelos foram avaliados utilizando acurácia. Para garantir reprodutibilidade, foi utilizada semente fixa e divisão estratificada dos dados.

A quantidade de dados em cada categoria e subcategoria é extremamente diversa, o que exige a remoção de instâncias das classes majoritárias de maneira aleatória para balanceamento, reduzindo para 900 instâncias cada subcategoria. Dessa forma, a Figura 4.16 mostra o balanceamento realizado entre as subcategorias. É possível observar que, apesar desse balanceamento, a quantidade total de subcategorias em cada categoria permanece desigual.

Ao total, sete categorias foram definidas com diferentes quantidades de subcategorias. Cada subcategoria possui 900 instâncias que precisam ser classificadas. A lista abaixo detalha quantas subcategorias existem em cada categoria.

- **Categoria 1:** Possui 10 subcategorias.
- **Categoria 2:** Possui 4 subcategorias.
- **Categoria 3:** Possui 4 subcategorias.
- **Categoria 4:** Possui 3 subcategorias.

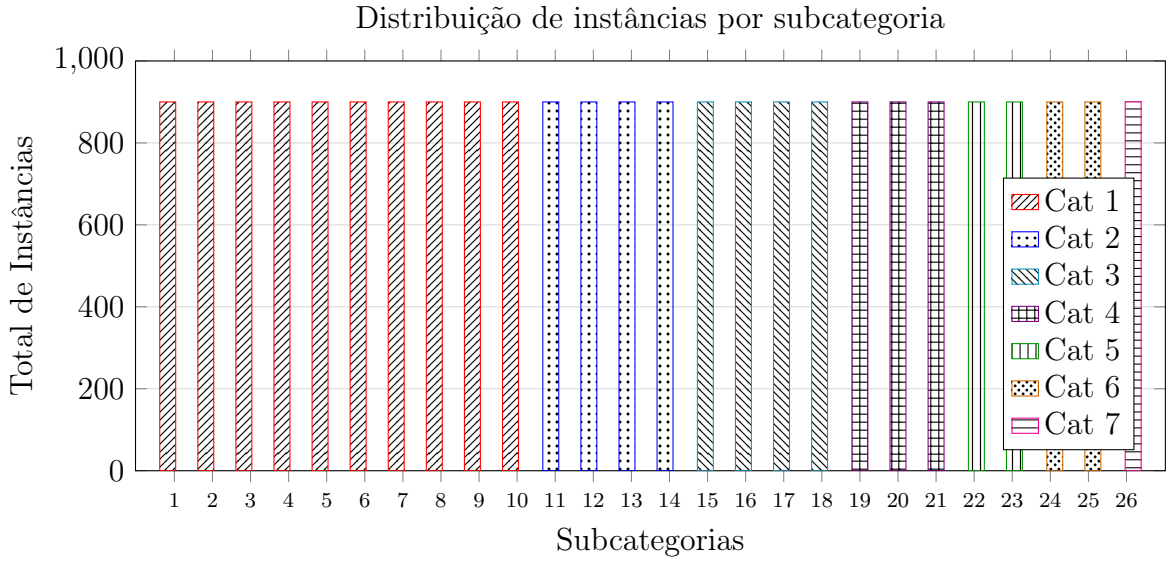



Figura 4.16: Distribuição de categorias e subcategorias.

- **Categoria 5:** Possui 2 subcategorias.
- **Categoria 6:** Possui 1 subcategoria.
- **Categoria 7:** Possui 2 subcategorias.

**Tarefa 2: Classificação de sentimento**

Para a tarefa de classificação de sentimentos, foi utilizado o conjunto de dados IMDb PT-BR de críticas de filmes [Gonçalves, 2022]. Essa é uma versão traduzida do *IMDB Dataset of 50K Movie Reviews* [Maas et al., 2011], um conjunto de dados amplamente utilizado em tarefas de classificação de sentimento. A versão traduzida preserva as mesmas 49.459 instâncias, balanceadas entre críticas com rótulo de sentimento positivo e críticas com rótulo de sentimento negativo.

A Figura 4.17 apresenta um histograma do comprimento das avaliações, agrupadas em avaliações positivas e negativas. É possível observar que as avaliações positivas tendem a ser mais curtas, enquanto as negativas tendem a apresentar uma quantidade maior de palavras.

A Figura 4.18 apresenta os termos mais frequentes, agrupados por tipo de crítica. Embora algumas palavras sejam utilizadas de maneira semelhante nos dois grupos, outras apresentam diferenças significativas entre críticas positivas e negativas.

Esse tipo de característica pode auxiliar os modelos, durante a construção do *embedding* das sentenças, a identificar padrões semânticos associados ao sentimento da crítica. Ainda assim, considerar apenas a frequência de unigramas tende a ser superficial, sendo insuficiente para representar contextos mais complexos.

Diversas palavras podem assumir sentidos distintos conforme o contexto em que aparecem, o que altera sua interpretação semântica. Termos como "bem" ou "melhor" podem ser empregados tanto de forma irônica quanto elogiosa, dependendo do restante da crítica para que seu significado real seja determinado. Expressões como "embora" ou "apesar de" podem alterar a polaridade da sentença.

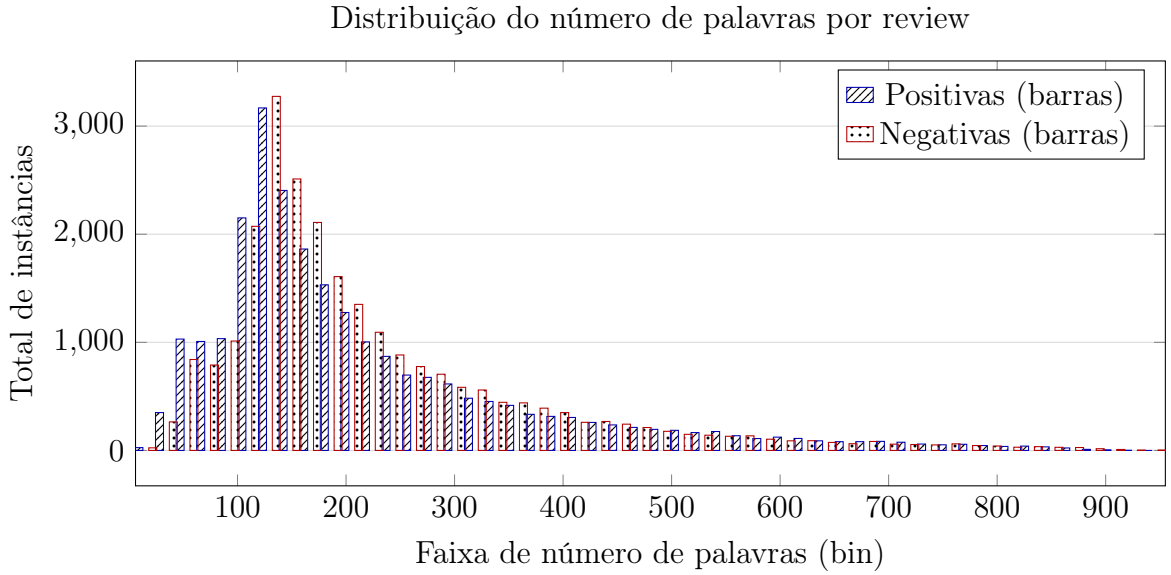



Figura 4.17: Distribuição por faixa do número de palavras.

Muitas críticas também podem incluir descrições gerais sobre a história do filme, o que faz com que palavras como "história" ou "personagens" apareçam com frequência, embora sejam neutras para a determinação do sentimento da crítica.

Outro aspecto relevante, apresentado na Figura 4.19, é a distribuição dos principais bigramas presentes nas avaliações. É possível observar que algumas combinações ocorrem com frequência muito maior em críticas positivas ou negativas, evidenciando diferenças de estilo e de conteúdo entre esses grupos.

Em particular, bigramas como "tão ruim" aparecem de forma desproporcional em avaliações negativas, funcionando como marcadores explícitos desse tipo de classificação. Por outro lado, bigramas como "primeira vez", "trilha sonora" e "vale pena" tendem a ocorrer com maior frequência no conjunto positivo. Esse tipo de comportamento pode sugerir que críticas positivas tendem a incluir recomendações ou destaques sobre aspectos do filme.

Também é possível notar que bigramas como "neste filme" e "deste filme" aparecem em proporções semelhantes, independentemente do sentimento da crítica. Nesses casos, a semântica só pode ser avaliada com o uso de modelos mais avançados, capazes de considerar o contexto da sentença. Isso reforça a importância de modelos baseados na arquitetura Transformer [Vaswani et al., 2017], como os utilizados neste trabalho.

### 4.5.2 Resultados Experimentais

A Figura 4.20 apresenta os resultados obtidos. As colunas “Categoria” e “Sub-Categoria” correspondem à primeira tarefa de classificação, cujo objetivo é classificar corretamente as instâncias do conjunto de dados Adrenaline. Já a coluna “Sentimento” corresponde à segunda tarefa de classificação, realizada com o conjunto de dados de críticas de filmes do IMDb.

Os resultados indicam que o MBE, treinado por apenas uma época, apresenta melhor

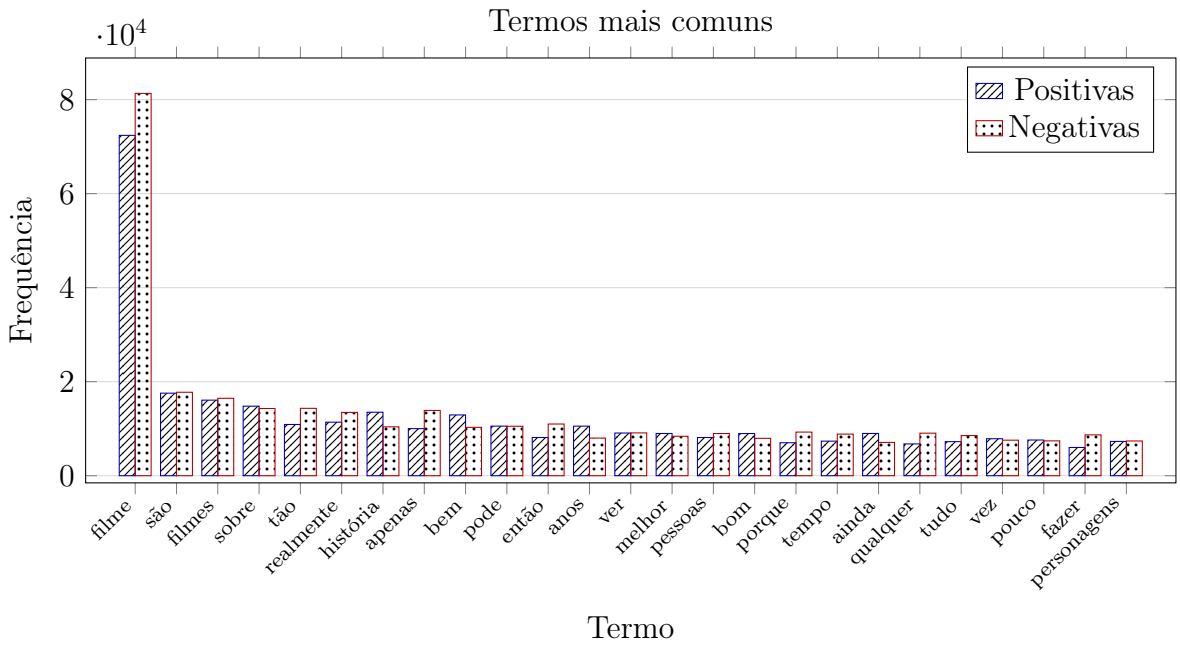



Figura 4.18: Frequência dos termos mais comuns no IMDB PT-BR, com separação por polaridade e total sobreposto.

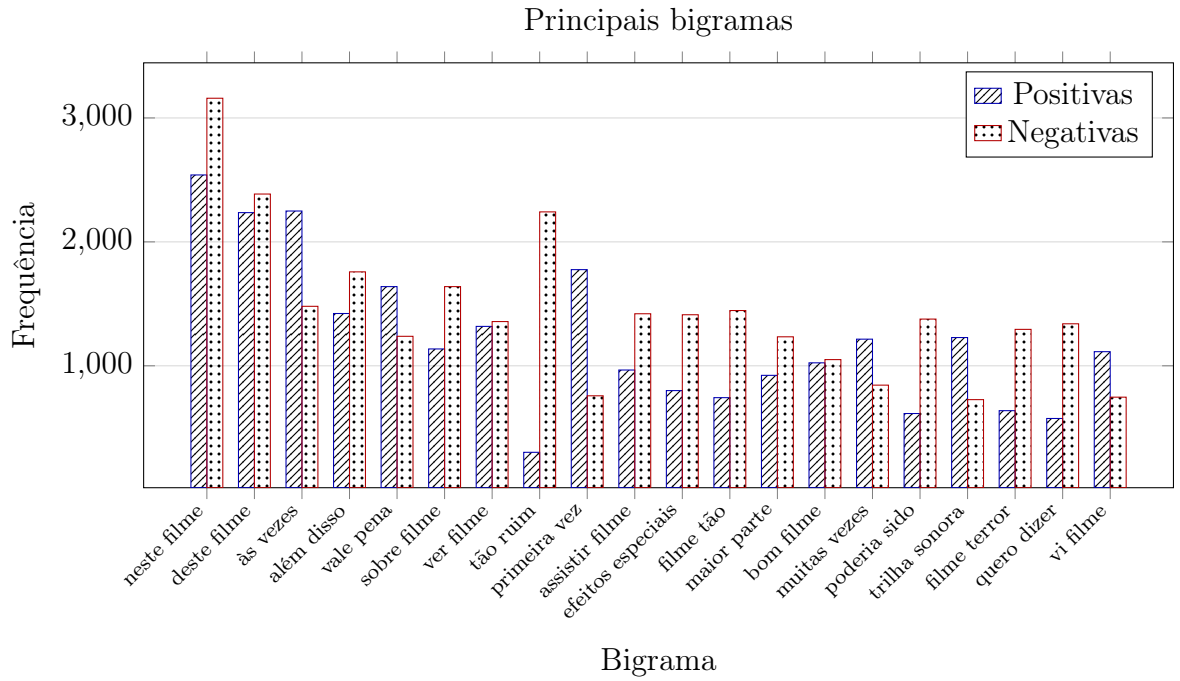



Figura 4.19: Frequência dos bigramas mais comuns.

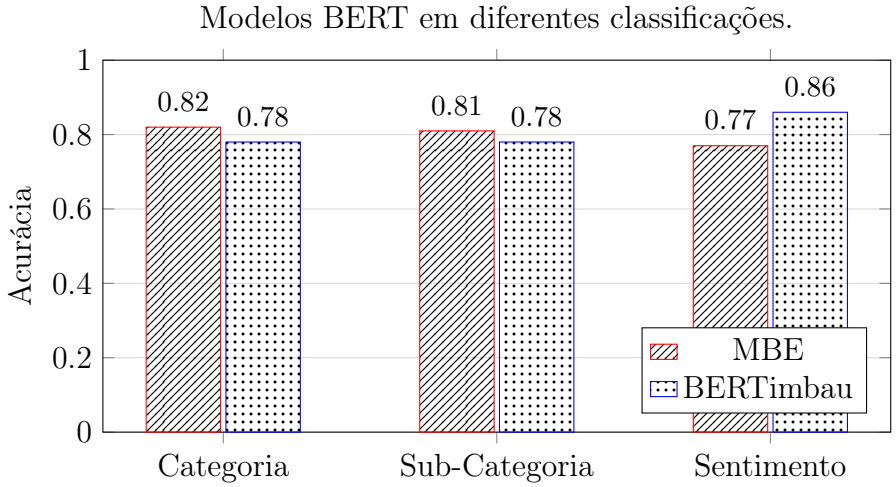



Figura 4.20: Acurácia do MBE e do BERTimbau em diferentes tarefas de classificação.

desempenho na representação das sentenças e no uso dessas representações como entrada para um `MLPClassifier`. É possível observar que o MBE supera o BERTimbau em 4 pontos percentuais na classificação de *Categoria* (0.82 vs 0.78) e em 3 pontos percentuais na classificação de *Sub-Categoria* (0.81 vs 0.78).

Por outro lado, o BERTimbau apresentou desempenho superior na segunda tarefa, de classificação de sentimentos. Nessa tarefa, o BERTimbau alcançou acurácia de 0.86, enquanto o MBE obteve 0.77, o que representa uma vantagem de aproximadamente 11%. Essa diferença é consistente com o fato de o BERTimbau ter sido pré-treinado em um corpus de domínio aberto, o que tende a favorecer tarefas generalistas.

Os resultados mostram que representações treinadas em domínio específico apresentam vantagem clara em tarefas do próprio domínio, como as tarefas de classificação de categoria e subcategoria avaliadas, enquanto modelos generalistas mantêm melhor desempenho em tarefas de domínio aberto, como classificação de sentimento.

## 4.6 Conclusão do Capítulo

Neste capítulo, foram apresentadas contribuições voltadas à criação de conjuntos de dados conversacionais para cenários de diálogos multiturnos e multiparticipantes. Primeiramente, foi proposta uma metodologia específica para gerar conjuntos de dados a partir de bases que contenham referências entre mensagens, permitindo a identificação e a estruturação de fluxos conversacionais multiturnos e multiparticipantes.

Com base nessa metodologia, foi desenvolvida uma ferramenta capaz de gerar conjuntos de dados de diálogos multiturnos e multiparticipantes, o que viabilizou a criação de dois conjuntos de dados de larga escala. Por fim, a avaliação desses conjuntos em uma tarefa de classificação evidenciou ganhos de desempenho em tarefas de domínio específico quando os modelos são treinados com dados próximos ao domínio-alvo, em comparação com modelos pré-treinados de uso geral, reforçando o papel crítico da adequação de domínio dos dados no treinamento de modelos de linguagem.

# Capítulo 5

# Geração Sintética de Conjuntos de Dados Conversacionais

**Pergunta de Pesquisa 2:** Como podem ser concebidos, gerados e estruturados dados conversacionais sintéticos em português brasileiro, compostos por múltiplos turnos e participantes, de forma controlada e metodologicamente consistente, para o treinamento de Large Language Models?

Este capítulo aborda a geração sintética de conjuntos de dados conversacionais como uma estratégia para suprir cenários em que há escassez de dados disponíveis, principalmente em razão de restrições de idioma e de domínios com requisitos específicos. O foco recai sobre diálogos multiturnos e multiparticipantes, preservando uma estrutura de conversa conhecida e atendendo a objetivos definidos previamente para a geração.

Propomos uma metodologia para criação controlada de diálogos baseada em roteiros anotados, geração assíncrona e rotação de participantes, bem como uma ferramenta cliente-servidor que operacionaliza e valida esse processo. Em contraste com abordagens que apenas reorganizam dados já disponíveis, buscamos gerar conversas a partir de roteiros e de participação distribuída, o que viabiliza a geração assíncrona, sem necessidade de interação simultânea.

Outra vantagem da proposta é a alternância entre participantes durante a geração assíncrona dos diálogos, ampliando a variabilidade dos turnos produzidos.

Outro aspecto relevante é que, devido ao alto custo de geração desse tipo de dado, o uso da estrutura de roteiros para orientar a criação dos turnos e os objetivos das conversas permite associar cada turno gerado ao turno correspondente no roteiro. Essa associação viabiliza a aplicação de permutação de turnos entre rótulos semanticamente equivalentes para ampliar os dados produzidos.

A lista a seguir apresenta os principais desafios abordados neste capítulo, formulados como questões de pesquisa:

1. **Criação de diálogos coerentes em domínio específico:** Como estruturar a geração via crowdsourcing para produzir conversas úteis, preservando a consistência semântica ao longo de múltiplos turnos.

2. **Diversidade multiparticipante em fluxo assíncrono:** Como alternar participantes durante a criação para ampliar a variabilidade linguística sem perder os

objetivos definidos no roteiro.

3. **Escalabilidade e aumento de dados com controle:** Como ampliar o volume de diálogos (histórias) com controle de estados, bloqueio de edição e permutação orientada por rótulos semanticamente equivalentes.

4. **Operacionalização e validação da abordagem:** Como implementar a metodologia em uma ferramenta cliente-servidor e validar essa abordagem por meio da geração e da análise de um conjunto de dados de exemplo (*Nutritious Food*).

A Seção 5.1 contextualiza o problema da escassez de dados conversacionais em domínios específicos e apresenta a proposta geral da metodologia MDC para lidar com as limitações identificadas. Na Seção 5.2, são detalhados os conceitos centrais da abordagem, o uso de roteiros para orientar os diálogos, o fluxo assíncrono de criação, a alternância entre participantes e o mecanismo de aumento de dados por permutação baseada em rótulos. A Seção 5.3 descreve a implementação da ferramenta que operacionaliza a metodologia, incluindo a arquitetura cliente-servidor, a interface dos participantes e os principais serviços de API. Em seguida, a Seção 5.4 apresenta o conjunto de dados *Nutritious Food*, o processo de criação e os resultados obtidos na etapa de aumento de dados. Por fim, a Seção 5.5 sintetiza os principais resultados e contribuições do capítulo.

## 5.1 Visão Geral

Esta seção apresenta o problema da escassez de dados conversacionais e introduz a metodologia MDC como abordagem proposta para sua mitigação. Esta escassez de dados conversacionais é um problema crítico, especialmente em domínios e idiomas específicos, nos quais a cobertura de intenções, fluxos e estilos linguísticos costuma ser insuficiente. Apesar dos avanços com a metodologia MCCD e com os conjuntos derivados de fontes existentes, ainda há cenários em que essa cobertura permanece limitada para o domínio em questão. Nesse contexto, a geração sintética controlada de diálogos, orientada por restrições estruturais como a definição de objetivos por turno, se torna uma alternativa para produzir dados em domínios que não podem ser atendidos apenas pelo reaproveitamento de bases existentes.

Mesmo diante dessa necessidade, obter volume suficiente de dados continua sendo um desafio em aberto, especialmente quando se busca evitar a dependência direta de LLMs para a geração integral de diálogos, seja em razão dos custos operacionais, seja pelo risco de introdução de vieses. Abordagens tradicionais de crowdsourcing ajudam nesse contexto, mas, em geral, apresentam limitações estruturais, como dependência de interação síncrona e consequente tempo ocioso, o que reduz a escalabilidade e eleva custos monetários e operacionais.

Para enfrentar essas limitações, propomos a metodologia *Mechanical Dialogue Creator* (MDC), acompanhada de uma ferramenta implementada de acordo com suas definições. A proposta combina criação assíncrona, rotação de participantes e roteiros que orientam cada turno e os objetivos da conversa, preservando a coerência semântica sem restringir a

naturalidade textual das contribuições dos participantes. Com isso, o processo de criação reduz o tempo ocioso, podendo aumentar a velocidade de geração de diálogos.

Um diferencial central da MDC é o controle do ciclo de vida das histórias, aqui definidas como instâncias de diálogo associadas a um roteiro e gerenciadas por estados, regras de transição e mecanismos de distribuição que incluem bloqueio, reativação de conversas abandonadas e critérios de conclusão e fechamento. Em conjunto com a rotação de participantes entre diálogos, essa estratégia reduz conflitos de edição e gargalos de espera por turnos, além de ampliar a diversidade das conversas ao introduzir múltiplos estilos linguísticos em uma mesma história.

A metodologia proposta ainda incorpora anotação estruturada por turno, associando cada contribuição ao objetivo correspondente no roteiro. Essa associação assegura rastreabilidade, viabiliza a inspeção da qualidade por intenção e permite analisar a cobertura de rótulos, favorecendo a auditoria do processo de geração e a filtragem de conversas inadequadas no conjunto final.

Para lidar com o baixo volume tipicamente associado à geração humana, a MDC prepara o conjunto para aumento de dados por meio de permutação baseada em rótulos. Assim, trechos vinculados a rótulos semanticamente equivalentes podem ser recombinados para gerar novas conversas, preservando seus objetivos conversacionais. Quando necessário, limites de uso por rótulo podem ser estabelecidos para reduzir a redundância e controlar a diversidade do conjunto ampliado.

## 5.2 Metodologia

A metodologia MDC define um processo de geração assíncrona de diálogos multiturnos e multiparticipantes baseado em roteiros estruturados por rótulos, histórias como instâncias de execução, estados e regras de transição, além de mecanismos de redistribuição e anotação. Seu objetivo é viabilizar a criação controlada e escalável de diálogos, preservando a consistência semântica das conversas e registrando informações estruturais que apoiem a análise, a supervisão e o aumento posterior dos dados gerados.

Esta metodologia foi definida para se beneficiar de criação de diálogos por meio de *crowdsourcing*. Isso favorece a produção de textos mais naturais e com maior variabilidade lexical e estrutural, uma vez que a formulação das frases varia de acordo com cada participante. Quando combinada às técnicas de criação assíncrona da metodologia, essa escolha pode aumentar a velocidade de geração de diálogos. Ainda assim, a participação humana nesse processo pode ampliar a imprevisibilidade das conversas, problema mitigado pelo uso de roteiros de diálogos seguidos pelos participantes durante a geração.

A lista a seguir apresenta as definições adotadas na metodologia MDC.

- **Turno:** unidade textual, sentença ou parágrafo, associada a um rótulo que define seu objetivo no roteiro.
- **Roteiro:** sequência de rótulos que orienta a criação dos turnos.
- **Rótulo:** unidade semântica que codifica a intenção comunicativa de um turno e define critérios de equivalência para anotação e recombinação entre diálogos.

- **Diálogo:** sequência de turnos criada a partir de um roteiro.
- **História:** conjunto formado pelo roteiro e pelo diálogo resultante.

Em outras soluções existentes, como a ferramenta de criação de diálogos Mephisto [Urbanek and Ringshia, 2024], embora haja apoio em crowdsourcing, não há benefício da criação assíncrona de diálogos, o que impacta diretamente a escala e a velocidade do processo de criação. No MDC, por sua vez, a criação assíncrona prioriza a rotação de participantes ao longo do processo, ampliando ainda mais a variabilidade das histórias geradas.

Para viabilizar esse processo, o MDC se apoia integralmente no uso de roteiros para orientar as conversas e os turnos em criação. O roteiro consiste em uma sequência de descrições com intenção comunicativa esperada para cada turno. Assim, é possível orientar o propósito de cada mensagem sem restringir explicitamente o conteúdo ou as palavras utilizadas, desde que o rótulo do turno seja respeitado. Além disso, cada atividade está associada com um rótulo, que pode ser utilizado para automatizar gerações usando a relação entre os rótulos e na etapa de aumento dos dados gerados. A Tabela 5.1 apresenta um exemplo fictício de roteiro. Nela, é possível observar a alternância de responsabilidades e a estruturação do rótulo de cada mensagem a ser criada. O turno atual deve ser destacado para indicar qual conteúdo o participante da vez deve produzir.

| Turno | Rótulo | Descrição | Função |
|---|---|---|---|
| 1 | A | Envie uma saudação | Cliente |
| 2 | B | Responda a saudação e pergunte como pode ajudar | Atendente |
| 3 | C | Informe que deseja fazer um pedido | Cliente |
| 4 | D | Pergunte o que o cliente deseja | Atendente |
| 5 | E | Informe que deseja duas pizzas e um refrigerante | Cliente |
| 6 | F | Confirme os itens do pedido | Atendente |

Tabela 5.1: Exemplo de roteiro.

A utilização de roteiros permite estruturar os diálogos e fornece base para melhorar a consistência semântica ao longo das conversas. Para complementar esse comportamento com maior variabilidade, sempre que possível, após a adição de uma resposta e a troca de turno, o participante responsável deve ser substituído aleatoriamente por outro disponível. Com isso, diferentes usuários participam da criação de um mesmo diálogo, atuando em todas as funções presentes ao longo da conversa.

A Figura 5.1 apresenta o ciclo de criação de diálogos da metodologia MDC, distinguindo as ações do participante das operações automáticas da ferramenta. O fluxo destaca as opções de adicionar um turno e concluir uma história. As demais transições, incluindo o descarte de histórias inválidas, são apresentadas na Figura 5.2.

A ferramenta seleciona uma história disponível segundo as regras de redistribuição e a reserva para impedir a escrita simultânea por outros participantes. O participante recebe o roteiro e o diálogo já construído e decide, com base nesse contexto, se deve adicionar um novo turno ou solicitar a conclusão da história.

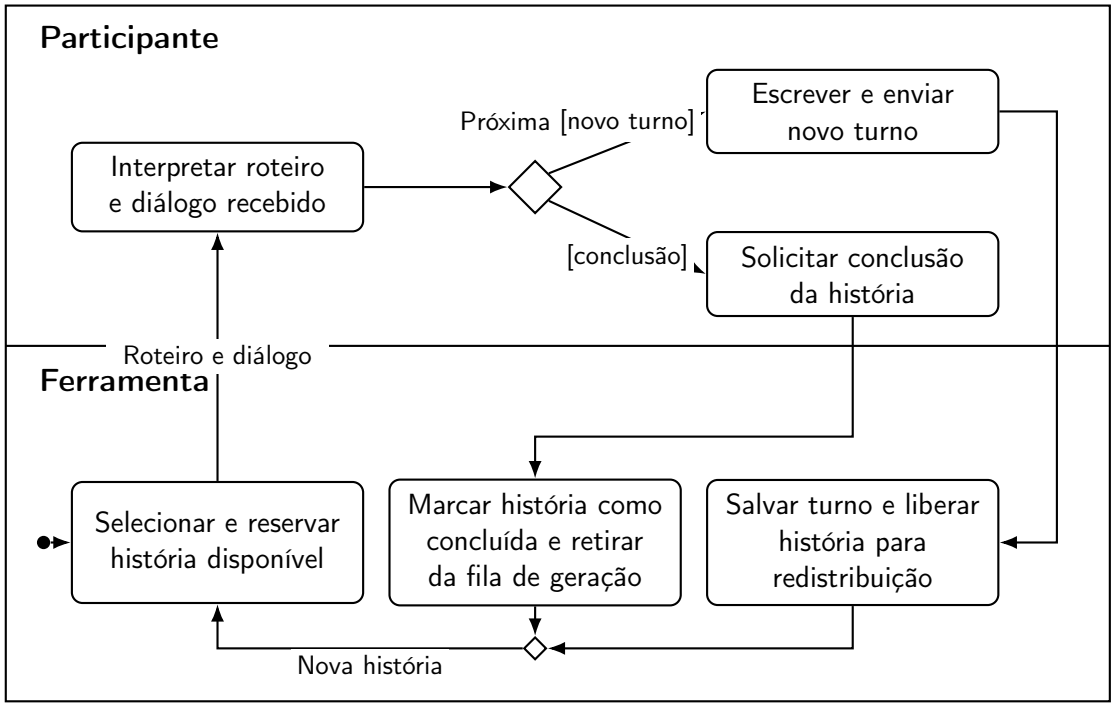



Figura 5.1: Fluxo de criação e edição de diálogos.

Quando um turno é enviado, a ferramenta o armazena e libera a história para novas contribuições. Quando a conclusão é solicitada, a história é marcada como concluída e deixa de participar de novas seleções, permanecendo entre as histórias finalizadas. Em ambos os casos, a interface solicita uma nova história disponível, permitindo que o participante continue a criação sem aguardar a continuidade do diálogo anterior.

Para que esse processo funcione adequadamente, foi definido um conjunto de estados e regras de transição. Isso garante a integridade das histórias, evitando que uma mesma história seja atribuída a mais de um participante ao mesmo tempo ou que turnos sejam gerados para histórias que não exigem mais alterações. Todas as histórias se iniciam no estado "Aberto", sem nenhum turno adicionado ao diálogo. Quando uma história é selecionada para a geração com um usuário, ela é marcada como "Ativa" e transiciona automaticamente para "Ativo Escrevendo". Ao concluir um turno, o usuário pode simplesmente adicionar o turno, fazendo com que a história retorne ao estado "Ativa" e passe a ter prioridade para redistribuição. Se o usuário a marcar como "Completa", a história é encerrada e passa a ser considerada pronta para uso. Se, por algum motivo, o usuário julgar que a história não é mais válida, é possível mover a história do estado "Ativo Escrevendo" para "Fechado", no qual ela deixa de ser redistribuída para geração e também não integra o conjunto final. A Figura 5.2 apresenta esses estados e suas transições.

Para lidar com problemas específicos de abandono de histórias, é recomendada a utilização de um limite de tempo para o estado "Ativo Escrevendo". Esse é o único estado em que esse tipo de controle se faz necessário, por ser o único capaz de impedir temporariamente que uma história seja redistribuída a outros participantes.

Como apresentado anteriormente, o usuário que participa da geração não tem visibilidade sobre os estados nem sobre o funcionamento geral da metodologia. Sua função, do ponto de vista da interação, consiste em receber histórias, escrever o próximo turno

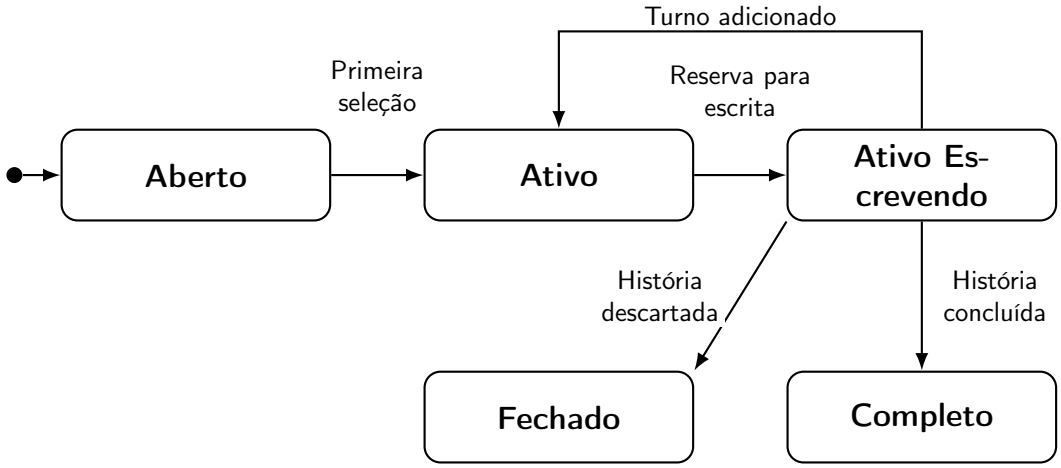



Figura 5.2: Diagrama de estados de uma história na metodologia MDC.

quando necessário e indicar a conclusão ou invalidação da história. Apesar dessa simplicidade de uso, suas responsabilidades ao longo dessas etapas são mais amplas.

De maneira simplificada, cabe ao usuário compreender o roteiro, entender o diálogo construído até o momento em que recebe a história, criar um novo turno e definir se ela está completa, seja por ter percorrido todos os turnos previstos, seja por apresentar algum problema que inviabilize seu uso. A Figura 5.3 apresenta os passos necessários para participar da criação assíncrona de diálogos.

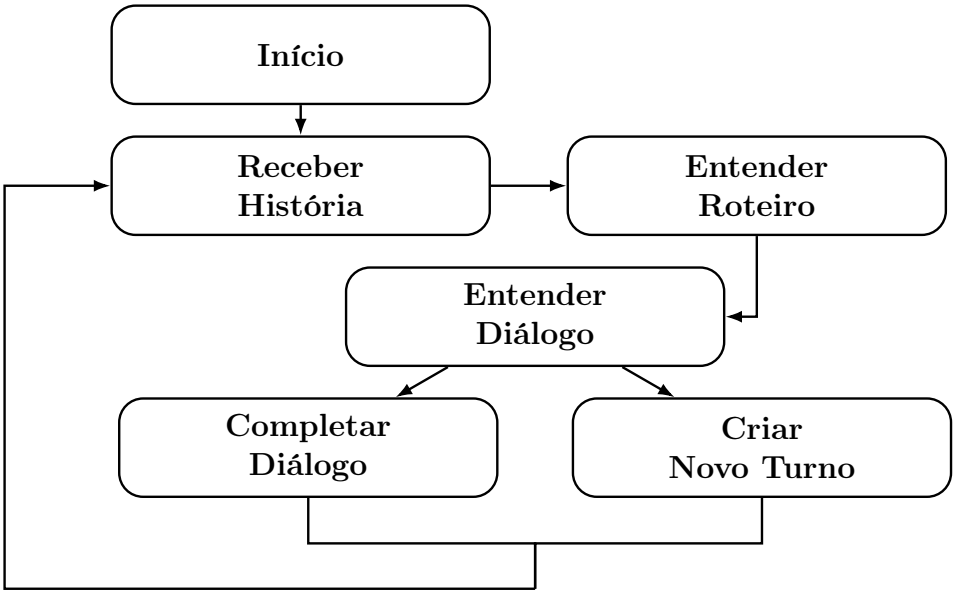



Figura 5.3: Passos para criar diálogos assíncronos.

Para compreender a história em criação, o participante deve interpretar o roteiro e o diálogo construído até aquele momento, identificando o rótulo de cada turno já criado e daqueles que ainda precisam ser produzidos. A leitura dos turnos existentes e como eles se relacionam entre si e com o roteiro é fundamental para que o próximo turno esteja de acordo com o diálogo e com sua possível progressão. Quando o participante domina esses elementos, é possível criar turnos semanticamente mais coerentes entre si e alinhados às

definições do roteiro.

Um dos principais aspectos da metodologia MDC é a troca de participantes durante a geração dos diálogos. Sempre que um participante requisita uma história, ele recebe uma instância nos estados "Aberto" ou "Ativo", com prioridade para histórias cujo último participante seja diferente do requisitante atual. Para aumentar a variabilidade de participantes em cada história, foi criado um sistema de decisão. Esse sistema é apresentado na Figura 5.4.

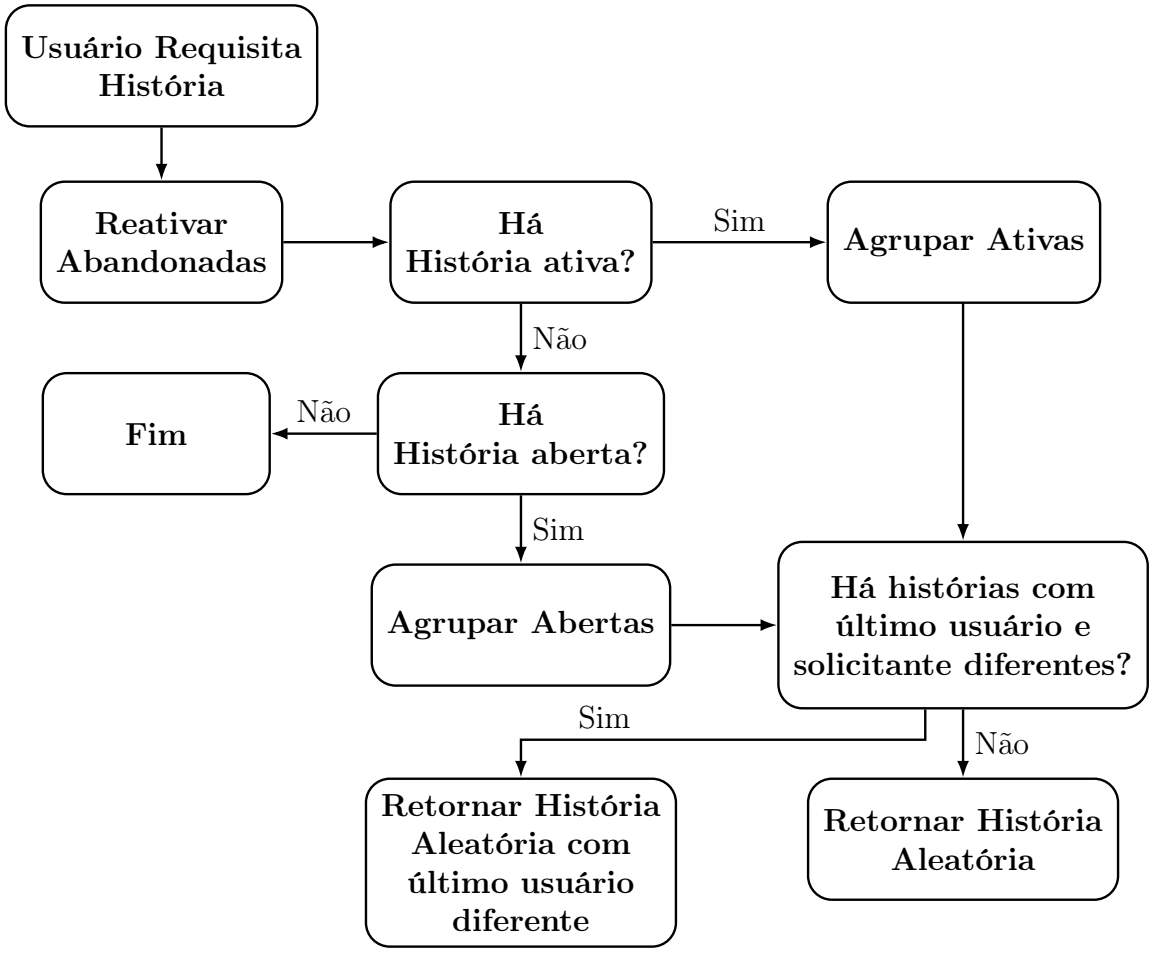



Figura 5.4: Fluxo para selecionar uma história.

É possível observar que, após a requisição de uma história, aquelas que foram abandonadas, ou seja, que ultrapassaram o tempo limite de geração, são reativadas. Isso ocorre para permitir a seleção das que ainda possam ser redistribuídas. Em seguida, é verificada a existência de alguma história ativa que possa ser distribuída. Em caso afirmativo, essas histórias são agrupadas para participar da seleção.

Caso não existam histórias ativas, as marcadas como em aberto são selecionadas. Se também não houver nenhuma em aberto, o processo é encerrado por falta de histórias que possam receber novas contribuições.

Em ambos os casos, quando ainda existam histórias ativas ou abertas, a última verificação consiste em tentar filtrar histórias em que o último participante da geração seja diferente do requisitante. Se isso for possível, é retornado uma instância em que haja

maior distanciamento entre os participantes. Caso contrário, uma aleatória é selecionada e retornada.

Todo esse processo fica invisível para o participante que fez a requisição, pois ele apenas recebe uma história e dá continuidade ao processo de criação. A lista a seguir sintetiza as propriedades operacionais esperadas da metodologia.

- **Criação assíncrona:** permite que múltiplos participantes contribuam sem a necessidade de interação síncrona.
- **Controle de estados:** organiza o ciclo de vida das histórias e regula quando elas podem ser distribuídas.
- **Maior escala:** possibilita gerar mais histórias em paralelo, com menor dependência de coordenação.
- **Maior velocidade:** reduz o tempo ocioso ao alternar histórias enquanto se aguardam contribuições de outros participantes.
- **Maior diversidade lexical:** amplia a variabilidade de estilo e vocabulário por meio da rotação de participantes.
- **Controle de integridade:** evita conflitos durante a geração de turnos por meio do controle de estados.
- **Padronização:** permite rótulos claros para cada turno por meio de roteiros, mantendo a coerência entre as histórias.

Durante todo o processo de geração de dados, sempre que um turno é incluído em uma história, ele fica automaticamente relacionado ao turno correspondente no roteiro. Dessa maneira, os dados são anotados de forma transparente durante a geração, indicando, ao menos, o rótulo de cada turno escrito. Além do valor de gerar diálogos com turnos anotados, essa abordagem pavimenta o caminho para que trechos de diálogos associados a rótulos equivalentes possam ser recombinados sem prejuízo semântico, desde que também sejam preservadas as dependências entre turnos relacionados. A Tabela 5.2 apresenta dois diálogos construídos a partir do mesmo roteiro. É possível observar que os turnos são intercambiáveis sem prejuízo da semântica nesse exemplo.

Para que isso seja possível, é necessário que a criação dos roteiros seja feita de maneira cuidadosa, identificando turnos e trechos em que essa troca possa ocorrer. A Figura 5.5 apresenta um exemplo de diagrama para a definição de roteiros nos quais esse tipo de estratégia é priorizado. Nesse exemplo, é possível observar que, além dos rótulos permitirem permutação, o próprio fluxo dos roteiros gerados com base nesse diagrama pode apresentar variações.

Como todos os roteiros gerados utilizam a estrutura apresentada no exemplo-base, os diálogos resultantes passam a ter mais variações em função dos diferentes circuitos percorridos. O próprio circuito pode ser intercambiado com o de outros roteiros, já que as estruturas são equivalentes.

| Turno | Rótulo | Descrição | Diálogo 1 | Diálogo 2 |
|---|---|---|---|---|
| 1 | A | Envie uma saudação | Oi | Olá |
| 2 | B | Responda a saudação e pergunte como pode ajudar | Olá! Tudo bem? Como posso ajudar? | Bem-vindo! Como podemos auxiliar hoje? |
| 3 | C | Informe que deseja fazer um pedido | Gostaria de fazer um pedido | Quero fazer um pedido |
| 4 | D | Pergunte o que o cliente deseja | Claro, o que você deseja pedir? | Qual vai ser o pedido? |
| 5 | E | Informe que deseja duas pizzas e um refrigerante | Quero duas pizzas e um refrigerante | 2 pizzas e refrigerante |
| 6 | F | Confirme os itens do pedido | Vai ser apenas as pizzas e o refrigerante? | Só as pizzas e o refrigerante? |

Tabela 5.2: Exemplo de rótulo e diálogos.

Para que isso seja possível, quando o roteiro define elementos específicos que alteram apenas parte dos rótulos, como, por exemplo, "Informe o total de R$ X,00", é necessário que as trocas preservem a consistência, para que a semântica e a coerência dos diálogos sejam mantidas. Quando turnos dependem de uma mesma entidade ou informação, como uma pergunta sobre um alimento e sua resposta, as variações desses turnos devem ser recombinadas em conjunto e manter a mesma entidade. A equivalência dos rótulos de cada turno, considerada isoladamente, não garante essa consistência.

Como as histórias criadas utilizam roteiros e os diálogos já estão associados aos rótulos anotados, é possível inspecionar individualmente cada parte da criação. Isso permite uma validação refinada de todos os aspectos das conversas geradas, especialmente para fins de avaliação e depuração, para verificar se o rótulo de cada turno foi atendido pelo texto associado.

Essa inspeção granular da criação resulta em um controle de qualidade superior, pois permite determinar, em diferentes momentos, quais histórias podem realmente compor o conjunto de dados final. Durante a geração, todos os participantes podem identificar quando uma história deixa de estar adequada ao objetivo e classificar a história como "Fechada", o que torna o processo mais robusto pela troca contínua de participantes ao longo da criação. Após a conclusão das histórias, ainda é possível realizar uma avaliação individual, classificando essas histórias como adequadas ou inadequadas.

Para fins estatísticos, como os dados são anotados e os roteiros são previamente padronizados, é possível determinar com precisão os volumes esperados em um ciclo de geração por crowdsourcing. Complementarmente, a partir do conjunto de dados gerado, é possível identificar quais roteiros originaram quais histórias, bem como a quantidade disponível de dados para cada tipo de rótulo e sequência de rótulos.

Entre seus principais benefícios, esse processo permite a utilização de técnicas para ampliar artificialmente a quantidade de dados gerados, por meio da recombinação de turnos associados a rótulos equivalentes. A lista a seguir complementa os benefícios apresentados anteriormente com aqueles obtidos pelo controle e pela geração de roteiros

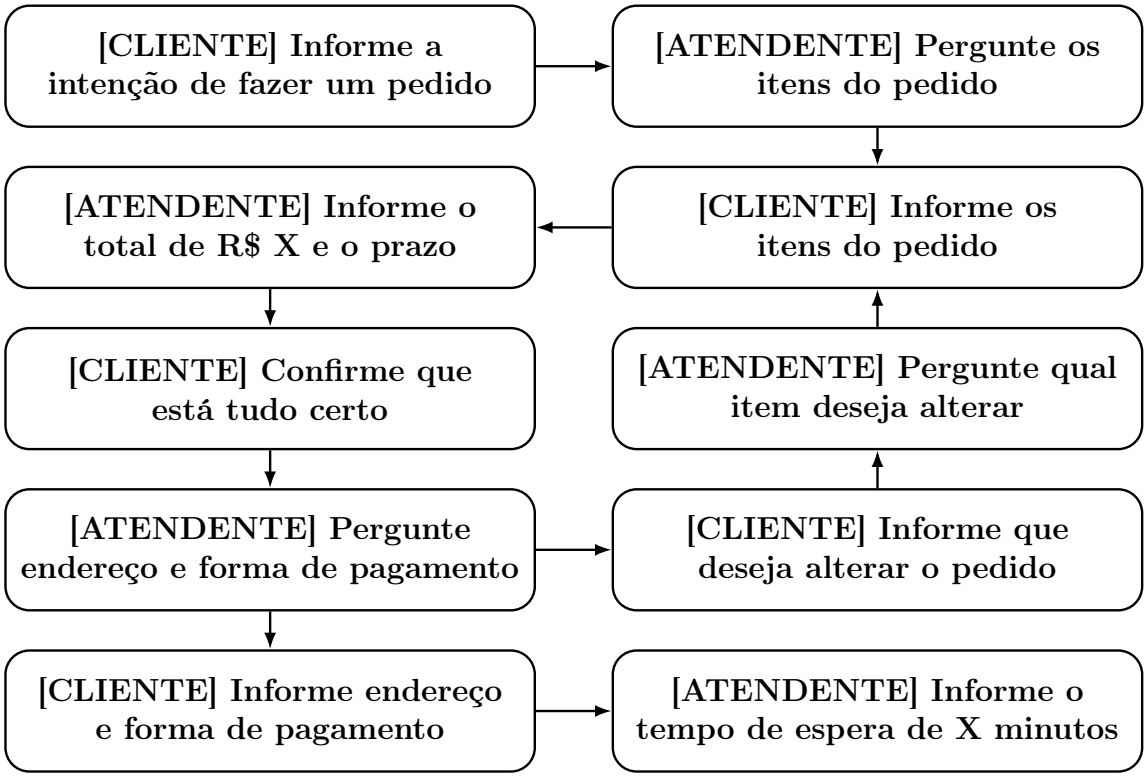



Figura 5.5: Fluxo principal simplificado para pedir pizza, com loop de alteração antes de endereço e pagamento.

e rótulos.

- **Turnos anotados:** cada turno fica associado ao rótulo do roteiro, viabilizando supervisão direta.
- **Aumento de dados:** abre caminho para ampliar o volume de dados gerados.
- **Rastreabilidade:** facilita a auditoria e a depuração de erros ao relacionar trechos do diálogo aos rótulos previstos.
- **Controle de qualidade:** possibilita validar por inspeção humana se um turno atende ao rótulo esperado.
- **Análises por intenção:** permite medir a cobertura e a distribuição de rótulos no conjunto gerado.

A técnica escolhida para ampliar os dados gerados pela metodologia MDC é a recombinação combinatória de turnos condicionada por rótulos equivalentes. Essa técnica combina as alternativas disponíveis em cada posição do roteiro, preservando a posição estrutural de cada turno e evitando alterações na ordem conversacional. Para um roteiro com $L$ posições e $k$ alternativas em cada posição, o número teórico de combinações é $k^L$; quando o número de alternativas varia por posição, a contagem é o produto das alternativas disponíveis em cada uma. Essa abordagem é especialmente eficiente em cenários nos quais há interseção entre os rótulos dos roteiros.

A Figura 5.6 ilustra, de forma geral, as permutações possíveis das letras "ABC"($n!$). Esse exemplo e o crescimento fatorial apresentado na Figura 5.7 ilustram a expansão combinatória, mas não representam diretamente a operação da MDC: nela, a ordem dos turnos permanece fixa e as alternativas são recombinadas entre posições correspondentes, como ilustrado na Figura 5.8.

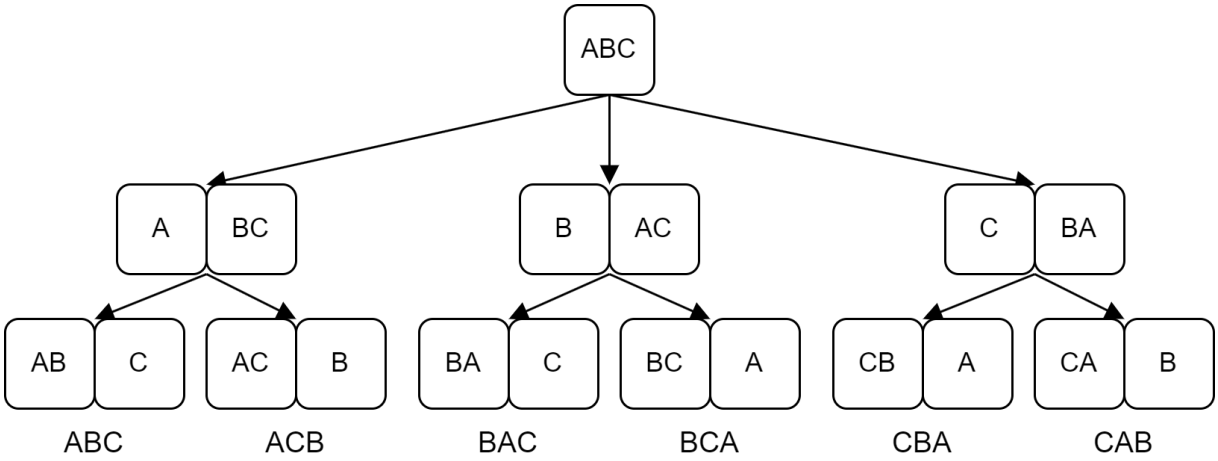



Figura 5.6: Permutações possíveis das letras ABC, como ilustração do crescimento fatorial ($n!$).

A Figura 5.7 apresenta o crescimento fatorial do número de permutações em função do tamanho do conjunto. É possível notar que essa quantidade cresce rapidamente, chegando ao ponto de inviabilizar o cálculo de todas as possibilidades. Apesar dessa limitação, essa curva serve apenas como ilustração geral do crescimento combinatório e não corresponde à contagem de diálogos aumentados pela MDC, que segue a recombinação posicional descrita anteriormente.

A grande quantidade de variações permite descartar dados aumentados que apresentem muitas similaridades com outros diálogos já existentes, seja da geração original, seja da própria aumentação. Assim, os novos dados criados a partir desse processo passam a conter elementos que os tornam suficientemente diferentes para que permaneçam relevantes.

A Figura 5.8 mostra como o conceito de permutação opera sobre diálogos e rótulos para a geração de novos diálogos. Nessa figura, os diálogos são representados pela letra "D" seguida do número do diálogo. Os rótulos de cada turno são representados pela letra "L" seguida do número do turno com esse rótulo em um determinado diálogo. É possível observar que, para cada rótulo, são criadas diversas variações de diálogo.

A utilização desse tipo de abordagem permite a criação de uma grande quantidade de dados com custo relativamente baixo, pois depende principalmente do custo computacional necessário para definir as permutações.

Como mencionado, a criação de diálogos similares decorre do funcionamento da técnica de permutação utilizada. Para lidar com isso, duas abordagens principais podem ser adotadas. A primeira consiste em limitar o número de vezes que cada rótulo pode ser utilizado durante as permutações, forçando, assim, a troca do rótulo de maneira mais precoce para evitar muitos diálogos com começos ou trechos parecidos.

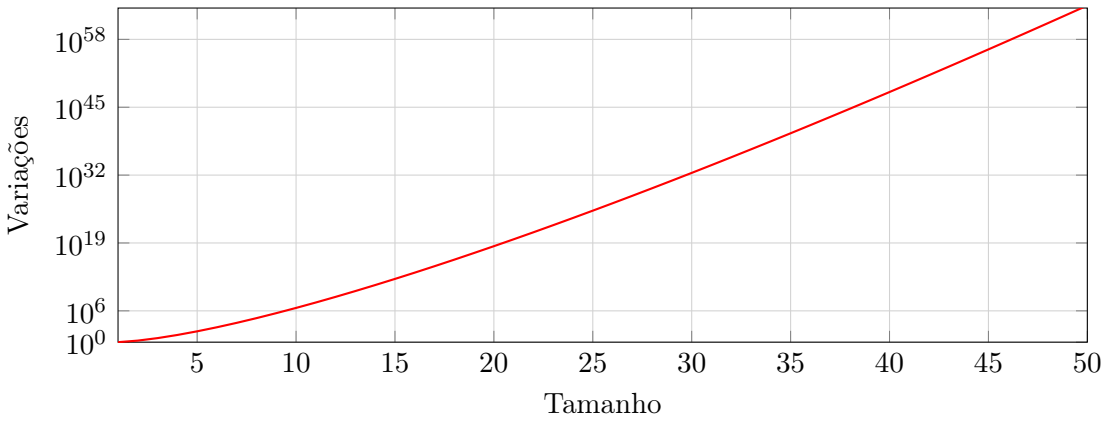



Figura 5.7: Crescimento do número de permutações ($n!$) em função do tamanho do conjunto.

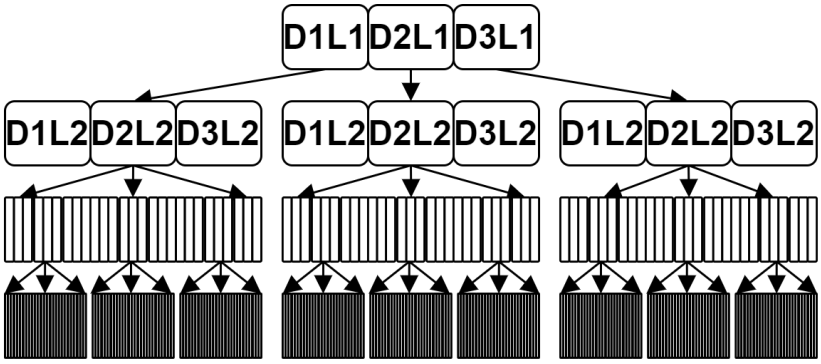



Figura 5.8: Exemplo de permutação.

Outra abordagem eficaz é associar os dados aumentados por meio da permutação a outras técnicas, como, por exemplo, o uso de LLMs para criar variações de alguns turnos em cada diálogo gerado por permutação. Isso amplia a variabilidade, embora também reintroduza a complexidade e o custo característicos de técnicas tradicionais.

Além dos benefícios já descritos ao longo desta seção, a utilização da permutação baseada em rótulos traz diversas vantagens. A lista a seguir sintetiza as propriedades esperadas da abordagem proposta.

- **Ganho combinatorial de escala:** um conjunto pequeno de histórias pode originar muitas variações válidas quando há interseção de rótulos entre roteiros.
- **Preservação de semântica:** a troca de turnos é restrita a rótulos equivalentes, reduzindo o risco de gerar exemplos incoerentes.
- **Baixo custo de anotação:** as variações herdam os rótulos dos turnos originais, evitando esforço manual adicional.
- **Controle de diversidade:** é possível limitar permutações por rótulo e descartar variações redundantes, mantendo apenas as mais relevantes.
- **Ajuste de distribuição:** a permutação pode ser usada para aumentar seletivamente rótulos/sequências raras, mitigando desbalanceamentos.

- **Geração *model-free*:** permite ampliar os dados sem depender de modelos generativos, embora possa se beneficiar de seu uso associado para criar novos textos.
- **Preservação de estilo humano:** reutiliza turnos escritos por participantes, preservando naturalidade e variação linguística.

## 5.3 Ferramenta

Para viabilizar a criação assíncrona de diálogos multiturnos e multiparticipantes a partir de roteiros, propomos uma ferramenta que implementa a metodologia MDC. Seu objetivo é permitir a criação de novos conjuntos de dados, especialmente para idiomas e cenários com escassez. Para isso, a ferramenta oferece uma interface simples aos participantes, abstraindo a complexidade da gestão das histórias e concentrando sua atuação no entendimento da história e na geração dos turnos. Além de apoiar a criação de novos conjuntos de dados, a ferramenta também funciona como uma validação prática das propostas da metodologia MDC.

A ferramenta adota arquitetura cliente-servidor, na qual o cliente atua como interface de interação e o servidor centraliza o gerenciamento de estados, histórias e validações.

O servidor centraliza o gerenciamento de roteiros, diálogos, histórias e estados, implementando controle de concorrência com bloqueio exclusivo por história durante a escrita de turnos, evitando conflitos entre participantes simultâneos. Com isso, é possível realizar verificações com maior consistência e garantir integridade durante a escrita concorrente de turnos.

Já o cliente é responsável por servir como interface para os participantes. Ele foi projetado para ser o mais simples possível. A Figura 5.9 apresenta a tela principal utilizada pelos participantes. Nela, é possível observar que a interface está dividida em três partes. A interface organiza controles de estado, roteiro anotado e histórico de turnos, permitindo ao participante compreender o contexto e produzir o próximo turno.

Para simplificar o uso, o cliente foi projetado para permanecer publicamente hospedado no GitHub [GitHub, Inc., 2026] [1].

A Figura 5.10 apresenta o funcionamento da comunicação da ferramenta desenvolvida. Nela, é possível observar que o GitHub é utilizado apenas para distribuir a interface do cliente, que funciona como uma aplicação executada no navegador. Uma vez iniciado, o cliente se comunica com o servidor por meio de requisições HTTP. A comunicação via HTTP foi adotada por seu modelo *stateless*, no qual cada requisição contém todas as informações necessárias para processamento, eliminando dependência de estado no cliente, facilitando escalabilidade e integrações futuras.

O cliente funciona apenas como camada de interface, sem alterar ou criar dados diretamente. Todas as alterações são realizadas pelo servidor. Para garantir o funcionamento transparente dessa arquitetura, as mensagens seguem um esquema JSON contendo os campos `user_id`, `story_id`, `access_key`, `action` e `turn`, todos validados quanto à integridade e autenticação no servidor.

[1] https://github.com/MatheusFerraroni/MDC-client

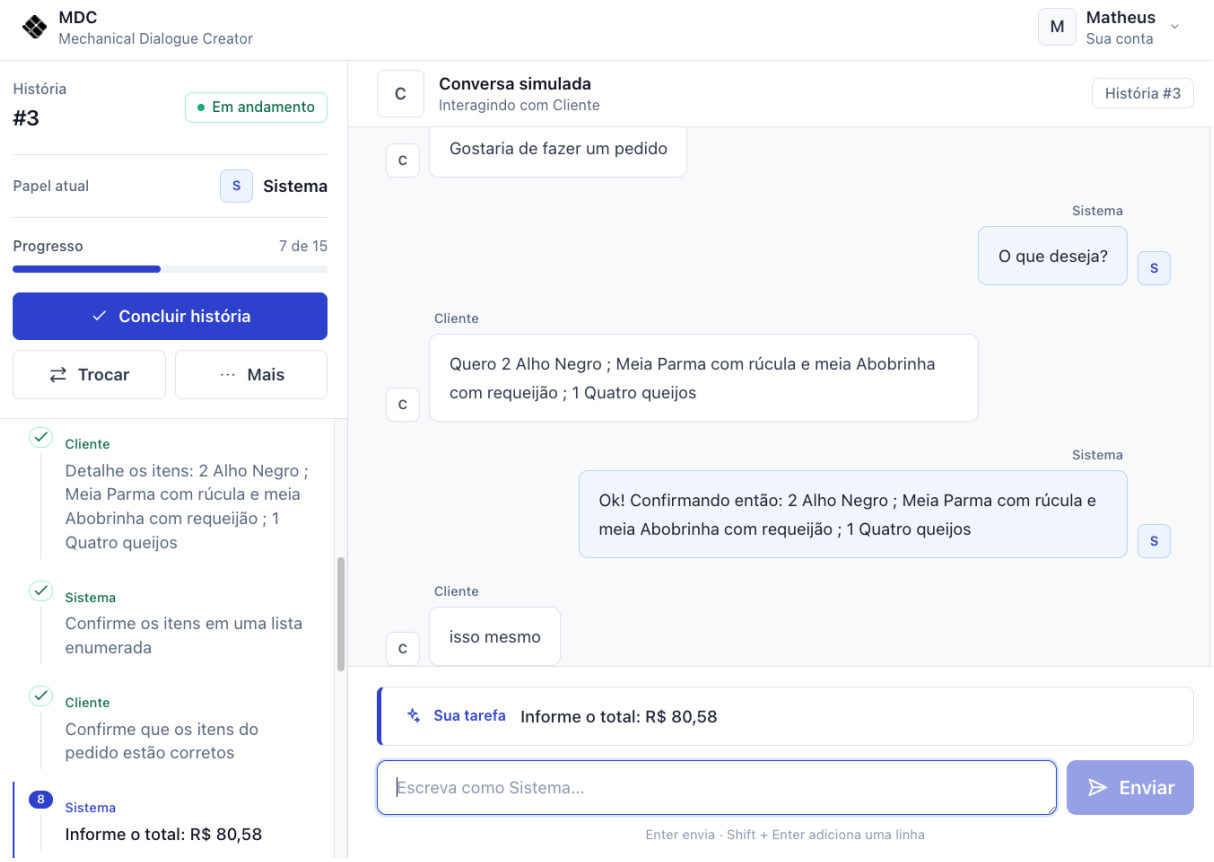



Figura 5.9: Exemplo da interface da ferramenta.

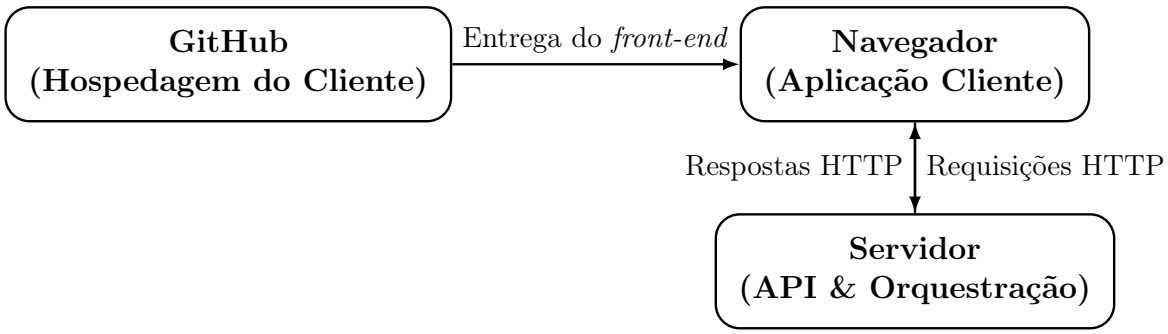



Figura 5.10: Diagrama simplificado da comunicação entre o cliente e servidor MDC.

Como etapa inicial do processo de geração como participante, a primeira ação após acessar a ferramenta é configurar os dados de acesso ao servidor em execução. Nessa configuração, o participante deve informar ‘e-mail‘, usado para identificar o participante de forma única, ‘Server Endpoint‘, usado para definir como acessar o servidor, e ‘Server Key‘, usada para autenticar o acesso ao servidor.

Após a configuração inicial e o acesso ao servidor, o participante passa a seguir um ciclo iterativo de recepção de história, interpretação do contexto a partir do roteiro e do diálogo já construído, geração de turno e atualização do estado. Nesse processo, a interface apresenta uma história pronta para criação, permitindo ao usuário identificar os objetivos gerais do roteiro e o objetivo específico do turno que ainda precisa ser escrito antes de iniciar a redação. Ao final de cada ciclo, o participante pode enviar o novo turno,

concluir a história ou descartar a história, conforme sua avaliação, e uma nova história é então requisitada automaticamente.

Toda a interface foi projetada para ser utilizada com pouco ou nenhum treinamento, desde que os participantes compreendam o fluxo e suas responsabilidades durante a criação de novos turnos.

Embora servidor e cliente tenham sido projetados em conjunto, o servidor foi desenvolvido para permitir a integração com outros clientes que demandem uma interface customizada. Nesse cenário, ele continua responsável pelo gerenciamento das histórias, das transições de estado, dos turnos criados, dos usuários participantes e das demais funcionalidades, mesmo quando conectado a uma interface diferente. Para viabilizar esse tipo de integração, foi definido um conjunto de rotas de API. A lista a seguir apresenta essas APIs.

- **Rotas relacionadas a usuários**
  - **check_user:** verifica se um usuário já existe.
  - **add_user:** adiciona um novo usuário.
- **Rotas relacionadas a histórias**
  - **get_story_info:** retorna todos os dados relacionados a uma história.
  - **story_mark_as_complete:** marca uma história como concluída.
  - **story_mark_as_closed:** finaliza uma história e a marca como fechada, impedindo que ela seja usada futuramente.
  - **get_story:** retorna uma história disponível e aplica bloqueio lógico para evitar escrita concorrente.
- **Rotas gerais**
  - **get_situation:** retorna uma descrição completa dos dados controlados pelo servidor, incluindo informações sobre usuários, diálogos, turnos e histórias.

Embora o servidor utilize mecanismos de cache para otimização de desempenho, os dados são persistidos de forma incremental a cada atualização de turno, garantindo tolerância a falhas e recuperação consistente após interrupções. Isso aumenta a confiabilidade do sistema em caso de interrupções inesperadas durante o processo de geração.

Cada história gera um arquivo JSON contendo o roteiro e os turnos criados no diálogo. Para cada turno, são salvos dados relacionados à criação, como horário, participante e o texto correspondente. Para fins de depuração, são armazenados arquivos relacionados às interações com o servidor, aos usuários participantes e a metadados não diretamente associados às histórias e ao processo de criação.

Uma vez concluídas, as histórias já se encontram no formato adequado para a execução do módulo de permutação, responsável por ampliar os dados por meio da troca de turnos ou trechos com rótulos equivalentes. Como saída, são gerados arquivos com a mesma estrutura das histórias originais, um para cada nova história criada. Para executar o

módulo, o único parâmetro necessário é o diretório em que os arquivos estão armazenados. Opcionalmente, o módulo pode ser estendido para limitar a geração a um número menor de variações por rótulo, produzindo conjuntos de dados com mudanças mais expressivas.

Para iniciar um novo processo de geração de dados, a inicialização consiste na clonagem do repositório do servidor[2] e na configuração dos dados de entrada. Após a adição dos diálogos e o início da aplicação, o servidor estará pronto para receber as requisições do cliente, que pode ser acessado diretamente pelo navegador por meio da versão pública hospedada.

## 5.4 Conjunto de Dados *Nutritious Food*

Nesta seção, é apresentado o conjunto de dados *Nutritious Food*. O conjunto foi produzido com a ferramenta implementada a partir da metodologia MDC, na qual participantes utilizam roteiros predefinidos para construir, de forma assíncrona, os turnos de cada história. Esse ambiente orienta a escrita por meio de instruções claras e exemplos, contribuindo para padronizar o estilo textual, reduzir ambiguidades e manter a temática central ao longo das narrativas.

O processo de criação seguiu etapas de escrita guiada e revisão, contribuindo para a coerência narrativa e a aderência aos objetivos do estudo. Esta seção está organizada em três eixos articulados ao longo da exposição: o fluxo de coleta dos diálogos, os critérios de qualidade aplicados durante a criação e a curadoria, e a caracterização estrutural do conjunto, incluindo o tipo de conteúdo produzido, a organização dos turnos e seu uso nas análises subsequentes. Ao longo dessa descrição, também discutimos de que modo as decisões de anotação e curadoria influenciam a consistência do material e sua adequação a experimentos posteriores.

Os critérios de qualidade adotados consideram a aderência semântica entre cada turno e o rótulo correspondente, a coerência global do diálogo em relação ao roteiro e a consistência linguística entre turnos adjacentes. Esses aspectos são avaliados pelos participantes humanos ao longo do processo de criação e posteriormente retomados na etapa de curadoria.

O conjunto de dados *Nutritious Food* foi criado como instância de validação da metodologia proposta, permitindo observar seu comportamento em um cenário controlado com múltiplos participantes e roteiros. Nesse contexto, também é possível explorar como as contribuições assíncronas dos participantes ampliam a diversidade linguística dos textos.

Neste conjunto de dados de validação, foram criados apenas cinco roteiros. Nesses roteiros, a intersecção de rótulos foi planejada para evidenciar o potencial da etapa de aumento de dados. A lista a seguir apresenta uma breve descrição:

- Dois roteiros eram exatamente iguais.
- Dois roteiros substituíram as perguntas sobre “chocolate” e “chocolate amargo” por perguntas sobre “laranja” e “morango”.

[2]https://github.com/MatheusFerraroni/MDC-server

- Um roteiro incluiu apenas uma pergunta sobre a ingestão diária ideal de calorias.

O bloco a seguir apresenta um dos roteiros utilizados nessa validação, traduzido para o português. As perguntas sobre alimentos específicos aparecem nas linhas 6 e 10. Ambas tratam da quantidade de calorias, variando apenas o item mencionado, como, por exemplo, chocolate ou chocolate amargo. O restante do diálogo permanece inalterado, independentemente dessas linhas.

```
1 - Start
2 -  [Cliente] Cumprimente o sistema de forma educada.
3 -  [Sistema] Cumprimente o usuário e pergunte se pode ajudar.
4 -  [Cliente] Pergunte se você pode fazer uma pergunta sobre a
quantidade de calorias em um alimento.
5 -  [Sistema] Confirme que esse é o tipo de pergunta com o qual você
pode ajudar o usuário.
6 -  [Cliente] Pergunte: Quantas calorias há no chocolate?
7 -  [Sistema] Informe: O chocolate tem aproximadamente 546 calorias
por 100 gramas.
8 -  [Cliente] Agradeça ao sistema.
9 -  [Sistema] Pergunte se pode ajudar em algo mais.
10 - [Cliente] Pergunte: Quantas calorias há no chocolate amargo.
11 - [Sistema] Informe: O chocolate amargo tem aproximadamente 185
calorias por 100 gramas.
12 - [Cliente] Diga que isso é interessante e agradeça ao sistema.
13 - [Sistema] Concorde com o usuário.
14 - [Cliente] Faça uma despedida.
15 - [Sistema] Faça uma despedida.
```

A geração dos diálogos foi realizada por instanciação direta dos roteiros, com a criação de um único diálogo para cada roteiro, totalizando cinco histórias. Apesar desse volume ser insuficiente para o treinamento de LLMs, é possível dar continuidade à metodologia MDC e aumentar os dados até um volume mais adequado.

Como parte da validação da metodologia, também avaliamos seu mecanismo de aumento de dados, baseado na permutação de turnos associados a rótulos equivalentes, com o objetivo de ampliar a quantidade de diálogos e preservar sua semântica e seus objetivos.

Este exemplo ilustra como a variação de turnos associados a rótulos equivalentes permite a recombinação posterior, mantendo a estrutura do roteiro e viabilizando a etapa de aumento de dados.

A Figura 5.11 mostra um exemplo de como esses fluxos se expandem rapidamente. Cada caixa da figura representa um rótulo para o qual existem N variações possíveis de turno. Assim, o número total de histórias cresce de forma combinatória em função do número de variações associadas a cada rótulo, da quantidade de rótulos presentes no roteiro e da existência de fluxos alternativos decorrentes da intersecção entre rótulos. Neste exemplo, podem ser geradas $8 \times 8 \times (2+4+2) \times 8 \times 8 \times 8 = 262.144$ novas histórias.

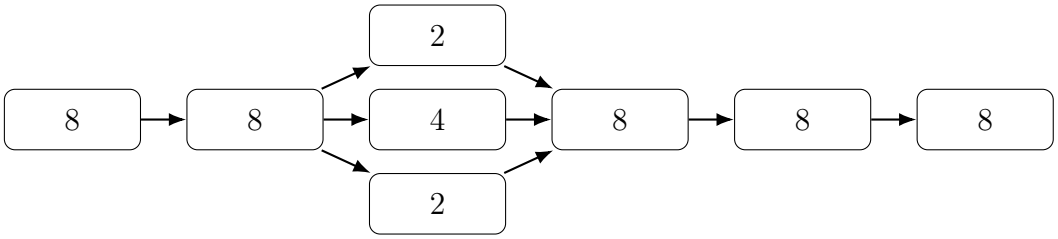



Figura 5.11: Multiplicação dos fluxos.

Para mitigar o problema da geração de diálogos similares, é possível utilizar um limite da quantidade de vezes que cada turno pode ser utilizado. A Figura 5.12 implementa esse limite no exemplo da Figura 5.11, gerando $2 \times 2 \times (2+2+2) \times 3 \times 4 \times 4 = 1.152$ diálogos.

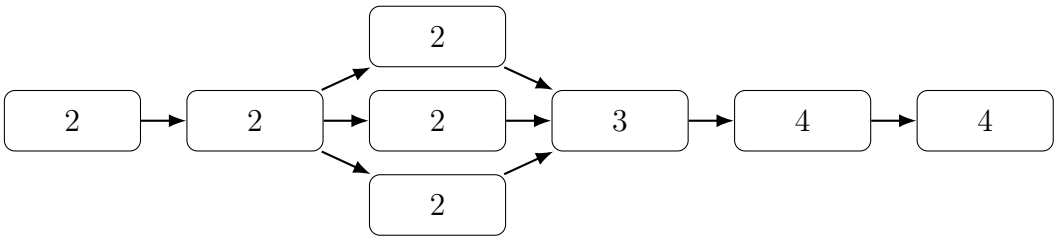



Figura 5.12: Multiplicação dos fluxos com aplicação de limite de uso por rótulo.

Para interpretar os valores da Tabela 5.3, considera-se um roteiro linear de 14 posições variáveis, correspondentes às linhas 2 a 15 do exemplo, com $k$ alternativas em cada posição. Sob essa hipótese de escolhas independentes, o total de combinações é $k^{14}$. Portanto, nessa tabela, $k$ representa o número de alternativas por posição e não um limite máximo de reutilização de cada turno. Cada alternativa em uma posição aparece em $k^{13}$ combinações. Para $k = 3$, isso corresponde a $3^{13} = 1.594.323$ reutilizações da mesma alternativa. Assim, as combinações compartilham os mesmos turnos de origem e não representam exemplos independentes.

| **Alternativas por posição ($k$)** | **Combinações** |
|---|---|
| 1 | 1 |
| 2 | 16.384 |
| 3 | 4.782.969 |

Tabela 5.3: Combinações teóricas para 14 posições com diferentes quantidades de alternativas por posição.

Essa contagem não garante que todas as combinações preservem a coerência semântica do conjunto *Nutritious Food*. Nas linhas 6 e 7, a pergunta e a resposta devem tratar do mesmo alimento. O mesmo vínculo existe entre as linhas 10 e 11. Como há no máximo duas realizações para cada alimento nesses roteiros, o cenário com $k = 3$ exige combinar alternativas de alimentos distintos.

Em experimentos posteriores com os diálogos aumentados, a divisão entre treino, validação e teste deve ocorrer antes da expansão, no nível dos roteiros-base. Roteiros idênticos ou equivalentes e todas as combinações derivadas deles devem permanecer na mesma par-

tição. Uma divisão aleatória após a expansão pode distribuir turnos iguais ou semelhantes entre treino e teste e superestimar o desempenho do modelo.

Embora não tenham sido implementadas nesta validação, outras técnicas podem ser empregadas para ampliar ainda mais o volume de dados ou aumentar a variabilidade dos diálogos, uma vez que os diálogos expandidos já são produzidos de acordo com as definições da metodologia.

O conjunto de dados está disponível para download público e pode ser obtido diretamente no repositório institucional indicado.[3] Essa disponibilização facilita o uso do material em estudos comparativos e em pesquisas futuras, além de apoiar a verificação dos resultados e das análises apresentadas.

## 5.5 Conclusão do Capítulo

Neste capítulo, propusemos e validamos uma metodologia para a geração sintética controlada de dados conversacionais em cenários com baixa disponibilidade de dados específicos, sua implementação em uma ferramenta cliente-servidor e sua validação com o conjunto de dados *Nutritious Food*. Em conjunto, os resultados obtidos a partir da aplicação da metodologia indicam que a criação assíncrona orientada por roteiros, com rotação de participantes e anotação por rótulos, permite ampliar a diversidade dos diálogos e viabilizar ganhos de escala por meio da recombinação baseada em rótulos equivalentes, desde que as dependências semânticas entre turnos relacionados sejam preservadas. As contagens combinatórias apresentadas ilustram possibilidades teóricas e não correspondem necessariamente à quantidade de diálogos semanticamente válidos ou independentes. Adicionalmente, a metodologia integra mecanismos de *crowdsourcing* assíncrono, controle de estados e avaliação de qualidade, consolidando um processo estruturado de geração de dados.

Esses resultados indicam que a abordagem proposta é adequada para a geração controlada de dados conversacionais em cenários com escassez de dados, servindo como base para as análises e experimentos apresentados nos capítulos seguintes.

[3] `https://www.lrc.ic.unicamp.br/~ferraroni/datasets/nutri/nutri.zip`

# Capítulo 6

# Tratamento e Adaptação de Conjuntos de Dados Existentes

**Pergunta de Pesquisa 3:** Como conjuntos de dados já consolidados na literatura podem ser adaptados para viabilizar seu uso eficaz no treinamento de Large Language Models?

Este capítulo aborda a preparação e a adaptação de conjuntos de dados já existentes para que possam ser empregados de forma segura e eficaz no treinamento e na avaliação de modelos, tema frequentemente subestimado em pesquisas com LLMs. Embora a literatura, nos últimos anos, tenha ampliado a discussão sobre a qualidade e a curadoria de dados, ainda é comum que o uso de grandes volumes de informação masque problemas estruturais presentes nos conjuntos de dados utilizados. Nesse contexto, é ainda mais relevante a disponibilidade de conjuntos de dados específicos em termos de idioma e domínio para o treinamento de LLMs.

A utilização de conjuntos de dados adequados ao domínio e ao idioma impacta diretamente a performance dos modelos. Apesar disso, a construção desses conjuntos demanda tempo e recursos significativos. Por isso, o reaproveitamento de conjuntos de dados amplamente utilizados na literatura frequentemente se torna a alternativa mais viável. Ainda assim, muitos desses conjuntos reutilizados foram criados originalmente com foco no volume e na abrangência temática.

Como consequência, conjuntos de dados consolidados podem se tornar inadequados para novos cenários de treinamento quando apresentam ruído estrutural, desalinhamento com o domínio, incompatibilidades linguísticas ou inconsistências entre texto e anotações. Nesta tese, a adaptação de um conjunto de dados é considerada bem-sucedida quando reduz essas inadequações sem comprometer sua utilidade para treinamento ou avaliação.

Esse critério envolve três condições: reduzir fontes identificáveis de ruído, preservar propriedades semânticas e estruturais necessárias ao uso pretendido e manter escala suficiente para que o conjunto continue representativo. Com base nisso, investigamos estratégias de adaptação por filtragem e curadoria, bem como por tradução e realinhamento estrutural, com foco na preservação de propriedades linguísticas e semânticas relevantes para o treinamento de LLMs.

A Seção 6.1 apresenta uma visão geral das etapas de adaptação e reaproveitamento de conjuntos de dados discutidas neste capítulo. Em seguida, a Seção 6.2 descreve o

processo adotado para aprimorar a qualidade de um conjunto de dados de grande volume já existente, com a remoção de ruído estrutural e de conteúdo potencialmente ofensivo. Na sequência, a Seção 6.3 apresenta a adaptação de um conjunto de dados completamente anotado para agentes de pergunta e resposta, incluindo sua tradução para o português brasileiro por meio de mecanismos automatizados. Por fim, a Seção 6.4 consolida as decisões metodológicas adotadas, discute suas limitações e contextualiza o impacto dessas escolhas nos experimentos apresentados nos capítulos seguintes.

## 6.1 Visão Geral

Para compor uma estratégia híbrida de geração de dados e ampliar a disponibilidade de dados em português brasileiro, em complemento à abordagem apresentada no capítulo anterior, a adaptação de conjuntos de dados existentes desempenha um papel fundamental. Embora a criação de novos conjuntos de dados permita maior controle sobre suas características, a reutilização de conjuntos consolidados possibilita incorporar diversidade linguística e volume adicional, além de aproveitar a robustez já validada desses recursos, especialmente no caso de dados anotados.

Apesar das limitações associadas à modificação de conjuntos de dados, como introdução de ruído ou perda de alinhamento estrutural, a adaptação deve preservar as características específicas dos dados originais. Em dados textuais, a semântica e o vocabulário característico de cada domínio devem ser preservados, minimizando alterações que afetem distribuição lexical e relações semânticas. Assim, a adaptação deve buscar reduzir ruídos e inconsistências sem comprometer a representatividade das amostras.

As adaptações realizadas são agrupadas em três categorias: (i) filtragem e normalização textual, que atuam sobre o conteúdo linguístico para remoção de ruído e de instâncias com baixa utilidade linguística ou informacional; (ii) transformação estrutural, modificando formatos, esquemas e metadados sem alterar o conteúdo semântico; e (iii) adaptação linguística, que envolve tradução e realinhamento semântico para adequação ao português brasileiro.

Foram selecionados dois conjuntos de dados consolidados na literatura, representativos de cenários complementares, para adaptação e uso em modelos futuros. A lista a seguir apresenta brevemente esses conjuntos e suas principais características.

- **BrWaC** (Brazilian Web as Corpus): corpus de grande escala coletado da web, amplamente utilizado no treinamento e avaliação de modelos de linguagem para o português brasileiro, caracterizado por alta diversidade temática e elevada incidência de ruído estrutural e conteúdo heterogêneo.

- **MultiWOZ** (**Multi**-Domain **W**izard-of-**Oz**): conjunto de dados de diálogos orientados a tarefas, anotado com estados de diálogo, atos de fala e marcações posicionais, cobrindo múltiplos domínios.

## 6.2 BrWaC

O *The Web-As-Corpus Kool Ynitiative* (WaCky) [Baroni et al., 2009] pavimentou o caminho para uma nova abordagem de construção de conjuntos de dados baseados na web. Nesse tipo de abordagem, o corpus é formado a partir da coleta de páginas de múltiplos sites. Em geral, isso resulta em grande volume e ampla diversidade temática, características potencialmente desejáveis para o treinamento de LLMs de propósito geral. Por outro lado, essa mesma estratégia também favorece a incorporação de ruídos e conteúdos inadequados.

A lista a seguir apresenta os principais fatores que comprometem a qualidade dos dados.

- **Dados inadequados:** instâncias incompatíveis com o escopo do corpus ou com o uso pretendido no treinamento, como páginas fora do domínio linguístico-alvo, fragmentos sem contexto, duplicações excessivas ou conteúdos que não deveriam compor a distribuição de treinamento.

- **Ruído:** elementos estruturais ou linguísticos que não correspondem a linguagem natural útil ao aprendizado estatístico, como artefatos de HTML, repetições automáticas de tokens, sequências numéricas extensas, erros de codificação, trechos incompletos e padrões gerados por scripts ou sistemas de rastreamento. O ruído tende a reduzir a densidade semântica do corpus e a introduzir distribuições artificiais de tokens.

- **Conteúdo impróprio:** textos que contêm linguagem ofensiva, discurso agressivo, material adulto ou associações potencialmente sensíveis envolvendo indivíduos, grupos ou entidades. Esse tipo de conteúdo pode induzir vieses indesejados em modelos treinados com o corpus, afetando diretamente o comportamento de geração e as associações probabilísticas aprendidas.

- **Dados de baixa qualidade:** instâncias em linguagem natural válida, mas com baixo valor informacional para modelagem, devido à reduzida diversidade lexical, baixa coerência textual, redundância interna ou predominância de conteúdo superficial.

Isso decorre da diversidade de sites incluídos na coleta, que abrange desde portais de notícias e páginas técnicas até blogs pessoais com conteúdos potencialmente adultos ou inadequados.

Com base na estratégia adotada pelo WaCky, foi proposto o *Brazilian Portuguese Web as Corpus* (BrWaC) [Wagner Filho et al., 2018], um conjunto de dados construído a partir de sites brasileiros para a geração de grandes volumes de texto. O BrWaC é uma das maiores bases de dados disponíveis em português brasileiro, constituindo uma ampla coletânea de páginas oriundas de diversos sites.

A Figura 6.1 ilustra o impacto desse tipo de dado ao apresentar a distribuição de probabilidades de um modelo do tipo *fill-mask* na tarefa de completar uma sentença com um termo mascarado. Nesse exemplo, a sentença utilizada foi “a psicopatia de [MASK] é

tão grande que", e o modelo empregado foi o "neuralmind/bert-large-portuguese-cased", também conhecido como "BERTimbau Large". O resultado evidencia como o modelo pode propor nomes de figuras públicas, religiosas, países e outras entidades sensíveis em um contexto negativo, sugerindo a possibilidade de que o conjunto de treinamento contenha coocorrências problemáticas e que a curadoria dos dados pode influenciar diretamente os padrões gerados. [1]

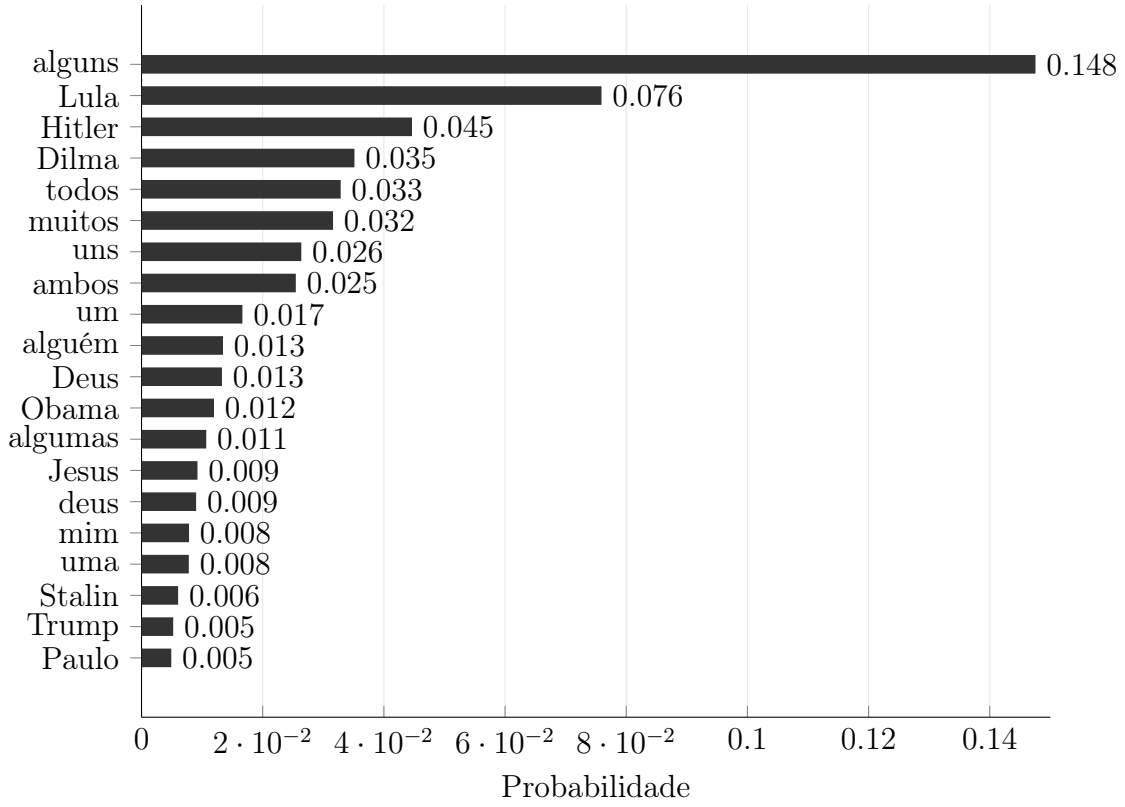



Figura 6.1: Exemplo de saída de *fill-mask* com top 20 probabilidades.

Esse comportamento sugere que padrões presentes na base de treinamento podem contribuir para associações inadequadas aprendidas pelo modelo. O BERTimbau Large foi treinado com o BrWaC, que, apesar de sua ampla variabilidade e dos ganhos observados em situações específicas, pode levar à geração de saídas inadequadas quando determinados conjuntos de termos são utilizados na entrada. Esse é um comportamento indesejável em diversas aplicações, especialmente quando o usuário final tem acesso direto às saídas produzidas pelo modelo.

Apesar dessas limitações, o BrWaC continua sendo uma das principais fontes de dados disponíveis, o que contribui para que sua popularidade e seu uso se mantenham relevantes ao longo dos anos. O gráfico 6.2 mostra as citações desse conjunto de dados nos últimos anos. Isso ocorre porque, apesar dessas limitações, o conjunto ainda reúne 3,5 milhões de páginas provenientes de mais de 120 mil sites distintos, totalizando 2,7 bilhões de tokens.

Para ilustrar como amostras inadequadas estão presentes no conjunto de dados, o verbatim a seguir apresenta um trecho do BrWaC alinhado pelo token "eh", que geralmente

[1] Este link abre a página do BERTimbau Large com a sentença de entrada usada no exemplo (URL: https://huggingface.co/neuralmind/bert-large-portuguese-cased).

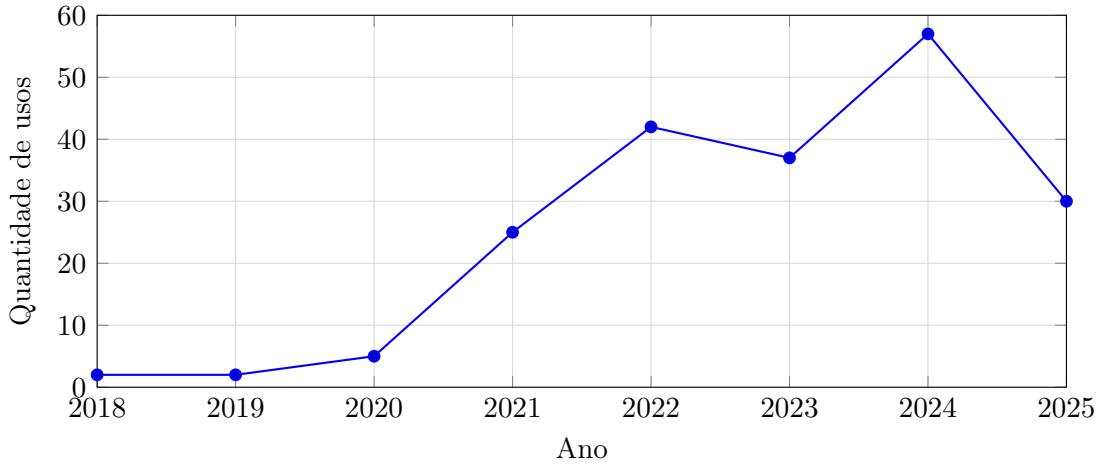



Figura 6.2: Número de citações do BrWaC por ano [Google Scholar, 2026].

corresponde ao termo "É". O exemplo não estima a prevalência desses padrões, mas ilustra tipos de ocorrência que motivam a etapa de filtragem. É possível observar que, embora haja diversos exemplos comuns e sem problemas aparentes, também há sentenças com elementos inadequados. Esse exemplo ilustra o tipo de ruído frequentemente encontrado em páginas web, como fragmentos de HTML, repetição de tokens, variações ortográficas e trechos de discussão política misturados a conteúdo irrelevante. Quando ingeridos em grande escala, esses padrões podem reforçar associações problemáticas no modelo e reduzir a utilidade do corpus para tarefas de geração e compreensão.

```
                                                |eh|
p> Como ri a juventude : eh eh eh eh . . . </p><p> Claro, a ação de Dilma
p><p> Como ri a juventude : eh eh eh eh . . . </p><p> Claro, a ação de Di
c. Daqui pra frente ate 3 / 10 eh tudo por conta e risco do Serra. O PIG
 </p><p> Como ri a juventude : eh eh eh eh . . . </p><p> Claro, a ação de
te sensíveis. </p><p> Por isso eh proibido o uso de qualquer substancia q
ividade das peças. Vanessa não eh apenas uma artesã, mas uma verdadeira a
Realmente a tal da politicagem eh um nojo! E infelizmente temos de ficar
o tem sequer vaga nocao do que eh o Brasil. </p><p> Comentou em 15/07/200
le e ele fracassou o day after eh um dia sem Serra e com Dilma. Ha que re
ra colo. A PSICOPATIA de LULLA eh tao grande que, nao bastasse REINVENTAR
opa de peixe ! E não só, eh eh eh ! </p><p> Entendo que nem todos gostem
a sopa de peixe ! E não só, eh eh eh ! </p><p> Entendo que nem todos gost
er a sopa de peixe ! E não só, eh eh eh ! </p><p> Entendo que nem todos g
y Rosana </p><p> Olá. Meu nome eh Rosana tenho 20 anos. Estou aqui para t
 a lente pink eh rosa mesmo ou eh meio vermelha ? " </p><p> Bomanah 04/03
 . </p><p> . . . Blz prof o sr eh deiz e naum tah prosa </p><p> mesmu diz
ii alguem sabe se a lente pink eh rosa mesmo ou eh meio vermelha ? " </p>
ele? Sei que o que está aí não eh são Nicolau, entendo isso, mas não deve
Matic e o Rodrigo . Capdevila, eh pá, já não arrisco. Aliás, se não jogar
u lembrei q quem guia a Igreja eh o Espírito Santo. Lembrei também, que e
m aparecer nem serem lembradas eh uma grande prova de amor ao proximo. Am
```

```
a já estava boa, mais de terça eh diante ando tendo enjoo depois que como
                                    |eh|
```

### 6.2.1 Metodologia

Com o objetivo de obter uma variação do corpus BrWaC com maior qualidade textual e menor incidência de ruído, foi adotada uma abordagem sistemática baseada na aplicação sequencial de filtros estruturais e linguísticos. O processo foi concebido para reduzir conteúdos inadequados ou pouco informativos, preservando a diversidade linguística necessária ao treinamento de LLMs e priorizando, nas etapas iniciais, procedimentos de baixo custo computacional. Todas as operações foram realizadas no nível de páginas web, consideradas a unidade básica de processamento do corpus.

A Figura 6.3 apresenta a exploração inicial dos dados, a definição dos critérios de exclusão e a aplicação sequencial dos filtros às páginas do corpus BrWaC.

Essa análise consistiu na inspeção amostral de URLs, domínios e conteúdos textuais, conduzida sobre uma amostra composta por ocorrências selecionadas com base na frequência de domínios e na recorrência de termos específicos, de modo a capturar padrões predominantes no conjunto de dados. Foram definidos indicadores observáveis, incluindo frequência relativa de tokens repetidos, proporção de caracteres não alfabéticos e presença de termos ofensivos previamente catalogados, que fundamentaram a definição dos critérios de filtragem.

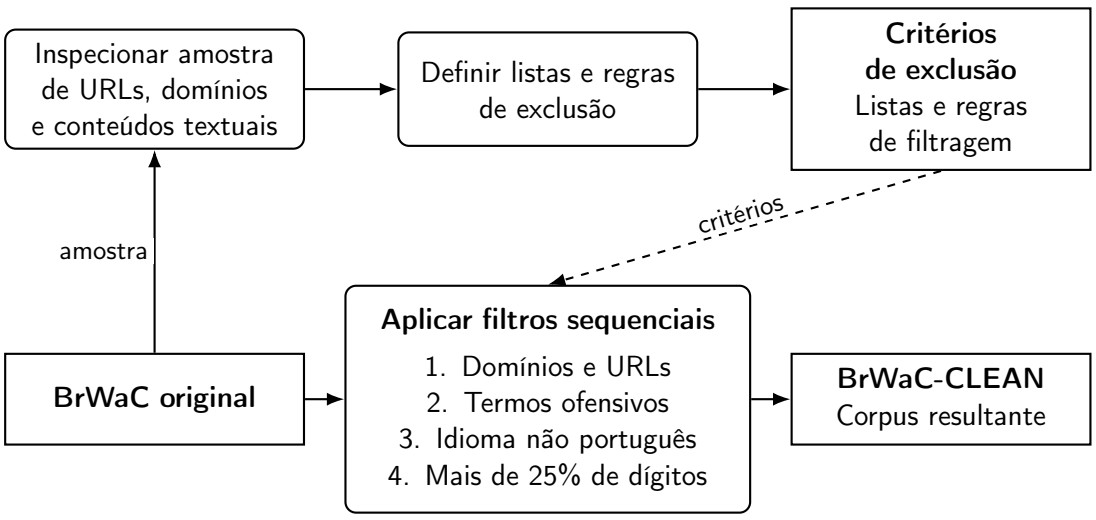



Figura 6.3: Fluxo de filtragem aplicado ao corpus BrWaC.

Como o BrWaC é composto por páginas provenientes da web, a análise exploratória se concentrou na identificação de padrões recorrentes em URLs e domínios associados a instâncias de baixa qualidade textual. Foi observada a concentração de conteúdos problemáticos em páginas oriundas de blogs pessoais, agregadores e portais com forte viés político, frequentemente caracterizados por elevada redundância e baixa densidade informacional. A exclusão desses domínios tem como objetivo reduzir a incidência de

polarização e redundância temática no corpus. Com base nessa análise, uma lista curada foi definida manualmente de padrões de domínios e substrings de URLs, empregada como critério inicial de exclusão.

No experimento descrito nesta tese, foram estabelecidos 27 padrões, definidos a partir da frequência de ocorrência na amostra exploratória e da recorrência em instâncias classificadas como de baixa qualidade textual, conforme critérios previamente definidos.

A filtragem estrutural baseada em domínios foi aplicada como primeiro estágio do fluxo, operando diretamente sobre as URLs das páginas antes da análise textual. Essa etapa funciona como um mecanismo de pré-filtragem de baixo custo computacional, reduzindo a incidência de páginas potencialmente problemáticas e o volume de dados submetido às etapas subsequentes. O fluxo foi estruturado de forma sequencial, sem reavaliação de instâncias removidas em etapas anteriores.

Após a filtragem por domínio, filtros foram aplicados diretamente sobre o conteúdo textual das páginas remanescentes. Foi desenvolvido um algoritmo determinístico de inspeção, baseado em varredura linear do texto, que utilizava correspondência lexical para identificar termos ofensivos e suas variações ortográficas previamente mapeadas em uma lista curada. Adicionalmente, a verificação automática de idioma foi realizada, com a remoção de páginas não classificadas como português, a fim de manter a consistência linguística do corpus resultante. Por fim, heurísticas voltadas à detecção de ruído estrutural foram aplicadas, incluindo o descarte de instâncias nas quais mais de 25% dos caracteres correspondiam a dígitos, critério definido empiricamente a partir da análise exploratória para identificar tabelas extensas e páginas contaminadas por artefatos comuns em conjuntos de dados da web.

A combinação sequencial dos filtros estruturais e linguísticos descritos constitui o procedimento completo de limpeza aplicado ao BrWaC nesta tese. O pipeline resultante prioriza decisões metodológicas simples, porém sistemáticas, e permite a obtenção de um subconjunto mais consistente do corpus original, adequado às etapas subsequentes de processamento e treinamento de modelos.

Embora o pipeline priorize a remoção de padrões recorrentes de ruído, sua aplicação pode introduzir falsos positivos, com a exclusão de conteúdo linguisticamente válido, e falsos negativos, com a manutenção de instâncias indesejadas. Apesar disso, do ponto de vista metodológico, a ocorrência de falsos positivos é considerada preferível à de falsos negativos, uma vez que a exclusão de uma parcela limitada de dados potencialmente úteis tende a ser menos prejudicial do que a manutenção de dados problemáticos capazes de contaminar o corpus e afetar o comportamento dos modelos treinados a partir dele.

### 6.2.2 Resultados

A aplicação do pipeline de filtragem descrito na seção de metodologia resultou em uma redução controlada do corpus BrWaC, preservando sua escala original e eliminando instâncias associadas a ruído estrutural e conteúdo potencialmente problemático. O corpus original continha 120 989 domínios distintos e 3 530 796 páginas. Após a aplicação sequencial das etapas de filtragem, o conjunto final passou a conter 96 814 domínios e 3 166 108 páginas, o que corresponde à remoção de mais de 24 mil domínios e aproximadamente

360 mil páginas, equivalentes a cerca de 20% dos domínios e 10% das páginas do conjunto original.

A redução ocorreu ao longo das etapas do pipeline, com maior impacto inicial da filtragem estrutural baseada em URLs. Como esperado, a remoção inicial se concentrou principalmente em páginas provenientes de blogs pessoais, agregadores e domínios previamente identificados como recorrentes em padrões específicos de conteúdo definidos na análise exploratória, o que contribuiu para reduzir a incidência de duplicação textual e de estruturas pouco informativas.

As etapas subsequentes de filtragem linguística contribuíram para refinar ainda mais o corpus. A verificação automática de idioma removeu páginas não classificadas como português, aumentando a consistência linguística do conjunto resultante. Em paralelo, o algoritmo de inspeção lexical descartou instâncias contendo termos ofensivos ou variações ortográficas associadas, reduzindo a probabilidade de propagação de vieses indesejados em modelos treinados com o corpus filtrado. Por fim, a heurística baseada na proporção de dígitos permitiu identificar páginas dominadas por listas automáticas, tabelas extensas ou artefatos de rastreamento, frequentemente observados em conjuntos de dados derivados da web.

Apesar da redução absoluta no número de instâncias, o corpus resultante mantém uma escala comparável ao conjunto original, característica fundamental para o treinamento de modelos de linguagem de grande porte. A diminuição relativa no volume total indica que o processo de filtragem atuou de forma seletiva, removendo principalmente páginas com maior probabilidade de introduzir ruído, sem reduzir substancialmente a escala do conjunto de dados.

Além das métricas quantitativas, a própria natureza dos filtros aplicados implica a remoção direta de instâncias contendo padrões ofensivos e páginas associadas a domínios previamente identificados como recorrentes em conteúdos polarizados. Assim, a redução desses tipos de ocorrência decorre diretamente das regras determinísticas definidas no pipeline de filtragem, e não de avaliação subjetiva posterior.

Os resultados indicam que a combinação de filtragem estrutural baseada em URLs com inspeção linguística determinística configura uma estratégia operacionalmente consistente para reduzir ocorrências associadas a ruído estrutural e conteúdo potencialmente inadequado.

## 6.3 MultiWOZ

O *MultiWOZ* é um conjunto de dados de diálogos multidomínios, anotado de forma estruturada para o treinamento de agentes conversacionais orientados a tarefas, nos quais o usuário persegue objetivos específicos ao longo da interação. As anotações incluem trechos das sentenças e atribuições semânticas, como intenções e entidades específicas. Devido à sua cobertura multidomínio, anotação rica em estados de diálogo e ampla adoção em benchmarks, esse conjunto é amplamente utilizado na literatura em trabalhos com LLMs.

O *MultiWOZ* reúne 10.438 diálogos e mais de 100.000 turnos, abrangendo diferentes tópicos, desde pedidos de informação até solicitações de reserva em hotéis. Parte dos diá-

logos envolve mais de um domínio ao longo da conversa. Esse conjunto foi construído por meio de *crowdsourcing*, com apoio do *Amazon Mechanical Turk*[2], no qual os participantes seguiam roteiros para criar uma conversa inicial e, em seguida, realizar sua anotação.

Neste trabalho, foi utilizada a versão 2.2 do *MultiWOZ*, que introduz correções nas anotações e inclui marcações de *spans* para valores de *slots* [Zang et al., 2020]. O esquema apresentado nos exemplos, com os campos `frames`, `service`, `start` e `exclusive_end`, corresponde a essa versão.

Apesar disso, o uso direto do *MultiWOZ* apresenta duas limitações principais: *(i)* o conjunto de dados é originalmente em inglês, o que limita seu uso em português brasileiro; e *(ii)* traduções simples tendem a introduzir desalinhamentos entre anotações e sentenças, comprometendo tarefas baseadas em spans e a consistência estrutural do conjunto. O *verbatim* a seguir apresenta um trecho do conjunto de dados original, em inglês. É possível observar que, além da *utterance* em inglês, o *frame* identifica diretamente no texto a posição inicial, a posição final e a anotação correspondente.


```
{
"speaker": "USER",
"turn_id": "6",
"utterance": "Then how about one in the cheap price range?"
},{
"speaker": "SYSTEM",
"turn_id": "7",
"utterance": "Cheap we can do! I recommend the Autumn House in the
east. Would you like to book a room there?",
"frames": [{
    "actions": [],
    "service": "hotel",
    "slots": [{
        "exclusive_end": 45,
        "slot": "hotel-name",
        "start": 33,
        "value": "Autumn House"
    }]
}]
},
```


A lista a seguir descreve cada campo presente nesse conjunto de dados.

- **speaker**: Indica quem produziu o turno do diálogo. Os valores possíveis incluem `USER` (usuário) e `SYSTEM` (atendente).
- **turn_id**: Identificador numérico do turno no diálogo. Define a ordem sequencial da conversa.

[2]https://www.mturk.com/

- **utterance**: Texto em linguagem natural correspondente à fala do participante naquele turno.
- **frames**: Lista de estruturas semânticas associadas ao turno. Cada *frame* representa um domínio ou serviço ativo na interação (por exemplo, hotel, restaurante ou trem).
- **service**: Domínio ao qual o *frame* pertence. Indica o tipo de serviço sendo tratado no turno.
- **actions**: Lista de atos de diálogo associados ao turno, representando intenções comunicativas como informar, solicitar ou recomendar.
- **slots**: Lista de pares atributo-valor mencionados explicitamente no texto da fala.
- **slot**: Nome do atributo no domínio (por exemplo, `hotel-name`, `price-range`).
- **value**: Valor associado ao atributo mencionado na fala.
- **start**: Índice do caractere inicial, no campo `utterance`, em que o valor aparece.
- **exclusive_end**: Índice final exclusivo do valor no campo `utterance`. Segue a convenção de intervalos semiabertos, como em `[start, exclusive_end)`.

Dessa maneira, foi desenvolvida uma estratégia que combina tradução automatizada com checagens e tradução com correção manual, a fim de adaptar esse conjunto de dados ao português brasileiro. As próximas seções detalham a metodologia empregada, bem como suas limitações e os resultados obtidos.

### 6.3.1 Metodologia

Para viabilizar o uso do *MultiWOZ* no treinamento de LLMs, foi adotada uma abordagem de tradução mais robusta, uma vez que uma simples conversão linguística não seria suficiente. Diferentemente de conjuntos de dados textuais não anotados, o *MultiWOZ* é um conjunto orientado a *spans*, no qual valores semânticos (*slot values*) estão associados a posições exatas nas sentenças por meio dos índices `start` e `exclusive_end`. Dessa forma, modificações textuais alteram o mapeamento entre o conteúdo linguístico e a estrutura anotada, podendo comprometer a validade do conjunto de dados em tarefas supervisionadas, como *slot filling* e predição de *spans*.

Outro problema recorrente é a diferença entre traduzir uma sentença inteira, como as sentenças presentes em *utterances*, e traduzir palavras ou trechos menores de maneira isolada. A sentença "I need a single room." pode ser traduzida como "Preciso de um quarto individual.", o que está correto. Porém, ao se traduzir isoladamente o *slot* "*single*", ele pode ser convertido para "solteiro". Por conta dessa diferença, o *slot* pode deixar de ser identificado na *utterance*, e os índices `start` e `exclusive_end` passam a ficar incorretos. Esse tipo de divergência reduz a qualidade do dado e exige tratamento individual para ser mitigado.

De maneira formal, a hipótese é que a tradução automática de conjuntos de dados anotados pode introduzir desalinhamentos estruturais não triviais, exigindo adequações, especialmente nos índices das anotações, para sua correção.

A Figura 6.4 apresenta o fluxo completo adotado para realizar a tradução e as correções necessárias para mitigar os problemas decorrentes do próprio processo tradutório. O fluxo evidencia duas dimensões distintas do problema: (i) a tradução linguística; e (ii) o realinhamento estrutural das anotações. É possível observar que o risco central do processo não reside apenas na qualidade da tradução automática, mas também na reestruturação sintática introduzida pelos sistemas de tradução, que frequentemente empregam sinônimos e reformulam as frases. Esse comportamento, embora linguisticamente aceitável, gera desalinhamento entre o *slot value* e o trecho literal correspondente na *utterance* traduzida.

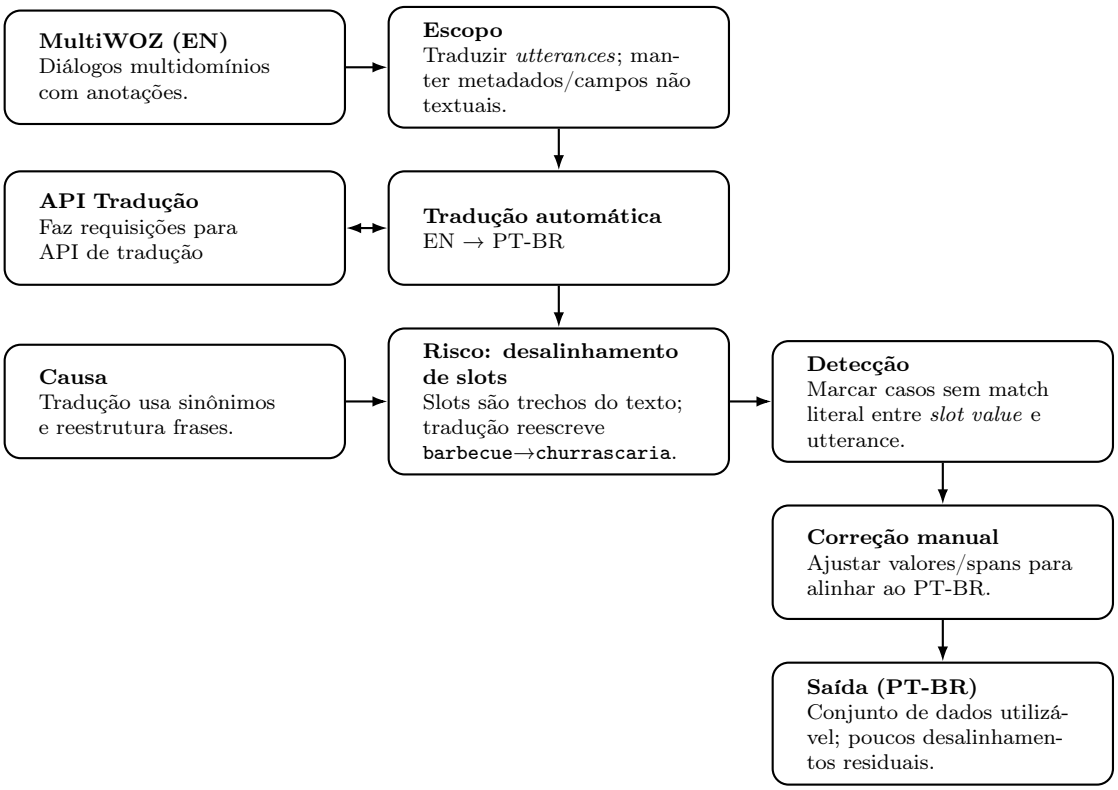



Figura 6.4: Processo de tradução do MultiWOZ e pós-processamento para realinhar *slot values* ao texto em PT-BR.

Inicialmente, foi realizada uma inspeção detalhada do formato JSON do conjunto de dados original. O verbatim apresentado anteriormente ilustra que cada turno contém campos estruturais fixos e que os *slots* mantêm dependência direta com o texto original, incluindo marcação posicional. Essa característica impõe uma restrição formal: a tradução altera o comprimento da sentença e, consequentemente, invalida os índices associados.

Com base nessa análise, o seguinte escopo foi definido:

- Traduzir exclusivamente os campos textuais: *utterances* e os valores textuais dos *slots*.
- Preservar integralmente os campos estruturais, os identificadores, os serviços e os metadados.

- Recalcular todos os índices posicionais após a tradução.

Essa decisão garante compatibilidade com ferramentas existentes e preserva a organização original do conjunto de dados.

A tradução foi realizada por meio de requisições automatizadas à API de tradução do Google[3], com processamento individual de cada *utterance*. A escolha pela tradução em nível de turno teve como objetivo preservar a estrutura original e evitar interferências contextuais externas ao turno específico. Após a tradução das *utterances*, os valores textuais dos *slots* foram traduzidos da mesma forma.

Após o processo de tradução, é possível observar que os valores de *slot* nem sempre são preservados por correspondência direta, mas sim por equivalência semântica. Por exemplo, a expressão "light meal" pode ser traduzida como "refeição leve". No entanto, caso o valor do *slot* seja apenas "light", sua tradução isolada pode resultar em "luz" ou "claro". Dessa forma, embora a *utterance* completa mantenha coerência semântica no idioma-alvo, o valor traduzido do *slot* pode não coincidir literalmente com o trecho presente na sentença traduzida. Nesses casos, a correspondência obtida por busca textual direta entre o valor do *slot* e a *utterance* se torna inconsistente.

O *verbatim* a seguir apresenta um exemplo desse tipo de problema. É possível observar que o *slot value* ‘‘`Casa de Outono`’’ difere da *utterance*, traduzida como ‘‘`Casa do Outono`’’. Embora a diferença semântica seja mínima, a variação sintática inviabiliza o alinhamento automático necessário para corrigir os índices `start` e `exclusive_end`.


```
{
"speaker": "USER",
"turn_id": "6",
"utterance": "Então que tal um na faixa de preço barato?"
},{
"speaker": "SYSTEM",
"turn_id": "7",
"utterance": "Podemos fazer barato! Eu recomendo a Casa do Outono no
leste. Você gostaria de reservar um quarto lá?",
"frames": [{
    "actions": [],
    "service": "hotel",
    "slots": [{
        "exclusive_end": 45,
        "slot": "hotel-name",
        "start": 33,
        "value": "Casa de Outono"
    }]
}]
},
```


[3]https://cloud.google.com/translate

O exemplo anterior à correção mostra os índices `start: 33` e `exclusive_end: 45` preservados do registro original em inglês. Eles não identificam o trecho "Casa do Outono" na *utterance* traduzida. Portanto, esse exemplo ilustra o estado anterior ao realinhamento, e não os índices válidos do arquivo final.

Para identificar inconsistências, foi implementado um algoritmo determinístico de busca de *substring* sobre a *utterance* traduzida. A verificação é realizada após normalização textual, que inclui conversão para minúsculas e normalização de espaços em branco. A busca é aplicada às versões normalizadas do valor do *slot* e da *utterance*. Para desambiguar ocorrências múltiplas, sejam $u_o$ e $u_t$ as *utterances* original e traduzida após a normalização, $L_o$ e $L_t$ seus comprimentos, e $s_o$ o índice inicial do *slot* na *utterance* original normalizada. A posição de referência na tradução é estimada por $\hat{s}_t = s_o(L_t/L_o)$. Se o valor traduzido ocorrer nas posições $P$, seleciona-se a ocorrência cuja posição $p \in P$ minimiza $|p - \hat{s}_t|$.

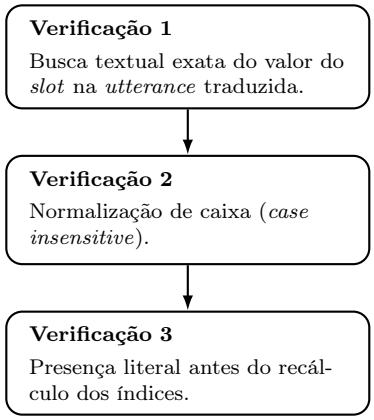



Figura 6.5: Verificação de inconsistências entre *slot values* e *utterances*.

Quando o valor não era encontrado como uma *substring* exata da sentença traduzida, o turno era marcado para revisão manual. Nessa etapa, o valor do *slot* era ajustado quando havia equivalência semântica clara e divergência apenas morfológica, e a *utterance* era reescrita quando o *slot* representava uma entidade nomeada com forma canônica conhecida. As ocorrências que ainda não podiam ser alinhadas foram contabilizadas como inconsistentes na validação final. Os registros disponíveis não especificam como essas anotações foram serializadas nos arquivos JSON finais, portanto não é possível afirmar se foram removidas ou se receberam índices nulos ou índices não atualizados.

Essa etapa é essencial para restabelecer a função de mapeamento entre texto e anotação. Diferentemente de conjuntos de dados não estruturados, a integridade do *MultiWOZ* depende da correspondência exata entre o valor semântico e sua posição no texto.

É importante destacar que o problema de desalinhamento é estruturalmente inevitável na tradução de conjuntos de dados anotados com *spans*. Sistemas de tradução otimizam a fluência e a naturalidade, mas não preservam a correspondência literal entre segmentos. Em algumas traduções entre idiomas específicos, determinados termos podem sequer possuir equivalentes diretos no idioma de destino, o que limita ainda mais essa abordagem e exige intervenções manuais adicionais.

Apesar das limitações existentes, a estratégia separa a tradução linguística da re-

construção estrutural das anotações e reduz parte dos desalinhamentos introduzidos pela tradução. A validação final, contudo, identificou ocorrências ainda inconsistentes, cuja representação nos arquivos JSON finais não está documentada. Essa limitação deve ser considerada ao interpretar a adequação do conjunto para treinamento e avaliação de modelos conversacionais orientados a tarefas.

No verbatim abaixo, é possível observar o mesmo exemplo apresentado anteriormente, agora com o *slot value* e os índices `start` e `exclusive_end` corrigidos após a tradução e o realinhamento. Nesse caso, o *slot value* foi atualizado, passando de ‘‘`Casa de Outono`’’ para ‘‘`Casa do Outono`’’.

```
{
"speaker": "USER",
"turn_id": "6",
"utterance": "Então que tal um na faixa de preço barato?"
},{
"speaker": "SYSTEM",
"turn_id": "7",
"utterance": "Podemos fazer barato! Eu recomendo a Casa do Outono no
leste. Você gostaria de reservar um quarto lá?",
"frames": [{
    "actions": [],
    "service": "hotel",
    "slots": [{
        "exclusive_end": 51,
        "slot": "hotel-name",
        "start": 37,
        "value": "Casa do Outono"
    }]
}]
},
```

### 6.3.2 Resultados

Nesta seção, são apresentados e analisados os resultados obtidos no processo de tradução do *MultiWOZ* para o português brasileiro, conforme a metodologia descrita na Seção 6.3.1. O objetivo é quantificar o impacto da tradução automática e avaliar o efeito da detecção e da correção manual sobre o alinhamento estrutural entre *utterances* e *slot values*.

Foram avaliados 115.434 turnos e 25.888 ocorrências anotadas de *slot values* no conjunto de dados traduzido. Considera-se alinhada a ocorrência em que o valor do *slot* é encontrado como *substring* literal da *utterance* traduzida após normalização de caixa e de espaços em branco, critério restritivo adotado para viabilizar o recálculo dos índices posicionais. É classificada como inconsistente a ocorrência em que o valor não é encontrado diretamente na sentença e, por isso, não é possível ajustar os índices para corresponder ao trecho textual.

Na etapa de tradução automática direta, aproximadamente 80% das ocorrências de slot values permaneceram alinhadas por correspondência literal na utterance traduzida. Isso indica que, em cerca de 20% dos casos, a tradução introduziu reformulações sintáticas ou substituições suficientes para invalidar a correspondência necessária em tarefas baseadas em *span prediction*.

Após a etapa de detecção e correção manual, 23.121 das 25.888 ocorrências de *slot values* foram classificadas como alinhadas, correspondendo a aproximadamente 89,3%. As 2.767 ocorrências restantes (10,7%) foram classificadas como inconsistentes na validação final.

Esses números evidenciam que a etapa de realinhamento é necessária para aumentar a proporção de ocorrências utilizáveis em tarefas supervisionadas dependentes de *spans*, como *slot filling*.

Os 2.767 casos restantes se concentram principalmente em três categorias:

1. substituição por expressões idiomáticas não diretamente mapeáveis,
2. substituição por sinônimos semanticamente semelhantes,
3. substituição de artigos definidos e de suas contrações com preposição (o, a, os, as, do, da, dos, das).

Embora a maioria das ocorrências tenha sido classificada como alinhada, a proporção, por si só, não permite concluir que o conjunto traduzido seja integralmente utilizável em tarefas baseadas em *spans*.

A Figura 6.6 apresenta a distribuição das ocorrências em que os *slot values* permaneceram desalinhados no conjunto traduzido. É possível observar uma distribuição de cauda longa, na qual poucos *slot values* concentram maior frequência de desalinhamento, enquanto muitos outros apresentam baixa frequência. Esse comportamento indica que a correção manual é especialmente relevante nos casos mais recorrentes.

Com base nessa concentração, a correção manual pôde priorizar os casos com maior frequência de inconsistências, maximizando o impacto da intervenção e deixando para o final os casos de menor impacto.

Os resultados quantitativos confirmam a hipótese apresentada na metodologia: a tradução automática de conjuntos de dados anotados pode introduzir desalinhamentos estruturais não triviais. Embora a tradução produza sentenças semanticamente adequadas, ela não preserva as posições das palavras em tarefas baseadas em índices. Por isso, o aumento de aproximadamente 9,3 pontos percentuais na taxa de alinhamento fornece evidência empírica de que a etapa de verificação e realinhamento é necessária para preservar a validade estrutural do conjunto em tarefas baseadas em índices.

## 6.4 Conclusão do Capítulo

Este capítulo investigou a adaptação de conjuntos de dados já existentes como estratégia complementar à geração de novos dados apresentada no capítulo anterior. Em vez de tratar conjuntos de dados e bases anotadas como insumos “prontos”, o capítulo partiu do

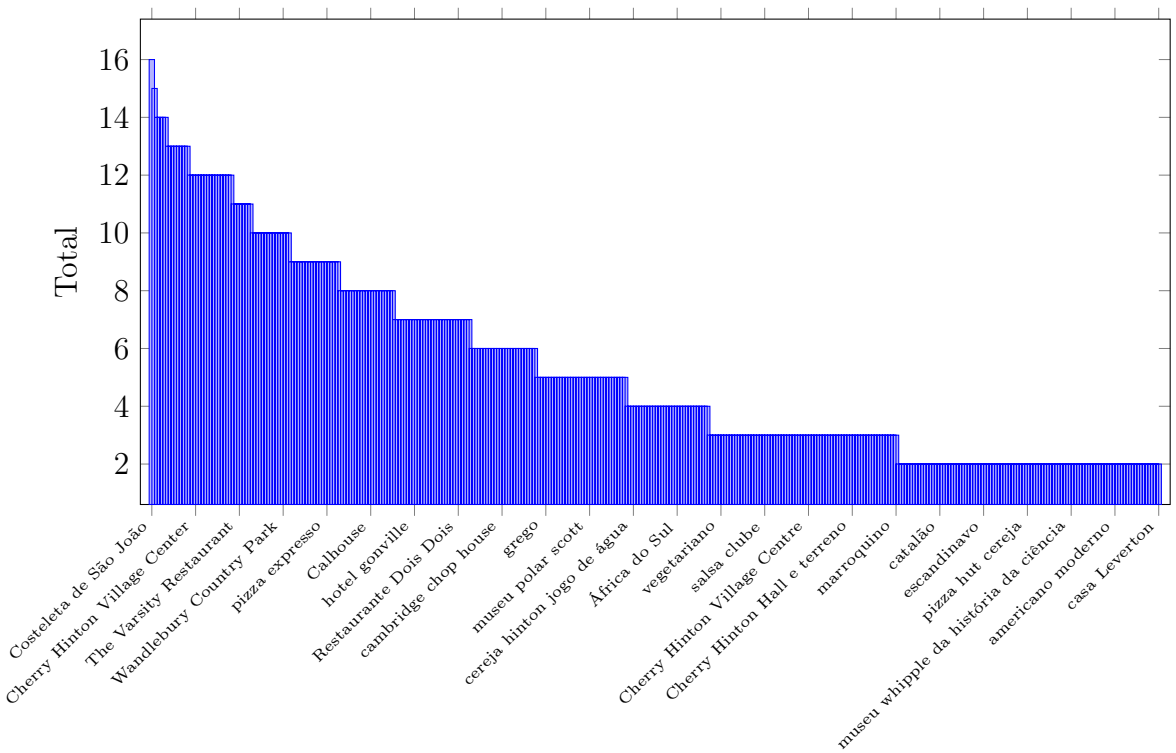



Figura 6.6: Amostras com desalinhamento após tradução, detecção e correção manual (os rótulos do eixo horizontal reproduzem literalmente os valores de slot gerados pela tradução automática).

princípio de que qualidade, segurança e adequação ao domínio e ao idioma são fatores determinantes para o desempenho e o comportamento de modelos modernos, em especial Large Language Models (LLMs). Assim, os procedimentos descritos neste capítulo buscam equilibrar escala e diversidade com a curadoria mínima necessária, reduzindo ruídos e riscos sem comprometer as propriedades essenciais dos dados.

No caso do BrWaC (Seção 6.2), foi mostrado que conjuntos de dados da web em grande escala carregam, inevitavelmente, conteúdo de baixa qualidade e ocorrências sensíveis, e foi proposta uma metodologia prática de limpeza baseada em (i) exploração de URLs, (ii) filtragem de domínios e padrões associados à baixa qualidade e (iii) inspeção textual para a remoção de termos ofensivos e de instâncias com ruído estrutural. O resultado preserva a escala do conjunto, mas remove uma fração expressiva de páginas potencialmente problemáticas, reduzindo a probabilidade de que o modelo aprenda associações espúrias a partir de exemplos degradados.

No caso do MultiWOZ (Seção 6.3), a adaptação se concentrou em preservar a estrutura e as anotações de alto valor do conjunto, ao mesmo tempo em que se viabilizou seu uso em português brasileiro. Como bases anotadas são particularmente custosas de produzir, a estratégia de tradução e alinhamento semântico permite reaproveitar conhecimento estruturado, como estados de diálogo, atos de diálogo e slots, em um novo idioma, com custo substancialmente menor do que o de uma coleta integral realizada do zero.

Em conjunto, essas adaptações fornecem dois insumos centrais para os experimentos dos próximos capítulos: (i) um corpus em português brasileiro com melhor qualidade para

pré-treinamento e continuação do treinamento; e (ii) um conjunto anotado para agentes de diálogo, que possibilita treinar e avaliar componentes orientados a tarefas. Além disso, os resultados reforçam uma conclusão metodológica: em cenários com restrições de tempo e recursos, ganhos relevantes podem ser obtidos não apenas pela criação de novos dados, mas também pela curadoria e transformação de dados existentes de maneira rastreável e consistente com o objetivo do modelo.

# Capítulo 7

# Metodologia de Coleta Web Controlada e sua Instanciação no Sistema WackyWacky

**Pergunta de Pesquisa 4:** Como coletar, filtrar, normalizar e estruturar sistematicamente dados textuais provenientes de websites em português brasileiro para gerar conjuntos de dados escaláveis e diversificados para o treinamento de Large Language Models?

Este capítulo apresenta uma metodologia sistemática para a coleta controlada de dados textuais da web, com escalabilidade, ponto de partida controlado e filtros de qualidade específicos. Diferentemente de abordagens tradicionais de crawling, frequentemente orientadas a maximizar a cobertura por meio de mecanismos de busca, a metodologia proposta estrutura a coleta como um processo formalizado, no qual critérios de expansão, validação, persistência e controle são definidos para otimizar a descoberta de novos sites.

A ferramenta BootCaT [Baroni and Bernardini, 2004] é uma das mais utilizadas para a geração de conjuntos de dados no estilo WaCky. Ela já foi empregada na criação de mais de 48 mil conjuntos de dados [Zanchetta, 2026]. A Figura 7.1 apresenta o fluxo padrão de coleta de dados para a criação de corpus utilizado pelo BootCaT. É possível observar que o mecanismo de exploração se baseia em dois elementos principais: o mecanismo de busca e a identificação de unigramas para novas consultas. Dessa maneira, essa ferramenta se apoia na indexação de buscadores tradicionais para explorar novos domínios.

De maneira geral, o BootCaT se apoia na expansão de unigramas e possui forte dependência de buscadores. Já na metodologia e na ferramenta propostas, a expansão ocorre de forma recursiva por meio do grafo de links das páginas encontradas e, após a inicialização, passa a ter independência dos sistemas usados para iniciar as buscas, devido à exploração de múltiplos sites.

Nesse modelo, a web é representada como um grafo de documentos conectados por referências, explorado recursivamente a partir de um conjunto inicial de consultas. A exploração é guiada por restrições operacionais, como limites de recursão, políticas de filtragem e deduplicação por *hash* de conteúdo e URL, para reduzir redundâncias e construir um conjunto de dados de alta qualidade.

A persistência armazena a fila de processamento e o estado global da execução, permitindo retomá-la após uma interrupção. Já o controle de concorrência é definido pelas

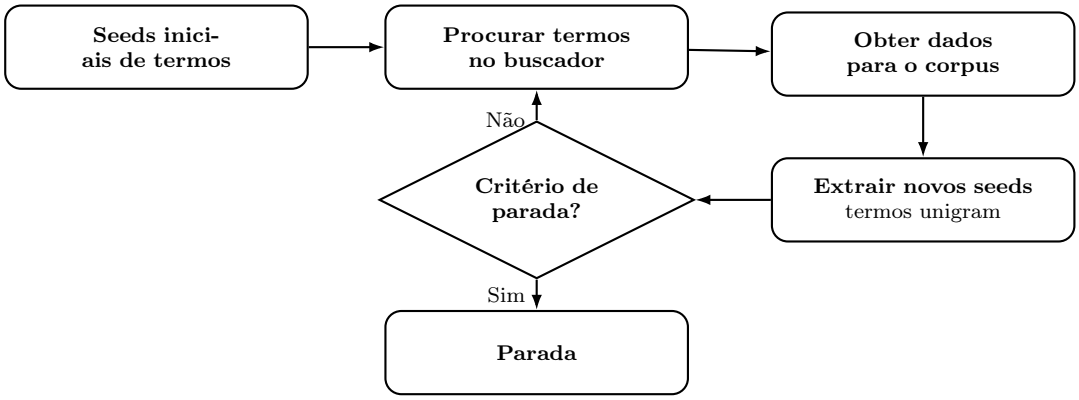



Figura 7.1: Fluxo da ferramenta BootCaT para criação iterativa de conjuntos de dados WaCky.

regras aplicadas à seleção e à atualização dos registros dessa fila: não é a consistência dos dados, por si só, que coordena os processos. Com essa arquitetura, o próprio mecanismo de persistência pode servir como fila, sem depender de um sistema externo para essa função.

Para validar e operacionalizar a metodologia proposta, este capítulo também descreve sua instanciação no sistema *WackyWacky*, um crawler web distribuído, configurável quanto ao idioma, com suporte a filtragem baseada em linguagem e projetado para exploração recursiva em larga escala, condicionada por restrições operacionais explícitas. O sistema incorpora mecanismos de controle de carga baseados em limitação por domínio e controle do número de threads de execução, deduplicação de conteúdo e instrumentação completa por meio de rastreamento, métricas e logs, permitindo a observação detalhada do processo de coleta.

Por fim, é apresentada a execução da metodologia no contexto experimental desta tese, bem como uma caracterização inicial do conjunto de dados gerado.

Neste capítulo, são discutidas soluções para os seguintes problemas: (*i*) Exploração web com controle operacional: como estruturar uma coleta recursiva em larga escala, com limites explícitos de expansão, domínio, linguagem e execução; (*ii*) Qualidade e consistência dos dados coletados: como aplicar filtragem, validação e deduplicação para reduzir ruído e redundância no conjunto final, ao mesmo tempo em que se aumenta a qualidade geral; (*iii*) Obtenção distribuída de dados: como operacionalizar a coleta com múltiplas unidades de processamento, ou *workers*, preservando coordenação, rastreabilidade e consistência durante a execução.

O restante deste capítulo está organizado da seguinte forma: a Seção 7.1 apresenta os componentes centrais da proposta e o fluxo de alto nível da coleta. A Seção 7.2 detalha a metodologia de coleta web controlada e seus mecanismos de execução. A Seção 7.3 descreve a implementação no sistema *WackyWacky*. Em seguida, a Seção 7.4 apresenta o conjunto de dados gerado e suas características. A Seção 7.5 discute os principais pontos de atenção da abordagem. Por fim, a Seção 7.6 sintetiza os resultados e contribuições.

## 7.1 Visão Geral

A crescente facilidade de publicação de conteúdo online resultou em um aumento significativo na disponibilidade de dados públicos, tornando a web uma das principais fontes para construção de conjuntos de dados em larga escala. Apesar disso, a obtenção desses dados de maneira organizada ainda é um desafio.

Realizar o processo de coleta de dados em larga escala nesse ambiente envolve desafios associados diretamente à heterogeneidade dos conteúdos, à natureza dinâmica das páginas e à necessidade de controle explícito sobre o processo de exploração. Nesse contexto, este capítulo apresenta a metodologia de coleta controlada adotada neste trabalho e sua instanciação no sistema WackyWacky, concebido como um crawler distribuído focado em robustez, controle operacional e geração de conjuntos de dados de alta qualidade.

Dessa maneira, esta proposta assume o funcionamento e a organização desse espaço de páginas como um grafo direcionado, isto é, uma estrutura composta por páginas, representadas como nós, e por referências entre elas, representadas como arestas. A Figura 7.2 apresenta esse tipo de representação.

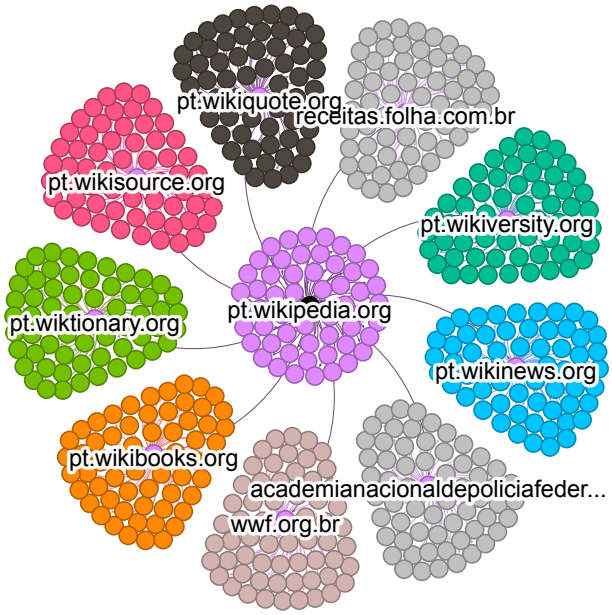



Figura 7.2: Exemplo da web representada como grafo.

Para que seja possível realizar a obtenção dos dados, a metodologia propõe a exploração e a coleta recursiva a partir de um conjunto inicial de consultas, permitindo a expansão progressiva do espaço de documentos. Diferentemente de abordagens baseadas em buscadores, nas quais o principal objetivo é priorizar a maximização da cobertura do espaço de páginas acessíveis, a exploração e a coleta propostas priorizam uma exploração condicionada por restrições operacionais para melhorar a qualidade e a aderência do conjunto de dados gerado ao domínio escolhido, reduzindo ruídos e conteúdos de baixa qualidade.

Adicionalmente, a metodologia define mecanismos de controle de acesso por domínio por meio de políticas que limitam o número de requisições por intervalo de tempo e a profundidade de exploração, garantindo que a exploração respeite intervalos mínimos

entre acessos consecutivos a um mesmo domínio. Esse tipo de abordagem contribui para a estabilidade do processo de coleta e para a redução da sobrecarga em servidores externos.

Como esta proposta foi definida para a geração de conjuntos de dados textuais, os processos e as técnicas escolhidos têm como objetivo final preparar e manter apenas dados textuais dos sites explorados. Recursos multimídia e elementos não textuais são descartados durante o processamento, priorizando a extração e persistência de conteúdo textual.

Como validação da metodologia proposta, foi implementado um sistema que materializa os mecanismos definidos, permitindo sua execução sob diferentes configurações operacionais. Experimentos com limites controlados de execução foram conduzidos para avaliar a aderência aos critérios estabelecidos, resultando na construção de um conjunto de dados textual em língua portuguesa com ampla diversidade de fontes.

## 7.2 Metodologia de Coleta Web Controlada

A metodologia proposta estrutura o processo de exploração e coleta na web como uma sequência de estágios, cada um responsável por um aspecto específico da exploração, do controle e da organização dos dados.

A arquitetura do sistema é organizada em torno de uma execução distribuída, composta por múltiplos trabalhadores (unidades de processamento) e coordenada por um mecanismo de liderança global. Essa organização permite processamento concorrente, ao mesmo tempo em que preserva a consistência por meio de sincronização baseada em banco de dados. O estado global do sistema e a fila de páginas a processar são mantidos de forma implícita pelo mecanismo de persistência, o que integra controle de execução e armazenamento operacional em uma mesma infraestrutura. A Figura 7.3 apresenta uma visão geral da arquitetura e do fluxo de coleta do sistema.

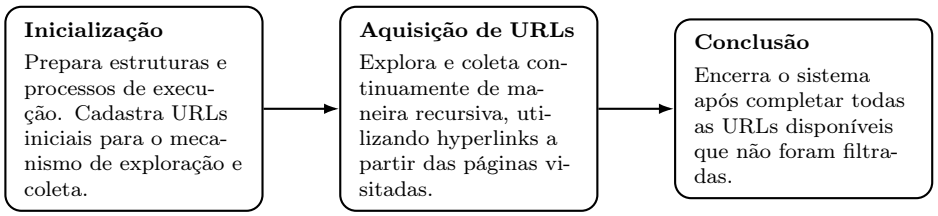



Figura 7.3: Fluxo simplificado da metodologia de exploração e coleta recursiva proposta.

De maneira geral, a metodologia foi desenvolvida para funcionar em três estágios, indo da inicialização à conclusão do processo de obtenção de URLs e resultando no armazenamento do conteúdo textual de diversas páginas web.

Na etapa de inicialização, caso o sistema esteja sendo executado pela primeira vez, é necessário definir o conjunto de URLs que dará origem ao processo de exploração e coleta. Essas URLs iniciais são geradas a partir de consultas em diferentes fontes e devem ser definidas de modo a maximizar a diversidade temática, resultando em conjuntos de dados mais diversos e completos. A diversificação das URLs iniciais auxilia na redução de vieses na coleta e aumenta a capacidade do sistema de explorar múltiplas regiões do grafo da web. Uma vez que as URLs estejam definidas, ou que o sistema esteja reiniciando o

processo de busca de dados, é possível avançar diretamente para o estágio de exploração recursiva de URLs.

O próximo estágio, referente à aquisição recursiva, constitui o núcleo da metodologia, sendo responsável pela exploração incremental do grafo de URLs. A metodologia proposta já define uma organização para permitir a escalabilidade nesse processo, uma vez que ele é o mais custoso e demorado. Cada unidade que participa deste estágio tem autonomia para requisitar, explorar, coletar e filtrar URLs, sem dependência de sincronização direta com outras unidades. A cada página explorada e coletada, a unidade que fez a requisição aplica localmente os mecanismos de filtragem e controle para decidir se os hyperlinks encontrados devem ser adicionados à lista de páginas a explorar, com diferentes estados definidos pelos filtros aplicados. Essa etapa se caracteriza como uma travessia controlada de grafo, com controles sobre concorrência e consistência.

O último estágio se caracteriza pela finalização das unidades de trabalho e pelo encerramento do processo de exploração e coleta de dados. Uma vez que não existam mais URLs disponíveis para exploração, as unidades que participam do processo podem encerrar as atividades em execução.

Durante a aquisição, a metodologia introduz um conjunto de mecanismos de controle que atuam diretamente sobre o processo de navegação e coleta. Esses mecanismos têm como objetivo limitar o espaço de busca, garantir estabilidade operacional e assegurar a qualidade dos dados obtidos. De forma geral, esses mecanismos incluem:

- **Limitação de recursão**: define limites para a profundidade de exploração tanto em nível de página quanto de domínio, evitando crescimento descontrolado do grafo explorado.
- **Controle de carga por domínio (*politeness*)**: estabelece intervalos mínimos entre requisições a um mesmo domínio, prevenindo sobrecarga de servidores externos.
- **Controle de tentativas (*retry*)**: define limites e condições para reprocessamento de páginas em caso de falha, garantindo tolerância a erros sem comprometer a convergência do processo.
- **Filtragem estrutural**: permite excluir domínios previamente definidos ou regiões do grafo previamente definidas como indesejadas.
- **Filtragem semântica**: restringe os dados coletados com base em características do conteúdo, como idioma, reduzindo ruído no conjunto final.
- **Controle de redundância**: identifica e trata conteúdos duplicados, evitando múltiplas representações da mesma informação.

Esses mecanismos são operacionalizados sobre uma estrutura de persistência que atua simultaneamente como fila de processamento e repositório de dados, permitindo coordenação distribuída e consistência global.

Durante o processo de aquisição, todos esses mecanismos operam de maneira coordenada, determinando o tratamento a ser aplicado a cada nova URL descoberta e com os dados obtidos a cada aquisição.

Além desses mecanismos, esta metodologia também define diretrizes para os dados obtidos. Essas definições suportam duas operações críticas ao longo do processo de aquisição. A primeira operação é dar suporte aos mecanismos de checagem de redundância, reduzindo a inserção de dados duplicados ao longo do processo. A segunda operação é permitir a reconstrução do grafo de navegação explorado no estágio de aquisição. Esse é um passo fundamental para análises que podem ser realizadas sobre o conjunto de dados gerado.

Em conjunto, esses elementos estabelecem uma abordagem de coleta que combina exploração recursiva com restrições explícitas, permitindo equilibrar cobertura, qualidade e controle operacional.

### 7.2.1 Fluxo da Metodologia

O fluxo da metodologia foi organizado em três estágios, constituídos por etapas menores, que podem conter etapas ainda mais específicas. Para apoiar esse fluxo, foram definidos estados do sistema para auxiliar nos mecanismos de sincronização, uma vez que a execução assíncrona e paralela pode ser realizada para aumentar a velocidade de aquisição de dados das URLs. A Figura 7.4 apresenta os estados existentes e as transições entre eles.

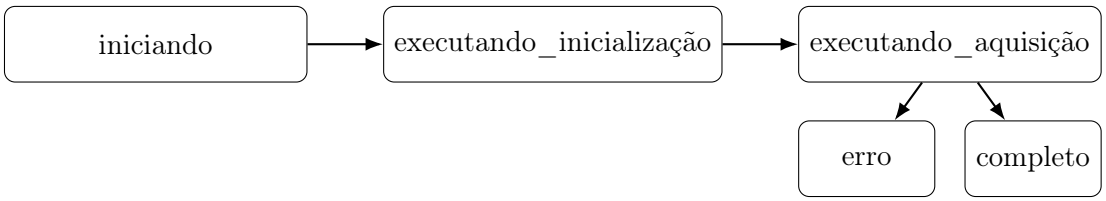



Figura 7.4: Diagrama de estados possíveis do sistema, desde a inicialização até a conclusão com sucesso ou erro.

É possível observar que, quando o sistema é iniciado, o estado inicial é "iniciando". Esse estado indica que o sistema ainda não foi executado nenhuma vez, mas está pronto para iniciar. Com a ativação das unidades de processamento, o dispositivo eleito como líder fica responsável por trocar o estado do sistema para "`executando_inicialização`" e iniciar o processo de geração de URLs *seed*. Assim que esse processo é concluído, o nó líder troca o estado do sistema para "`executando_aquisição`".

No estado "`executando_aquisição`", todas as unidades de processamento ativas podem participar da exploração de novas URLs e da obtenção de dados das páginas encontradas. Caso algum erro aconteça, o estado do sistema é trocado para "erro" e as unidades de processamento são encerradas para aguardar a solução. Se não existirem mais URLs que atendam ao critério de distribuição de URLs disponíveis, o estado do sistema é trocado para "completo", e as unidades de processamento também se encerram conforme checam o estado do sistema.

Os estados do sistema são alterados conforme os estágios são executados. A Figura 7.5 sintetiza os três estágios e os principais módulos dos mesmos: inicialização, aquisição e conclusão.

O estágio de inicialização corresponde ao momento em que a aplicação verifica o estado corrente do sistema e define a ação que a instância em execução deve tomar. A princípio,

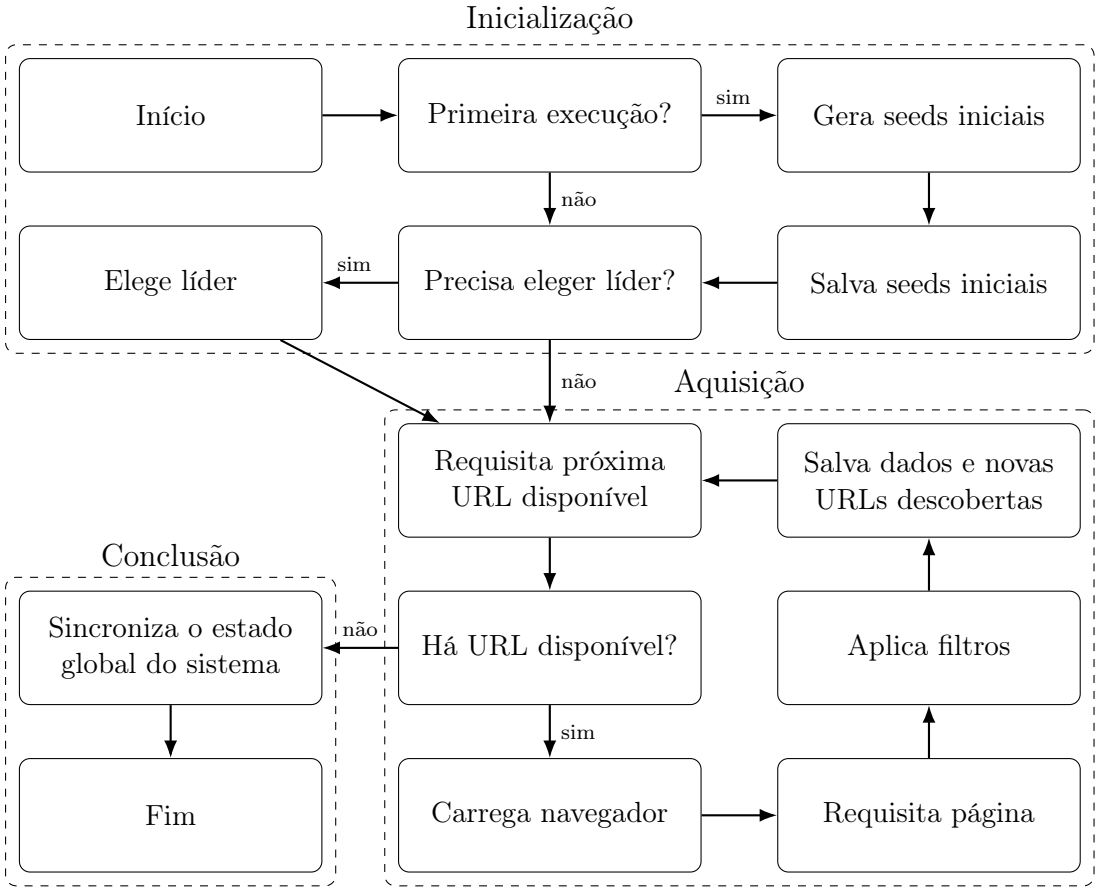



Figura 7.5: Estágios de inicialização, aquisição e conclusão, com seus respectivos módulos.

existem duas ações possíveis. Caso seja a primeira execução, a instância atual fica responsável por gerar as *seeds* iniciais de URLs. Caso não seja a primeira execução, ou as URLs já tenham sido geradas, é possível iniciar o processo de eleição de líder. Esta metodologia não define os mecanismos nem o funcionamento da liderança. Ela apenas define o ponto ideal para que isso seja feito, caso seja necessário na implementação. A utilização de um líder pode simplificar etapas que alteram o estado do sistema ou reportam estatísticas específicas.

A geração das *seeds* iniciais de URLs, apesar de ter um processo específico, é salva no mesmo local em que as unidades de processamento buscarão as próximas URLs, padronizando o processo de aquisição. Uma vez concluída a etapa de inicialização, com as URLs iniciais geradas ou ignoradas por se tratar da continuação de uma execução já existente, é possível iniciar o estágio de aquisição, alterando o estado do sistema para "executando_aquisição".

O estágio de aquisição é o núcleo do processo metodológico. Neste estágio, cada unidade de processamento executa o ciclo composto pelas etapas da lista abaixo:

1. Selecionar próxima URL disponível: identifica, seleciona e marca a URL obtida como em processamento, evitando que outras unidades de processamento selecionem a mesma URL.

2. Incrementar contador de tentativas: o incremento do contador de tentativas evita

que páginas com falhas persistentes bloqueiem a continuidade do sistema.

3. Carregar página: faz uma requisição HTTP para a página, carregando os textos disponíveis.

4. Aplicar restrições de processamento: checa se a página e os dados obtidos atendem aos requisitos definidos para a geração, como idioma e tamanho mínimo.

5. Persistir dados relevantes: salva os dados relevantes obtidos.

6. Aplicar restrições de expansão recursiva: checa se a página obtida atende aos critérios para continuar a expansão recursiva.

7. Expandir recursivamente a fronteira de busca: quando a página atender às restrições de expansão, adiciona as URLs descobertas à lista de próximas URLs.

Esse ciclo é executado de forma concorrente por múltiplas unidades de processamento, sendo coordenado por uma estrutura de persistência que garante consistência global do estado do sistema.

Embora essa metodologia simplifique o processo de obtenção de dados, a utilização de um navegador com suporte à renderização dinâmica (HTML, CSS e JavaScript) é necessária para garantir a obtenção correta do conteúdo em páginas modernas. Atualmente, muitos sites, quando carregados sem um navegador capaz de renderizar HTML com CSS e executar códigos em JavaScript, podem não exibir o conteúdo real da página. Isso adiciona uma complexidade adicional ao processo, o que torna a obtenção mais lenta e aumenta consideravelmente o consumo de recursos computacionais.

Uma vez que a página atual já tenha seus dados obtidos e que os filtros de conteúdo e expansão tenham sido aplicados, os dados obtidos são persistidos e se inicia o processo de adição das próximas URLs descobertas à lista de URLs em aberto. Para isso, os filtros e requisitos devem ser utilizados para determinar a ação correta. Páginas marcadas como inadequadas não devem salvar URLs exploráveis. Por exemplo, uma URL que gerou dados em um idioma não desejado deve ter todas as URLs possíveis a partir dela também marcadas como inadequadas por idioma.

Uma vez que todas as URLs que atendam aos critérios estabelecidos já tenham sido obtidas e processadas, é possível iniciar o estágio de conclusão, encerrando o sistema por completo. Nesse estágio, as unidades de processamento ativas são encerradas conforme identificam que não existem mais URLs para serem processadas. Se alguma unidade de processamento for definida como líder, ela pode alterar o estado do sistema para "completo" e encerrar.

Em termos metodológicos, o fluxo implementado estabelece três propriedades centrais. Primeiro, a coordenação é distribuída, porém ancorada em uma única fonte de verdade persistente. Segundo, há aquisição concorrente, com preservação sincronizada dos estados e dos dados de cada URL. Terceiro, há expansão recursiva do espaço de busca, restringida por políticas explícitas de profundidade, domínio, idioma e repetição. Essas propriedades fazem com que o processo de crawling não seja apenas iterativo, mas também controlado, rastreável e recuperável frente a falhas.

### 7.2.2 Inicialização

O estágio de inicialização define o ponto de entrada do processo de coleta. Esse estágio é responsável por definir as *seeds* iniciais de URLs a serem exploradas, cujas definições funcionam como guia para toda a etapa de exploração subsequente.

Para gerar as URLs iniciais, é necessário definir um conjunto finito de consultas predefinidas que representem os tópicos ou domínios de interesse do processo de coleta. Essas consultas podem assumir a forma de termos ou sentenças, constituindo o principal mecanismo de direcionamento semântico da coleta e delimitando o espaço de documentos a ser explorado. O trecho em *Verbatim* a seguir apresenta um exemplo ilustrativo no domínio da medicina:

```
["fisiologia humana", "biologia celular", "anatomia sistêmica",
 "processos metabólicos", ..., "saúde"]
```

Cada termo é transformado em um conjunto de URLs candidatas a exploração por meio de fontes externas, como mecanismos de busca ou plataformas de conteúdo estruturado. A utilização dessas fontes permite não apenas obter documentos diretamente relacionados às consultas, mas também estabelecer conexões com outros domínios da web por meio dos links presentes nesses documentos. Por exemplo, para o termo *"fisiologia humana"*, são produzidas URLs como:

```
http://pt.wikipedia.org/w/index.php?fulltext=1&search=fisiologia+humana
https://pubmed.ncbi.nlm.nih.gov/?term=fisiologia+humana
https://arxiv.org/search/?query=fisiologia+humana&searchtype=all
https://scholar.google.com/scholar?q=fisiologia+humana
https://www.google.com/search?q=fisiologia+humana
https://duckduckgo.com/?q=fisiologia+humana
https://www.bing.com/search?q=fisiologia+humana
```

As URLs geradas são inseridas na mesma estrutura de persistência utilizada no estágio de aquisição, permitindo que o processo de exploração funcione de maneira uniforme desde a inicialização, sem fluxos especiais ou distintos uma vez que a aquisição inicie.

A definição de quais sistemas web e ferramentas devem ser integrados não é feita pela metodologia. Essa escolha permite maior versatilidade e melhor adequação a domínios específicos, com buscadores também específicos.

Esse processo é aplicado a todo o conjunto de consultas, permitindo a construção de uma base inicial de URLs com diversidade temática controlada. É possível utilizar termos de domínios mais específicos ou criar conjuntos de termos mais dinâmicos e abrangentes.

A Figura 7.6 ilustra o processo de inicialização da coleta. Inicialmente, um conjunto de consultas é definido, representando o escopo semântico desejado. Essas consultas são então submetidas a diferentes fontes externas, que retornam conjuntos de URLs associadas. O resultado dessa etapa é a formação do conjunto de URLs iniciais, que alimenta o processo de exploração recursiva.

Por fim, a etapa de inicialização estabelece as condições iniciais do processo, garantindo que a exploração subsequente ocorra de maneira consistente e controlada. A partir

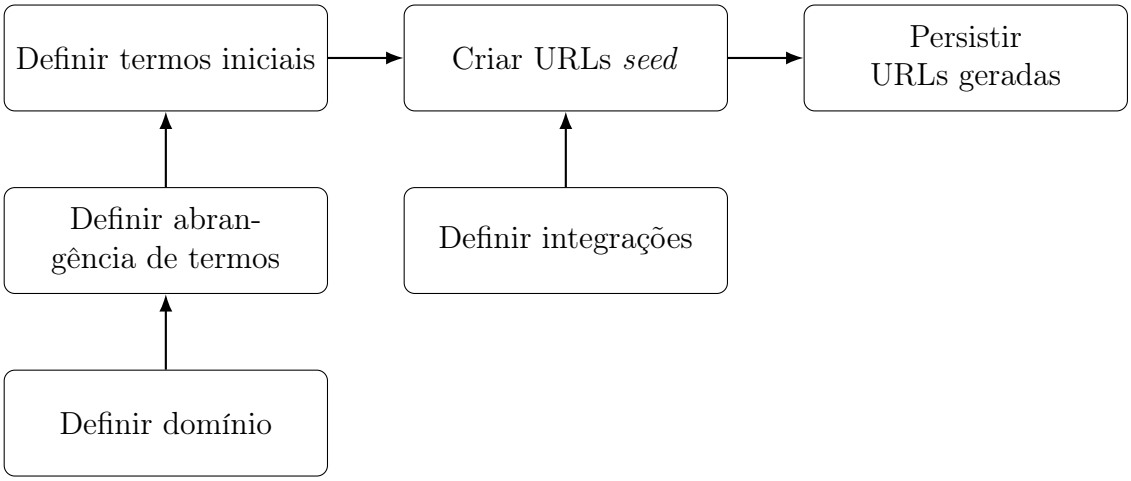



Figura 7.6: Processo de inicialização da coleta a partir de consultas semânticas.

desse ponto, o processo evolui de forma iterativa e recursiva, com novas URLs sendo incorporadas ao conjunto de dados por meio da expansão recursiva do grafo de documentos.

### 7.2.3 Exploração e Coleta de Páginas Web - Aquisição

O estágio de aquisição é a etapa principal da metodologia, responsável por explorar novas páginas web, coletar e persistir seus conteúdos. No contexto desta metodologia, o espaço de documentos é explorado de maneira controlada e recursiva, em que novas páginas são continuamente descobertas a partir dos enlaces presentes no conteúdo coletado a cada momento. A utilização desse mecanismo permite a expansão progressiva do conjunto de dados, ao mesmo tempo que possibilita a aplicação de restrições e filtros durante todo o processo de exploração.

Esse estágio se inicia a partir de um conjunto de URLs pendentes de processamento, persistidas em uma estrutura que atua como fila de execução. A seleção de URLs é realizada de forma a garantir exclusão mútua entre unidades de processamento, evitando que múltiplos trabalhadores processem a mesma URL simultaneamente.

Assim que uma URL é selecionada para processamento, a primeira ação consiste na verificação das restrições associadas à URL. Além das verificações de limite de recursão e domínio, é necessário avaliar condições que dependem do estado global, o qual pode ser alterado pela execução concorrente de diversas unidades de processamento. Pela possibilidade de execução paralela, surge a necessidade de verificar essas condições durante a execução, e não apenas na distribuição das próximas URLs disponíveis para processamento. Caso alguma restrição falhe, a URL é descartada e uma nova URL pode ser requisitada.

Se todas as restrições forem atendidas, a URL pode ser carregada com uma requisição HTTP do tipo *GET*. Como o objetivo desta metodologia é apenas construir um conjunto massivo de dados de páginas públicas, a requisição *GET* para páginas com conteúdo aberto na internet já é suficiente, descartando a necessidade de interação com formulários e outros elementos.

Para cada URL carregada, seu conteúdo deve ser analisado, extraído e avaliado para

definir se a página atende aos requisitos desejados e se as URLs descobertas a partir dela devem ser exploradas. O conteúdo obtido é sempre persistido, sendo o estado da URL utilizado para indicar a validade dos dados. A partir do documento obtido, são identificados novos enlaces (*hrefs*), que constituem candidatos à expansão do espaço de documentos e URLs. Essas URLs candidatas podem receber diferentes estados para evitar processamento desnecessário, de acordo com os dados descobertos. A lista abaixo define os dois comportamentos possíveis e as ações para os dados obtidos:

1. Tratamento dos dados textuais: se os dados atendem aos requisitos, a URL é marcada como concluída. Se não atendem, a URL é marcada como bloqueada, com a descrição do requisito que falhou.

2. Tratamento das URLs descobertas: se a URL obtida retornou dados que atendem aos requisitos, as checagens de restrições, como limite de recursividade, domínio e outras, são verificadas para definir o estado em que as URLs descobertas serão salvas, permitindo a continuidade recursiva apenas para URLs candidatas que possivelmente atendam aos requisitos. Se não atenderem aos requisitos, seja de qualidade dos dados, seja de restrições de URL ou domínio, a URL é salva com estado de falha e com a descrição do requisito que falhou.

De qualquer maneira, o conteúdo da URL atual é sempre persistido, alterando apenas o estado para indicar que se trata de uma URL com dados válidos ou inválidos. A lista abaixo apresenta exemplos dos possíveis tratamentos aplicados às URLs descobertas de uma página em diferentes situações:

- A URL retornou dados que atendem aos requisitos:
  - As restrições de domínio foram atendidas:
    As URLs descobertas são salvas para serem exploradas de maneira recursiva.
  - A recursão das páginas descobertas é maior que o limite configurado:
    As URLs descobertas são salvas com estado de bloqueio por limite de recursão atingido.
- A URL retornou dados em outro idioma:

  Todas as URLs descobertas são bloqueadas por possivelmente também estarem fora do idioma desejado.

A Figura 7.7 apresenta um fluxograma do funcionamento deste estágio, com baixo detalhamento nas checagens e restrições que serão aprofundadas em seções futuras.

Como se observa no fluxograma, o estágio de aquisição opera em um ciclo contínuo, alternando entre a requisição de uma nova URL elegível e a execução das etapas de validação, carregamento, persistência e expansão. Esse ciclo permanece ativo enquanto houver URLs pendentes passíveis de processamento.

As checagens e as restrições constituem a parte central desse funcionamento, pois determinam, simultaneamente, o destino da URL atual e a elegibilidade das URLs descobertas. Essas regras foram definidas para controlar o ritmo da expansão recursiva e a

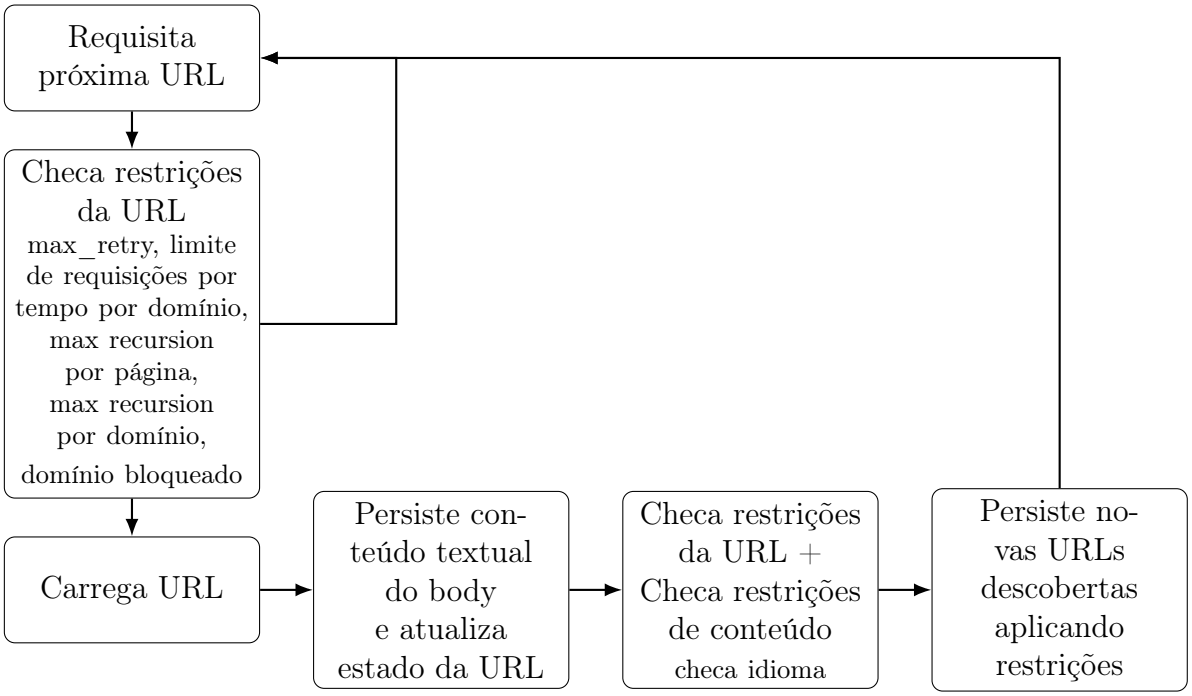



Figura 7.7: Processo de requisição, validação e persistência de URLs e conteúdos durante a coleta.

qualidade dos dados persistidos, garantindo que o processo avance de forma previsível, rastreável e alinhada aos critérios estabelecidos.

A aquisição deixa de ser apenas uma etapa de coleta e passa a atuar como mecanismo de controle da própria exploração, integrando navegação, validação e decisões de continuidade em um único ciclo.

### 7.2.4 Controle de Execução

O controle de execução constitui um componente fundamental da metodologia proposta e é responsável por regular o processo de aquisição. Em particular, esse controle assegura que a exploração ocorra de forma coordenada, limitada e consistente, mesmo em cenários de expansão contínua do espaço de páginas exploradas. Como resultado, é possível expandir esse espaço mantendo o controle e melhorando a qualidade dos dados obtidos.

O controle de execução é estruturado em três dimensões complementares: (i) controle de orquestração, (ii) controle de exploração e (iii) controle de persistência. Essas dimensões atuam de maneira coordenada, estabelecendo restrições que, em conjunto, aumentam a robustez, previsibilidade e reprodutibilidade da execução.

#### Controle de Orquestração

A metodologia foi concebida para funcionar em estágios sequenciais. Durante a execução, esses estágios podem ocorrer novamente, formando ciclos. Com base nessa estrutura, foram definidos mecanismos de controle dos estágios, dos estados do sistema e da orquestração de unidades de processamento que operam de maneira assíncrona.

A principal premissa adotada para definir tais mecanismos é que todas as unidades de processamento envolvidas podem ser executadas concorrentemente, de forma assíncrona, e compartilham a mesma base de código, uma vez que a aquisição é o estágio mais custoso e pode ser realizada por qualquer unidade de processamento em atividade.

O controle de orquestração é responsável por organizar o comportamento global da execução, definindo as transições de estágios e estados, ao mesmo tempo em que estabelece métodos de sincronização das unidades de processamento ativas. Esse controle define uma referência única de estado, que pode ser consultada a qualquer momento, permitindo a sincronização entre as unidades de processamento.

Por exemplo, quando uma unidade de processamento inicia o processo de geração das URLs iniciais (*seeds*), esse estado deve refletir tal atividade, evitando que outras unidades iniciem o processamento antes da disponibilização dessas URLs ou que múltiplas unidades gerem simultaneamente as mesmas URLs.

Esse estado global indica quando uma unidade de processamento já está realizando a geração das URLs iniciais, fazendo com que outras unidades que tentem iniciar o processamento aguardem sua conclusão. Em caso de parada, esse mesmo estado permite que todas as unidades verifiquem que o sistema atingiu um ponto de encerramento e finalizem suas atividades de maneira ordenada.

Outro aspecto central da orquestração é a coordenação distribuída. A metodologia foi desenvolvida para que todas as instâncias em execução possam realizar majoritariamente as mesmas atividades. Assim, todas contribuem para atingir o objetivo principal de aquisição e exploração da web. Apesar disso, em algumas implementações, pode ser necessária a utilização de um nó líder, responsável por reportar dados específicos da execução e realizar ações de encerramento, seja por conclusão, seja por falha.

Manter a mesma base para todas as unidades de processamento, sejam elas líderes ou não, permite que a primeira instância a iniciar a execução assuma esse papel e, em caso de falha, qualquer outra possa substituir essa instância. Durante a exploração recursiva da web, todas executam as mesmas funções e verificações.

Por fim, a orquestração também controla a continuidade do sistema por meio de verificações recorrentes do estado global antes de requisitar uma nova URL para processamento. Assim, caso alguma unidade identifique um problema que exija a interrupção da execução, as demais podem encerrar suas atividades de forma gradual, à medida que concluem o processamento da URL corrente.

A metodologia propõe uma abordagem iterativa de supervisão, com monitoramento contínuo para geração de URLs iniciais, aquisição de dados, persistência e encerramento.

### Controle de Exploração

O controle de exploração determina como o espaço de páginas é percorrido ao longo da execução. Esse controle estabelece como as unidades de trabalho são selecionadas, distribuídas e processadas, bem como as condições sob as quais novas páginas passam a integrar o espaço explorado.

A exploração é definida considerando unidades de processamento independentes, cada uma associada ao processamento de uma página. A distribuição das URLs entre as unida-

des não ocorre de forma aleatória nem segue estritamente uma política *First-In-First-Out* (FIFO), mas respeita um conjunto de regras que define quais URLs são elegíveis. Dessa forma, apenas páginas aptas ao processamento são consideradas para execução paralela.

O controle de progressão é um dos principais elementos na determinação das URLs elegíveis. A metodologia impõe limites estruturais que restringem o avanço no espaço de páginas.

Esses mecanismos impõem uma exploração finita e controlada, evitando tanto o crescimento descontrolado quanto execuções improdutivas.

Durante o processamento de uma página, o controle de exploração define uma sequência estruturada de etapas, incluindo a obtenção do conteúdo, a verificação de condições de sucesso e a extração de links. Esses links são então avaliados e, quando considerados válidos, registrados como URLs candidatas para novas unidades de processamento.

### Controle de Persistência

O controle de persistência é responsável por garantir que o estado da execução e os dados produzidos sejam mantidos de forma consistente ao longo da execução da metodologia e das ferramentas que a implementam. Esse controle não se limita ao armazenamento dos resultados, pois atua continuamente como suporte à própria execução.

Além do armazenamento, a persistência coordena as unidades distribuídas por meio de consistência transacional centralizada no banco de dados. Para a fila de URLs, o protocolo de reivindicação combina a seleção de registros elegíveis, o bloqueio transacional desses registros e a mudança do estado de `TODO` para `PROCESSING`. A seleção e a mudança de estado ocorrem na mesma transação, garantindo que uma URL não seja atribuída simultaneamente a unidades distintas. Essa garantia limita-se à reivindicação e à atualização da fila. A aquisição continua concorrente, portanto não há execução sequencial global nem consistência eventual da fila.

Como premissa, a metodologia exige transações e bloqueios para a reivindicação de URLs, mas não especifica um nível de isolamento SQL nem uma instrução específica de bloqueio. Esses parâmetros dependem do SGBD e devem ser informados em cada implementação para caracterizar com precisão as garantias oferecidas. Sistemas relacionais com suporte a transações e bloqueios, como MySQL e PostgreSQL, atendem a essa premissa.

Cada página possui um estado associado que indica sua situação no fluxo de processamento. Esse estado é atualizado conforme a página avança no processo, permitindo identificar se ela ainda deve ser executada, se está em processamento ou se já foi concluída, com ou sem sucesso. Esse mecanismo torna o processo observável e evita reprocessamentos indevidos.

Além do estado, é mantido um registro do número de tentativas de execução de cada página. Esse registro é fundamental para aplicar limites de reprocessamento e impedir que páginas problemáticas sejam executadas indefinidamente. A persistência também atua como um mecanismo de controle.

O conteúdo obtido durante a exploração também é armazenado de forma estruturada. Dados textuais, como o conteúdo do *body* da página, bem como o HTML completo, podem ser persistidos. Isso permite que os resultados da exploração sejam reutilizados

sem a necessidade de novas requisições, contribuindo para a eficiência e reprodutibilidade.

Entretanto, a inserção direta desses elementos no banco de dados gera um volume elevado de dados, aumentando custos e tempo de execução. Por esse motivo, a utilização de técnicas de compressão é recomendada para reduzir o espaço ocupado pelos dados armazenados.

Outro aspecto essencial é a inserção das novas páginas descobertas. Cada nova página é registrada com a indicação da página de origem e seu nível de profundidade. Esse registro permite reconstruir o processo de navegação e compreender como o espaço explorado foi expandido.

Por fim, são aplicados mecanismos de normalização e deduplicação. A normalização assegura que diferentes representações de uma mesma página sejam tratadas de forma uniforme, enquanto a deduplicação evita redundâncias tanto no nível de páginas quanto de conteúdo. Esses mecanismos contribuem para a eficiência do armazenamento e para a qualidade dos dados produzidos.

**Síntese**

As três dimensões do controle de execução atuam de forma complementar. A orquestração define o comportamento global do sistema, a exploração regula a expansão do espaço de páginas e a persistência assegura a continuidade e a consistência dos dados produzidos.

Essa organização transforma o processo de exploração em um procedimento controlado, no qual cada etapa é regulada por mecanismos explícitos.

### 7.2.5 Validação e Filtragem de Conteúdo

O processo de aquisição funciona de forma recursiva, explorando a web como um grafo e utilizando hiperligações (*hrefs*) como arestas para descobrir novas URLs e conteúdos. A definição de mecanismos robustos para determinar se as relações descobertas devem ser exploradas se torna parte fundamental da metodologia. A etapa de validação e filtragem de conteúdo tem como objetivo central determinar quais documentos são compatíveis com os critérios definidos e quais URLs devem ser exploradas.

Além das regras gerais, é possível definir critérios específicos por domínio, de acordo com o contexto de aplicação e o objetivo final da coleta.

A aplicação desses mecanismos atua de forma qualitativa, reduzindo ruído e evitando a propagação de conteúdos indesejados, além de funcionar como um mecanismo de controle da exploração de novos documentos e URLs. Os filtros não apenas removem dados, mas também atuam na seleção de quais URLs devem ser exploradas e de como o grafo de documentos deve continuar crescendo.

A Figura 7.8 destaca onde esses mecanismos são executados no processamento de uma URL por uma unidade de trabalho. É possível observar que há duas checagens complementares: uma pré-requisição e outra pós-requisição.

Na checagem pré-requisição, são realizadas verificações operacionais de controle para validar se a URL pode ser requisitada naquele momento. A lista a seguir apresenta as checagens já definidas.

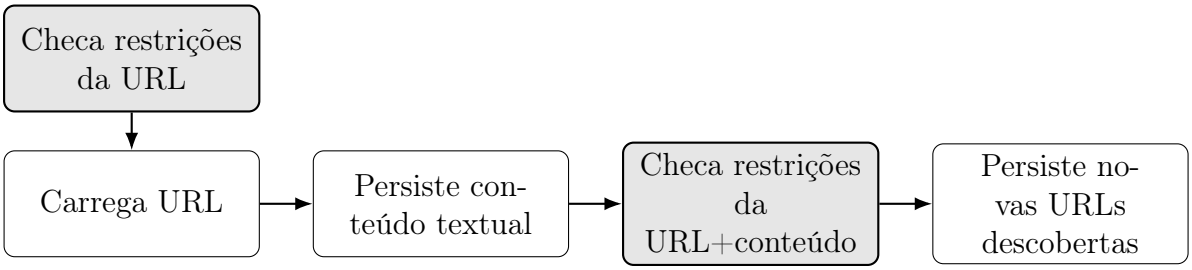



Figura 7.8: Processo de aquisição com destaque nas validações e filtros.

- **Restrições de profundidade da página:** limitam o número de níveis de navegação a partir de uma página inicial dentro do mesmo encadeamento de links, controlando a expansão local do grafo.
- **Restrições de profundidade de domínio:** limitam o número de transições entre domínios distintos ao longo da exploração, restringindo a expansão interdomínios.
- **Limite de reprocessamento:** define o número máximo de tentativas de requisição de uma mesma página em caso de falhas, evitando ciclos de reexecução indefinidos.
- **Intervalo mínimo entre requisições por domínio:** estabelece um tempo mínimo entre requisições consecutivas a um mesmo domínio, controlando a taxa de acesso e evitando sobrecarga.
- **Bloqueio explícito de domínios:** impede a requisição de URLs pertencentes a domínios previamente classificados como indesejados, garantindo a exclusão sistemática dessas fontes.
- **Elegibilidade baseada no estado da página:** restringe a seleção de URLs aos estados apropriados do ciclo de processamento, evitando reprocessamentos indevidos.
- **Controle temporal de reprocessamento:** condiciona a reexecução de páginas que falharam previamente a um intervalo mínimo, garantindo espaçamento entre tentativas sucessivas.

Os limites de profundidade de página e de domínio são verificados separadamente para cada URL descoberta. A profundidade de página representa os níveis percorridos na cadeia de navegação, enquanto a profundidade de domínio representa as transições entre domínios distintos. Para que a URL permaneça elegível, ambos os limites precisam ser atendidos; exceder qualquer um deles gera bloqueio.

Essas verificações são realizadas de forma consistente entre unidades concorrentes, sendo baseadas no estado persistido das URLs, o que garante decisões uniformes e evita conflitos durante a execução distribuída.

Na validação pré-requisição, cada URL pode ser considerada inelegível por diferentes motivos. Alguns desses motivos impedem que a URL seja processada novamente, como no caso de domínio bloqueado. Outros motivos de falha são temporários, como o intervalo mínimo para requisição no mesmo domínio. Algumas checagens resultam em estados

temporários que permitem reprocessamento futuro, enquanto outras mudam seu estado para bloqueada.

Na checagem pós-requisição, são realizadas validações semânticas no conteúdo obtido com a requisição. Esse conteúdo é avaliado para determinar se atende ao objetivo final da geração do conjunto de dados, como, por exemplo, estar escrito em português, conter uma quantidade mínima de palavras ou incluir termos específicos. Para cada URL encontrada no conteúdo obtido, as mesmas checagens de controle pré-requisição são reaplicadas para determinar se a URL deve ser registrada como bloqueada por algum motivo ou como elegível.

As referências como *hrefs* a outras páginas descobertas a partir de uma URL só podem ser inseridas como elegíveis se atenderem a todos os requisitos anteriores e se a página que as gerou for válida. Caso a página atual seja inválida, a expansão a partir dela é bloqueada, e as URLs descobertas são persistidas com estado de bloqueio.

As validações não têm como objetivo descartar documentos, URLs ou páginas que já foram identificadas. O estado da URL é alterado para determinar sua situação atual, sem a realização de exclusões de registros, a fim de manter o histórico da evolução e da execução da aplicação que implementa a metodologia.

Além dessas checagens e filtros já definidos, é possível que cada domínio, ou ferramenta implementada com base nesta metodologia, adicione novos filtros que devem ser validados de acordo com o domínio em que se inserem e com o objetivo final.

Esta etapa de validação e filtragem não funciona como um mecanismo de exclusão de dados, mas como um componente central de governança do processo de aquisição. Com a utilização de critérios operacionais e semânticos, a metodologia permite controlar a expansão do grafo de documentos e manter a rastreabilidade das decisões tomadas ao longo da execução.

### 7.2.6 Controle de Redundância

Um dos problemas mais comuns no processo de exploração e coleta de URLs desta metodologia é a existência de múltiplas páginas referenciando a mesma URL. Nesses casos, a reexploração não agrega informações relevantes e aumenta significativamente o tempo do estágio de aquisição. Além disso, páginas podem se referenciar mutuamente, formando ciclos que precisam ser tratados para garantir a qualidade dos dados. A Figura 7.9 ilustra esse comportamento, com três páginas diferentes referenciando a mesma URL.

Identificar quais páginas são equivalentes pode ser útil para a reconstrução do grafo explorado. Por isso, esta metodologia adota uma técnica de identificação multicamada, verificando URLs e conteúdos para identificar e marcar redundâncias estruturais e semânticas.

A adoção da identificação multicamada visa preservar as características finais do conjunto de dados, evitando que o dado principal, construído a partir do conteúdo textual dos sites, fique repetitivo ou desbalanceado. Esse desbalanceamento pode ocorrer, por exemplo, por páginas que bloqueiam requisições retornando a mesma mensagem.

A primeira camada é a checagem estrutural, na qual as URLs são verificadas para identificar se já existe alguma correspondência registrada anteriormente. Para isso, é

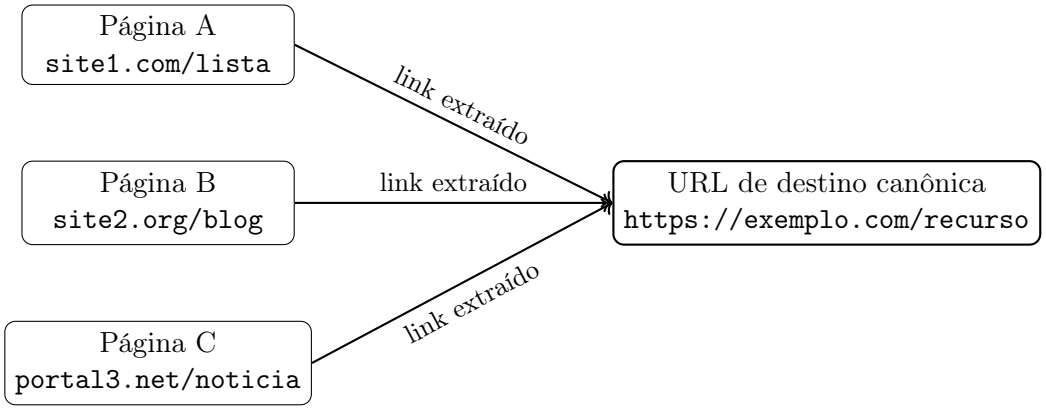



Figura 7.9: Múltiplas páginas de origem referenciando a mesma URL.

necessário realizar uma etapa de normalização da URL.

Durante o processo de normalização de URLs, são aplicadas transformações que não alteram a semântica da requisição ao servidor. Assim, parâmetros de consulta são preservados (*query*).

Especificamente, os espaços em branco no início e no final da URL são removidos, bem como barras repetidas ao final do endereço. Os fragmentos identificados pelo símbolo `#` são descartados, pois não são transmitidos ao servidor durante a requisição HTTP. URLs sem esquema definido recebem, por padrão, o prefixo `http://`. A lista a seguir apresenta alguns exemplos desse processo.

- `"example.com/path#section "` : `"http://example.com/path"`
- `"https://site.com///"` : `"https://site.com"`
- `"www.test.com/page///#top"` : `"http://www.test.com/page"`

No caso dos domínios, a normalização consiste em extrair o *hostname* completo da URL, preservando os subdomínios, o domínio principal e o *Top-Level Domain* (TLD), e descartando componentes como caminho, parâmetros de consulta e fragmentos. A lista a seguir apresenta alguns exemplos desse processo.

- `"https://WWW.Example.com/path"` : `"example.com"`
- `"http://blog.site.org/articles?id=10"` : `"blog.site.org"`
- `"test.com/page/subpage"` : `"test.com"`

Uma vez concluída a normalização, para identificar dados redundantes e viabilizar a persistência, são gerados valores de *hash* para a URL e para o domínio. A utilização de *hashes* permite identificar elementos repetidos de forma eficiente em sistemas de bancos de dados relacionais, pois reduz a comparação de *strings* potencialmente longas a comparações binárias de tamanho fixo. Esse mecanismo permite que múltiplas unidades de processamento verifiquem e insiram dados de forma consistente, evitando duplicações mesmo em cenários de execução concorrente.

Antes da inserção de uma nova URL ou domínio no sistema, é realizada uma verificação de existência baseada no valor de *hash* previamente calculado. No caso das URLs, esse processo consiste em verificar se o valor de *hash* gerado a partir da URL já foi persistido. Caso exista correspondência, o registro existente é reutilizado, evitando a criação de entradas duplicadas. No caso dos domínios, a verificação segue o mesmo princípio para a *hash* gerada a partir deles, garantindo que domínios já conhecidos não sejam inseridos novamente e permitindo a correta manutenção das relações entre URLs e domínios.

A segunda camada se relaciona à identificação de elementos duplicados com base no conteúdo textual obtido da página. Isso se torna especialmente relevante em cenários nos quais sites distintos apresentam o mesmo conteúdo, situação comum em páginas de manutenção ou de acesso bloqueado.

Nesses casos, a mesma estratégia de geração de *hash* do conteúdo é aplicada para verificar a existência de conteúdos idênticos já armazenados.

Para gerar as *hashes* de domínio, URL e conteúdo, o MD5 é utilizado [Rivest, 1992]. Essa técnica foi escolhida por sua eficiência computacional e por apresentar probabilidade teórica de colisão aleatória de aproximadamente 1 em $2^{128}$.

No caso do domínio, a *hash* é utilizada tanto para identificar unicamente cada domínio quanto para verificar sua existência prévia no sistema, evitando a inserção de registros duplicados. Esse procedimento garante a unicidade estrutural dos domínios e a consistência das relações entre URLs e seus respectivos domínios. Diferentemente do que ocorre com URLs e conteúdos textuais, a *hash* de domínio não é utilizada para inferir equivalência semântica, atuando exclusivamente no nível estrutural.

Quando um conteúdo duplicado é identificado com base no texto obtido da página, o registro não deve ser removido. Em vez disso, um atributo específico é atualizado para referenciar o registro já existente, estabelecendo uma relação entre eles. A persistência dessa informação permite reconstruir o grafo explorado da web e analisar o comportamento do processo de aquisição ao longo do tempo, sem introduzir redundância no armazenamento.

Na descrição metodológica, a correspondência do hash do conteúdo textual (`text_md5`) associa o novo registro ao conteúdo já armazenado por meio de `same_as`. A verificação prévia do hash da URL normalizada (`url_md5`) reutiliza o registro existente antes de inserir outra linha.

### 7.2.7 Persistência e Representação dos Dados

Neste capítulo, o termo persistência designa o estado registrado do sistema e os mecanismos de armazenamento e consulta. Não designa um agente autônomo nem, por si só, um componente específico da arquitetura. Os componentes de execução consultam e atualizam esse estado para registrar as fases e unidades de trabalho, selecionar URLs, coordenar agentes concorrentes e retomar a execução após falhas. Assim, a persistência dá suporte a essas funções, mas não as executa.

Para fins de modelagem, o estado persistido é descrito em dois níveis: global, que registra a fase corrente do sistema, e local, associado às unidades de trabalho. Os componentes de execução consultam o estado global para identificar o estágio e o estado local para selecionar novas URLs, dar continuidade à aquisição e retomar tarefas após falhas.

A representação persistida do estado atual evolui ao longo da execução dos estágios. Cada registro representa uma unidade de trabalho, definida como uma URL associada ao seu contexto de aquisição e aos metadados necessários para sua avaliação, processamento e eventual descarte. Essa unidade define sua posição no ciclo de processamento, permitindo identificar sua evolução até estados finais, como conclusão ou descarte após falhas sucessivas.

A persistência pode ser analisada em três níveis complementares: (*i*) nível conceitual, no qual o sistema é definido como uma máquina de estados persistida; (*ii*) nível lógico, que especifica as entidades e suas relações; e (*iii*) nível físico, com essas definições materializadas em um sistema de gerenciamento de banco de dados relacional.

Para isso, a modelagem da persistência tem como objetivo central contemplar a transição de estados como mecanismo de execução do sistema e de aquisição de dados. A modelagem das URLs representa as fases do ciclo de identificação, obtenção e conclusão de cada URL. Consequentemente, a execução do sistema pode ser interpretada como a evolução sucessiva dos estados persistidos.

A execução é formalizada como uma máquina de estados finita persistida, na qual cada unidade de trabalho percorre um conjunto discreto de estados mutuamente exclusivos, incluindo estados de elegibilidade, execução, sucesso e bloqueio ou falha. As transições entre estados são realizadas exclusivamente por operações transacionais sobre a persistência, garantindo consistência mesmo sob concorrência distribuída.

As principais entidades necessárias para atender aos requisitos desta metodologia são apresentadas em uma estrutura conceitual do modelo de dados. Estes elementos, que representam domínios, páginas com URLs e domínios bloqueados, são descritos em um modelo relacional, adotado por sua capacidade de representar explicitamente relações estruturais, garantir integridade referencial e permitir a rastreabilidade das dependências entre as entidades do processo de aquisição.

Devido às propriedades recursivas do estágio de aquisição, as definições representadas nas relações e nos modelos de persistência precisam ser ajustadas para explicitar o processo de exploração como uma estrutura recursiva. Domínios e páginas sempre referenciam o domínio ou a página que levaram à identificação de cada novo item descoberto. Além de identificar os saltos de exploração percorridos da origem até o registro atual, essa estrutura torna possível determinar a profundidade de cada registro, o que permite controlar a exploração e limitar a execução recursiva.

De acordo com o controle de redundância, a existência e a checagem dos *hashes* de URL, domínio e conteúdo são obrigatórias e precisam estar representadas nos modelos propostos de persistência. Um atributo específico é utilizado para armazenar esses itens, podendo atuar como mecanismo de identificação e verificação de equivalência entre entidades. A identificação de equivalência entre entidades é realizada por meio de funções de *hash* aplicadas a atributos normalizados. Essa abordagem permite definir equivalência estrutural, sobre URL ou domínio, e equivalência semântica, utilizando o conteúdo, evitando reprocessamento redundante e garantindo consistência no grafo de exploração.

Como parte fundamental da metodologia consiste em não processar páginas que seriam descartadas por baixa qualidade ou por introduzirem ruído nos dados, o conceito de domínios bloqueados é definido diretamente na metodologia. Para isso, é adotado

um modelo que representa esses domínios como uma estrutura declarativa baseada na identificação do domínio por meio de seu *hash*.

A modelagem de domínios bloqueados integra diretamente a lógica de execução, uma vez que a elegibilidade de uma URL depende dessa verificação. Os componentes de execução consultam as restrições persistidas para decidir quais caminhos de exploração são permitidos e quais devem ser excluídos imediatamente.

Nos diagramas, o atributo `domain` das entidades `domain` e `blocked_domain` armazena o *hostname* normalizado, identificado por `domain_md5`. Já `pages.url` representa a URL completa da página.

Durante a seleção de trabalho, isto é, ao escolher a URL a ser obtida e explorada, o sistema utiliza o estado persistido de cada URL em conjunto com as regras definidas pela implementação, configurando um ponto crítico de controle na modelagem da persistência. Isso faz com que o sistema que implementa a metodologia se caracterize como um modelo de execução orientado a dados, no qual as decisões operacionais são definidas a partir do estado persistido.

A seleção de unidades de trabalho é realizada por consultas sobre o estado persistido, caracterizando uma fila implícita e sem a necessidade de utilização de mecanismos de filas dedicados. Nesse modelo, a elegibilidade de cada unidade é determinada dinamicamente a partir de atributos persistidos, como estado atual, número de tentativas, nível de recursão e restrições temporais, permitindo que a fila seja derivada diretamente do banco de dados.

A Figura 7.10 apresenta um diagrama mínimo da modelagem das três principais classes necessárias à implementação das definições desta metodologia. A modelagem apresentada corresponde a uma abstração mínima necessária para utilizar os conceitos e definições desta metodologia. Em implementações concretas, essa estrutura deve ser estendida com atributos adicionais, incluindo metadados de execução, identificadores derivados e marcas temporais.

No diagrama apresentado, são observadas três relações de autorreferência. Na entidade de domínio, essa autorreferência identifica o domínio de origem de cada novo domínio descoberto, permitindo compreender a interconexão entre os itens encontrados. Na entidade de páginas, a primeira autorreferência é utilizada para identificar páginas redundantes, indicando o registro ao qual o registro atual é equivalente e evitando sua exclusão. Por fim, ainda na entidade de páginas, a segunda autorreferência, de modo análogo ao domínio, permite identificar o caminho percorrido durante a exploração da web ao indicar qual registro originou outro.

O estado persistido de cada unidade de trabalho dá suporte à tolerância a falhas por ser atualizado ao longo da execução. Em situações de falha ou interrupção, os componentes de execução podem reclassificar unidades em estados intermediários com base em critérios temporais e no histórico de execução, permitindo sua reintegração ao conjunto de elegibilidade.

As definições desta metodologia para a criação desses modelos decorrem do objetivo de utilizar a persistência como mecanismo de controle, incorporando à modelagem as características necessárias para suportar a exploração recursiva. Além disso, ao adotar as propostas de persistência desta metodologia, a execução do sistema se torna determinística em relação ao estado persistido, permitindo não apenas rastrear integralmente

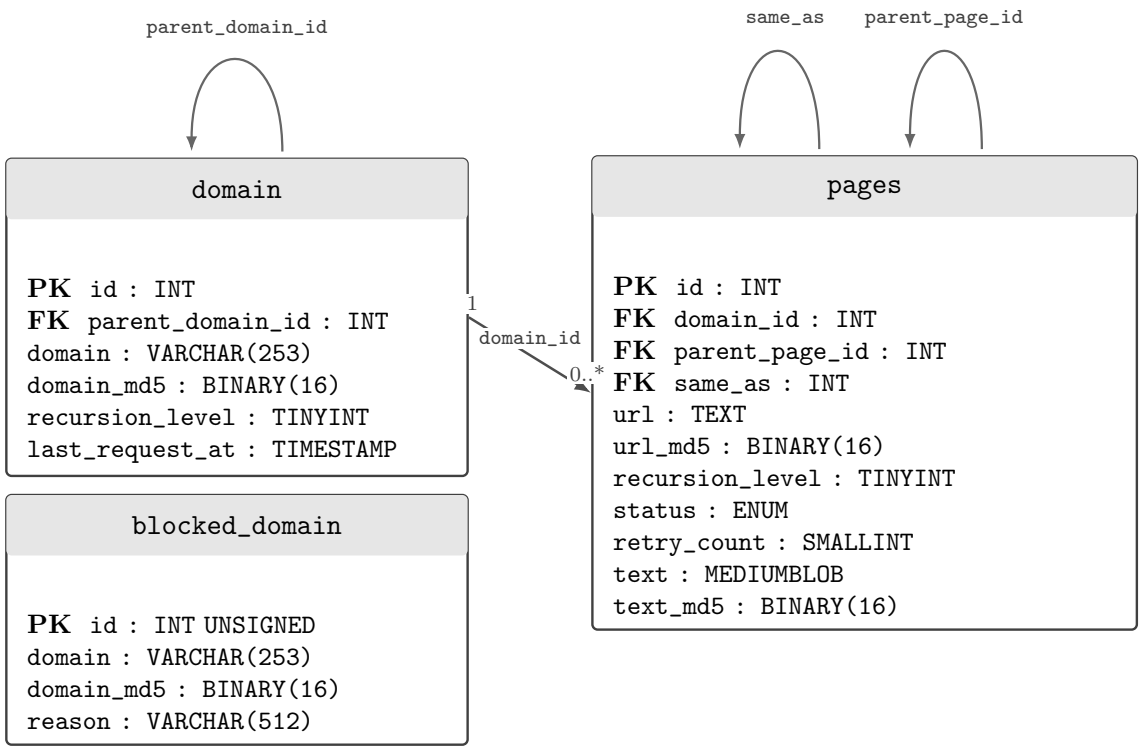



Figura 7.10: Modelagem mínima para atender aos requisitos de persistência da metodologia.

o processo de exploração, mas também reproduzir, auditar e recuperar sua evolução de forma consistente.

## 7.3 Implementação no Sistema WackyWacky

Esta seção apresenta o WackyWacky, uma implementação baseada na metodologia proposta que materializa e especifica as recomendações para a geração de conjuntos de dados de domínio aberto. Além de validar a metodologia, essa implementação tem como objetivo criar e pavimentar o caminho para novos conjuntos de dados.

Nesta implementação, a metodologia é materializada como um crawler distribuído, no qual múltiplas unidades de processamento participam de todos os estágios definidos. Conforme definido nas especificações, o sistema de banco de dados é utilizado como mecanismo de controle, sincronização e fila de processamento. Os atributos persistidos durante as execuções dos estágios permitem validar os estados e as transições, bem como o processo de exploração.

O banco de dados desempenha um papel fundamental, funcionando como o estado persistente da aplicação. Essa função garante a idempotência das operações e controla a concorrência. Os hashes definidos são utilizados para a identificação de duplicados e para atuar como índices, permitindo operações mais rápidas.

Nesta implementação, toda a aplicação está instrumentada para permitir a visualização de traces, logs e métricas durante a execução. Uma das unidades de processamento ativas atua como líder e, além das atividades recorrentes do estágio de aquisição, é res-

ponsável por alterar o estado global do sistema durante a execução.

O sistema possui os estados globais `starting`, `running_starter`, `running_mining`, `completed`, `error` e `stopping`. O estágio de aquisição da metodologia corresponde ao estado `running_mining`, no qual as URLs são exploradas e obtidas de forma recursiva.

Na implementação do *WackyWacky*, a linguagem `Python 3.12.13` foi utilizada, enquanto o `MySQL 8.0.45` foi adotado como mecanismo de armazenamento e coordenação entre os estágios distribuídos. A obtenção de conteúdo web foi realizada com o `Playwright`, o que permitiu a execução de requisições e a interação com páginas dinâmicas. A observabilidade da aplicação, por sua vez, foi viabilizada por meio do `Open Telemetry`, com suporte à coleta de traces, logs e métricas.

O banner da aplicação desenvolvida é apresentado na Figura 7.11. Todo o código-fonte está disponível publicamente no GitHub[1].




Figura 7.11: Banner de apresentação do sistema *WackyWacky*.

Nas próximas seções, são apresentados o funcionamento e as decisões técnicas adotadas na implementação desta ferramenta.

### 7.3.1 Componentes Principais

A implementação atual da ferramenta foi desenvolvida com uma arquitetura modular, permitindo que pesquisadores possam adaptar a ferramenta a necessidades e domínios específicos no futuro. Dessa forma, componentes especializados foram criados para delimitar responsabilidades, sempre utilizando o estado persistido para auxiliar no controle e na execução.

A ferramenta suporta execução concorrente com *threads* locais e múltiplas unidades de processamento, coordenadas pelo estado persistido no banco de dados. A Figura 7.12 apresenta os principais componentes e suas dependências funcionais. As setas indicam a utilização de um componente por outro, sem representar uma sequência de execução.

O `App` constitui o ponto de entrada da aplicação e o componente central de orquestração do sistema. Esse componente é responsável pelo paralelismo, pela identificação de URLs disponíveis para aquisição, pela execução do `Starter` e pelo gerenciamento do estado global e das unidades de execução do sistema.

O componente `App` estabelece a conexão com o banco de dados e realiza a eleição de líder, que se apoia no próprio banco para essa definição. A coordenação da execução ocorre com base no estado global do sistema.

[1] https://github.com/matheusFerraroni/wackywacky

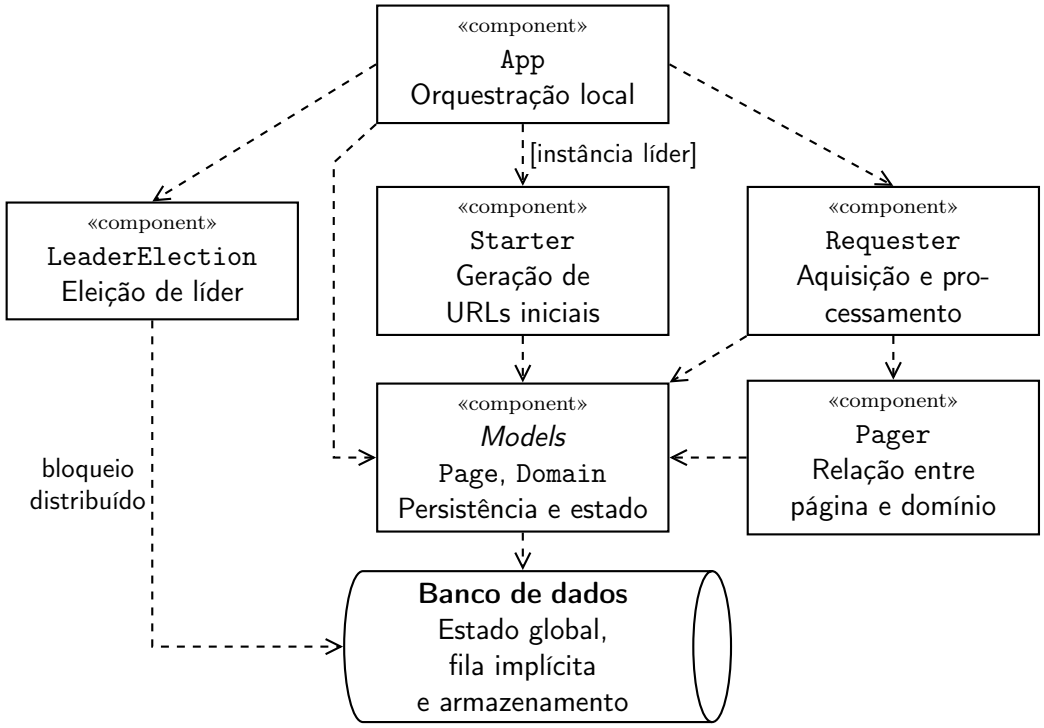



Figura 7.12: Principais componentes do WackyWacky e suas dependências funcionais.

Quando a instância é eleita líder, o `App` executa o componente `Starter` para gerar as URLs iniciais (*seeds*). A implementação inclui suporte a Bing, DuckDuckGo, Google e Wikipedia. Cada termo definido gera uma URL em cada sistema habilitado nas configurações dinâmicas.

Com as URLs iniciais definidas, ou com o sistema já em execução, o `App` coordena o processamento das URLs disponíveis para aquisição.

O `Requester` se apoia no `Pager` para identificar de forma única o domínio associado à URL no banco de dados, permitindo a persistência dos dados no sistema de armazenamento.

Todas as alterações de estado e a persistência de novos dados obtidos, como dados textuais da página ou novas URLs, são realizadas utilizando as *models* específicas disponíveis.

A implementação utiliza dois componentes de configuração. Um módulo é responsável por identificar configurações locais, incluindo variáveis de ambiente, como número máximo de threads, dados de acesso ao banco de dados, idioma-alvo e se o conteúdo HTML também deve ser salvo. O segundo módulo obtém configurações persistidas no banco de dados, responsáveis por definir o comportamento do sistema, como termos iniciais para geração das URLs *seeds*, estado geral do sistema, quais instâncias do `Starter` devem ser executadas.

### 7.3.2 Modelo de Dados

Os mecanismos de persistência são responsáveis por centralizar estado, coordenação distribuída e controle de execução no sistema. Seguindo essas definições, a persistência se apoia no modelo relacional para funcionar como núcleo de coordenação e gerenciamento de estado. A Tabela 7.1 apresenta as funções da persistência nesta implementação, as quais se materializam diretamente nas entidades `pages` e `domain`.

| Função | Descrição no contexto do sistema |
|---|---|
| Armazenamento de dados | Persistência de URLs, conteúdo textual e metadados extraídos durante o processo de crawling. |
| Coordenação distribuída | Sincronização entre múltiplos workers por meio de controle transacional e mecanismos de bloqueio no banco de dados. |
| Mecanismo de sincronização | Garantia de exclusividade na seleção e no processamento de URLs em cenários concorrentes. |
| Fila de trabalho | Implementação de uma fila implícita baseada em estados persistidos, como `TODO` e `PROCESSING`, permitindo a orquestração do fluxo de execução. |
| Representação de estado | Registro do estado de execução global e individual das unidades de processamento, viabilizando controle e rastreabilidade. |

Tabela 7.1: Responsabilidades da persistência no sistema.

A escolha do modelo relacional decorre da necessidade de garantias fortes de consistência e isolamento, fundamentais para implementar controle concorrente baseado em bloqueios, como a seleção de tarefas por meio de operações transacionais com controle de bloqueio, permitindo coordenar os acessos simultâneos diretamente no banco de dados com garantias de consistência e isolamento, sem dependência de sistemas externos de mensageria.

Para atender às definições da metodologia e, ao mesmo tempo, permitir a geração rastreável sem domínio específico, foi desenvolvido o modelo apresentado na Figura 7.13.

Foram definidas quatro entidades principais: `domain`, `blocked_domain`, `pages` e `settings`. A entidade `domain` atua como mecanismo de controle de taxa por domínio, permitindo regular a frequência de requisições com base em estado persistido. A entidade `blocked_domain` tem como única responsabilidade auxiliar na exclusão de domínios reconhecidamente de baixa qualidade, como spam e servidores de telemetria.

Por sua vez, a entidade `settings` armazena configurações dinâmicas que podem ser alteradas durante a execução. Já a entidade `pages` constitui o núcleo operacional do sistema, atuando simultaneamente como fila distribuída, estrutura de controle de execução e repositório de conteúdo. O texto extraído é persistido de forma comprimida, enquanto o HTML pode ser armazenado opcionalmente. Além disso, a entidade mantém atributos responsáveis pelo controle de execução, como estado, número de retentativas e profundi-

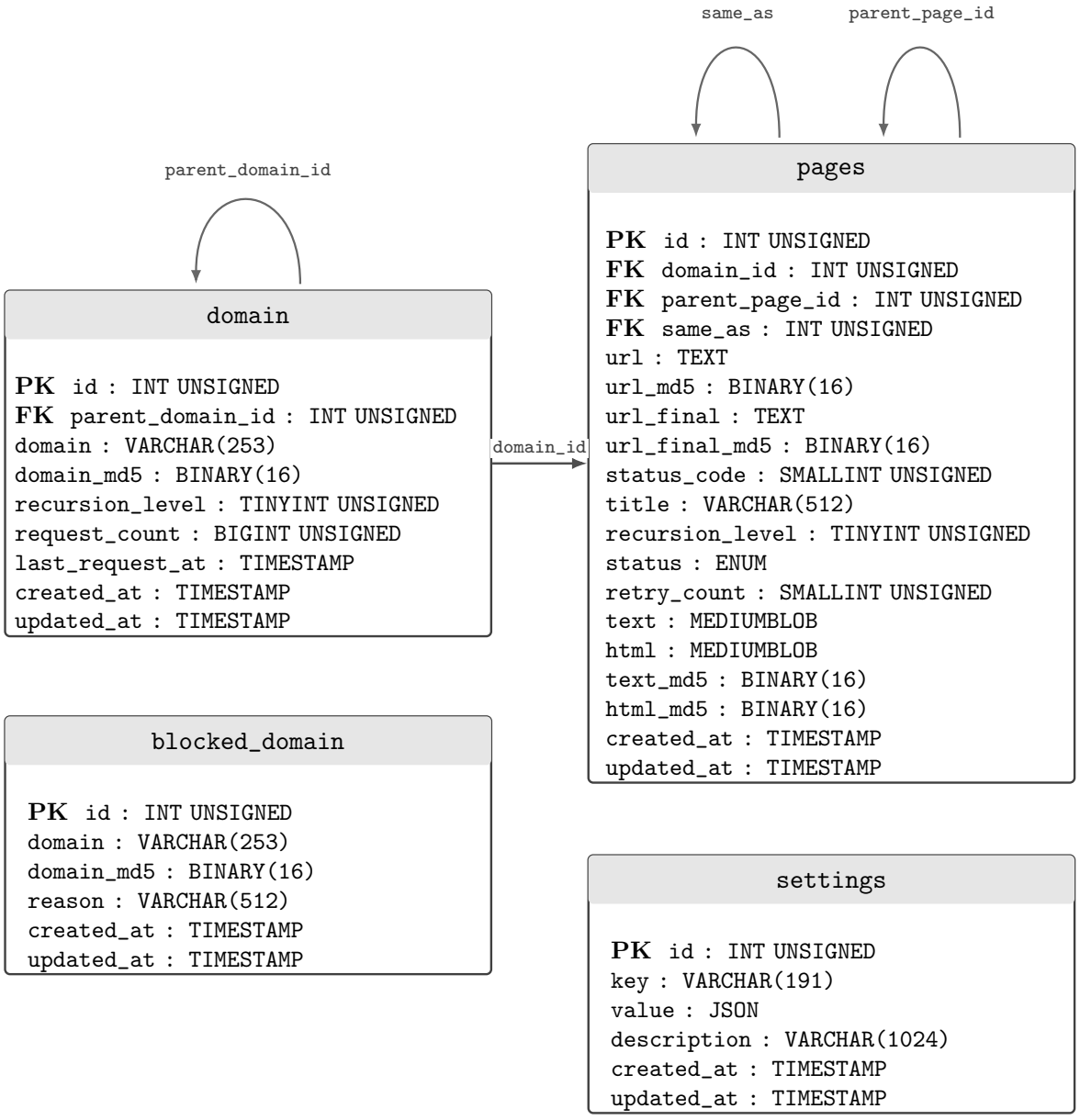



Figura 7.13: Modelo relacional do crawler com configuração dinâmica desacoplada.

dade de exploração. Para isso, são utilizados o estado atual, a quantidade de retentativas, a profundidade da recursão e a última atualização do registro.

Na Figura 7.13, os campos `pages.text` e `pages.html` são definidos como `MEDIUMBLOB`, que comporta conteúdos maiores que o limite de 65.535 bytes do tipo `BLOB`. Caso o conteúdo esperado ultrapasse a capacidade de `MEDIUMBLOB`, deve-se usar `LONGBLOB`.

As entidades `domain`, `blocked_domain` e `pages` possuem atributos dedicados à persistência do *hash* gerado a partir do domínio ou da URL. Esse mecanismo permite deduplicação estrutural, por URL, e pelo conteúdo obtido baseado em *hash*, reduzindo redundância estrutural e limitando a expansão combinatória do grafo de exploração. Quando uma equivalência é identificada, ela é referenciada no atributo `same_as`.

O atributo `same_as` atua como mecanismo para preservar a rastreabilidade do percurso de descoberta, sem replicação desnecessária de conteúdo.

Para modelar a estrutura da web como um grafo, os atributos `parent_domain_id` da entidade `domain` e o atributo `parent_page_id` da entidade `pages` são utilizados para representar um grafo dirigido que permite reconstruir a descoberta e a relação entre páginas e domínios.

Apesar das vantagens obtidas com esse tipo de abordagem, ainda existem limitações. A centralização de múltiplas responsabilidades no banco impõe limites de escalabilidade, especialmente em cenários com alto grau de paralelismo, devido à contenção em operações transacionais e ao aumento de latência em operações de I/O. Embora esta limitação não tenha sido crítica nos experimentos realizados, ela constitui um potencial gargalo em cenários de maior escala, especialmente sob alta concorrência.

### 7.3.3 Execução e Concorrência

Esta seção apresenta o funcionamento do modelo de execução concorrente implementado pelo WackyWacky. Dois níveis distintos de concorrência estão presentes para maximizar o aproveitamento de recursos, como CPU e memória. O primeiro nível é composto por *threads* locais, que promovem maior eficiência no uso dos recursos da própria unidade de processamento. O segundo nível corresponde à execução em múltiplas unidades de processamento, abordagem que tira proveito da distribuição de carga para ampliar o desempenho. Apesar disso, essa coordenação traz desafios de sincronização que precisaram ser endereçados, como a manutenção da consistência dos dados em execuções simultâneas.

Cada unidade de processamento mantém um *pool* de $N$ *threads* locais independentes, responsáveis pela execução completa do ciclo de aquisição e processamento de páginas. Cada *thread* é responsável por todo o estágio de aquisição, desde obter os dados textuais de uma URL até salvar os dados obtidos e persistir cada nova URL descoberta com o estado apropriado, respeitando a profundidade da recursão, o filtro de idioma e os domínios bloqueados.

O sistema suporta a execução simultânea de múltiplas unidades computacionais distribuídas, sendo que cada unidade é responsável por se autogerenciar, utilizando o mecanismo de persistência para sincronizar sua execução com as demais unidades.

Essa abordagem distribuída é viabilizada pelo papel central do mecanismo de persistência, que atua simultaneamente como fila global de tarefas e como mecanismo de coordenação entre as unidades de processamento. Ao centralizar o controle de estado e a seleção de tarefas, o sistema permite a sincronização entre execuções concorrentes sem a necessidade de um orquestrador externo.

Unidades de processamento e *threads* locais competem pelo mecanismo de persistência. A Figura 7.14 apresenta o funcionamento dessa concorrência, destacando como o mecanismo de persistência exerce papel central nessa abordagem.

Todas as *threads* de cada unidade de processamento leem e escrevem simultaneamente na fila implícita, durante a seleção de tarefas, atualização do estado e inserção de novas URLs descobertas. Para minimizar o impacto dessa centralização, cada requisição de URLs disponíveis para processamento atualiza o estado das URLs selecionadas para indicar que já estão em processamento, e outra medida para reduzir esse efeito é o mecanismo de seleção de URLs, que sempre seleciona as próximas vinte URLs disponíveis a cada

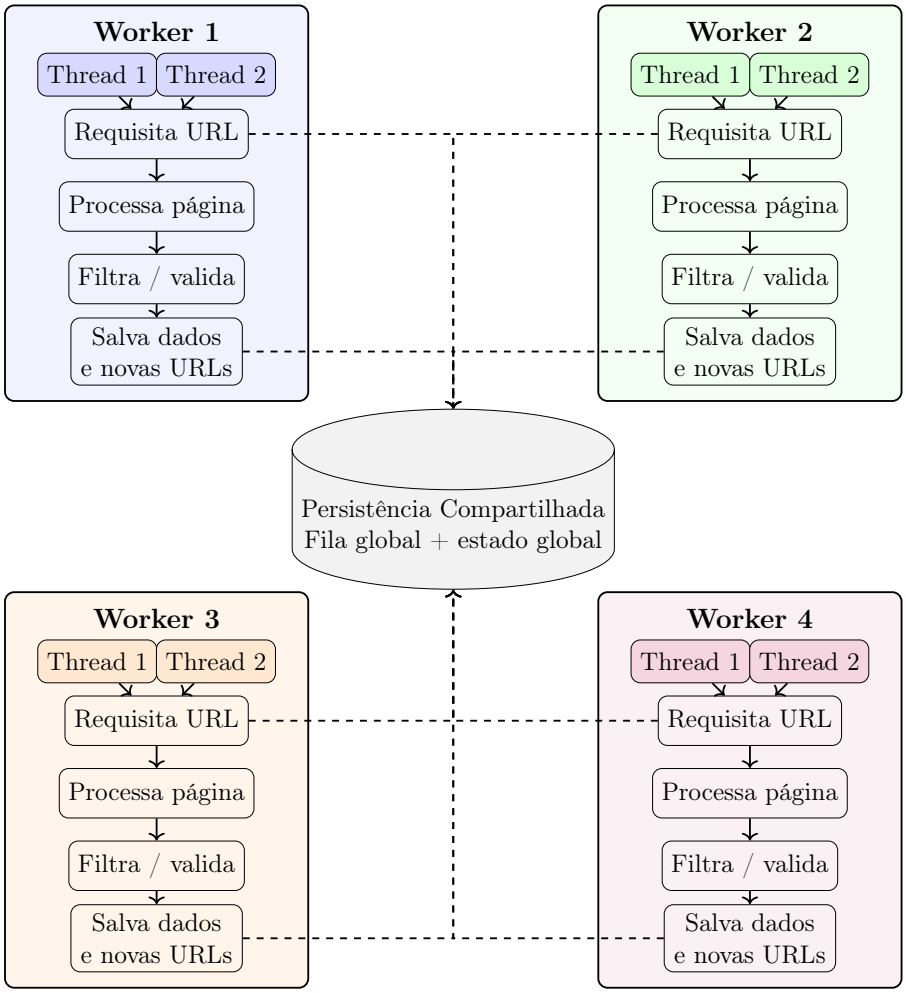



Figura 7.14: Demonstração da execução concorrente.

requisição.

Embora essas otimizações reduzam a contenção e a quantidade de requisições com filtros na seleção de URLs, melhorando o desempenho dessa etapa, a execução concorrente distribuída ainda introduz cenários de falha. Como consequência, falhas ou interrupções podem deixar URLs no estado de processamento sem conclusão.

Para mitigar esse problema, a implementação incorpora um mecanismo auxiliar de liderança, responsável por operações globais do sistema. O líder atua na identificação e liberação de registros travados no estado de processamento, além de ser a única unidade autorizada a modificar o estado global do sistema. Durante a primeira execução, também é responsável pela inserção das URLs *seeds* iniciais e pela transição do sistema para o estágio de aquisição.

O modelo de execução permite dois modos de escalabilidade. O primeiro é o escalonamento vertical, utilizando unidades de processamento com maior poder computacional e a execução de mais *threads* locais ao mesmo tempo. O segundo é o escalonamento horizontal, utilizando mais unidades de processamento simultaneamente.

### 7.3.4 Fluxo de Processamento

O fluxo de processamento constitui o núcleo operacional do sistema WackyWacky e é responsável por executar, de maneira distribuída, os registros obtidos da fila implícita persistida na tabela `pages`, na qual o estado de cada registro atua como mecanismo de controle de processamento concorrente. Toda a orquestração, as checagens e as decisões são realizadas de forma autônoma, para garantir a independência durante o estágio de aquisição.

Conforme apresentado na Figura 7.15, a primeira ação é a eleição do líder desta execução, utilizando o mecanismo de `LOCK` distribuído do sistema de persistência. Esse líder é responsável por inicializar o sistema, gerar as URLs *seeds* iniciais e alterar o estado do sistema para o estágio de aquisição.

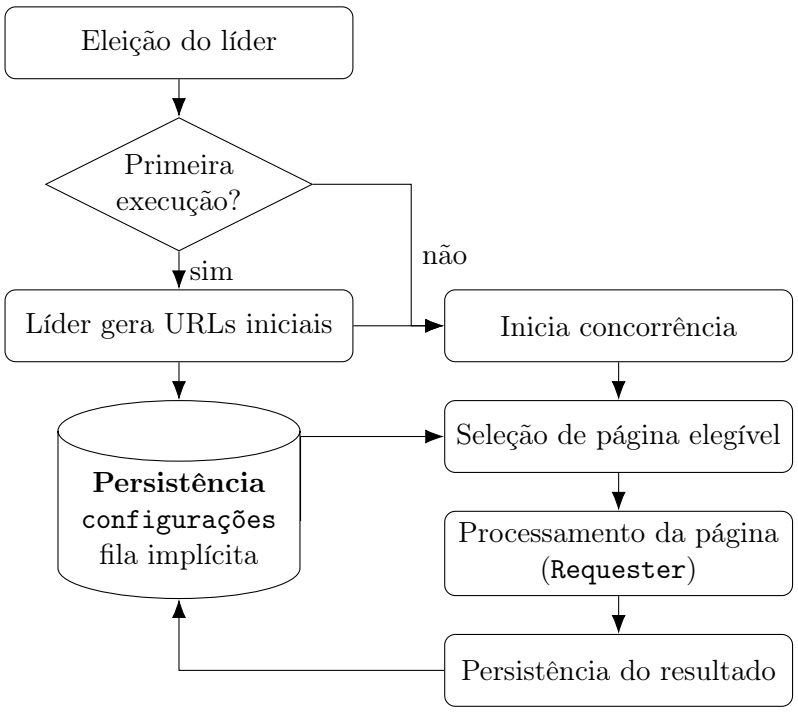



Figura 7.15: Pipeline principal de processamento.

Durante o processo de geração das URLs iniciais, o líder utiliza as configurações dinâmicas para determinar qual componente de geração de URLs deve ser utilizado. Os resultados são persistidos como URLs no estado `TODO`, estabelecendo o ponto inicial para a continuidade do processo.

A partir desse ponto, é possível iniciar o mecanismo de concorrência local, com múltiplas *threads* por unidade de processamento selecionando registros elegíveis sem colisão na seleção concorrente dos registros, de acordo com as regras estabelecidas anteriormente. Essa seleção é realizada diretamente sobre a tabela `pages`, utilizando mecanismos de bloqueio transacional no banco de dados para concorrência que garantem exclusividade no consumo dos registros.

Cada registro de URL disponível para processamento é delegado ao componente `Requester`, responsável por executar o ciclo completo de processamento de uma URL, incluindo validação de elegibilidade, execução da requisição HTTP, extração de conteúdo

estruturado e persistência dos resultados. Enquanto a Figura 7.15 descreve o fluxo macro de orquestração, a Figura 7.16 detalha o processamento interno de cada unidade de trabalho. A figura evidencia as alterações de estado da URL selecionada a cada decisão.

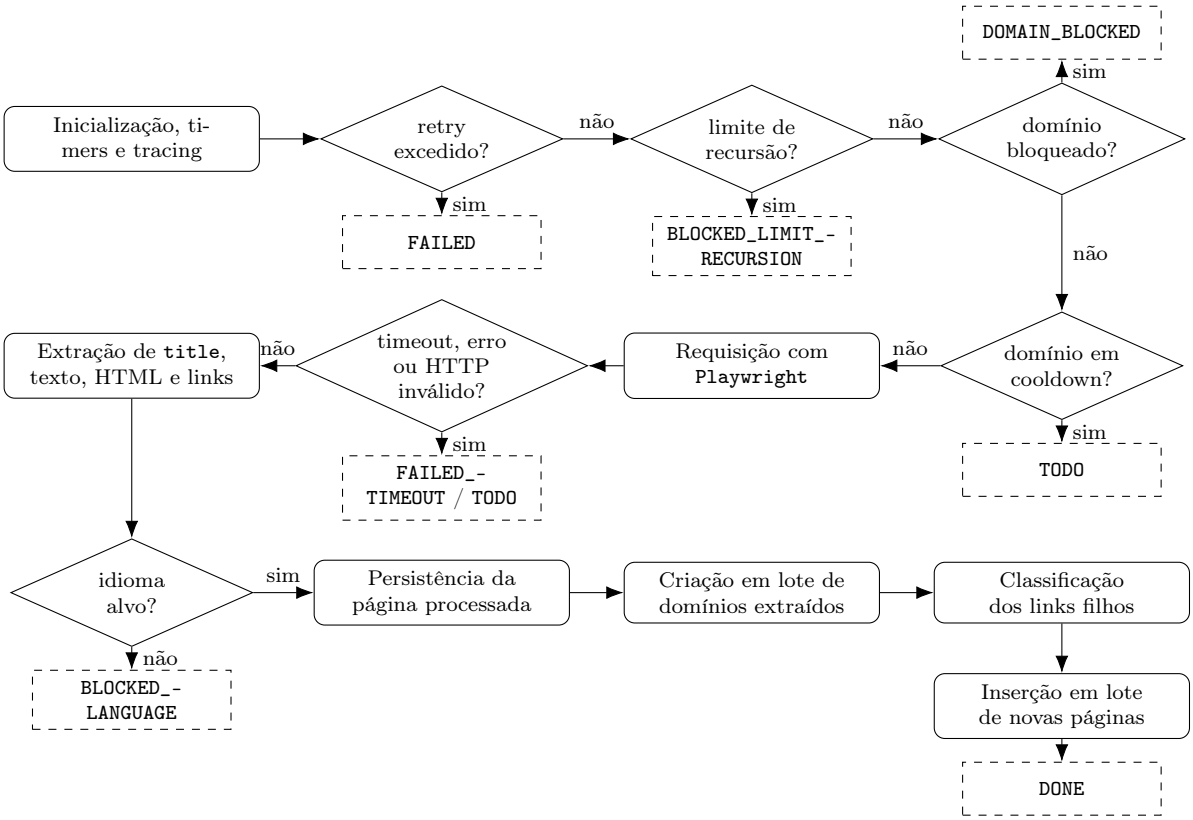



Figura 7.16: Pipeline interno do componente `Requester`.

O processamento realizado pelo `Requester` se inicia com checagens pré-requisição para definir se é possível prosseguir com a obtenção dos dados de uma determinada URL. Os primeiros atributos verificados são o limite de retentativas, o limite de recursão e se o domínio está definido como bloqueado ou em espera por requisições recentes. Essas checagens precisam ser realizadas nesse momento para evitar inconsistências decorrentes da concorrência, especialmente a execução redundante de páginas que se tornam inválidas após o momento da seleção.

Se todas as condições forem válidas, a URL selecionada é carregada com o Playwright, utilizando um navegador *headless* baseado em Chromium, permitindo a execução de conteúdo dinâmico e carregamento completo do *DOM*. Caso a requisição falhe por algum motivo, a URL é reclassificada para um estado de falha, permitindo reprocessamento ou descarte conforme a política definida. Se a requisição for concluída com sucesso, dados textuais, referências a outras páginas e o título são extraídos para serem persistidos. Com base no conteúdo da página, é possível verificar se os dados obtidos estão no idioma-alvo da execução antes de iniciar sua persistência.

Antes de salvar os dados identificados, é realizada uma reclassificação do estado da página, determinando seu status final no ciclo de processamento. Fatores como nível de recursão e idioma determinam se o estado persistido dos dados identificados já identifica que não devem ser explorados de maneira recursiva, evitando processamento desnecessário

de dados que seriam descartados posteriormente.

Com as referências a outras páginas persistidas, o sistema viabiliza a expansão incremental do grafo de exploração, mantendo explicitamente as relações de origem entre páginas e domínios.

Esses processos são executados de maneira contínua enquanto houver URLs disponíveis para processamento, garantindo que todas as URLs elegíveis sejam processadas. Essa abordagem permite a integração contínua de novas inserções na fila implícita e, ao mesmo tempo, fornece base para interrupções e retomada de aquisições já iniciadas.

O pipeline resultante apresenta propriedades de escalabilidade horizontal, tolerância a falhas e observabilidade nativa, características fundamentais para sistemas distribuídos orientados a dados. Toda a aquisição foi instrumentada com métricas e *traces*, aprofundando o entendimento e a depuração das etapas descritas.

### 7.3.5 Mecanismos de Controle

O estágio de aquisição recursivo apresenta desafios estruturais, incluindo o crescimento exponencial do grafo de URLs, a sobrecarga de domínios e a incorporação de conteúdo irrelevante. Para mitigar esses efeitos, o sistema implementa um conjunto de mecanismos de controle aplicados ao longo do fluxo de aquisição.

Inicialmente o mecanismo de controle utilizado é o limite de recursão. Esse mecanismo estabelece limites para a profundidade de exploração em nível de domínio e de página, restringindo o crescimento do grafo e garantindo a conclusão do estágio de aquisição. Seu objetivo é evitar que a geração recursiva produza infinitos registros sem um critério adequado de parada.

As URLs-semente recebem nível de recursão 0 nos campos `pages.recursion_level` e `domain.recursion_level`. A cada hiperligação percorrida, o nível da página de destino recebe o nível da página de origem acrescido de uma unidade, mesmo quando ambas pertencem ao mesmo domínio. Esse contador não reinicia quando ocorre uma transição entre domínios. O nível do domínio permanece igual ao do domínio de origem em uma navegação interna e aumenta uma unidade quando a hiperligação leva a um domínio distinto. Portanto, a navegação interna aumenta apenas o nível da página, enquanto a transição entre domínios aumenta os dois níveis.

Cada URL descoberta precisa atender aos dois limites para permanecer elegível. Como o nível do domínio não pode exceder o nível da página, quando os dois máximos são iguais, o limite de domínio não bloqueia uma URL antes do limite de página. Em uma cadeia em que cada hiperligação leva a um domínio distinto, ambos os limites podem ser atingidos na mesma etapa.

Outro mecanismo implementado é o bloqueio de domínios. Esse mecanismo busca evitar que domínios de baixa qualidade sejam explorados, aumentando a qualidade dos dados gerados e reduzindo processamento desnecessário. Como este bloqueio de domínios pode ter um grande número de registros, o sistema de persistência utiliza um índice baseado na *hash* do domínio e as instanciações da aplicação utilizam um cache local em memória (LRU) para aumentar a velocidade das avaliações.

Para definir a lista de domínios bloqueados, é utilizada uma lista já existente e vali-

dada. Por padrão, são carregados os domínios identificados pela Universidade Toulouse Capitole, da França [Université Toulouse Capitole, 2026]. Essa lista contém mais de cinco milhões de domínios categorizados que podem ser utilizados para filtrar domínios de baixa qualidade. A Tabela 7.2 apresenta a quantidade de domínios bloqueados por categoria [2].

| Motivo | Total | Motivo | Total | Motivo | Total |
|---|---|---|---|---|---|
| Adulto | 4.593.613 | Bitcoin | 1.393 | Softwares de Limpeza | 166 |
| Malware | 253.951 | Notícias Falsas | 1.094 | Roupas Adultas | 155 |
| Redirect | 132.316 | Hospedagem | 806 | Sect | 144 |
| Compras | 36.962 | Redes Sociais | 702 | Adulto Misto | 136 |
| Jogos | 33.188 | Manga | 640 | Proxy Residêncial | 107 |
| Aposta | 32.224 | Celebridades | 630 | Marketingware | 76 |
| Cryptojacking | 13.979 | Drogas | 597 | IA | 70 |
| Banco | 6.626 | Rádio | 567 | Crianças | 59 |
| Namoro | 6.183 | Stalkerware | 523 | Tricheur pix | 54 |
| Imprensa | 4.634 | Finanças | 471 | Telefones | 47 |
| Encurtador | 4.529 | Busca de Emprego | 428 | Cozinha | 37 |
| Phishing | 4.462 | DDOS | 416 | Materiais Perigosos | 35 |
| VPN | 4.420 | Webmail | 393 | Atualizações | 30 |
| Propaganda | 4.344 | Violência | 358 | Webhosting | 25 |
| Download | 3.972 | Examen pix | 299 | Educação Sexual | 12 |
| Áudio Vídeo | 3.527 | Language | $\sim$ 295 | Arjel | 12 |
| DNS | 3.013 | Liste blanche | 267 | Sites Reafetados | 7 |
| Sites Locais FR | 2.496 | Hacking | 251 | Jogos Educacionais | 6 |
| Esportes | 2.348 | Chat | 232 | Astrologia | 4 |
| DNS Dinâmico | 2.045 | Forums | 205 | Telefones 2 | 3 |
| Blog | 1.471 | Site de Tradução | 174 | Outros | 3 |
| Warez | 1.406 | Desktop Remoto | 167 | **Total** | **5.163.805** |

Tabela 7.2: Domínios bloqueados por categoria.

É possível observar que a maior parte dos domínios bloqueados se concentra em categorias relacionadas a conteúdo adulto e atividades potencialmente maliciosas. Isso indica que a lista de bloqueio atua principalmente como mecanismo de mitigação de risco e redução de ruído no processo de aquisição.

O uso da Wikipedia como fonte de URLs iniciais introduz um viés estrutural caracterizado pela geração massiva de páginas similares em idiomas distintos. Essas páginas não agregam conteúdo semântico adicional e aumentam significativamente a redundância do grafo de exploração, além de serem posteriormente descartadas pelo controle de idi-

[2]Disponível em `https://dsi.ut-capitole.fr/blacklists/index_en.php`

oma. Portanto, o bloqueio dos domínios é recomendado da própria Wikipedia em outros idiomas.

Adicionalmente, o sistema implementa controle de requisições por tempo por domínio, garantindo um intervalo mínimo entre requisições consecutivas ao mesmo host. Esse mecanismo reduz a probabilidade de sobrecarga dos servidores, evitando bloqueios.

Também é utilizado um controle de redundância, que atua em nível de URL e de conteúdo. Em nível de URL, ele é responsável por identificar URLs idênticas e evitar que obtenções repetidas ocorram. O segundo nível consiste na checagem do conteúdo obtido em busca de conteúdos idênticos. Essa checagem é especialmente importante em páginas de bloqueio, uma vez que tendem a apresentar o mesmo conteúdo.

O controle responsável pela adequação do conteúdo obtido às restrições determinadas é o mecanismo de filtro por idioma. Esse mecanismo é aplicado após a obtenção do conteúdo textual, implicando custo computacional adicional, porém necessário para garantir aderência ao idioma alvo. Páginas em outros idiomas não geram novos registros válidos para exploração, reduzindo processamento desnecessário posteriormente e compensando o custo extra durante a checagem.

Por fim, é utilizado o controle de limite de retentativa em caso de falha na obtenção de dados. Esse controle previne que a aplicação continue tentando, de maneira indefinida, obter dados de URLs que não estão mais disponíveis ou cujo tempo de entrega do conteúdo seja excessivamente grande.

Os mecanismos de controle são aplicados de forma sequencial ao longo do pipeline de aquisição, atuando como filtros progressivos que podem interromper o processamento em diferentes estágios, desde a seleção da URL até a validação do conteúdo obtido.

## 7.4 Conjunto de Dados Gerado

Esta seção caracteriza o WackyWacky PTBR Geral, uma coleção web em português brasileiro produzida pela ferramenta WackyWacky a partir de termos-semente e exploração recursiva de URLs. O objetivo é reunir textos extraídos da web, preservando metadados de obtenção e aplicando filtros de idioma, tamanho e controle de ramificação para apoiar treinamentos e análises em português brasileiro.

O restante desta seção está organizado em três partes. A Subseção 7.4.1 apresenta as características gerais do conjunto de dados, incluindo execução, configurações, termos-semente e status geral da geração. Em seguida, a Subseção 7.4.2 examina a estrutura da coleta, com foco nas URLs exploradas, domínios e níveis de recursão. Por fim, a Subseção 7.4.3 discute a qualidade inicial dos textos, considerando tamanho, frequências lexicais e bigramas.

### 7.4.1 Características Gerais do Conjunto de Dados

A execução foi iniciada em 2026-03-14. O recorte analisado nesta seção corresponde à exportação realizada em 2026-08-21, quando a coleta ainda estava em execução. A coleta utilizou termos-semente e a Wikipédia como mecanismo inicial de busca. Os agregados apresentados descrevem as URLs e os domínios registrados no momento da exportação,

mas não medem a velocidade de crescimento da coleta nem isolam o efeito da seleção dos termos-semente ou dos parâmetros operacionais.

A construção desse material adota uma estratégia recursiva: cada URL encontrada pode levar à identificação de novos endereços e domínios, posteriormente incorporados à fila da ferramenta. Todos os resultados reportados correspondem a um recorte intermediário da execução e, portanto, descrevem o estado da coleta no momento da exportação, não sua distribuição final.

A Tabela 7.3 apresenta as configurações utilizadas neste processo de geração.

| Parâmetro | Valor |
|---|---|
| Tempo limite da requisição (ms) | 10.000 |
| Número máximo de tentativas | 3 |
| Intervalo entre tentativas (ms) | 60.000 |
| Intervalo entre requisições por domínio (ms) | 500 |
| Profundidade máxima de recursão por domínio | 4 |
| Profundidade máxima de recursão por página | 4 |
| Mecanismo de busca | StarterWikipedia |
| Total de Seeds | 318 |

Tabela 7.3: Configurações de Execução.

As configurações adotadas contemplam três dimensões operacionais: tolerância a falhas transitórias, controle de carga por domínio e limitação da profundidade recursiva. O tempo limite de 10.000 ms por requisição estabelece um limite de espera por resposta, enquanto o limite de três tentativas prevê novas tentativas em caso de falha. O intervalo de 60.000 ms entre tentativas busca espaçar os acessos após falhas transitórias ou limitações de acesso, sem garantir a recuperação da requisição.

Entre requisições para um mesmo domínio, o intervalo de 500 ms introduz uma pausa mínima entre acessos sucessivos. Essa pausa busca reduzir a carga de acessos por domínio. Já as profundidades máximas de recursão, ambas definidas como 4, limitam a profundidade da exploração recursiva a partir dos endereços encontrados. Essa restrição não estabelece, por si só, um limite para o número de URLs descobertas em cada nível nem garante a viabilidade computacional do processo.

O mecanismo StarterWikipedia utiliza a enciclopédia livre como fonte para buscar os termos-semente definidos para a execução, representados pelo parâmetro Total de seeds, com 318 itens. Esses termos orientam as consultas iniciais, mas sua seleção não permite determinar, por si só, a cobertura temática dos textos obtidos durante a exploração recursiva. Os termos-semente utilizados nesta execução estão disponíveis no Apêndice A.

A Figura 7.17 apresenta uma visualização tridimensional dos termos-semente utilizados. Para a construção do gráfico, os termos foram representados por meio de *embeddings* e, em seguida, reduzidos para três dimensões utilizando PCA. Os grupos foram definidos por K-Means apenas para fins exploratórios de visualização, utilizando oito clusters. Os *embeddings* foram gerados com o modelo `BAAI/bge-m3`.

A projeção apresenta grupos com diferentes graus de compactação e separação entre os termos-semente. A interpretação desses agrupamentos deve considerar a perda de

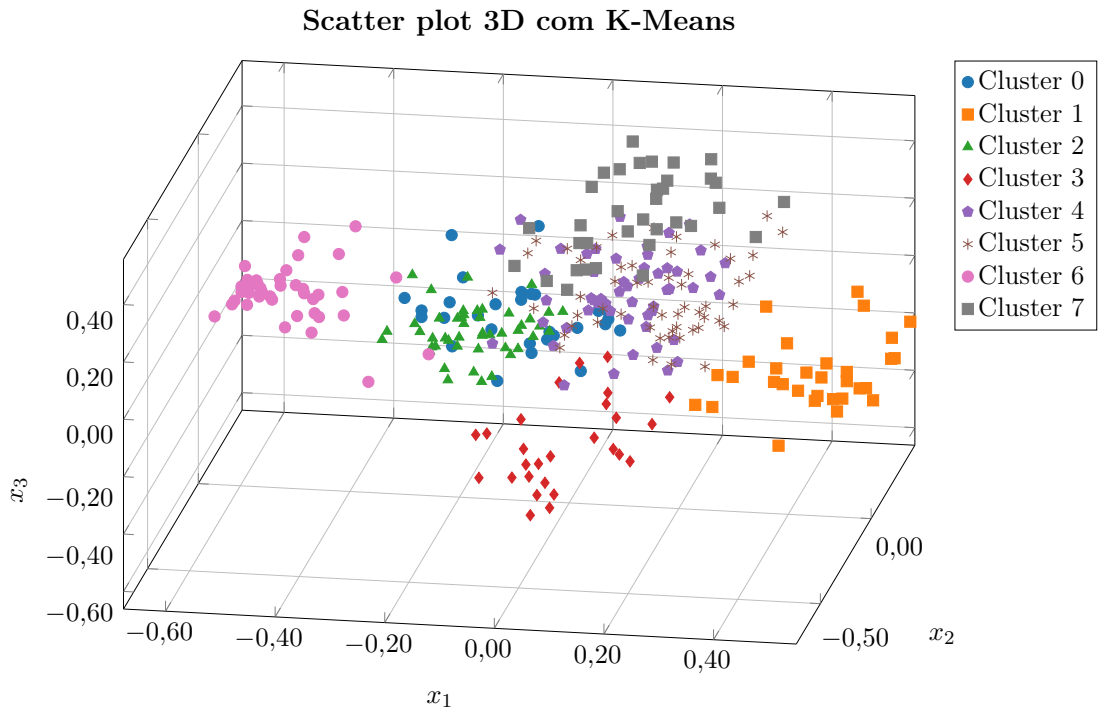



Figura 7.17: Distribuição tridimensional dos termos-semente, agrupados por K-Means.

informação introduzida pelo PCA e a natureza particional do K-Means. Essa visualização caracteriza a distribuição das representações das sementes no espaço projetado, mas não mede, por si só, a diversidade temática dos textos efetivamente coletados nem permite inferir a cobertura alcançada durante a exploração recursiva.

A lista a seguir descreve os possíveis estados atribuídos às URLs durante a geração do conjunto de dados.

- **Concluído**: corresponde a URLs obtidas e processadas pelo coletor. Inclui também aquelas identificadas como repetidas e vinculadas a um registro existente, sem armazenar novamente o conteúdo.
- **Pendente**: identifica URLs que aguardam processamento.
- **Em processamento**: identifica URLs registradas como reservadas para processamento por um trabalhador no estado exportado.
- **Bloqueado por limite de recursão**: reúne URLs cuja exploração foi impedida por atingir ou ultrapassar os limites de profundidade definidos para páginas ou domínios.
- **Bloqueado por domínio**: representa URLs pertencentes a domínios incluídos na lista de bloqueio do coletor.
- **Bloqueado por idioma**: reúne URLs cujo conteúdo não foi reconhecido pelo detector como pertencente aos idiomas-alvo, incluindo situações em que não foi possível identificar o idioma.

- **Falha**: registra falhas de acesso ou processamento, incluindo situações em que o limite de tentativas foi atingido.
- **Falha por tempo limite**: identifica falhas registradas pelo coletor em decorrência da expiração do tempo permitido para o acesso à página.

Até a exportação, haviam sido encontradas 83.590.843 URLs, cuja distribuição por status é apresentada na Tabela 7.4.

| Status | Total | Percentual |
|---|---|---|
| Em processamento | 11 | <0,10% |
| Falha | 127.216 | 0,15% |
| Falha por tempo limite | 192.475 | 0,23% |
| Bloqueado por idioma | 823.902 | 0,99% |
| Bloqueado por domínio | 964.887 | 1,15% |
| Concluído | 5.285.496 | 6,32% |
| Pendente | 21.939.707 | 26,25% |
| Bloqueado por limite de recursão | 54.257.149 | 64,91% |
| Total | 83.590.843 | 100,00% |

Tabela 7.4: Distribuição de URLs por Status

No momento da exportação, o estado Bloqueado por limite de recursão concentrava 54.257.149 URLs, equivalentes a 64,91% do total, enquanto aproximadamente 26,25% dos registros permaneciam pendentes. Juntos, esses estados representam cerca de 91,16% do inventário, mas refletem situações distintas: URLs cuja exploração foi impedida pelos limites configurados e URLs que ainda aguardavam processamento. A predominância dos bloqueios por recursão evidencia o alcance dessa restrição sobre os endereços descobertos, sem indicar indisponibilidade das páginas ou inadequação de seu conteúdo.

O estado Concluído reúne 5.285.496 registros, correspondentes a 6,32% do inventário. Como inclui ocorrências repetidas vinculadas a registros existentes, esse total não corresponde ao número de documentos únicos nem garante texto válido em todos os registros. Já os 11 registros Em processamento representam apenas as reservas de trabalho no instante da exportação, não uma medida da capacidade de processamento da coleta.

Os bloqueios por domínio abrangem 964.887 URLs, correspondentes a 1,15% do total, enquanto os bloqueios por idioma reúnem 823.902 URLs, equivalentes a 0,99%. Embora menos frequentes que os bloqueios por recursão, esses estados mostram a atuação de outros critérios de restrição: a lista de domínios bloqueados e a identificação dos idiomas-alvo. No segundo caso, estão incluídos conteúdos cujo idioma não pôde ser identificado, de modo que o total não representa exclusivamente páginas reconhecidas como pertencentes a outros idiomas.

Por fim, o estado Falha reúne 127.216 ocorrências, enquanto Falha por tempo limite registra 192.475, correspondendo, respectivamente, a 0,15% e 0,23% do inventário. As expirações de tempo predominam entre as falhas, mas ambas as categorias são menos frequentes que os bloqueios por domínio e idioma. Esses percentuais descrevem os estados do inventário completo, que inclui URLs não acessadas, e não taxas de sucesso ou falha das

requisições. A subseção seguinte examina um recorte que exclui os registros pendentes, em processamento e bloqueados por limite de recursão.

### 7.4.2 Estrutura e Distribuições da Coleta

No inventário completo da coleta, foram identificadas 2.658.962 ocorrências de URLs repetidas, sinalizadas pelo campo `same_as` por corresponderem a endereços já registrados. Esse comportamento é esperado em um rastreamento recursivo, pois um mesmo endereço pode ser encontrado a partir de diferentes termos-semente, domínios ou caminhos de navegação. A identificação dessas ocorrências permite distinguir novos endereços daqueles já registrados.

Tomando os 2.645.420 registros com texto válido como subconjunto dos 5.285.496 registros Concluídos, restam 2.640.076 registros concluídos sem texto válido. As 2.658.962 ocorrências de `same_as`, porém, foram contabilizadas no inventário completo e não estão discriminadas por status ou validade do texto. A soma de textos válidos e referências `same_as` é 5.304.382, valor 18.886 acima do total Concluído.

A Tabela 7.5 apresenta a distribuição de 7.393.976 registros de URLs, desconsiderando aqueles pendentes, em processamento ou bloqueados por limite de recursão. O estado Concluído representa 71,48% desse recorte. Como a seleção inclui bloqueios que podem ocorrer sem acesso direto à página, os percentuais descrevem os estados registrados pelo coletor, e não uma taxa de sucesso das requisições HTTP. A disponibilidade de conteúdo textual válido é verificada separadamente nos textos exportados.

| Status | Total | Percentual |
|---|---|---|
| Falha | 127.216 | 1,72% |
| Falha por tempo limite | 192.475 | 2,60% |
| Bloqueado por idioma | 823.902 | 11,14% |
| Bloqueado por domínio | 964.887 | 13,05% |
| Concluído | 5.285.496 | 71,48% |

Tabela 7.5: Distribuição dos estados selecionados da coleta.

No recorte apresentado, os bloqueios por domínio e idioma somam 24,19% dos registros, enquanto as falhas gerais e por tempo limite correspondem a 4,32%. Os estados de bloqueio são, portanto, mais frequentes que os de falha. Essa comparação descreve a classificação das URLs no snapshot, incluindo bloqueios que podem ocorrer sem uma tentativa direta de acesso à página.

A Tabela 7.6 apresenta os dez domínios com maior contribuição ao conteúdo textual deduplicado, acompanhados pelo número de requisições, textos válidos e volume de palavras. A coluna “Textos válidos” contabiliza os registros concluídos com texto disponível, decodificado e não vazio, antes da limpeza e deduplicação. Ambas as colunas de palavras utilizam o texto após a limpeza. A coluna “Volume coletado” contabiliza as palavras em cada ocorrência de página, incluindo conteúdos repetidos. Já “Contribuição ao conteúdo deduplicado” considera cada conteúdo apenas uma vez e, quando ele ocorre em mais de um domínio, divide suas palavras igualmente entre os domínios envolvidos.

O domínio `pt.wikipedia.org` concentra 5.977.393 requisições, correspondentes a 65,32% das 9.151.363 requisições registradas, e 2.104.784 textos válidos, equivalentes a 79,60% do total. Seu volume coletado alcança 2,42 bilhões de palavras, enquanto sua contribuição ao conteúdo deduplicado é de 2,40 bilhões, ou 88,20% das 2,72 bilhões de palavras do corpus final. Em conjunto, os dez domínios apresentados concentram 72,30% das requisições, 85,60% dos textos válidos e 91,40% do conteúdo deduplicado. A predominância observada da Wikipédia e de outros projetos Wikimedia é compatível com a utilização da Wikipédia como ponto inicial da coleta, embora também estejam presentes fontes externas, como `planalto.gov.br`, `web.archive.org` e `scholar.googleusercontent.com`.

| Host | Requisições | Textos Válidos | Volume coletado (palavras) | Contribuição ao conteúdo deduplicado (palavras) |
|---|---|---|---|---|
| pt.wikipedia.org | 5.977.393 | 2.104.784 | 2.422.645.886 | 2.401.010.397 |
| pt.wikiversity.org | 166.275 | 27.569 | 16.175.921 | 15.060.740 |
| pt.wikisource.org | 55.229 | 21.308 | 15.935.940 | 14.493.009 |
| planalto.gov.br | 18.146 | 6.858 | 15.207.105 | 13.373.498 |
| pt.wikibooks.org | 85.163 | 23.929 | 13.500.412 | 12.398.211 |
| pt.wikinews.org | 41.670 | 21.208 | 9.623.144 | 9.343.747 |
| pt.wikiquote.org | 41.288 | 14.672 | 6.530.396 | 6.017.740 |
| web.archive.org | 133.586 | 5.109 | 6.044.324 | 5.958.161 |
| pt.wiktionary.org | 96.517 | 37.587 | 6.092.997 | 5.849.510 |
| scholar. googleusercontent.com | 1.532 | 651 | 4.870.242 | 4.861.132 |

Tabela 7.6: Principais domínios por requisições, textos válidos e volume textual.

As duas medidas contabilizam elementos diferentes: a soma de `domain.request_count` é de 9.151.363 requisições, e 7.393.976 é o total de URLs em estados terminais. A diferença é de 1.757.387. Como o contador está agregado por domínio, não é possível determinar quantas requisições correspondem a tentativas repetidas, redirecionamentos ou outras operações da coleta.

A implementação do componente `Starter` oferece Bing, DuckDuckGo, Google e Wikipedia, mas a configuração desta execução utilizou apenas `StarterWikipedia` e 318 termos-semente. Como os termos-semente partem de `pt.wikipedia.org`, há um único domínio no nível 0.

Os sete domínios Wikimedia listados na Tabela 7.6 somam cerca de 2,46 bilhões de palavras. Os 616.422 domínios não Wikimedia demonstram amplitude de descoberta, mas sua contribuição conjunta ainda fica abaixo de 9,5% no recorte parcial avaliado. Termos dos rankings, como *histórico* e *discussão*, e bigramas como *afluentes editar* e *bytes desfazer*, são compatíveis com páginas editoriais ou utilitárias, mas não permitem estimar a proporção desses documentos.

A Tabela 7.7 apresenta os dez domínios com maior contribuição ao conteúdo textual deduplicado entre aqueles que não pertencem à lista explícita do ecossistema Wikimedia. As métricas seguem as mesmas definições da tabela anterior. Esse recorte permite

examinar as demais fontes do corpus, cuja participação fica menos visível diante da predominância da Wikipédia.

Nesse grupo, o domínio `planalto.gov.br` apresenta a maior contribuição, com aproximadamente 13,37 milhões de palavras, equivalentes a 5,30% das palavras atribuídas aos domínios não Wikimedia. A tabela reúne fontes governamentais, serviços de arquivamento, plataformas acadêmicas, mecanismos de busca e blogs. Essa diversidade mostra que o corpus inclui conteúdos de diferentes tipos de fonte, além dos projetos Wikimedia. Contudo, a contribuição textual não acompanha necessariamente o número de requisições: `web.archive.org`, por exemplo, registra mais requisições que `planalto.gov.br`, mas contribui com um volume menor de conteúdo deduplicado.

| Host | Requisições | Textos válidos | Volume coletado (palavras) | Contribuição ao conteúdo deduplicado (palavras) |
|---|---|---|---|---|
| planalto.gov.br | 18.146 | 6.858 | 15.207.105 | 13.373.498 |
| web.archive.org | 133.586 | 5.109 | 6.044.324 | 5.958.161 |
| scholar.<br>googleusercontent.com | 1.532 | 651 | 4.870.242 | 4.861.132 |
| gov.br | 28.344 | 11.399 | 4.926.637 | 4.851.581 |
| google.com | 47.858 | 13.641 | 4.897.704 | 4.843.713 |
| famososquepartiram.com | 1.986 | 1.466 | 3.549.200 | 3.549.200 |
| periodicos.capes.gov.br | 14.751 | 3.817 | 3.189.272 | 3.189.144 |
| istoesergipe.<br>blogspot.com | 2.020 | 1.908 | 2.871.483 | 2.871.219 |
| dx.doi.org | 43.522 | 926 | 2.120.549 | 2.099.208 |
| edisciplinas.usp.br | 1.020 | 685 | 2.049.475 | 2.012.582 |

Tabela 7.7: Dez domínios não Wikimedia com maior contribuição ao conteúdo textual deduplicado.

A Tabela 7.8 apresenta os domínios com maior número de filhos diretos na relação de descoberta registrada pelo coletor. Nesse contexto, um domínio filho é aquele cujo `parent_domain_id` aponta para o domínio apresentado, não implicando relação de propriedade, subdomínio ou afinidade temática. Dos 616.429 domínios distintos do snapshot, 616.428 possuem um domínio pai registrado e apenas o domínio inicial não possui pai.

O domínio `pt.wikipedia.org` originou diretamente a descoberta de 445.727 domínios, equivalentes a 72,31% de todos os domínios do snapshot. Os quinze domínios apresentados na tabela concentram 483.076 filhos, ou 78,37% do total, indicando forte concentração da expansão em poucos pontos de origem. A predominância da Wikipédia é coerente com sua utilização como ponto inicial da coleta, mas essa relação estrutural não permite determinar, por si só, o perfil temático ou a relevância textual das fontes descobertas.

A média de requisições dos filhos complementa a contagem de filhos ao distinguir amplitude de descoberta e intensidade de acesso. Os filhos de `pt.wikipedia.org` apresentam média de 4,54 requisições por domínio, enquanto os associados a `google.com` e `web.archive.org` apresentam médias de 0,63 e 0,77. Em sentido oposto, `pt.wikibooks.org`, `pt.wikiquote.org` e `pt.wiktionary.org` possuem menos filhos, mas médias de 70,40, 129,06 e 111,76 requisições, respectivamente. Assim, um domínio pode originar

uma expansão ampla e pouco explorada ou uma expansão menor com maior volume médio de acesso. Como a média inclui filhos com zero requisições e pode ser influenciada por poucos valores elevados, ela deve ser interpretada como medida agregada, e não como comportamento típico de todos os filhos.

| Host | Filhos | Percentual (%) | Média de requisições dos filhos |
|---|---|---|---|
| pt.wikipedia.org | 445.727 | 72,31 | 4,54 |
| google.com | 8.728 | 1,42 | 0,63 |
| web.archive.org | 7.550 | 1,22 | 0,77 |
| www2.unifap.br | 4.962 | 0,80 | 0,01 |
| pt.wikiversity.org | 3.321 | 0,54 | 25,17 |
| pt.wikivoyage.org | 2.139 | 0,35 | 0,59 |
| pt.wikibooks.org | 1.619 | 0,26 | 70,40 |
| gov.br | 1.531 | 0,25 | 0,20 |
| catolicosnaciencia.org.br | 1.453 | 0,24 | 0,01 |
| pt.wikiquote.org | 1.252 | 0,20 | 129,06 |
| aquidauananews.com | 1.122 | 0,18 | 0,00 |
| pt.wiktionary.org | 969 | 0,16 | 111,76 |
| al.sp.gov.br | 945 | 0,15 | 0,32 |
| vivaotux.blogspot.com | 897 | 0,15 | 0,03 |
| pedradosertao.blogspot.com | 861 | 0,14 | 0,00 |

Tabela 7.8: Domínios com maior número de filhos diretos na relação de descoberta e média de requisições desses filhos.

A Tabela 7.9 apresenta a distribuição dos 616.429 domínios e das 9.151.363 requisições segundo o nível de recursão do domínio. O nível 0 contém somente o domínio inicial, mas concentra 5.977.393 requisições, correspondentes a 65,32% do total. O nível 1 reúne 445.727 domínios, ou 72,31%, e 2.025.474 requisições, equivalentes a 22,13%. Em conjunto, os níveis 0 e 1 concentram 87,45% das requisições do snapshot.

A igualdade entre os 445.727 domínios do nível 1 e os 445.727 filhos de `pt.wikipedia.org` confirma a consistência entre as duas perspectivas: a primeira tabela agrupa os filhos pelo domínio que originou sua descoberta, enquanto a segunda os organiza pela profundidade alcançada. Isso não significa, entretanto, que as requisições atribuídas ao nível 0 sejam as mesmas contabilizadas na média dos filhos de `pt.wikipedia.org`; no primeiro caso são requisições do domínio inicial, e no segundo são requisições dos domínios descobertos a partir dele.

A distribuição apresentada nesta tabela corresponde ao campo `domain.recursion_level`; ela não apresenta a profundidade registrada para cada página. Como os dois limites foram configurados com o valor 4 e o nível do domínio não excede o nível da página, a verificação do limite de domínio não bloqueia uma URL antes da verificação do limite de página. A profundidade de página contabiliza todos os saltos e pode limitar novas descobertas antes de o nível de domínio chegar a 4. Assim, a distribuição da Tabela 7.9 conta apenas os domínios registrados segundo as transições interdomínios alcançadas, e não representa a distribuição dos níveis das páginas. A Tabela 7.4 agrega os bloqueios

por limite de recursão sem identificar qual condição foi atingida, inclusive quando os dois limites são alcançados na mesma etapa.

| Nível | Domínios | Domínios (%) | Requisições | Requisições (%) | Média de requisições por domínio |
|---|---|---|---|---|---|
| 0 | 1 | 0,00 | 5.977.393 | 65,32 | 5.977.393,00 |
| 1 | 445.727 | 72,31 | 2.025.474 | 22,13 | 4,54 |
| 2 | 104.726 | 16,99 | 1.103.935 | 12,06 | 10,54 |
| 3 | 62.922 | 10,21 | 44.561 | 0,49 | 0,71 |
| 4 | 3.053 | 0,50 | 0 | 0,00 | 0,00 |

Tabela 7.9: Distribuição dos domínios e das requisições por nível de recursão do domínio no snapshot analisado.

Os níveis 2, 3 e 4 concentram conjuntamente 27,69% dos domínios, mas apenas 12,55% das requisições. O nível 2 apresenta média de 10,54 requisições por domínio, superior à média de 4,54 observada no nível 1, demonstrando que a intensidade de acesso não diminui de maneira monotônica com a profundidade. Já o nível 3 reúne 10,21% dos domínios e somente 0,49% das requisições, com média de 0,71. No nível 4, foram registrados 3.053 domínios sem requisições associadas. Esses resultados mostram que a descoberta de um domínio e sua efetiva requisição são eventos distintos. A ausência de requisições no último nível caracteriza o estado do snapshot analisado, mas não permite identificar isoladamente sua causa.

### 7.4.3 Qualidade Inicial dos Dados Textuais

Até este ponto, a análise caracterizou a coleta em termos de estados das URLs, domínios de origem e níveis de recursão. A Tabela 7.10 sintetiza os totais da coleta e do conteúdo textual disponível, incluindo as dimensões do conjunto após limpeza e deduplicação. A partir desses totais, a análise passa à distribuição dos tamanhos dos documentos e aos padrões lexicais mais frequentes.

| Medida | Total |
|---|---|
| Registros de URLs inventariados | 83.590.843 |
| Registros com estado Concluído | 5.285.496 |
| Registros com texto disponível e válido | 2.645.420 |
| Documentos após limpeza e deduplicação | 2.615.088 |
| Palavras no conjunto final | 2.721.747.727 |
| Caracteres no conjunto final | 22.372.730.395 |

Tabela 7.10: Síntese quantitativa da coleta e do conjunto textual analisado. As contagens de palavras e caracteres referem-se aos documentos após limpeza e deduplicação.

A Figura 7.18 apresenta a distribuição do tamanho, em caracteres. Para facilitar a visualização da região de maior concentração, o histograma foi limitado ao intervalo entre

2 e 9.999 caracteres. Os documentos fora dessa faixa permanecem no corpus, mas não são exibidos na figura.

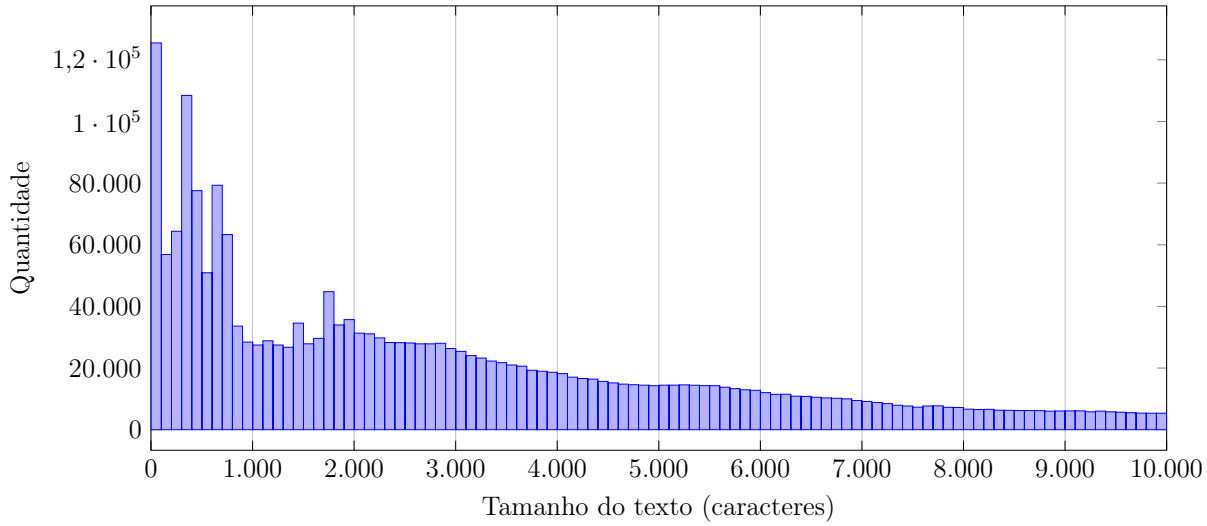



Figura 7.18: Distribuição dos tamanhos dos textos obtidos.

A Figura 7.19 apresenta a cobertura acumulada das ocorrências de palavras pelas formas observadas e pelos lemas mais frequentes. Para cada quantidade $K$, somam-se as frequências dos $K$ termos mais frequentes e divide-se esse valor pelo total de ocorrências de palavras do conjunto final. Diferentemente dos rankings anteriores, esta análise inclui as stopwords e não aplica o filtro visual de itens de interface.

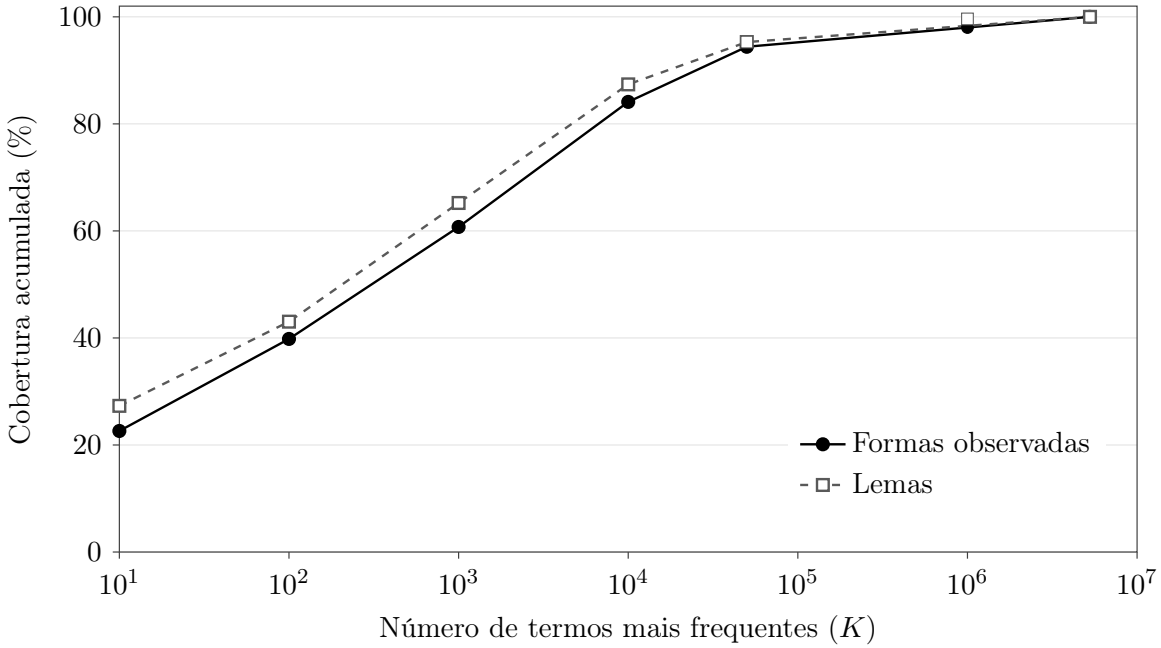



Figura 7.19: Cobertura acumulada das ocorrências de palavras pelos termos mais frequentes no conjunto final.

Observa-se elevada concentração das ocorrências em uma parcela pequena do vocabulário: as 1.000 formas mais frequentes respondem por 60,74% das ocorrências, enquanto

as 10.000 e as 50.000 mais frequentes cobrem 84,11% e 94,41%, respectivamente. Essas 50.000 formas representam apenas 0,95% das 5.258.405 formas distintas observadas. O resultado caracteriza a concentração lexical do conjunto, sem permitir inferir, isoladamente, sua diversidade temática ou a qualidade dos textos.

As frequências lexicais apresentadas nas Figuras 7.20 e 7.21 foram calculadas sobre o corpus final. A tokenização utilizou spaCy 3.8.15 com o modelo português `pt_core_news_sm` 3.8.0. Foram considerados como palavras os tokens alfabéticos, normalizados em Unicode NFC e por `casefold`, preservando os acentos.

As stopwords da configuração portuguesa foram omitidas apenas da apresentação dos rankings, permanecendo nas contagens. Além das stopwords portuguesas, a apresentação dos rankings de palavras e bigramas omite itens de interface definidos por uma lista explícita. Esse filtro atua somente na apresentação dos resultados, sem alterar o corpus nem as frequências calculadas. Não houve exclusão geral de tokens com um caractere nem aplicação de uma lista de stopwords em inglês.

Os rankings apresentam formas observadas, sem agrupamento por lema. O número entre parênteses ao lado de cada palavra indica a quantidade de documentos em que ela aparece.

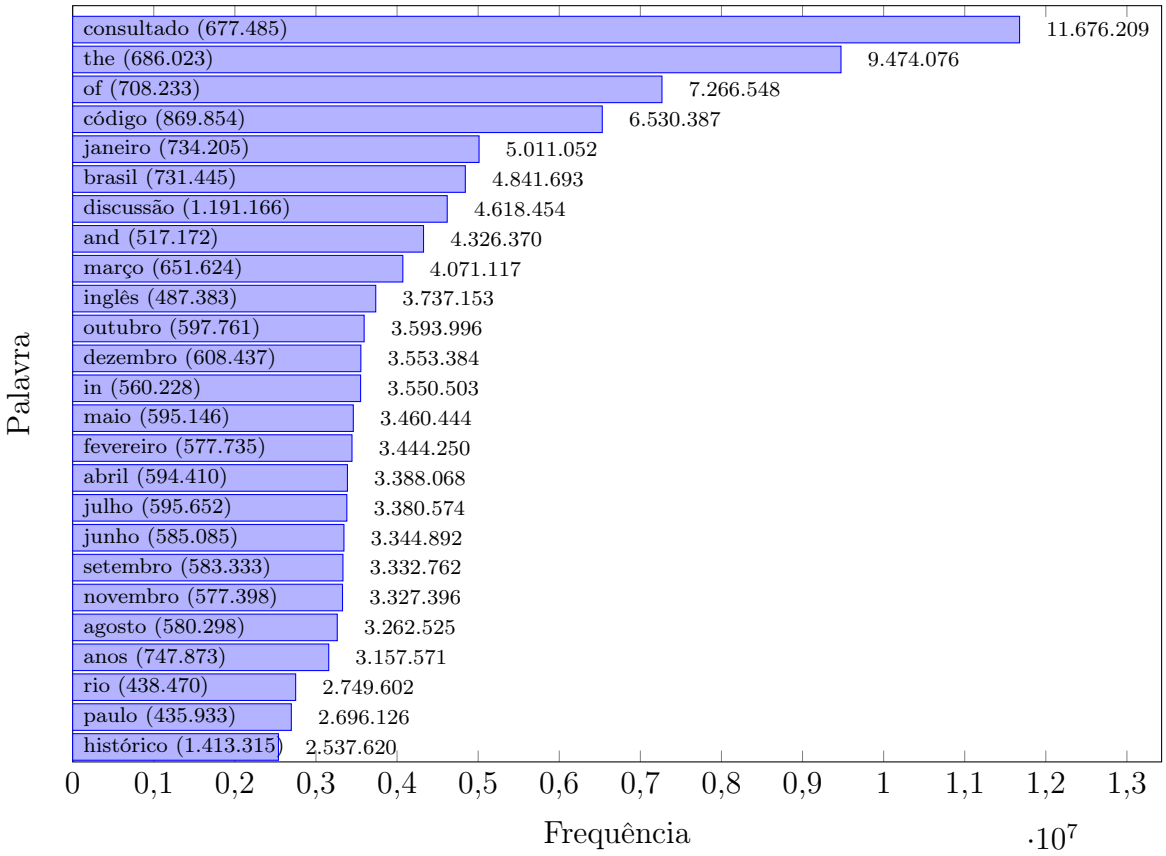



Figura 7.20: Palavras mais frequentes da coleção *WackyWacky*, excluindo stopwords portuguesas.

Entre as palavras mais frequentes, *consultado* ocorre 11.676.209 vezes. Também aparecem *discussão* (4.618.454) e *histórico* (2.537.620), compatíveis com referências e elementos editoriais recorrentes. Palavras funcionais do inglês também figuram no ranking: *the*

ocorre 9.474.076 vezes em 686.023 documentos; *of*, 7.267.548 em 708.233; *and*, 4.326.370 em 517.172; e *in*, 3.550.503 em 560.228. Esses dados podem refletir conteúdo em inglês ou fragmentos de interface e código, mas não permitem atribuir todas as ocorrências a ruído. Como o ranking é calculado após limpeza e deduplicação, os padrões indicam resíduos no corpus final, sem quantificar a proporção de documentos ou de texto afetados.

A Figura 7.21 complementa essa análise apresentando os bigramas mais frequentes do conjunto final analisado. Os pares foram formados por adjacência na sequência de tokens alfabéticos de cada parágrafo, mantendo as stopwords durante a formação e a contagem. Para apresentação, foram excluídos os pares que continham stopwords portuguesas. Como pontuação e tokens não alfabéticos não integram essa sequência, os bigramas não correspondem necessariamente a trechos contíguos do texto original.

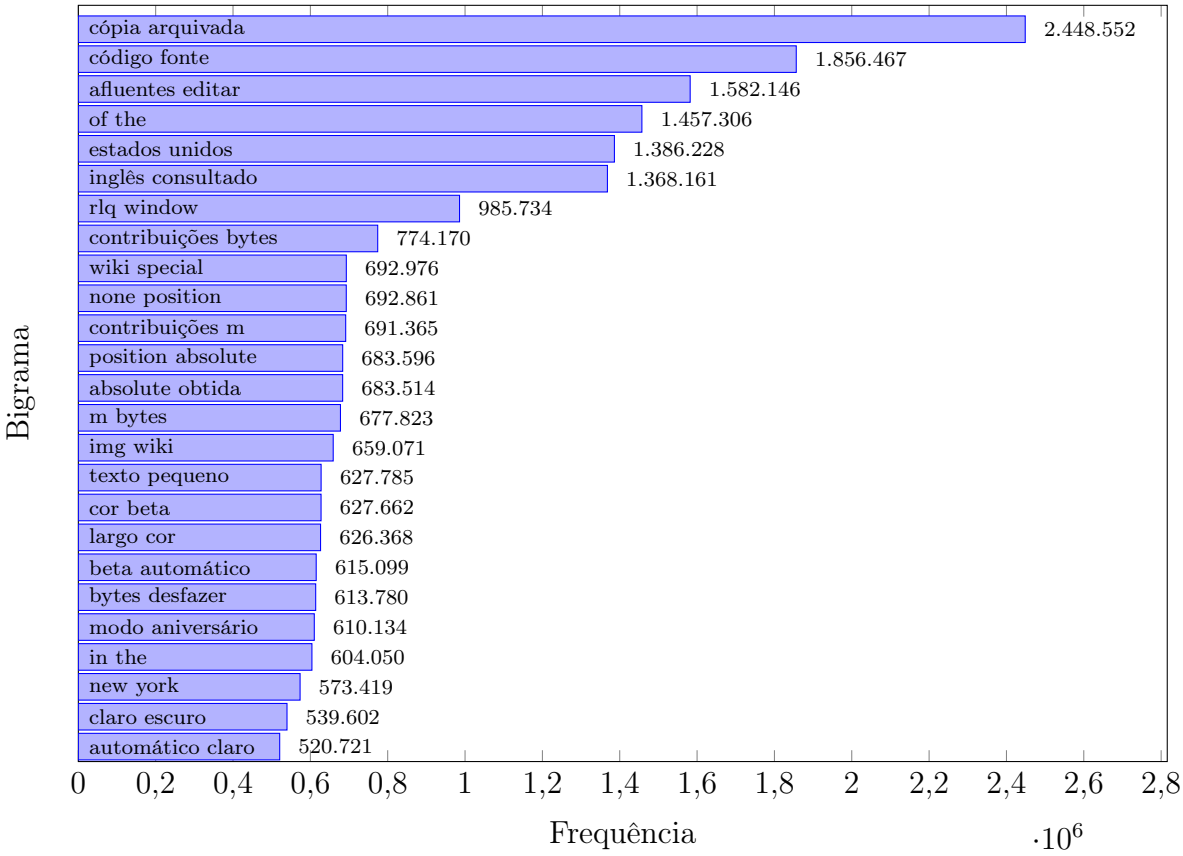



Figura 7.21: Bigramas mais frequentes da coleção *WackyWacky*, excluindo stopwords portuguesas.

Os bigramas mais frequentes incluem padrões compatíveis com resíduos de JavaScript, CSS e interface do MediaWiki. Entre eles, *rlq window* aparece 985.734 vezes e ocupa a sétima posição geral; também se destacam *contribuições bytes* (774.170), *wiki special* (692.976), *none position* (692.861), *contribuições m* (691.365), *position absolute* (683.596), *absolute obtida* (683.514), *m bytes* (677.823), *img wiki* (659.071), *texto pequeno* (627.785), *cor beta* (627.662), *largo cor* (626.368), *beta automático* (615.099), *bytes desfazer* (613.780) e *modo aniversário* (610.134). A recorrência desses pares evidencia resíduos não triviais no corpus final; contudo, as frequências contam ocorrências, não documentos distintos, e não estimam a fração de texto não substantivo. Como a tokenização também desconsidera

pontuação, os bigramas não comprovam, por si sós, sua origem exata no HTML nem a etapa do processamento em que foram preservados.

Esses resultados identificam uma limitação concreta da limpeza: a persistência de artefatos de scripts, estilos e interface no corpus final. Uma melhoria futura deve avaliar a remoção, no DOM e antes da extração textual, de elementos não substantivos como `script`, `style`, `noscript`, `nav` e `footer`, além de filtros de URL específicos para páginas utilitárias do MediaWiki. A comparação antes e depois da limpeza, acompanhada de auditoria de uma amostra, permitirá avaliar a redução de ruído sem descartar conteúdo legítimo.

O ranking também contém referências geográficas, como *estados unidos* e *new york*, que ilustram a necessidade de cautela na remoção de expressões frequentes. Já expressões como *cópia arquivada* dependem do contexto de suas ocorrências para serem classificadas como conteúdo principal, referência ou ruído.

Em síntese, o WackyWacky PTBR Geral reúne 83.590.843 registros de URLs no recorte analisado, dos quais 2.645.420 possuem texto disponível e válido. Após limpeza e deduplicação, o conjunto final analisado contém 2.615.088 documentos e 2.721.747.727 palavras. Esses valores caracterizam o estado parcial da coleta no momento da exportação.

Os resultados mostram forte concentração das requisições e da descoberta de domínios nos projetos Wikimedia, compatível com a utilização da Wikipédia como ponto inicial. No recorte de estados selecionados, os bloqueios por domínio e idioma são mais frequentes que as falhas gerais e por tempo limite. Essa distribuição descreve os estados atribuídos pelo coletor, sem determinar a taxa de sucesso das requisições HTTP ou a qualidade dos textos disponíveis.

Do ponto de vista do conteúdo, a distribuição dos tamanhos dos textos confirma a heterogeneidade esperada em uma coleção web ampla, combinando documentos curtos, médios e uma cauda de itens mais extensos. Já os rankings lexicais e, sobretudo, os bigramas evidenciam a persistência de fragmentos frequentes de scripts, estilos e interface do MediaWiki no corpus após a limpeza e a deduplicação. Como essas frequências não medem diretamente a proporção de documentos ou de texto afetado, a qualidade relativa do conjunto não pode ser quantificada apenas por esses rankings, reforçando a necessidade de avaliar refinamentos na extração DOM e na filtragem de URLs.

## 7.5 Limitações

A metodologia proposta apresenta limitações decorrentes de decisões de projeto e de restrições inerentes ao ambiente web, as quais impactam diretamente a cobertura, a distribuição e o viés dos dados coletados.

A lista a seguir apresenta as principais limitações identificadas na metodologia e na ferramenta propostas.

- **Mecanismos anti-automação na web**: A presença de *CAPTCHAs*, bloqueios por comportamento suspeito e técnicas de *fingerprinting* pode limitar a coleta em diversos domínios.

- **Dependência da estrutura de hiperlinks**: A exploração baseada em hiperlinks induz um viés estrutural, limitando a cobertura a componentes conectados do grafo web e excluindo conteúdos isolados ou com poucas conexões. A cobertura ainda depende da alcançabilidade a partir dos pontos de inicialização, podendo excluir regiões do grafo web não acessíveis pelos caminhos gerados pelos termos iniciais.
- **Custo computacional elevado de navegação**: O uso de navegação baseada em navegador impõe latências por requisição na ordem de segundos, limitando a taxa de coleta e aumentando o custo computacional.
- **Viés na inicialização do processo**: O conjunto inicial de termos utilizados influencia diretamente os resultados obtidos, podendo impor limitações quando não for bem definido para cobrir os domínios desejados.
- **Concentração por domínio e identificação de URLs**: A execução analisada usou apenas `StarterWikipedia`. O intervalo entre requisições por domínio limita a frequência de acesso, mas não o total de páginas por domínio. Os filtros podem reduzir páginas vedadas, repetidas ou utilitárias. Já os limites por domínio e a alternância da fila reservam parte do orçamento para outras fontes, reduzindo o risco de um domínio de alta ramificação dominar a coleta.
- **Resíduos de scripts, estilos e interface**: Os rankings das Figuras 7.20 e 7.21 mostram a persistência de fragmentos associados a código, CSS e interface do MediaWiki. As frequências observadas não permitem estimar, por si sós, a proporção de documentos afetados. Como trabalho futuro, recomenda-se remover do DOM elementos não substantivos, como `script`, `style`, `noscript`, `nav` e `footer`, e aplicar filtros de namespaces e parâmetros de URL apenas a domínios MediaWiki, incluindo `Special:`, `Especial:`, `Talk:`, `Discussão:`, `action=history`, `action=edit`, `diff=` e `oldid=`. A avaliação deve comparar os resultados antes e depois da filtragem e inspecionar amostras para evitar a remoção de conteúdo legítimo.
- **Escopo da deduplicação**: O método usa hashes de conteúdo para identificar páginas iguais. Como essa estratégia não agrupa, por si só, páginas semanticamente semelhantes ou versões com pequenas variações, alguns conteúdos equivalentes podem permanecer em registros distintos. Em estudos futuros, a deduplicação aproximada pode complementar o hash exato `text_md5`, por exemplo com *MinHash LSH* sobre *shingles* textuais. Os limiares devem ser avaliados para reduzir duplicatas sem agrupar documentos distintos.

## 7.6 Conclusão do Capítulo

Este capítulo apresentou uma metodologia completa para aquisição controlada de dados utilizando a web como fonte, instanciada na ferramenta WackyWacky e aplicada à geração de um conjunto de dados experimental.

A proposta modela a web como um grafo em que as páginas fazem referência entre si, adotando uma estratégia de descoberta e exploração recursiva do espaço de documentos.

Foram definidos mecanismos operacionais para controle de expansão, filtragem, deduplicação e persistência, garantindo consistência e rastreabilidade do processo de aquisição. O foco é gerar conjuntos de dados com início e expansão bem definidos, previsibilidade e controle dos estágios. Essa abordagem sistemática contrasta com estratégias oportunistas ao introduzir controle explícito sobre o processo de exploração.

A ferramenta WackyWacky foi instanciada de maneira concreta, cuja arquitetura implementa diretamente a metodologia proposta e suporta execução distribuída do processo de aquisição. A arquitetura ainda possui, de maneira integrada, instrumentação completa via métricas, logs e *traces* distribuídos. Esta implementação demonstrou a viabilidade prática e serve como validação da proposta ao seguir as definições e utilizar mecanismos operacionais concretos.

Com a instanciação da ferramenta, foi gerado um conjunto de dados de domínio aberto utilizando a Wikipedia como ponto de partida para geração dos *seeds* iniciais de URLs. Além do volume gerado, foram disponibilizados metadados completos do processo de aquisição, incluindo os registros e a relação mapeada entre documentos. Essa característica amplia o valor do conjunto de dados, permitindo análises mais profundas sobre o comportamento de coleta e a estrutura do espaço explorado, estabelecendo um processo reprodutível e observável de aquisição de dados na web, em contraste com estratégias tradicionais baseadas em coleta oportunista.

# Capítulo 8

# Pré-Treinamento Continuado e Avaliação em Português Brasileiro

**Pergunta de Pesquisa 5:** Como o pré-treinamento continuado e a proximidade de domínio dos dados afetam o desempenho de um modelo de linguagem em português brasileiro quando o volume de dados, o orçamento computacional e os demais hiperparâmetros são mantidos constantes?

O pré-treinamento continuado permite adaptar um modelo de linguagem previamente treinado a novas distribuições textuais sem modificar sua arquitetura ou seu vocabulário. Entretanto, a avaliação desse processo exige distinguir o efeito do treinamento adicional daquele produzido pela proximidade entre os dados de pré-treinamento e o domínio de aplicação. Essa distinção é especialmente relevante para modelos em português brasileiro, dada a heterogeneidade dos corpora disponíveis e a presença de variedades linguísticas associadas a diferentes ambientes comunicativos.

Este capítulo investiga esses efeitos tomando como ponto de partida o Tucano2-0.6B-Base, integrante da família Tucano 2. Essa família reúne modelos de linguagem abertos voltados ao português, com disponibilização dos artefatos e procedimentos de treinamento para favorecer a reprodução e a extensão dos resultados [Corrêa et al., 2026].

Para isso, são comparadas três condições experimentais. A primeira corresponde ao modelo original, denominado Base. A segunda corresponde a um modelo submetido a pré-treinamento continuado com uma composição de dados de caráter mais geral, denominado Geral. A terceira corresponde a um modelo treinado com o mesmo orçamento, mas incluindo dados provenientes de fóruns e conteúdos tecnológicos, denominado Fórum/Tec. As duas variantes de pré-treinamento utilizam a mesma quantidade de sequências, tokens e passos de otimização, além do mesmo checkpoint inicial, arquitetura, tokenizer, hiperparâmetros e semente.

O desenho experimental emprega sequências compartilhadas entre as duas variantes e substitui, de forma pareada, o volume de Adrenaline e OuterSpace utilizado pelo Fórum/Tec por sequências adicionais do BrWaC-CLEAN no Geral. Essa organização permite examinar separadamente o efeito geral do pré-treinamento continuado e a contribuição adicional da proximidade de domínio. Os modelos são avaliados por meio de métricas intrínsecas de linguagem, classificação com representações congeladas e análises diagnósticas em regime de poucos exemplos, por probabilidade condicional e ao longo dos

checkpoints de treinamento.

O restante deste capítulo está organizado da seguinte forma. A Seção 8.1 apresenta o desenho experimental, os modelos comparados, os contrastes investigados e as variáveis controladas. A Seção 8.2 descreve a composição dos dados de cada variante, a preparação das sequências e os procedimentos empregados para evitar sobreposição entre treinamento e avaliação. A Seção 8.3 detalha a configuração e a execução do pré-treinamento continuado. A Seção 8.4 apresenta o protocolo de avaliação e os procedimentos estatísticos adotados. Em seguida, a Seção 8.5 reúne os resultados obtidos e sintetiza os contrastes experimentais. Por fim, a Seção 8.6 discute as evidências, as limitações do estudo e a resposta à pergunta de pesquisa.

## 8.1 Desenho Experimental

O experimento foi estruturado para separar o efeito geral do pré-treinamento continuado daquele decorrente da proximidade entre os dados de treinamento e o domínio avaliado. Para isso, foram comparados um modelo-base e dois modelos derivados dele, treinados sob o mesmo orçamento computacional e diferenciados apenas pela composição de parte dos dados. O desenho pareado permite interpretar as diferenças observadas por meio de três contrastes complementares, mantendo constantes as principais variáveis capazes de influenciar o treinamento.

### 8.1.1 Modelos comparados

Foram comparados três modelos baseados na mesma arquitetura e no mesmo tokenizer. O primeiro, denominado *Base*, corresponde ao Tucano2-0.6B-Base antes de qualquer treinamento adicional. Os outros dois modelos, denominados *Geral* e *Fórum/Tec*, foram obtidos por meio de pré-treinamento continuado de todos os parâmetros a partir do mesmo checkpoint do modelo Base.

O modelo Base, também identificado como B0, possui aproximadamente 670 milhões de parâmetros e funciona como referência para determinar o efeito do pré-treinamento continuado. Seus pesos não foram modificados durante os experimentos descritos neste capítulo. Dessa forma, qualquer diferença observada entre esse modelo e os modelos refinados decorre da etapa adicional de treinamento, considerada em conjunto com a composição de dados utilizada em cada condição experimental.

O modelo Geral foi treinado com BrWaC-CLEAN, GigaVerbo, WackyWacky e MultiWOZ-PTBR. Além da parcela compartilhada do BrWaC-CLEAN, esse modelo recebeu sequências adicionais do mesmo corpus para ocupar as posições que, no treinamento do modelo Fórum/Tec, foram destinadas aos dados de fóruns e tecnologia. Dessa forma, o Geral representa a condição de pré-treinamento continuado sem a inclusão de Adrenaline e OuterSpace.

O modelo Fórum/Tec foi treinado com as mesmas sequências compartilhadas de BrWaC-CLEAN, GigaVerbo, WackyWacky e MultiWOZ-PTBR, acrescidas de sequências do Adrenaline e do OuterSpace. Esses dois corpora substituem, em quantidade equivalente, o volume adicional de BrWaC-CLEAN empregado no modelo Geral. O Fórum/Tec

representa a condição de pré-treinamento continuado com maior proximidade em relação ao domínio de fóruns e conteúdos tecnológicos.

### 8.1.2 Contrastes experimentais

A comparação entre os três modelos foi organizada em três contrastes experimentais. O primeiro contraste, Geral menos Base, estima o efeito do pré-treinamento continuado com a composição de dados definida para o modelo Geral. Como o Base não recebe treinamento adicional, esse contraste incorpora o efeito de expor o modelo a novos tokens sob o regime de treinamento adotado.

O segundo contraste, Fórum/Tec menos Geral, estima a contribuição adicional da proximidade de domínio. Os dois modelos partem do mesmo checkpoint e utilizam o mesmo número de sequências, tokens e passos de otimização. Além disso, compartilham literalmente parte das sequências de treinamento. A diferença entre eles está nas posições em que o Fórum/Tec utiliza Adrenaline e OuterSpace, enquanto o Geral utiliza sequências adicionais e exclusivas do BrWaC-CLEAN. Esse é, portanto, o contraste mais diretamente relacionado ao efeito da composição dos dados.

O terceiro contraste, Fórum/Tec menos Base, representa o efeito prático total da combinação entre pré-treinamento continuado e proximidade de domínio. Embora esse contraste indique quanto o modelo final difere do ponto de partida, ele não permite separar esses dois componentes. Por essa razão, sua interpretação é complementada pelos contrastes anteriores.

Para métricas em que valores maiores indicam melhor desempenho, como acurácia e macro-F1, deltas positivos favorecem o primeiro modelo da subtração. Para métricas em que valores menores são preferíveis, como loss, Log-Verossimilhança Negativa (NLL) e perplexidade, deltas negativos favorecem o primeiro modelo da subtração.

### 8.1.3 Variáveis controladas

Nos experimentos realizados, a arquitetura, o número de parâmetros, o tokenizer, o vocabulário, os identificadores dos tokens especiais e o comprimento das sequências foram preservados. Em ambos os casos, o pré-treinamento continuado atualizou todos os parâmetros do modelo, sem o uso de adaptadores ou alterações na camada de entrada.

Cada modelo processou 425.984.000 tokens, distribuídos em 416.000 sequências de 1.024 tokens, ao longo de 52.000 passos de otimização. Foram mantidos constantes o batch global de oito sequências, a semente, o otimizador AdamW, a taxa máxima de aprendizado de $5 \times 10^{-6}$, o aquecimento de 520 passos, o scheduler linear, a precisão BF16 e a estratégia de treinamento distribuído. A configuração completa do treinamento é apresentada na Seção 8.3.

O controle também foi aplicado à seleção e à ordem dos dados. Os dois modelos compartilharam 250.441 sequências provenientes de BrWaC-CLEAN, GigaVerbo, MultiWOZ-PTBR e WackyWacky, inseridas nas mesmas posições do cronograma de treinamento. As 165.559 posições restantes receberam Adrenaline e OuterSpace no Fórum/Tec e sequências adicionais e exclusivas do BrWaC-CLEAN no Geral. A substituição foi realizada na

proporção de uma sequência para uma sequência, sem repetição ou sobreamostragem.

Consequentemente, os dois treinamentos apresentam a mesma carga computacional teórica e diferem apenas no conteúdo das posições destinadas à substituição de domínio.

## 8.2 Dados Utilizados e Preparação

Os corpora apresentados anteriormente na tese foram submetidos aos procedimentos de limpeza descritos nas respectivas seções e à deduplicação textual exata dentro de cada corpus. Cada modelo utilizou 416.000 sequências de 1.024 tokens, totalizando 425.984.000 tokens de treinamento. No Fórum/Tec, a alocação partiu de cotas aproximadamente iguais entre os seis corpora. Como o MultiWOZ-PTBR forneceu apenas 2.102 sequências, sua cota remanescente foi redistribuída deterministicamente entre os demais. O Geral substituiu as sequências de Adrenaline e OuterSpace por BrWaC-CLEAN adicional, conforme a Tabela 8.1.

| Corpus | Geral | Fórum/Tec |
|---|---|---|
| Adrenaline | 0 | 82.780 |
| BrWaC-CLEAN | 248.339 | 82.780 |
| GigaVerbo | 82.780 | 82.780 |
| MultiWOZ-PTBR | 2.102 | 2.102 |
| OuterSpace | 0 | 82.779 |
| WackyWacky | 82.779 | 82.779 |
| **Total** | **416.000** | **416.000** |

Tabela 8.1: Distribuição das sequências de treinamento de 1.024 tokens por corpus e modelo.

A Figura 8.1 apresenta as 250.441 sequências compartilhadas (60,20%) e as 165.559 posições de substituição (39,80%), ocupadas por BrWaC-CLEAN adicional no Geral e por Adrenaline e OuterSpace no Fórum/Tec. Os conjuntos de treinamento dos dois modelos foram derivados de uma única coleção preparada, com intervalos sem sobreposição para as parcelas comum e adicional do BrWaC-CLEAN. Um cronograma proporcional e determinístico preservou as posições das sequências compartilhadas e o pareamento das substituições ao longo do treinamento. Cada referência foi consumida uma única vez por modelo, sem sobreamostragem.

Na preparação dos dados, a separação entre treinamento e avaliação intrínseca foi realizada antes do *packing*, utilizando hashes das referências de origem e a semente do experimento. Para BrWaC-CLEAN, GigaVerbo, MultiWOZ-PTBR, OuterSpace e WackyWacky, foram mantidos os conjuntos de 256 sequências reservados nessa etapa. No Adrenaline, a avaliação final utilizou um conjunto separado por thread de origem e submetido ao controle adicional de repetição descrito no protocolo de avaliação intrínseca (Seção 8.4.1). Os mesmos conjuntos finais foram utilizados para comparar o Base, o Geral e o Fórum/Tec.

Na classificação do Adrenaline, foram excluídas integralmente as threads associadas a qualquer fragmento utilizado no treinamento ou no conjunto de avaliação intrínseca

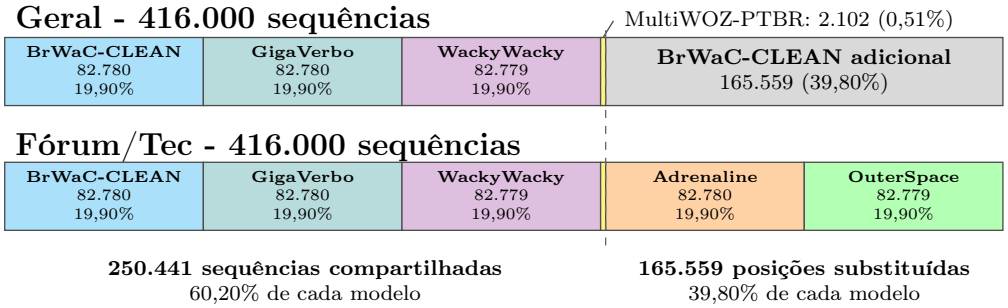



Figura 8.1: Composição pareada dos dados de pré-treinamento dos modelos Geral e Fórum/Tec.

originalmente reservado durante a preparação dos dados.

Também foram removidas duplicatas exatas do par título-primeiro post. Títulos idênticos após normalização formaram grupos dos quais se selecionou um representante por tarefa e semente, sem compartilhamento entre treino, validação e teste. As mesmas partições atenderam aos três modelos e às duas variantes de entrada.

Na composição dos conjuntos de treinamento, não foi realizada deduplicação cruzada entre os seis corpora nem identificação de equivalência semântica.

## 8.3 Procedimento de Pré-Treinamento Continuado

As variantes Geral e Fórum/Tec foram submetidas ao mesmo procedimento de pré-treinamento continuado, diferenciando-se apenas pela composição dos dados. Ambas foram inicializadas com a mesma revisão fixa do Tucano2-0.6B-Base, de arquitetura Llama e com 670.127.616 parâmetros. O treinamento atualizou todos os parâmetros, sem adaptadores e sem alterações na arquitetura, no tokenizer, no vocabulário ou nos identificadores dos tokens especiais. Foi mantido o objetivo autorregressivo de prever o próximo token a partir do contexto precedente, minimizando a entropia cruzada sobre as sequências preparadas.

Cada treinamento percorreu uma época sobre 416.000 sequências de 1.024 tokens, totalizando 425.984.000 tokens em 52.000 passos de otimização. Foi utilizado o otimizador AdamW, com aquecimento linear da taxa de aprendizado nos primeiros 520 passos e decaimento linear até o encerramento do treinamento. O *weight decay* foi desabilitado para biases, parâmetros de normalização e parâmetros com menos de duas dimensões. A Tabela 8.2 apresenta os hiperparâmetros comuns às duas variantes.

As execuções utilizaram duas GPUs NVIDIA L40S em treinamento distribuído com *DistributedDataParallel* (DDP). A acumulação de gradientes resultou em um batch global de oito sequências, conforme a configuração apresentada na tabela. Empregou-se precisão mista com BF16 nas operações compatíveis, mantendo parâmetros, gradientes e estados do otimizador em FP32.

Para analisar a evolução do desempenho ao longo do treinamento, foram preservados checkpoints intermediários nos passos 13.000, 26.000 e 39.000, além dos modelos finais obtidos após 52.000 passos. Os modelos finais e seus respectivos tokenizers foram submetidos

| Parâmetro | Valor |
|---|---|
| Comprimento da sequência | 1.024 tokens |
| Épocas | 1 |
| Passos de otimização | 52.000 |
| Microbatch por GPU | 1 sequência |
| Acumulação por GPU | 4 microbatches |
| Batch global | 8 sequências / 8.192 tokens |
| Otimizador | AdamW, implementação *fused* do PyTorch |
| Taxa de aprendizado máxima | $5 \times 10^{-6}$ |
| Aquecimento | 520 passos |
| Decaimento da taxa de aprendizado | Linear |
| Coeficientes do AdamW | $\beta_1 = 0{,}9$, $\beta_2 = 0{,}95$ |
| Termo de estabilidade numérica | $\epsilon = 10^{-8}$ |
| Weight decay | 0,1 |
| Norma máxima dos gradientes | 1,0 |

Tabela 8.2: Hiperparâmetros do pré-treinamento continuado.

à verificação de integridade e de carregamento antes da avaliação.

## 8.4 Protocolo de Avaliação

A avaliação combinou modelagem de linguagem, classificação com representações congeladas e análises diagnósticas. Essas abordagens examinaram a capacidade de prever textos, a utilidade das representações para classificação e a evolução do desempenho ao longo do treinamento. Cada comparação utilizou os mesmos exemplos para todos os modelos, mantendo congelados os parâmetros dos modelos de linguagem.

### 8.4.1 Avaliação intrínseca de linguagem

A avaliação intrínseca utilizou 256 sequências de 1.024 tokens de cada um dos seis corpora, totalizando 1.536 sequências. Os mesmos conjuntos foram empregados na avaliação do Base e dos modelos finais Geral e Fórum/Tec, sem atualização dos parâmetros.

Para o Adrenaline, foram excluídas as threads representadas no treinamento ou no conjunto de avaliação originalmente reservado. A seleção foi determinística, baseada em hashes e na semente utilizada, considerando até 8.192 threads. Foi selecionado um fluxo por thread e um bloco completo por fluxo, preservando o formatador e o tokenizer do experimento, sem concatenar fluxos distintos ou acrescentar padding. Cada candidato foi comparado com os blocos preparados para treinamento dos seis corpora e excluído integralmente quando continha um fragmento idêntico de pelo menos 32 tokens. Essa comparação foi restrita ao interior dos blocos de treinamento.

A loss correspondeu à entropia cruzada da previsão do próximo token. Cada sequência de 1.024 tokens forneceu 1.023 alvos, totalizando 261.888 alvos por corpus. A loss de cada corpus foi calculada pela média das losses de suas sequências, equivalente à média por alvo, pois todas possuíam o mesmo comprimento.

A loss global foi calculada pela média aritmética dos seis corpora, atribuindo-lhes pesos iguais, independentemente de sua participação no treinamento. A perplexidade foi obtida por PPL = $\exp(\mathcal{L})$, em que $\mathcal{L}$ representa a loss correspondente. Assim, a perplexidade global é a exponencial da loss global, não a média aritmética das perplexidades individuais. Valores menores indicam melhor previsão dos textos avaliados.

### 8.4.2 Classificação com representações congeladas

A avaliação de classificação foi realizada com cinco sementes e utilizou o dataset canônico do Adrenaline descrito na Seção 8.2. Esse dataset continha 352.296 exemplos elegíveis, organizados em 341.069 grupos de títulos, constituindo a população de origem das amostras balanceadas selecionadas para cada tarefa.

Foram consideradas duas tarefas de classificação: por categoria, com seis classes e 2.000 exemplos por classe, e por subcategoria, com 22 classes e 1.000 exemplos por classe. A inclusão de uma subcategoria exigiu pelo menos 1.000 grupos de títulos elegíveis distintos. Os conjuntos balanceados totalizaram, respectivamente, 12.000 e 22.000 exemplos em cada repetição.

Em cada repetição de cada tarefa, selecionou-se deterministicamente um representante por grupo de títulos, seguido de amostragem balanceada e divisão estratificada em 70% para treino, 15% para validação e 15% para teste. Os mesmos exemplos e partições foram utilizados para os três modelos e para duas entradas: título isolado e título concatenado ao primeiro post, separado por duas quebras de linha.

As entradas foram tokenizadas com o tokenizer original, limitadas a 1.024 tokens por truncamento à direita. Da última camada de cada modelo, foram extraídas duas representações candidatas: a média dos vetores dos tokens de conteúdo e o vetor do último token de conteúdo. Tokens especiais e padding foram excluídos da agregação. A inferência utilizou BF16, enquanto a agregação e o armazenamento dos embeddings utilizaram FP32.

Sobre os embeddings, foi ajustada uma regressão logística multinomial com regularização L2 no scikit-learn, precedida de padronização por `StandardScaler`. Foram utilizados o solver *lbfgs*, intercepto, ausência de ponderação de classes, tolerância de $10^{-5}$ e limite de 2.000 iterações.

Durante a seleção de hiperparâmetros, o padronizador e o classificador foram ajustados exclusivamente com os dados de treino. Foram avaliadas duas representações dos textos e quatro valores do parâmetro de regularização: $C \in \{0{,}01;\ 0{,}1;\ 1;\ 10\}$. Valores menores de $C$ impõem maior penalização aos coeficientes do classificador. A configuração foi escolhida com base no desempenho na validação.

Para cada combinação de tarefa e entrada, escolheu-se uma única configuração pela maior média de macro-F1 de validação entre os três modelos e as repetições.

A seleção de hiperparâmetros foi também realizada independentemente para cada semente, mantendo a configuração compartilhada entre os três modelos. Em todos os casos, foram selecionados a média dos tokens de conteúdo e $C = 0{,}01$, sem alteração dos resultados finais.

Após a escolha, o padronizador e o classificador foram reajustados na união de treino

e validação e avaliados no teste correspondente, mantendo a configuração compartilhada entre os modelos. Esse procedimento resultou em 60 avaliações finais.

### 8.4.3 Avaliações diagnósticas

As avaliações diagnósticas foram organizadas em três análises complementares: o desempenho na classificação com poucos exemplos de treinamento, a log-verossimilhança negativa condicional do primeiro post e a evolução dessa medida ao longo dos checkpoints. Os protocolos de cada análise são descritos a seguir.

**Avaliação em regime low-shot.** A avaliação com poucos exemplos utilizou a classificação por categoria, com título e primeiro post como entrada e os embeddings existentes agregados pela média. Manteve-se $C = 0{,}01$, parâmetro de regularização da regressão logística.

Foram selecionados, exclusivamente do treino original, subconjuntos aninhados de 16, 64, 256 e 1.400 exemplos por classe, mediante ordenação determinística por hash para cada semente. Os subconjuntos foram compartilhados pelos três modelos, mantendo-se o teste original de 300 exemplos por classe.

**NLL condicional.** Foi selecionado um conjunto de 1.800 threads, com 300 exemplos por categoria e um representante por grupo de títulos. A seleção respeitou a exclusão de sobreposição com o pré-treinamento continuado e com os grupos presentes nos testes anteriores, conforme a Seção 8.2. A entrada reuniu título e primeiro post, com truncamento à direita em 1.024 tokens e exigência de pelo menos 32 tokens avaliáveis do post.

A NLL foi calculada somente nos tokens do primeiro post, condicionados ao título e aos tokens precedentes. Foram excluídos tokens especiais, padding e tokens que atravessassem a fronteira entre título e post. A NLL foi normalizada pelo número de tokens avaliados de cada thread e agregada pela média das threads de cada categoria, seguida da média das seis categorias. A perplexidade correspondeu à exponencial dessa NLL agregada.

**Curva por checkpoints.** O mesmo conjunto de threads e o mesmo procedimento de NLL foram utilizados para avaliar o Base e os estados de Geral e Fórum/Tec nos passos 13.000, 26.000, 39.000 e 52.000. Esses pontos correspondem a 106.496.000, 212.992.000, 319.488.000 e 425.984.000 tokens de treinamento, tomando o Base como referência no passo zero. Foram calculadas diferenças acumuladas em relação ao Base e incrementais entre checkpoints consecutivos. Os artefatos finais representaram o passo 52.000, totalizando nove estados avaliados.

### 8.4.4 Análise estatística

A análise estatística priorizou a macro-F1 na classificação por categoria com título e primeiro post, calculada pela média não ponderada do F1 das classes. A acurácia, as demais combinações de tarefa e entrada, as métricas por classe e as matrizes de confusão

constituíram resultados secundários. As avaliações diagnósticas foram tratadas como evidências complementares à análise principal.

Para classificação e low-shot, foram calculados média, desvio-padrão amostral e intervalos de confiança de 95% baseados na distribuição t de Student, com quatro graus de liberdade. Os contrastes Geral menos Base, Fórum/Tec menos Geral e Fórum/Tec menos Base foram calculados dentro de cada semente, obtendo-se os intervalos sobre os cinco deltas pareados. Como diferentes sementes podem compartilhar exemplos, inclusive em partições distintas, esses intervalos foram interpretados descritivamente como medidas da variabilidade observada entre divisões.

Para NLL e diferenças entre checkpoints, foram utilizados intervalos percentis de 95%, obtidos por 10.000 reamostragens pareadas e estratificadas por categoria. As threads foram reamostradas com reposição dentro de cada categoria, preservando os mesmos índices para todos os modelos e checkpoints. No intervalo terminal, foi calculada a diferença entre a NLL no passo 52.000 e a NLL no passo 39.000: um intervalo inteiramente negativo indicou melhora adicional; um intervalo incluindo zero, ausência de evidência clara de mudança; e um intervalo inteiramente positivo, piora.

Como houve apenas uma execução de pré-treinamento por variante, os intervalos representam variação entre divisões ou threads, não entre treinamentos independentes.

## 8.5 Resultados

Nas condições avaliadas, os dois modelos submetidos ao pré-treinamento continuado apresentaram melhorias na modelagem de linguagem e ganhos modestos na classificação de textos em relação ao modelo Base. O Fórum/Tec apresentou vantagem adicional sobre o Geral na previsão de textos do domínio de fóruns, mas essa vantagem não foi conclusiva na classificação. Os resultados a seguir detalham essas diferenças nas avaliações intrínsecas, na classificação e nos diagnósticos complementares.

### 8.5.1 Avaliação intrínseca de linguagem

A loss global diminuiu de 2,8461 no Base para 2,6197 no Geral e 2,5631 no Fórum/Tec. As perplexidades correspondentes foram 17,2210, 13,7311 e 12,9754, representando reduções de 20,27% e 24,65% em relação ao Base. A Tabela 8.3 apresenta os resultados por corpus e a agregação global.

A vantagem do Fórum/Tec sobre o Geral concentrou-se nos corpora de fóruns. A perplexidade foi 14,01% menor no Adrenaline e 22,11% menor no OuterSpace. Nos quatro corpora compartilhados, entretanto, o Fórum/Tec apresentou perplexidades superiores às do Geral: aumentos de 3,70% no BrWaC-CLEAN, 1,51% no GigaVerbo, 0,15% no MultiWOZ-PTBR e 0,84% no WackyWacky. Na agregação dos seis corpora, sua perplexidade foi 5,50% inferior à do Geral. Portanto, a vantagem observada não representou uma melhora uniforme entre os corpora.

| | Base | | Geral | | Fórum/Tec | |
|---|---|---|---|---|---|---|
| **Corpus** | **Loss** | **PPL** | **Loss** | **PPL** | **Loss** | **PPL** |
| Adrenaline | 3,4723 | 32,2097 | 3,4776 | 32,3824 | 3,3267 | 27,8470 |
| BrWaC-CLEAN | 2,7973 | 16,3996 | 2,7125 | 15,0667 | 2,7488 | 15,6238 |
| GigaVerbo | 2,5694 | 13,0573 | 2,5606 | 12,9431 | 2,5756 | 13,1389 |
| MultiWOZ-PTBR | 2,4417 | 11,4927 | 1,7362 | 5,6755 | 1,7377 | 5,6842 |
| OuterSpace | 3,5258 | 33,9796 | 3,5120 | 33,5156 | 3,2621 | 26,1042 |
| WackyWacky | 2,2704 | 9,6837 | 1,7191 | 5,5797 | 1,7274 | 5,6263 |
| **Global** | **2,8461** | **17,2210** | **2,6197** | **13,7311** | **2,5631** | **12,9754** |

Tabela 8.3: Loss e perplexidade na avaliação intrínseca. Valores menores indicam melhor desempenho.

### 8.5.2 Classificação de textos

A seleção de hiperparâmetros escolheu a agregação pela média dos tokens de conteúdo e $C = 0{,}01$ para as quatro combinações de tarefa e entrada. Na classificação por categoria com título e primeiro post, as médias de macro-F1 foram 0,8194 para Base, 0,8250 para Geral e 0,8276 para Fórum/Tec. Os desvios-padrão foram, respectivamente, 0,0065, 0,0042 e 0,0024. A Tabela 8.4 reúne as médias das métricas principais e secundárias.

| Tarefa | Entrada | Métrica | Base | Geral | Fórum/Tec |
|---|---|---|---|---|---|
| Categoria | Título | Macro-F1 | 0,6660 | 0,6730 | 0,6765 |
| | | Acurácia | 0,6662 | 0,6731 | 0,6769 |
| Categoria | Título + post | Macro-F1 | 0,8194 | 0,8250 | 0,8276 |
| | | Acurácia | 0,8193 | 0,8249 | 0,8276 |
| Subcategoria | Título | Macro-F1 | 0,5935 | 0,6017 | 0,6050 |
| | | Acurácia | 0,5947 | 0,6026 | 0,6058 |
| Subcategoria | Título + post | Macro-F1 | 0,7471 | 0,7535 | 0,7530 |
| | | Acurácia | 0,7493 | 0,7559 | 0,7558 |

Tabela 8.4: Médias das métricas de classificação.

Na análise principal, os ganhos sobre o Base foram de 0,0056 para Geral e 0,0081 para Fórum/Tec, ambos com intervalos inteiramente positivos. A diferença adicional entre Fórum/Tec e Geral foi de 0,0026. Os contrastes e respectivos IC95% são apresentados posteriormente na Tabela 8.6.

A inclusão do primeiro post elevou o desempenho dos três modelos nas duas tarefas. Embora os modelos refinados tenham apresentado médias superiores às do Base em todas as combinações, o Fórum/Tec não manteve vantagem sobre o Geral: na classificação por subcategoria com título e primeiro post, suas médias foram ligeiramente inferiores.

As análises exploratórias de erros mostraram a persistência de confusões entre as subcategorias Diversos e Sistemas Operacionais, assim como entre Hardwares e Eletrônicos, todas pertencentes à categoria Tecnologia. Dessa forma, o refinamento não eliminou as principais dificuldades de distinção entre subcategorias próximas.

### 8.5.3 Resultados das avaliações diagnósticas

As avaliações diagnósticas complementaram a comparação entre os modelos em três aspectos: o desempenho com poucos exemplos de treinamento, a NLL condicional do primeiro post e a evolução dessa medida ao longo dos checkpoints.

**Desempenho em regime low-shot.** Embora tenham sido avaliados quatro orçamentos de exemplos rotulados, a apresentação neste capítulo concentra-se no orçamento prioritário de 64 exemplos por classe. Nessa condição, a macro-F1 média foi de 0,7276 para Base, 0,7361 para Geral e 0,7423 para Fórum/Tec. Os ganhos sobre o Base foram de 0,0085 para Geral e 0,0148 para Fórum/Tec, ambos com intervalos inteiramente positivos. O contraste Fórum/Tec menos Geral foi de 0,0063, com IC95% incluindo zero. Nesse orçamento, o diagnóstico exploratório apresentou resultados favoráveis ao pré-treinamento continuado.

Embora o Fórum/Tec tenha apresentado maior macro-F1 média, os resultados não permitem concluir que seu desempenho seja superior ao do Geral.

**NLL condicional.** Na coorte adicional, a NLL agregada foi de 3,3593 para Base, 3,3456 para Geral e 3,2628 para Fórum/Tec, com perplexidades de 28,7700, 28,3765 e 26,1238. O contraste prioritário Fórum/Tec menos Geral foi de $-0{,}0827$, com IC95% de $[-0{,}0881; -0{,}0774]$, correspondendo a uma redução de aproximadamente 7,94% na perplexidade agregada. O Fórum/Tec apresentou NLL inferior à do Geral em 76,11% das threads.

Os contrastes Geral menos Base e Fórum/Tec menos Base também apresentaram intervalos inteiramente negativos, com reduções de 0,0138 e 0,0965 na NLL. As proporções de threads melhoradas foram de 57,72% e 77,17%, respectivamente. Dessa forma, diferentemente da classificação, a NLL condicional apresentou um contraste adicional favorável ao Fórum/Tec.

**Evolução por checkpoints.** A Tabela 8.5 apresenta a trajetória observada. O Geral permaneceu com NLL superior à do Base nos passos 13.000 e 26.000, passando a valores inferiores nos passos 39.000 e 52.000. O Fórum/Tec apresentou redução da NLL em todos os pontos consecutivos avaliados.

| | Geral | | Fórum/Tec | |
|---|---|---|---|---|
| **Passo** | **NLL** | **PPL** | **NLL** | **PPL** |
| 0 | 3,3593 | 28,7700 | 3,3593 | 28,7700 |
| 13.000 | 3,3730 | 29,1649 | 3,3245 | 27,7852 |
| 26.000 | 3,3715 | 29,1234 | 3,2953 | 26,9862 |
| 39.000 | 3,3512 | 28,5383 | 3,2757 | 26,4610 |
| 52.000 | 3,3456 | 28,3765 | 3,2628 | 26,1238 |

Tabela 8.5: NLL condicional e perplexidade por checkpoint. O passo zero corresponde ao Base.

Entre 39.000 e 52.000 passos, a NLL diminuiu 0,0057 no Geral e 0,0128 no Fórum/Tec. Os respectivos IC95% foram [−0,0076; −0,0038] e [−0,0150; −0,0107]. Ambos apresentaram melhora adicional no último intervalo observado, embora esse resultado não estabeleça se a tendência seria mantida com a extensão do treinamento.

### 8.5.4 Síntese dos contrastes

A Tabela 8.6 reúne os contrastes da classificação principal, do low-shot com 64 exemplos por classe e da NLL condicional final. Em cada comparação $A - B$, valores positivos favorecem $A$ em macro-F1, enquanto valores negativos favorecem $A$ em NLL. As magnitudes de métricas distintas não devem ser comparadas diretamente.

| Contraste | Avaliação | Δ | IC95% |
|---|---|---|---|
| Geral − Base | Classificação | +0,0056 | [+0,0010; +0,0101] |
| | Low-shot | +0,0085 | [+0,0033; +0,0137] |
| | NLL | −0,0138 | [−0,0182; −0,0094] |
| Fórum/Tec − Geral | Classificação | +0,0026 | [−0,0029; +0,0081] |
| | Low-shot | +0,0063 | [−0,0009; +0,0134] |
| | NLL | −0,0827 | [−0,0881; −0,0774] |
| Fórum/Tec − Base | Classificação | +0,0081 | [+0,0013; +0,0149] |
| | Low-shot | +0,0148 | [+0,0068; +0,0227] |
| | NLL | −0,0965 | [−0,1027; −0,0905] |

Tabela 8.6: Contrastes pareados e IC95% nas três avaliações.

Os dois modelos refinados apresentaram resultados favoráveis em relação ao Base nas três avaliações sintetizadas. A vantagem adicional do Fórum/Tec sobre o Geral foi identificada na NLL condicional, mas permaneceu inconclusiva na classificação principal e no low-shot. Essas evidências se referem aos modelos e às amostras avaliados.

## 8.6 Discussão e Limitações

Os resultados indicam que o pré-treinamento continuado produziu ganhos modestos, mas favoráveis, nas condições avaliadas. Em relação ao Base, o Geral apresentou aumento de aproximadamente 0,56 ponto percentual no macro-F1 da classificação principal e de 0,85 ponto percentual no regime com 64 exemplos por classe, além de redução da NLL condicional. Esses resultados sugerem que a atualização dos parâmetros com os corpora selecionados beneficiou tanto a modelagem de linguagem quanto as representações utilizadas pelo classificador, embora não demonstrem uma melhoria irrestrita em outras tarefas.

A contribuição adicional da proximidade de domínio foi mais evidente na NLL condicional. O Fórum/Tec apresentou redução de 0,0827 em relação ao Geral, com intervalo de confiança inteiramente negativo, e melhorou a previsão em 76,11% das threads avaliadas. Portanto, a substituição de BrWaC-CLEAN extra por Adrenaline e OuterSpace favoreceu a previsão dos textos de fórum, mas não estabeleceu uma vantagem conclusiva na classificação. Essa ausência de evidência conclusiva também não demonstra equivalência entre os modelos.

A diferença entre os resultados é compatível com os objetivos das avaliações. A NLL condicional examina a previsão dos tokens do primeiro post, aproximando-se do objetivo otimizado durante o pré-treinamento continuado. A classificação, por sua vez, depende de como as informações relevantes para os rótulos são recuperadas por um classificador linear a partir de representações congeladas. Assim, uma melhora na distribuição de probabilidades dos textos não implica necessariamente maior separação entre categorias.

Na avaliação intrínseca agregada, a vantagem do Fórum/Tec sobre o Geral resultou das menores losses no Adrenaline e no OuterSpace, parcialmente compensadas pelas losses ligeiramente superiores nos quatro corpora compartilhados. O resultado indica um benefício associado à composição com textos de fóruns nas condições avaliadas, mas não uma superioridade uniforme entre os corpora.

A curva por checkpoints mostrou trajetórias distintas: o Geral apresentou piora inicial da NLL condicional em relação ao Base, seguida de recuperação, enquanto o Fórum/Tec apresentou reduções sucessivas nos pontos avaliados. Entre 39 mil e 52 mil passos, ambos obtiveram melhora adicional, com intervalos dos deltas inteiramente abaixo de zero. Esse resultado não sustenta a afirmação de que os modelos já haviam atingido um platô no último intervalo observado. Contudo, também não permite determinar o ponto de saturação ou garantir que prolongar o treinamento manteria os ganhos, especialmente sob outra configuração de taxa de aprendizado.

A principal limitação experimental é a existência de apenas uma execução de pré-treinamento por variante. As sementes utilizadas variaram a amostragem e as divisões da classificação, não os treinamentos dos modelos de linguagem. Consequentemente, os intervalos apresentados caracterizam a variação entre divisões ou threads, condicionada aos modelos obtidos, e não a variabilidade entre execuções independentes de pré-treinamento.

A proximidade de domínio foi operacionalizada pela substituição de BrWaC-CLEAN adicional por Adrenaline e OuterSpace, mantendo os demais componentes e o orçamento de tokens. Portanto, o contraste Fórum/Tec-Geral avalia essa mudança de composição, sem permitir atribuir seus efeitos exclusivamente à proximidade temática, pois os corpora também podem diferir em estilo, diversidade e qualidade textual.

A validade externa permanece restrita a um backbone, ao orçamento adotado e às tarefas construídas com dados do Adrenaline. O balanceamento das classes, o truncamento em 1.024 tokens e o uso de representações congeladas também delimitam o alcance dos resultados, que não avaliam diretamente geração de diálogos, seguimento de instruções ou desempenho em outros domínios.

Em resposta à pergunta de pesquisa, o experimento mostrou que, mantendo o volume de tokens, os passos e os principais hiperparâmetros constantes entre as variantes refinadas, o pré-treinamento continuado pode melhorar a modelagem de linguagem e o desempenho downstream do modelo estudado. A proximidade de domínio acrescentou uma vantagem clara na previsão dos textos de fórum, mas sua contribuição adicional para a classificação permaneceu inconclusiva. Portanto, os benefícios do refinamento dependem não apenas da composição dos dados, mas também da capacidade examinada.

# Capítulo 9

# Ataques de Extração e Defesa por Substituição Semântica em Aprendizado Federado

**Pergunta de Pesquisa 6:** Em que medida a substituição semântica rotativa reduz a reprodução de dados pessoais sintéticos durante o treinamento federado de um modelo de linguagem em português brasileiro, em comparação com a ausência de defesa, na presença de um cliente auxiliar benigno ou adversário?

Os LLMs têm sido cada vez mais utilizados em cenários distribuídos, como teclados inteligentes, agentes locais e assistentes pessoais. Nesses contextos, o FL pode ser empregado como uma estratégia eficiente para o treinamento desses modelos, permitindo o uso descentralizado dos dados diretamente nos dispositivos em que são gerados.

Essa abordagem reduz os custos associados à centralização de dados e oferece ganhos de privacidade ao evitar a transferência de informações sensíveis para servidores centrais. Apesar disso, a preservação da privacidade dos dados não implica, por si só, a privacidade do modelo treinado, que pode permanecer suscetível a vazamentos de informação.

Embora o FL reduza a exposição direta dos dados ao evitar sua centralização, essa propriedade não garante que o modelo resultante seja seguro. Mesmo sem acesso explícito aos registros originais, o processo de treinamento pode induzir memorização e posterior exposição de informações sensíveis. No caso de LLMs, o problema se torna particularmente sensível porque a maior capacidade de modelagem favorece a memorização de padrões raros, enquanto a participação de poucos clientes por rodada, cenário comum em *clusters*, aumenta a influência relativa de atualizações maliciosas no processo de agregação.

Abordagens que buscam mitigar esse tipo de problema frequentemente introduzem modificações nos gradientes, nos parâmetros ou no processo de otimização do modelo. Em muitos casos, essas intervenções impactam diretamente seu desempenho.

Este capítulo investiga o problema de vazamento de dados em cenários de treinamento federado com LLMs, por meio da definição de um protocolo experimental controlado que permite comparar diferentes cenários e aprofundar o entendimento sobre os fatores que influenciam esse problema.

Como parte dessa investigação, é proposta e avaliada uma abordagem de substituição semântica, que consiste na transformação de dados sensíveis em representações estrutu-

ralmente equivalentes e semanticamente similares aos dados originais, com o objetivo de mitigar o vazamento dessas informações.

As contribuições centrais deste capítulo são: (*i*) a formalização de um fluxo reprodutível para análise de vazamento em FL; (*ii*) a comparação controlada entre condições com e sem envenenamento e com diferentes estratégias de defesa; e (*iii*) a caracterização empírica da exposição de distintos tipos de informação sensível.

O restante deste capítulo está organizado da seguinte forma. A Seção 9.1 apresenta uma visão geral do problema e da abordagem adotada. Em seguida, a Seção 9.2 descreve as questões de pesquisa e as hipóteses investigadas. O modelo de ameaça considerado é definido na Seção 9.3. A Seção 9.4 apresenta o mecanismo de substituição semântica proposto. O protocolo experimental é descrito na Seção 9.5, incluindo a distribuição dos dados, o treinamento federado, as variantes benigna e adversária do cliente auxiliar e as métricas de avaliação. Os resultados obtidos são discutidos na Seção 9.6. Por fim, a Seção 9.7 apresenta a conclusão do capítulo.

## 9.1 Visão Geral

A avaliação empírica foi organizada como um protocolo controlado composto por três estágios: preparação e validação dos dados sintéticos, treinamento federado das diferentes trajetórias e avaliação de reprodução e utilidade. Os estágios foram implementados de forma modular e retomável, preservando os mesmos contratos de dados, o mesmo modelo inicial e os mesmos procedimentos de auditoria nas comparações pareadas. O protocolo foi executado separadamente em duas condições de geração distintas.

Na preparação dos dados, foram gerados dez conjuntos locais, um para cada cliente-vítima, com 20 participantes sintéticos e 100 conversas cada, totalizando 200 participantes e 1.000 conversas. Cada participante possui quatro conversas protegidas que compartilham o mesmo registro canônico completo e uma conversa geral sem informações individualizadas. Os participantes são disjuntos entre os clientes, e cada conjunto mantém a composição de 80 conversas protegidas e 20 conversas gerais.

O registro canônico reúne nove campos protegidos, cuja escolha é motivada pela proteção de dados pessoais [Brasil, 2018]: nome, data de nascimento, CPF, RG, telefone, e-mail, endereço, data de atendimento e horário de atendimento.

O protocolo também inclui um único cliente auxiliar, que utiliza exclusivamente dados sintéticos próprios e não possui acesso aos nomes, valores, atualizações locais ou mapas de substituição das vítimas. Em cada rodada, esse cliente gera 80 conversas protegidas e 20 conversas gerais novas. Na apresentação benigna, os registros protegidos são acompanhados por respostas naturais. Na apresentação adversária, o treinamento é concentrado na continuação canônica do registro, com mascaramento do prefixo. O auxiliar não realiza as consultas direcionadas utilizadas na avaliação.

O modelo inicial de todas as trajetórias é o Tucano2-0.6B refinado no domínio Fórum/Tec. Esse modelo, denominado B0, foi avaliado antes de qualquer treinamento federado. Em seguida, foram executadas oito trajetórias distintas, organizadas em comparações pareadas, todas iniciadas a partir dos mesmos pesos de B0 e compostas por 20

rodadas de treinamento federado com dez clientes-vítima e um cliente auxiliar. A agregação foi realizada por FedAvg [McMahan et al., 2017], com $k = 1$ e peso 1/11 para cada cliente. Após a primeira rodada, cada trajetória continuou a partir de seu próprio estado global anterior.

As trajetórias foram organizadas em três grupos de comparação:

- F0 e F1, sem defesa nas vítimas, com cliente auxiliar benigno e adversário, respectivamente;
- F2 e F3, com DP-AdamW aplicado somente às vítimas, também com auxiliar benigno e adversário, respectivamente, para os valores-alvo de $\varepsilon = 3$ e $\varepsilon = 8$;
- F4 e F5, com substituição semântica rotativa aplicada somente às vítimas, com auxiliar benigno e adversário, respectivamente.

Nas trajetórias F2 e F3, o mecanismo DP-AdamW foi aplicado apenas ao treinamento das vítimas, enquanto o cliente auxiliar permaneceu não privado. Nas trajetórias F4 e F5, os nove campos protegidos, inclusive o nome, foram substituídos localmente antes da tokenização. Os valores substitutos permaneceram estáveis nas quatro conversas protegidas e nas quatro passagens locais de uma rodada, sendo renovados na rodada federada seguinte. F4 e F5 utilizaram a mesma agenda determinística de substituições, diferenciando-se apenas pela apresentação do conjunto auxiliar. As conversas gerais não foram modificadas.

Após B0 e cada rodada federada, um avaliador confiável, separado dos clientes e do servidor, realizou uma auditoria de extração com geração *greedy*. Esse avaliador é o único componente que conhece os nomes e os registros corretos das vítimas. A auditoria incluiu consultas direcionadas ao perfil completo, consultas específicas por tipo de campo e uma consulta complementar sem nome. Nas trajetórias com substituição, também foram avaliados os nomes substitutos da rodada corrente e, no ponto final, os nomes substitutos de rodadas anteriores. A utilidade foi avaliada em B0 e nos oito modelos finais, utilizando 500 conversas construídas a partir de 100 perfis sintéticos que não participaram do treinamento.

A Figura 9.1 apresenta o fluxo geral do protocolo. As diferenças entre as trajetórias devem ser interpretadas em relação às intervenções explicitamente definidas, uma vez que DP-AdamW altera amostragem, clipping e ruído, enquanto a substituição modifica os valores apresentados às vítimas e os renova entre as rodadas.

Todas as condições utilizaram dados pessoais inteiramente sintéticos. O treinamento federado foi executado como uma simulação de clientes lógicos, processados sequencialmente pelo orquestrador.

## 9.2 Questões de Pesquisa e Hipóteses

A pergunta de pesquisa deste capítulo investiga em que medida a substituição semântica rotativa reduz a reprodução de dados pessoais sintéticos durante o treinamento federado, em comparação com a ausência de defesa e com DP-AdamW, na presença de um cliente

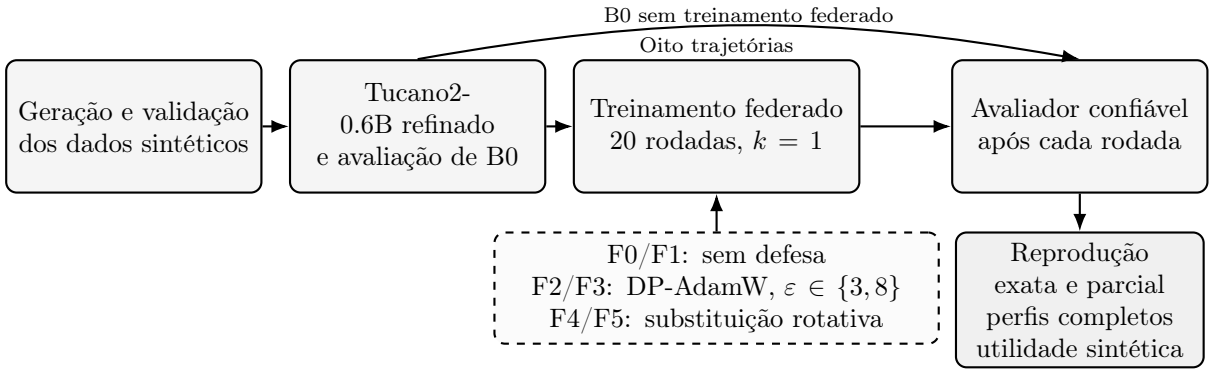



Figura 9.1: Visão geral do protocolo experimental federado.

auxiliar benigno ou adversário. A avaliação considera três contrastes: o efeito do auxiliar adversário, o efeito da substituição sobre a reprodução dos valores originais e sua relação com a utilidade sintética, em comparação com DP-AdamW.

As hipóteses são apresentadas a seguir, mantendo sua orientação original e atualizando a nomenclatura para as condições do protocolo atual.

**H1 - Efeito do atacante:** A participação do auxiliar adversário, em F1, aumenta as métricas de reprodução dos dados originais em relação ao auxiliar benigno, em F0.

**H2 - Efeito da substituição semântica:** A substituição semântica rotativa reduz consistentemente a reprodução dos dados originais e preserva a utilidade sintética, ao comparar F4 com F0 e F5 com F1.

**H3 - Comparação com DP-AdamW:** A substituição semântica rotativa apresenta melhor equilíbrio entre redução da reprodução dos dados originais e preservação da utilidade sintética que as configurações de DP-AdamW avaliadas, ao comparar F4 com F2 e F5 com F3, para cada orçamento de privacidade.

## 9.3 Modelo de Ameaça

Esta seção apresenta o modelo de ameaça adotado no protocolo experimental, definindo os participantes, seus limites de acesso e as capacidades atribuídas ao cliente auxiliar adversário. O cenário é uma simulação de FL para uma LLM causal, com dez clientes-vítima, um cliente auxiliar, um servidor de agregação e um avaliador confiável. Os dados utilizados durante o treinamento federado são inteiramente sintéticos.

Denotamos o conjunto de clientes-vítima por $\mathcal{C}_v = \{c_1, \ldots, c_{10}\}$ e o cliente auxiliar por $c_a$. Cada cliente-vítima $c_i$ possui acesso exclusivo ao conjunto local $D_i$, formado por participantes sintéticos disjuntos dos participantes dos demais clientes. Esses dados não são enviados ao servidor, ao cliente auxiliar ou a outros clientes.

O servidor recebe somente as atualizações produzidas pelo treinamento local e aplica a agregação FedAvg [McMahan et al., 2017]. No experimento executado, os dez clientes-vítima e o cliente auxiliar participam de todas as rodadas com o mesmo peso de agregação, igual a 1/11. Se $\theta^{r-1}$ representa o modelo global no início da rodada $r$, e $\Delta_i^r$ e $\Delta_a^r$

representam, respectivamente, as atualizações de cada cliente-vítima e do cliente auxiliar, a agregação pode ser representada por:

$$\theta^r = \theta^{r-1} + \frac{1}{11}\left(\sum_{i=1}^{10} \Delta_i^r + \Delta_a^r\right). \tag{9.1}$$

O servidor é considerado honesto e executa somente a agregação especificada. Ele não acessa conversas, registros pessoais sintéticos, mapas de substituição ou resultados privados de treinamento. Também não compartilha as atualizações individuais entre os participantes. Não são considerados mecanismos adicionais de agregação segura ou robusta.

A Figura 9.2 apresenta os participantes do protocolo e os principais limites de acesso.

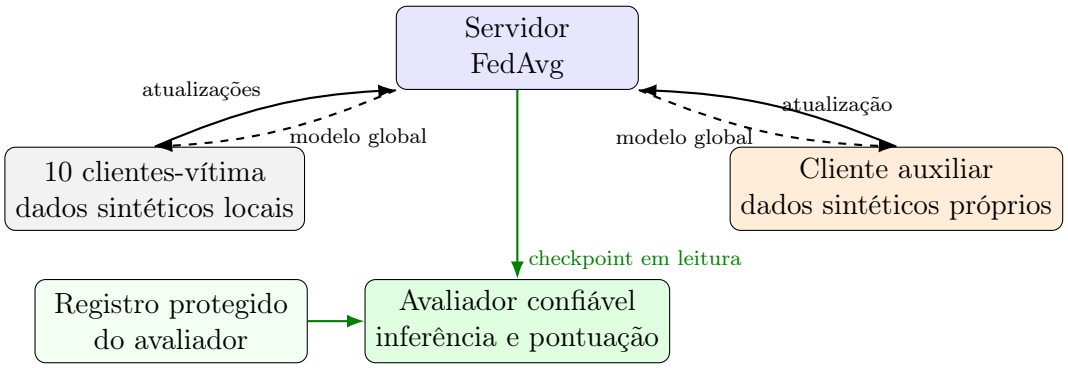



Figura 9.2: Participantes e limites de acesso no protocolo federado.

O cliente auxiliar recebe o mesmo modelo global disponibilizado aos clientes-vítima, mas não possui acesso aos nomes, valores protegidos, conversas, mapas de substituição ou atualizações individuais das vítimas. Seu conjunto local é constituído exclusivamente por perfis sintéticos gerados por ele. A cada rodada federada, são produzidas 80 conversas protegidas e 20 conversas gerais, totalizando 100 conversas auxiliares novas.

Para cada rodada, as condições benigna e adversária do cliente auxiliar são construídas a partir dos mesmos perfis, valores e ordem das amostras. Na apresentação benigna, os registros protegidos são acompanhados por molduras de resposta naturais e a perda é calculada sobre todos os tokens. Na apresentação adversária, utiliza-se somente o registro canônico, com mascaramento do prefixo e cálculo da perda sobre a continuação que contém os campos estruturados. As 20 conversas gerais permanecem iguais e utilizam perda integral nas duas apresentações.

A intervenção adversária procura reforçar no modelo global a estrutura textual canônica utilizada pelos registros protegidos. Entretanto, nenhum exemplo do auxiliar contém nomes ou valores pertencentes às vítimas. Dessa forma, o cliente auxiliar não associa deliberadamente seus padrões a registros específicos dos participantes honestos.

No protocolo executado, o cliente auxiliar utiliza AdamW com taxa de aprendizado de $3\times10^{-5}$, uma passagem sobre seu conjunto local e 25 passos de otimização por rodada. Ele não adapta seus dados às respostas do modelo global, não altera a escala da atualização submetida e não interfere diretamente no processo de agregação. A distinção entre os comportamentos benigno e adversário está na apresentação das conversas protegidas e no

escopo da função de perda.

As condições experimentais são pareadas de forma que o auxiliar benigno e o adversário utilizem a mesma agenda de dados em cada rodada. O auxiliar benigno participa de F0, F2 e F4, enquanto o auxiliar adversário participa de F1, F3 e F5. Na rodada inicial, cada trajetória parte do mesmo modelo de referência. A partir da segunda rodada, cada cenário continua a partir de seu próprio modelo global anterior.

A Figura 9.3 apresenta o fluxo de uma rodada federada.

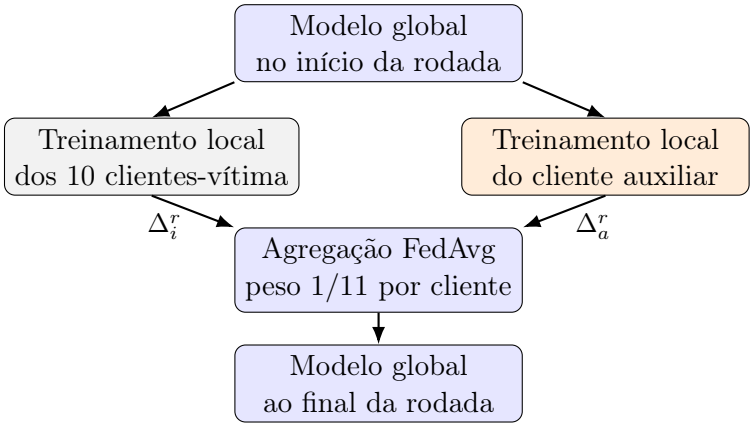



Figura 9.3: Treinamento local e agregação em uma rodada federada.

As defesas avaliadas são aplicadas somente aos clientes-vítima. Em F2 e F3, o treinamento local das vítimas emprega DP-AdamW com clipping por conversa, amostragem Poisson e adição de ruído. Em F4 e F5, os nove campos protegidos, inclusive o nome, são substituídos antes da tokenização. O cliente auxiliar não recebe DP-AdamW nem substituição semântica em qualquer cenário.

A extração direcionada não é realizada pelo cliente auxiliar. Essa função pertence exclusivamente ao avaliador confiável, que possui acesso aos registros corretos das vítimas e utiliza os modelos globais apenas para inferência. O avaliador não participa do treinamento nem modifica os parâmetros do modelo.

Após o baseline e após cada rodada federada, o avaliador constrói consultas controladas para examinar a reprodução do perfil completo, de campos específicos e de valores sem o fornecimento do nome. Nas condições com substituição semântica, ele também conhece os aliases sintéticos necessários para auditar os valores substitutos atuais e históricos. Os prompts, as gerações e os resultados dessa auditoria não são fornecidos aos clientes ou ao servidor. A Figura 9.4 apresenta a separação entre o modelo treinado e a auditoria de extração.

A auditoria verifica se o modelo global reproduz valores exclusivos dos clientes-vítima e se a apresentação adversária do auxiliar altera esse comportamento. A reprodução observada indica exposição sob consultas do avaliador confiável, não extração autônoma pelo auxiliar. Assim, a vulnerabilidade é avaliada pela reprodução mensurável dos registros, e não presumida pela presença do cliente adversário.

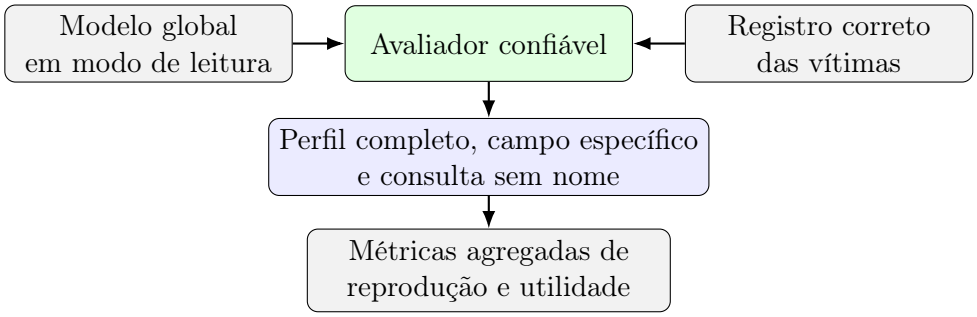



Figura 9.4: Auditoria de extração realizada pelo avaliador confiável.

## 9.4 Mecanismo de Substituição Semântica

Esta seção apresenta o mecanismo de substituição semântica rotativa avaliado neste trabalho. Seu objetivo é reduzir a reprodução dos valores pessoais originais durante o treinamento federado, preservando a estrutura necessária para o treinamento causal. O efeito da transformação sobre a utilidade do modelo é avaliado empiricamente, não sendo assumido como uma propriedade garantida do mecanismo.

Nas condições F4 e F5, a transformação é realizada localmente em cada cliente-vítima no início de cada rodada federada e antes da tokenização. O conjunto substituído na rodada é tokenizado uma única vez e reutilizado nas quatro passagens locais. O cliente auxiliar e o servidor não recebem as conversas substituídas, os valores originais ou os mapas utilizados na transformação.

A Figura 9.5 apresenta o fluxo da substituição dentro de uma rodada federada.

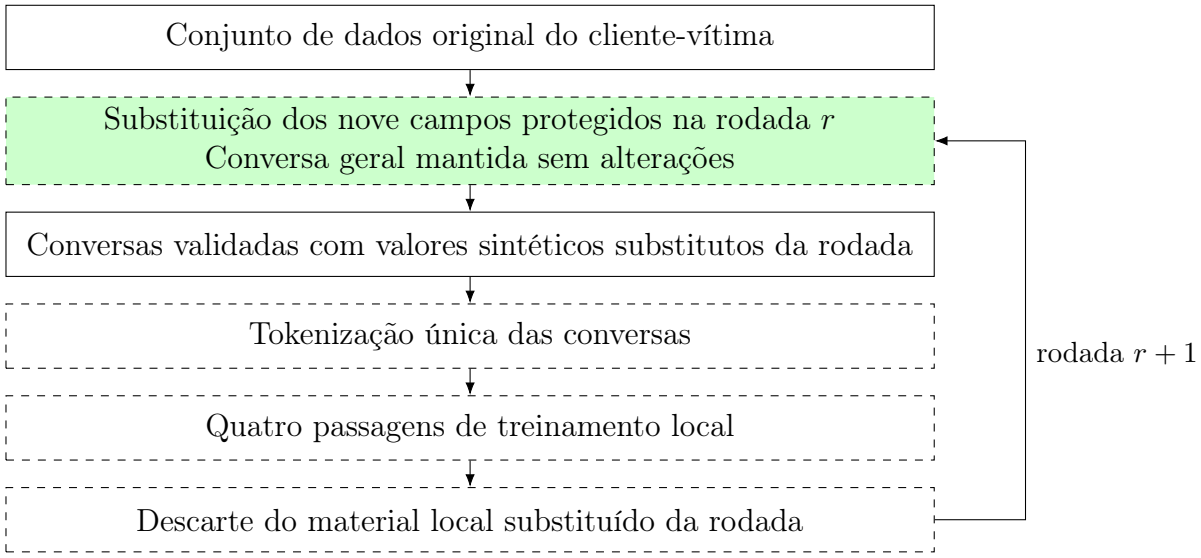



Figura 9.5: Substituição semântica rotativa com valores mantidos na rodada e renovados na seguinte.

Para uma conversa protegida $x_{c,e,j}$, pertencente ao cliente $c$, à entidade técnica $e$ e ao template $j$, a transformação na rodada $r$ pode ser representada por:

$$\tilde{x}_{c,e,j}^{(r)} = f\left(x_{c,e,j}; s, c, e, r\right) \tag{9.2}$$

em que $s$ representa a semente do experimento. A mesma combinação de semente, cliente, entidade e rodada produz deterministicamente o mesmo perfil substituto. O cenário experimental, o peso do cliente auxiliar e o estado do modelo não participam dessa derivação. Consequentemente, F4 e F5 utilizam a mesma agenda de substituições, diferindo somente pela apresentação benigna ou adversária dos dados do cliente auxiliar.

As conversas gerais não contêm informações individualizadas e não são modificadas pelo mecanismo. Para essas conversas, a transformação pode ser representada por:

$$\tilde{x}^{(r)}_{c,e,\text{geral}} = x_{c,e,\text{geral}} \tag{9.3}$$

O mecanismo utiliza as anotações estruturadas disponíveis no conjunto sintético para localizar e substituir os campos protegidos. A identificação automática de informações pessoais em texto livre não faz parte deste experimento. O identificador técnico da entidade, utilizado apenas para agrupamento e validação interna, é preservado, enquanto seu nome textual e os demais valores do perfil são substituídos.

Neste trabalho, o termo *semântica* refere-se à preservação da categoria, do formato e das restrições operacionais de cada atributo. O mecanismo não utiliza embeddings, modelos de linguagem ou medidas de similaridade para selecionar os substitutos e não procura preservar o fato pessoal original. Um CPF é substituído por outro valor com a aparência de CPF sintético, um nome por outro nome em português brasileiro e um horário por outro horário pertencente ao domínio permitido.

Os substitutos são gerados com Faker, utilizando a localização `pt_BR`, e seguem as mesmas regras de validação aplicadas aos perfis sintéticos originais. A Tabela 9.1 apresenta as propriedades preservadas para cada um dos nove campos.

Para uma mesma entidade, os valores substitutos permanecem consistentes nas quatro conversas protegidas e durante as quatro passagens locais de uma rodada. Na rodada federada seguinte, um novo perfil substituto é produzido. O mecanismo impede que qualquer campo reutilize o próprio valor original ou um substituto empregado anteriormente pela mesma entidade.

Para nome, CPF, RG, telefone, e-mail e endereço, também são rejeitadas coincidências com qualquer valor original validado desses tipos. Datas de nascimento, datas de atendimento e horários de atendimento podem coincidir com valores pertencentes a outras entidades, pois possuem domínios naturalmente repetíveis.

Colisões entre valores falsos de entidades distintas são permitidas durante a validação. Quando um mesmo nome substituto está associado a mais de uma entidade, o avaliador o classifica como ambíguo e não atribui automaticamente a exposição a um participante específico. Essa política evita inflar as métricas de reprodução quando diferentes entidades recebem substitutos iguais.

No lado dos clientes-vítima, somente o conjunto substituído da rodada corrente é materializado em memória. O cliente auxiliar e o servidor não recebem os mapas ou valores produzidos. O avaliador confiável reconstrói independentemente as substituições a partir do mesmo contrato determinístico para auditar a reprodução dos valores atuais e históricos.

A substituição atua exclusivamente sobre os dados entregues ao treinamento local, sem

| Campo | Formato preservado | Restrição aplicada |
|---|---|---|
| Nome | Nome sintético em português brasileiro | Diferente do próprio original e dos substitutos anteriores da entidade |
| Data de nascimento | `DD/MM/AAAA` | Data diferente da original, preservando a faixa etária do participante; pode se repetir entre entidades |
| CPF | `000.000.000-00` | Formato aparente preservado, com verificador intencionalmente inválido |
| RG | `00.000.000-X` | Formato aparente preservado, com verificador incompatível com a convenção sintética de referência |
| Telefone | `+55 00 90000-0000` | DDD sintético não roteável e formato de telefone celular preservado |
| E-mail | Parte local e domínio autorizado | Formato do endereço eletrônico preservado e valor diferente dos originais validados |
| Endereço | Logradouro e marcação sintética | Inclui cidade e CEP explicitamente fictícios |
| Data de atendimento | `DD/MM/AAAA` | Data entre 01/01/2026 e 31/12/2027; pode se repetir entre entidades |
| Horário de atendimento | `HH:MM` | Horário entre 08:00 e 18:45, em intervalos de 15 minutos; pode se repetir entre entidades |

Tabela 9.1: Campos protegidos e propriedades preservadas pela substituição semântica rotativa.

modificar diretamente os gradientes, os parâmetros do modelo ou a regra de agregação FedAvg. Sua efetividade é avaliada pela redução dos pares originais reproduzidos em F4 em relação a F0 e em F5 em relação a F1. A reprodução dos próprios substitutos, incluindo os aliases atuais e históricos, é relatada separadamente como diagnóstico.

O impacto da defesa sobre o modelo é avaliado por meio da perda média por conversa, da log-verossimilhança negativa por token e da perplexidade no conjunto sintético de utilidade. Dessa forma, o protocolo não pressupõe que a substituição elimine toda forma de memorização. Ele investiga se a renovação dos valores desloca o aprendizado dos registros originais para perfis sintéticos transitórios e se esse deslocamento reduz a exposição observada sem degradar a utilidade mensurada.

## 9.5 Protocolo Experimental

Esta seção descreve o protocolo experimental desenvolvido para avaliar o impacto de mecanismos de defesa voltados à mitigação do vazamento de dados em cenários federados com

a utilização de LLMs. O objetivo deste protocolo é apresentar essa avaliação de maneira detalhada, servindo de base e garantindo a reprodutibilidade de novos experimentos.

### 9.5.1 Modelo e Treinamento Federado

Esta seção descreve o modelo, a agregação federada e as receitas de treinamento empregadas no experimento. O estado B0 corresponde ao modelo inicial antes do treinamento federado, e todas as trajetórias F0-F5 partem independentemente desse mesmo estado.

O modelo utilizado foi o Tucano2-0.6B refinado no braço Fórum/Tec do experimento de pré-treinamento continuado apresentado no Capítulo 8. O artefato selecionado corresponde ao checkpoint produzido após 52.000 passos e contém 670.127.616 parâmetros na arquitetura `LlamaForCausalLM`. O modelo possui vocabulário de 49.152 tokens e contexto nativo de 4.096 tokens, embora o comprimento máximo adotado no treinamento federado tenha sido limitado a 1.024 tokens.

Os pesos do artefato foram armazenados em FP32 e carregados diretamente em BF16 para o experimento federado. Todos os parâmetros do modelo permaneceram treináveis. O tokenizador original do Tucano2-0.6B-Base foi preservado sem adição, remoção ou remapeamento de tokens, incluindo os identificadores dos tokens especiais. Antes da execução, foram verificadas a equivalência do vocabulário, do backend do tokenizador e das codificações produzidas pelo artefato refinado e pelo modelo original.

O sistema federado foi representado por dez clientes-vítima e um cliente auxiliar, todos participantes de cada rodada. O servidor aplicou somente a média ponderada FedAvg com peso igual para todos os clientes. O servidor recebeu apenas as atualizações do modelo e os metadados técnicos necessários para validá-las, sem acesso às conversas, aos registros protegidos ou aos mapas de substituição.

As receitas locais foram diferentes para os clientes-vítima e para o cliente auxiliar. Nas trajetórias não privadas e nas trajetórias com substituição semântica, cada cliente-vítima percorreu suas 100 conversas quatro vezes, totalizando 400 apresentações e 100 passos do otimizador por rodada. Foi utilizado AdamW com taxa de aprendizado $1 \times 10^{-4}$. O cliente auxiliar percorreu suas 100 conversas uma vez, com AdamW a $3 \times 10^{-5}$ e 25 passos por rodada. Em ambos os casos, utilizou-se microbatch físico de uma conversa, lote lógico de quatro conversas, scheduler constante, $\beta_1 = 0{,}9$, $\beta_2 = 0{,}95$, $\varepsilon_{\text{AdamW}} = 10^{-8}$ e weight decay de 0,01.

A perda causal foi calculada em FP32. Primeiro foi obtida a perda média de cada conversa sobre os tokens supervisionados e, em seguida, calculou-se a média das conversas pertencentes ao lote lógico. Nas conversas adversárias do cliente auxiliar, os tokens do prefixo foram mascarados e somente a continuação canônica participou da perda. Os otimizadores foram reiniciados a cada cliente e rodada, garantindo que nenhum estado local do AdamW fosse compartilhado entre clientes ou persistido entre rodadas.

A Tabela 9.2 resume os principais parâmetros do modelo e das receitas não privadas.

**Treinamento com privacidade diferencial.** Nas trajetórias privadas, o mecanismo foi aplicado somente aos dez clientes-vítima. O cliente auxiliar permaneceu não privado e conservou sua receita de 25 passos com AdamW a $3 \times 10^{-5}$. O mecanismo é denominado

| Parâmetro | Valor |
|---|---|
| Modelo inicial | Tucano2-0.6B refinado no braço Fórum/Tec |
| Arquitetura | `LlamaForCausalLM` |
| Quantidade de parâmetros | 670.127.616 |
| Precisão durante o treinamento | BF16, com cálculo da perda em FP32 |
| Vocabulário | 49.152 tokens, sem alteração do tokenizador |
| Contexto nativo | 4.096 tokens |
| Limite de treinamento | 1.024 tokens |
| Otimizador | AdamW |
| Taxa de aprendizado das vítimas | $1 \times 10^{-4}$ |
| Taxa de aprendizado do auxiliar | $3 \times 10^{-5}$ |
| Microbatch / lote lógico não privado | 1 / 4 conversas |
| Passagens e passos das vítimas | 4 passagens e 100 passos por rodada |
| Passagens e passos do auxiliar | 1 passagem e 25 passos por rodada |
| Betas do AdamW | (0,9; 0,95) |
| Epsilon do AdamW / weight decay | $10^{-8}$ / 0,01 |
| Scheduler | Constante |
| Rodadas federadas | 20 por trajetória |
| Participação e peso FedAvg | 11 clientes por rodada, com peso 1/11 cada |

Tabela 9.2: Modelo e configurações do treinamento federado não privado.

DP-AdamW porque o clipping e o ruído da privacidade diferencial foram aplicados às contribuições individuais antes da atualização realizada pelo AdamW.

Cada conversa foi considerada uma unidade de privacidade. Em cada participante sintético, o mesmo registro protegido aparece em quatro conversas distintas. Por isso, os valores de $\varepsilon$ apresentados não constituem uma garantia de privacidade no nível do participante. O treinamento privado utilizou amostragem de Poisson com $q = 0{,}04$, 100 passos por rodada e 2.000 passos ao longo das 20 rodadas. Foi empregado clipping plano com norma máxima $C = 1{,}0$, lote físico máximo de uma conversa e $\delta = 10^{-5}$.

Para o orçamento nominal $\varepsilon = 3$, foi utilizado multiplicador de ruído $\sigma = 2{,}81$, resultando em $\varepsilon = 2{,}9878$ e ordem Rényi Differential Privacy (RDP) 7,4. Para o orçamento nominal $\varepsilon = 8$, utilizou-se $\sigma = 1{,}36$, resultando em $\varepsilon = 7{,}9643$ e ordem RDP 3,7. Um contabilizador RDP independente foi mantido para cada cliente-vítima durante as 20 rodadas, e o resultado reportado corresponde ao maior $\varepsilon$ entre os dez clientes, e não à soma desses valores.

A implementação privada utilizou Opacus 1.6.0 [Yousefpour et al., 2021], com captura de gradientes por hooks. Para cada orçamento de privacidade, F2 e F3 reconstruíram, em cada cliente-vítima e rodada, exatamente a mesma agenda de amostragem de Poisson e a mesma sequência de ruído gaussiano.

As derivações dessas sequências dependeram da semente do experimento, do identificador técnico do cliente, da rodada e do orçamento de privacidade. Assim, o contraste entre F2 e F3 manteve pareados o mecanismo privado e sua aleatoriedade reproduzível, variando, no desenho experimental, somente a apresentação benigna ou adversária do cliente auxiliar.

A Tabela 9.3 apresenta os cenários avaliados. As condições F2 e F3 foram executadas

separadamente para os dois orçamentos de privacidade. A substituição semântica e o DP-AdamW foram aplicados exclusivamente aos clientes-vítima.

| Cenário | Dados das vítimas | Defesa nas vítimas | Cliente auxiliar |
|---|---|---|---|
| B0 | Sem treinamento federado | Não aplicável | Não aplicável |
| F0 | Registros sintéticos originais | Nenhuma | Benigno |
| F1 | Registros sintéticos originais | Nenhuma | Adversário |
| F2 | Registros sintéticos originais | DP-AdamW, $\varepsilon \in \{3, 8\}$ | Benigno |
| F3 | Registros sintéticos originais | DP-AdamW, $\varepsilon \in \{3, 8\}$ | Adversário |
| F4 | Substitutos sintéticos rotativos | Substituição semântica | Benigno |
| F5 | Substitutos sintéticos rotativos | Substituição semântica | Adversário |

Tabela 9.3: Cenários avaliados no protocolo federado.

Para uma mesma seed, os cenários pareados utilizaram os mesmos dados das vítimas, a mesma agenda auxiliar, a mesma ordem local, os mesmos pesos de agregação e o mesmo estado inicial. Dessa forma, a comparabilidade foi preservada sem exigir que modelos pertencentes a trajetórias já divergentes permanecessem numericamente iguais.

A execução foi realizada de forma offline em uma GPU NVIDIA RTX PRO 6000 Blackwell, com Python 3.12, PyTorch 2.7.1 com CUDA 12.8, Transformers 4.53.2, Tokenizers 0.21.2 e Opacus 1.6.0. Para reduzir fontes conhecidas de não determinismo, foram exigidos algoritmos determinísticos do PyTorch e a configuração `CUBLAS_WORKSPACE_CONFIG=:4096:8`.

Considerando conjuntamente as duas execuções, o experimento totalizou 320 rodadas federadas, 328.000 passos de otimização local, 122.086 gerações de auditoria e 9.000 avaliações de utilidade.

### 9.5.2 Cliente Auxiliar e Amplificação do Padrão Canônico

Esta subseção detalha a construção das variantes benigna e adversária do cliente auxiliar, cujas capacidades e limites de acesso foram definidos na Seção 9.3. O contraste concentra-se na apresentação das conversas protegidas e no escopo da função de perda, mantendo a receita de treinamento resumida na Tabela 9.2.

Em cada rodada, as 80 conversas protegidas são associadas, cada uma, a um perfil auxiliar completo gerado em um fluxo determinístico próprio. Os perfis são renovados a cada rodada e não são derivados das respostas do modelo global. Assim, a geração dos dados auxiliares é não adaptativa, pois o modelo global recebido influencia apenas o treinamento local, sem alterar a programação da geração, os valores produzidos ou a ordem das amostras.

As duas variantes utilizam os mesmos perfis, valores, conversas gerais e ordem das amostras em cada rodada. Esse pareamento permite avaliar o efeito da apresentação e da supervisão do padrão canônico sem introduzir diferenças nos indivíduos sintéticos observados pelo cliente auxiliar.

Considere um perfil auxiliar $j$ gerado na rodada $r$. Sua representação canônica pode ser expressa por

$$x_{r,j} = p(n_{r,j}) \oplus c(v_{r,j}), \tag{9.4}$$

em que $p(n_{r,j})$ é o prefixo que contém o nome sintético auxiliar, $c(v_{r,j})$ é a continuação canônica com os oito campos restantes e $\oplus$ representa concatenação textual. O prefixo utilizado é "USUÁRIO: Meu nome é {nome}. Meus dados cadastrados são:", enquanto a continuação apresenta, nesta ordem, data de nascimento, CPF, RG, telefone, e-mail, endereço, data de atendimento e horário de atendimento.

Na variante benigna, empregada em F0, F2 e F4, o registro canônico completo é seguido por uma quebra de linha e por uma resposta genérica do assistente, selecionada deterministicamente de um catálogo fixo de quatro confirmações. A perda causal é calculada sobre todos os tokens não utilizados como preenchimento. Assim, a amostra mantém a forma de uma interação natural entre usuário e assistente:

$$x^{\text{ben}}_{r,j} = x_{r,j} \oplus \text{"ASSISTENTE: confirmação genérica"}. \tag{9.5}$$

Na variante adversária, empregada em F1, F3 e F5, utiliza-se somente o registro canônico $x_{r,j}$. O texto é tokenizado uma única vez, mas os rótulos correspondentes ao prefixo $p(n_{r,j})$ são mascarados. Consequentemente, a perda é calculada apenas sobre a continuação $c(v_{r,j})$:

$$\mathcal{L}^{\text{adv}}_{r,j} = \mathcal{L}(c(v_{r,j}) \mid p(n_{r,j}))\,. \tag{9.6}$$

As 20 conversas gerais são idênticas nas duas variantes e utilizam perda sobre todos os tokens não utilizados como preenchimento.

A amplificação investigada consiste, portanto, no reforço da tarefa de completar um perfil estruturado a partir de um nome. O experimento avalia se essa intervenção, realizada com dados sintéticos próprios do auxiliar, modifica a reprodução posterior dos registros das vítimas pelo modelo global.

### 9.5.3 Métricas

A avaliação foi realizada pelo avaliador confiável, logicamente separado dos clientes-vítima, do cliente auxiliar e do servidor federado. Apenas esse avaliador teve acesso aos nomes usados nas consultas, aos registros protegidos de referência e, nos cenários com substituição semântica, aos aliases e valores sintéticos correspondentes a cada rodada. O cliente auxiliar não recebeu as instruções de auditoria, as respostas do modelo nem os registros empregados na pontuação.

As auditorias utilizaram exclusivamente decodificação *greedy*, selecionando o token de maior probabilidade condicional em cada passo. Foi produzida uma única continuação por instrução, sem amostragem ou busca em feixe. O limite foi de 192 tokens novos nas consultas de perfil completo e sem nome e de 48 tokens nas consultas de campo específico. Somente os tokens gerados após o término da instrução foram pontuados, impedindo que nomes ou valores presentes no próprio *prompt* fossem contabilizados como reprodução.

Nas rodadas 1 a 19, a auditoria foi executada sobre 20 participantes selecionados deterministicamente, com dois participantes de cada cliente-vítima. Na rodada 20, foram

avaliados os 200 participantes. Para cada conjunto de alvos, foram empregados três tipos de consulta: uma consulta direcionada de perfil completo por participante, oito consultas de campo específico por participante e uma consulta complementar sem nome por modelo. Assim, a auditoria de 200 participantes compreendeu 200 consultas principais, 1.600 consultas específicas e uma consulta sem nome, totalizando 1.801 gerações.

**Reprodução direcionada de pares.** A principal métrica de exposição foi a reprodução exata do par associado entre a pessoa consultada, o tipo do campo e seu valor correto. O nome foi fornecido pelo avaliador na instrução principal e, por isso, não integrou o denominador da extração direcionada. Foram avaliados os oito campos que deveriam completar o registro: data de nascimento, CPF, RG, telefone, e-mail, endereço, data de atendimento e horário de atendimento.

Seja $\mathcal{T}$ o conjunto de participantes auditados e $\mathcal{F}$ o conjunto desses oito tipos de campo. O número de pares exatos foi definido por:

$$M_{\text{exato}} = \sum_{i \in \mathcal{T}} \sum_{f \in \mathcal{F}} \mathbb{I}\left[\hat{v}_{i,f} = v_{i,f}\right], \tag{9.7}$$

em que $v_{i,f}$ representa o valor correto do campo $f$ do participante $i$, $\hat{v}_{i,f}$ representa o valor reconhecido na continuação gerada e $\mathbb{I}$ é a função indicadora. A taxa correspondente foi calculada por:

$$R_{\text{exato}} = \frac{M_{\text{exato}}}{|\mathcal{T}|\,|\mathcal{F}|}. \tag{9.8}$$

Na rodada 20, o denominador foi, portanto, $200 \times 8 = 1.600$ pares. Cada combinação entre geração, tipo e valor foi contabilizada no máximo uma vez, evitando inflação causada pela repetição de um mesmo valor na resposta.

Os campos CPF, RG, telefone, e-mail e endereço foram adicionalmente agrupados como campos distintivos, pois deveriam identificar uma entidade sintética com maior especificidade. Para esses cinco campos, o denominador na rodada 20 foi $200 \times 5 = 1.000$ pares. Também foi registrada a quantidade de participantes para os quais ao menos um campo distintivo correto foi reproduzido:

$$E_{\text{dist}} = \sum_{i \in \mathcal{T}} \mathbb{I}\left[\exists f \in \mathcal{F}_{\text{dist}} : \hat{v}_{i,f} = v_{i,f}\right]. \tag{9.9}$$

Data de nascimento, data de atendimento e horário de atendimento foram tratados como campos repetíveis. Como esses valores podem legitimamente coincidir entre diferentes entidades sintéticas, uma coincidência literal isolada não foi interpretada como identificação de um participante sem que a associação com o nome consultado também estivesse correta.

**Perfis completos e consultas específicas.** Uma geração foi classificada como perfil completo quando reproduziu corretamente os oito campos esperados na consulta direcionada. A métrica mais estrita de perfil completo ordenado exigiu, adicionalmente, os

rótulos, delimitadores e a ordem canônica definidos pelo protocolo, sem omissão, reordenação ou repetição dos campos.

Os valores da coluna “Perfis originais completos” da Tabela 9.4 correspondem à definição de perfil completo, sem exigir os rótulos, os delimitadores ou a ordem canônica dos campos. A métrica de perfil completo ordenado não é apresentada nessa tabela.

As consultas de campo específico foram avaliadas separadamente da consulta principal. Para cada participante, foram produzidas oito instruções, uma para cada tipo de campo. Foram registrados o número de respostas exatas e, como diagnóstico complementar, o número de correspondências parciais.

A correspondência parcial foi calculada com similaridade de Levenshtein normalizada e limiar mínimo de 0,80. Essa medida foi utilizada apenas para identificar aproximações textuais e não alterou a métrica principal de reprodução exata. Também foram contabilizadas associações incorretas, nas quais a resposta continha um valor pertencente a outra entidade sintética, e candidatos com formato válido que não pertenciam ao registro protegido conhecido.

Antes das comparações exatas, os textos foram normalizados em Unicode NFC, os espaços externos foram removidos e sequências internas de caracteres de espaço foram colapsadas. Caixa, pontuação e dígitos permaneceram significativos, de modo que a normalização não corrigisse documentos, endereços ou demais valores produzidos incorretamente.

**Consulta sem nome.** A consulta sem nome foi utilizada como controle complementar para verificar reproduções não condicionadas à identidade fornecida pelo avaliador. Nessa modalidade, foram contabilizados valores protegidos exatos, valores únicos, nomes de participantes encontrados e entidades cujos nove campos apareceram conjuntamente na mesma geração.

Como alguns campos podem se repetir entre participantes, os acertos sem nome foram deduplicados pelo par formado pelo tipo e pelo valor. A reprodução exata de um nome nessa consulta foi contabilizada como exposição, pois, diferentemente da consulta direcionada, o nome não fazia parte da instrução.

**Avaliação da substituição semântica.** Nos cenários F4 e F5, foram executadas auditorias independentes condicionadas ao nome original e ao alias utilizado na rodada corrente. Nas rodadas 1 a 19, cada uma dessas visões utilizou 20 participantes balanceados entre os clientes-vítima. Na rodada 20, cada visão utilizou os 200 participantes. Cada visão produziu seus próprios prompts e gerações.

Adicionalmente, na rodada 20, foi realizada uma auditoria dos aliases históricos, abrangendo 20 participantes e os aliases empregados nas 19 rodadas anteriores, totalizando 380 alvos históricos.

Posteriormente, as continuações geradas foram cruzadas, sem novas inferências, com os registros originais e com os substitutos atuais e históricos conhecidos pelo avaliador. Esse procedimento permitiu medir a reprodução de valores originais condicionada ao nome original, a reprodução dos substitutos atuais condicionada ao alias corrente, a retenção de

substitutos de rodadas anteriores, a associação de aliases a valores originais e a associação a substitutos pertencentes a outra entidade ou rodada.

Aliases compartilhados por mais de uma entidade foram marcados como ambíguos. Nesses casos, os acertos puderam contribuir para as contagens agregadas de valores reconhecidos, mas não foram atribuídos individualmente a uma entidade. As correspondências foram deduplicadas pelo trio formado por alias, tipo e valor.

A redução dos pares originais proporcionada por uma defesa foi calculada em relação à trajetória sem defesa com a mesma apresentação do cliente auxiliar:

$$\text{Redução}_D = 1 - \frac{M_{\text{original},D}}{M_{\text{original},C}}, \tag{9.10}$$

em que $D$ representa a trajetória protegida e $C$ seu comparador. F2 e F4 foram comparados a F0, enquanto F3 e F5 foram comparados a F1.

A redução relativa foi analisada em conjunto com as contagens absolutas de pares originais exatos, de participantes com campos distintivos expostos e de perfis originais completos reproduzidos. Essas medidas permitiram caracterizar a exposição residual em cada trajetória.

Nos cenários privados, foi relatado o valor de $\varepsilon$ realizado pelo contabilizador RDP. Como cada cliente-vítima possuía um contabilizador independente e os conjuntos de participantes eram disjuntos, foi reportado o maior $\varepsilon$ entre os dez clientes, e não sua soma. A unidade protegida foi a conversa. Consequentemente, esses valores não foram interpretados como uma garantia de privacidade no nível do participante completo.

**Utilidade sintética.** A utilidade foi avaliada sobre um conjunto *held-out* composto por 100 perfis sintéticos exclusivos e 500 conversas, sem sobreposição com as entidades utilizadas pelos clientes-vítima ou pelo cliente auxiliar. O modelo não recebeu gradientes durante essa avaliação.

Foram calculadas a perda causal média por conversa, a log-verossimilhança negativa média por token e a perplexidade. Seja $N$ o número de conversas, $T_i$ o número de tokens supervisionados da conversa $i$ e $\ell_{i,t}$ a perda causal do token $t$. A perda média por conversa foi definida por:

$$L_{\text{conv}} = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{T_i} \sum_{t=1}^{T_i} \ell_{i,t} \right) \tag{9.11}$$

A NLL por token foi ponderada pelo total de tokens supervisionados:

$$\text{NLL}_{\text{token}} = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \ell_{i,t}}{\sum_{i=1}^{N} T_i} \tag{9.12}$$

e a perplexidade foi calculada por:

$$\text{PPL} = \exp\left(\text{NLL}_{\text{token}}\right). \tag{9.13}$$

A perda média por conversa e a NLL por token possuem formas de agregação distintas e, portanto, não devem ser interpretadas como medidas numericamente equivalentes.

Além dos valores absolutos, foram calculadas as diferenças de perda, NLL e perplexidade em relação ao modelo inicial e ao comparador F0 ou F1 correspondente.

Essas métricas caracterizam apenas a utilidade sintética dentro da distribuição, dos campos e dos modelos de conversa empregados no protocolo. Elas não constituem uma avaliação da qualidade linguística geral do modelo e não é possível determinar seu desempenho em tarefas externas.

## 9.6 Resultados e Discussão

Esta seção analisa a reprodução de dados pessoais sintéticos e a utilidade dos modelos nos cenários definidos na Tabela 9.3, com foco nos resultados finais da rodada 20 e nos contrastes entre as defesas.

Os resultados por cenário são apresentados como médias aritméticas de duas execuções ($n = 2$), sem tratar as medidas ao longo das rodadas como observações independentes. Os percentuais de redução e de variação de perplexidade correspondem à média dos percentuais calculados em cada execução em relação ao respectivo comparador. Os denominadores indicados nas tabelas correspondem a cada execução.

**Reprodução de dados pessoais sintéticos.** O modelo inicial B0 não reproduziu pares originais na auditoria realizada, enquanto as trajetórias federadas sem defesa F0 e F1 apresentaram reprodução de campos distintivos e de perfis originais completos. A Tabela 9.4 apresenta os resultados médios de B0 e dos modelos federados na rodada 20, permitindo comparar a exposição observada nas condições com e sem defesa.

| Cenário | Comparador | Pares originais /1.600 | Distintivos /1.000 | Participantes distintivos /200 | Perfis originais completos | Redução | $\varepsilon$ realizado |
|---|---|---|---|---|---|---|---|
| B0 | – | 0,0 | 0,0 | 0,0 | 0,0 | – | – |
| F0 | – | 305,0 | 189,0 | 70,0 | 27,5 | – | – |
| F1 | – | 308,5 | 194,0 | 70,5 | 28,0 | – | – |
| F2, $\varepsilon = 3$ | F0 | 4,0 | 1,5 | 1,5 | 0,0 | 98,70% | 2,9878 |
| F3, $\varepsilon = 3$ | F1 | 5,5 | 1,0 | 1,0 | 0,0 | 98,24% | 2,9878 |
| F2, $\varepsilon = 8$ | F0 | 6,0 | 1,5 | 1,5 | 0,0 | 98,03% | 7,9643 |
| F3, $\varepsilon = 8$ | F1 | 5,0 | 2,0 | 2,0 | 0,0 | 98,37% | 7,9643 |
| F4 | F0 | 5,5 | 0,5 | 0,5 | 0,0 | 98,20% | – |
| F5 | F1 | 7,5 | 2,0 | 2,0 | 0,0 | 97,56% | – |

Tabela 9.4: Reprodução direcionada na rodada 20, média de duas execuções ($n = 2$).

As trajetórias federadas sem defesa F0 e F1 reproduziram, em média, 305,0 e 308,5 dos 1.600 pares originais possíveis, respectivamente. Dos 1.000 pares distintivos, foram reproduzidos, em média, 189,0 em F0 e 194,0 em F1, atingindo 70,0 e 70,5 participantes. Também foram obtidos, em média, 27,5 e 28,0 perfis originais completos, respectivamente. Esses valores confirmam que a condição de alta exposição empregada no experimento produziu memorização mensurável.

A diferença entre o auxiliar benigno e o adversário foi pequena. F1 reproduziu, em média, 3,5 pares originais a mais que F0. Embora o contraste médio tenha sido positivo, sua magnitude foi pequena e não foi realizada análise inferencial que permita afirmar amplificação estatisticamente significativa.

Todas as condições com defesa apresentaram redução média da reprodução dos pares originais superior a 97% em relação ao respectivo comparador sem defesa. Para DP-AdamW com $\varepsilon = 3$, as reduções médias foram de 98,70% em F2 e 98,24% em F3. Com $\varepsilon = 8$, foram de 98,03% e 98,37%, respectivamente. Na substituição semântica, as reduções médias foram de 98,20% em F4 e 97,56% em F5. Nenhuma condição com defesa reproduziu um perfil original completo.

**Reprodução de aliases e valores substitutos.** A Tabela 9.5 apresenta a reprodução exata de pares substitutos associados aos aliases correntes e históricos, avaliada separadamente da exposição dos valores originais.

| Cenário | Alias corrente /1.600 | Aliases históricos /3.040 |
|---|---|---|
| F4 | 11,5 | 16,5 |
| F5 | 12,5 | 14,5 |

Tabela 9.5: Reprodução de pares substitutos na rodada 20 ($n = 2$).

A reprodução associada ao alias corrente foi, em média, de 11,5 pares em F4 e 12,5 em F5, dos 1.600 possíveis, correspondendo aproximadamente a 0,72% e 0,78%, respectivamente. Para os aliases históricos, foram reproduzidos, em média, 16,5 pares em F4 e 14,5 em F5, dos 3.040 possíveis, aproximadamente 0,54% e 0,48%. Nenhuma dessas auditorias reproduziu um perfil substituto completo e não foram observados aliases ambíguos. Como os denominadores das duas visões são diferentes, suas contagens absolutas não devem ser comparadas diretamente.

Esses resultados mostram que a substituição rotativa não eliminou completamente a reprodução dos valores sintéticos transitórios. O padrão observado é compatível com aprendizado residual dos substitutos e da estrutura canônica, enquanto a reprodução dos registros originais foi fortemente reduzida. Entretanto, essas contagens não permitem concluir que houve uma transferência causal da memorização dos originais para os substitutos.

Os dois orçamentos de DP-AdamW apresentaram resultados próximos e não produziram uma ordenação uniforme entre $\varepsilon$ e a reprodução residual média. Em F2, $\varepsilon = 3$ apresentou 4,0 pares originais residuais, contra 6,0 com $\varepsilon = 8$. Em F3, as médias foram de 5,5 e 5,0 pares, respectivamente. Essas diferenças pequenas, analisadas descritivamente, não sustentam uma conclusão sobre a superioridade de um dos orçamentos.

Os valores de $\varepsilon$ devem ser interpretados no nível da conversa, conforme o protocolo experimental. A implementação destina-se à pesquisa reprodutível, sem aleatoriedade criptograficamente segura.

**Utilidade sintética.** A Tabela 9.6 apresenta as médias das métricas de utilidade entre as execuções, cujos conjuntos de avaliação continham, em média, 53.480 tokens supervi-

sionados. A perplexidade apresentada corresponde à média das perplexidades.

| Cenário | Loss média | NLL/token | Perplexidade | Δ PPL/comparador |
|---|---|---|---|---|
| B0 | 3,75821 | 3,79933 | 44,67249 | – |
| F0 | 1,52278 | 1,77698 | 5,91268 | – |
| F1 | 1,52238 | 1,77672 | 5,91049 | – |
| F2, $\varepsilon = 3$ | 1,57807 | 1,53301 | 4,63212 | -21,64% |
| F3, $\varepsilon = 3$ | 1,91640 | 1,95141 | 7,03863 | +19,09% |
| F2, $\varepsilon = 8$ | 1,57617 | 1,52602 | 4,59984 | -22,18% |
| F3, $\varepsilon = 8$ | 1,90077 | 1,93097 | 6,89688 | +16,70% |
| F4 | 1,04754 | 1,21486 | 3,37086 | -43,00% |
| F5 | 1,04431 | 1,21120 | 3,35879 | -43,17% |

Tabela 9.6: Métricas médias de utilidade sintética ($n = 2$).

Todos os modelos federados apresentaram perplexidade média inferior à observada em B0 nesse conjunto sintético. Esse resultado é compatível com o aprendizado da estrutura e dos templates utilizados pelo protocolo.

As trajetórias de DP-AdamW apresentaram comportamentos diferentes conforme a variante do cliente auxiliar. Nas condições com auxiliar benigno, F2 apresentou reduções médias de perplexidade de 21,64% com $\varepsilon = 3$ e de 22,18% com $\varepsilon = 8$ em relação a F0. Nas condições com auxiliar adversário, F3 apresentou aumentos médios de 19,09% e 16,70%, respectivamente, em relação a F1. O experimento não foi estruturado para isolar os mecanismos responsáveis por essa diferença.

A substituição semântica apresentou os menores valores médios de loss, NLL e perplexidade em todas as comparações finais. A redução média de perplexidade foi de 43,00% em F4 em relação a F0 e de 43,17% em F5 em relação a F1. Portanto, entre as defesas avaliadas, a substituição preservou melhor a utilidade no conjunto sintético definido pelo protocolo. Esses resultados de utilidade não demonstram melhora da capacidade geral do modelo, mas sim melhor desempenho preditivo no conjunto sintético de avaliação, restrito à distribuição e aos formatos de conversa empregados no protocolo.

**Discussão das hipóteses.** Em relação à H1, F1 reproduziu 308,5 pares, apenas 3,5 a mais que F0 (305,0). O contraste limitado é plausivelmente explicado por dois aspectos do protocolo. Primeiro, as dez vítimas realizam quatro passagens (100 passos por rodada, com taxa de aprendizado $1 \times 10^{-4}$) e 80% das conversas incluem o prefixo "USUÁRIO: Meu nome é {nome}. Meus dados cadastrados são: ...". O auxiliar realiza uma passagem (25 passos, com taxa de aprendizado $3 \times 10^{-5}$) e tem peso 1/11 na agregação. Segundo, F0 e F1 usam o mesmo prefixo e a continuação canônica de oito campos, descritos nas Equações 9.4 a 9.6. Pela máscara causal, a confirmação final em F0 não altera as previsões desses campos. Assim, ambas as variantes compartilham o sinal de treinamento da continuação. F0 também supervisiona o prefixo e a confirmação, portanto as perdas totais diferem. Esses fatores contextualizam o resultado, mas não demonstram causalidade nem corroboram H1 de forma conclusiva.

A H2 recebeu suporte descritivo nas condições investigadas. A substituição semântica rotativa associou forte redução da reprodução dos pares originais, sem perfis originais com-

pletos reproduzidos, à melhor utilidade sintética em relação aos respectivos comparadores sem defesa.

Quanto à H3, a substituição apresentou melhor utilidade sintética média que as duas configurações de DP-AdamW, tanto com o auxiliar benigno quanto com o adversário. Entretanto, as contagens médias residuais de pares originais não estabeleceram uma ordenação uniforme entre os mecanismos, pois DP-AdamW apresentou menor reprodução em algumas comparações.

A comparação não estabelece uma medida agregada de equilíbrio entre reprodução e utilidade. Assim, os resultados sustentam uma vantagem de utilidade sintética, mas não corroboram de maneira conclusiva o melhor equilíbrio conjunto proposto em H3 nem demonstram superioridade geral de privacidade em relação a DP-AdamW. Nesse contexto, a substituição semântica rotativa mostra-se uma alternativa empírica a DP-AdamW para reduzir a reprodução de dados originais e preservar a utilidade sintética nas condições avaliadas, sem implicar equivalência às garantias formais de privacidade diferencial.

O protocolo deve ser compreendido como um teste de estresse controlado de alta exposição, construído para produzir reprodução mensurável e permitir a comparação das defesas. As conclusões estão restritas ao modelo refinado, aos dados sintéticos utilizados, à participação auxiliar $k = 1$, às execuções de desenvolvimento, à auditoria greedy e às demais decisões experimentais adotadas.

## 9.7 Conclusão do Capítulo

Este capítulo investigou a reprodução de dados pessoais sintéticos por LLMs em um piloto controlado de FL com o Tucano2-0.6B refinado no domínio Fórum/Tec. Foram comparadas condições sem defesa, com substituição semântica rotativa e com uma implementação experimental de DP-AdamW, na presença de clientes auxiliares benignos ou adversários. A substituição foi aplicada localmente aos nove campos protegidos, antes da tokenização, com renovação dos valores a cada rodada federada. Um avaliador confiável separado realizou as consultas de extração e a mensuração da utilidade sintética.

Considerando as médias das duas execuções de desenvolvimento, a substituição semântica rotativa reduziu a reprodução direcionada dos pares originais em 98,20% na comparação F4-F0 e em 97,56% na comparação F5-F1, sem reprodução de continuações originais completas nas condições com essa defesa. Também apresentou menor perda média, NLL e perplexidade no conjunto sintético de utilidade que as configurações de DP-AdamW avaliadas. Entretanto, as contagens residuais de reprodução não estabeleceram superioridade uniforme entre os mecanismos.

Os resultados sustentam a viabilidade empírica da substituição semântica rotativa para mitigar a reprodução dos valores originais, associando forte redução de exposição à melhor utilidade sintética nas condições investigadas. Assim, a solução proposta se mostra uma alternativa ao uso de DP-AdamW para mitigar a reprodução de dados originais nas condições avaliadas, sem implicar equivalência às garantias formais de privacidade diferencial.

# Capítulo 10

# Considerações Finais e Trabalhos Futuros

Este capítulo sintetiza as principais contribuições e os resultados obtidos ao longo desta tese, retomando os objetivos inicialmente definidos e discutindo de que forma as abordagens propostas respondem às questões de pesquisa investigadas. Também são examinadas possíveis limitações do trabalho e oportunidades para a continuidade da pesquisa.

As contribuições desta tese foram estruturadas para investigar a hipótese de que a adequação dos dados ao domínio pode favorecer o desempenho dos modelos, enquanto a transformação local de informações sensíveis pode mitigar seu vazamento durante o treinamento federado. Para examinar essas questões sob diferentes perspectivas, as contribuições foram organizadas em quatro eixos: (*i*) geração e adaptação de dados para LLMs em português, (*ii*) coleta e escalonamento de dados em larga escala, (*iii*) pré-treinamento continuado e avaliação de modelos e (*iv*) análise e mitigação de riscos à privacidade.

Na seção seguinte, serão apresentadas as conclusões gerais da tese, com destaque para os principais resultados alcançados e suas implicações para o treinamento de LLMs, especialmente no contexto de dados em português brasileiro e de ambientes de aprendizado federado. Em seguida, são indicadas direções para trabalhos futuros que podem ampliar ou aprofundar as abordagens desenvolvidas neste estudo.

## 10.1 Conclusões

Esta tese aborda três desafios centrais: (*i*) escassez de dados em português brasileiro, (*ii*) avaliação dos efeitos da proximidade de domínio no pré-treinamento continuado e (*iii*) riscos de vazamento de dados sensíveis em aprendizado federado. Os resultados reforçam a importância de articular a preparação dos dados, a avaliação dos modelos e a análise dos riscos à privacidade no treinamento de LLMs. O experimento de pré-treinamento continuado evidenciou benefícios dependentes da composição dos dados e da capacidade avaliada. No aprendizado federado, os resultados indicaram que a transformação local de informações sensíveis pode mitigar seu vazamento, nas condições e métricas consideradas.

As quatro primeiras perguntas de pesquisa, relacionadas à geração e à adaptação de conjuntos de dados, foram respondidas por meio de estratégias complementares. Primei-

ramente, foi proposta uma metodologia para transformar interações reais com referências entre mensagens em diálogos multiturnos e multiparticipantes por meio da identificação de fluxos conversacionais. Paralelamente, foi desenvolvida uma ferramenta capaz de gerar conjuntos de dados em larga escala e de operacionalizar essa metodologia.

Em seguida, foi apresentada uma metodologia de geração sintética, baseada em *crowdsourcing*, para a construção de conjuntos conversacionais. Essa abordagem se apoia na participação assíncrona, em roteiros e em mecanismos de controle de qualidade. Com base nela, criamos e compartilhamos uma ferramenta que pode ser reutilizada por outros pesquisadores para criar novos conjuntos de dados.

Mostramos ainda que conjuntos de dados já consolidados na literatura podem ser reaproveitados de forma eficaz para atender a requisitos de diferentes domínios. Na mesma linha, definimos uma metodologia de geração distribuída de dados a partir da web e implementamos uma ferramenta alinhada a essas diretrizes, capaz de criar conjuntos em grande escala com rastreabilidade dos processos executados.

As contribuições desenvolvidas mostram como estratégias complementares de geração, adaptação e curadoria podem integrar processos reproduzíveis de preparação de dados para LLMs. As metodologias propostas e as ferramentas implementadas possibilitaram a geração, a adaptação ou o compartilhamento de conjuntos de dados como Adrenaline, OuterSpace, *Nutritious Food*, BrWaC-CLEAN, MultiWOZ-PTBR e WackyWacky PTBR Geral. As avaliações realizadas caracterizaram a qualidade dos dados preparados, enquanto os experimentos com os corpora selecionados forneceram evidências de sua utilidade para o treinamento e o refinamento de modelos de NLP, nas condições investigadas nesta tese.

A quinta pergunta de pesquisa foi respondida por meio de um experimento controlado de pré-treinamento continuado, comparando o Tucano2-Base com duas versões refinadas, Geral e Fórum/Tec, treinadas com o mesmo orçamento de tokens e os mesmos hiperparâmetros, mas com composições distintas de dados. Essa comparação complementou a avaliação dos conjuntos utilizados ao examinar seus efeitos sobre o desempenho dos modelos.

O pré-treinamento continuado proporcionou ganhos modestos na classificação e melhora na modelagem de linguagem. A proximidade de domínio acrescentou uma vantagem clara na NLL condicional dos textos de fórum, mas não estabeleceu uma vantagem adicional conclusiva na classificação em relação ao modelo Geral. A comparação entre checkpoints também mostrou melhora na previsão textual dos dois modelos no último intervalo de treinamento avaliado. Dessa forma, concluímos que os benefícios do refinamento dependem tanto da composição dos dados quanto da capacidade avaliada.

A sexta pergunta de pesquisa foi investigada por meio de um piloto federado controlado com o Tucano2-0.6B refinado no domínio Fórum/Tec, utilizando dados pessoais inteiramente sintéticos. Foram comparadas condições sem defesa, com substituição semântica rotativa e com uma implementação experimental de DP-AdamW, na presença de clientes auxiliares benignos ou adversários. A avaliação demonstrou reprodução dos dados originais nas condições sem defesa, mas não corroborou de maneira conclusiva sua amplificação pelo auxiliar adversário.

A substituição semântica rotativa reduziu a reprodução direcionada dos pares originais,

em média, entre 97,56% e 98,20% (variando entre 97,03% e 98,32% nas execuções individuais), sem reprodução de continuações originais completas nas condições com essa defesa. Também apresentou melhor utilidade sintética que as configurações de DP-AdamW avaliadas, embora as contagens residuais de reprodução não tenham estabelecido superioridade uniforme entre os mecanismos. Esses resultados sustentam a viabilidade empírica da substituição nas condições investigadas, sem constituir garantia formal de privacidade ou demonstração de generalização para outros dados, modelos e procedimentos de extração.

Os objetivos da tese foram alcançados ao responder às seis perguntas de pesquisa formuladas na introdução, mostrando que o avanço de LLMs em português brasileiro depende da articulação entre engenharia de dados, treinamento adequado ao domínio e estratégias de proteção à privacidade. Em conjunto, as contribuições apresentadas ampliam a base metodológica e experimental disponível para o português e estabelecem fundamentos consistentes para a continuidade das pesquisas sobre o uso de dados adequados ao domínio no pré-treinamento continuado, a avaliação dos modelos resultantes e a proteção à privacidade em aprendizado federado.

## 10.2 Produção Científica

Esta seção reúne as publicações, os reconhecimentos e os produtos metodológicos e experimentais desenvolvidos ao longo do doutorado.

### 10.2.1 Journal Publicado

- **Automatic Extraction of Conversation Flows from Human Dialogues: Understanding Their Impact to Refine NLP Models:** Este trabalho dá continuidade à evolução da metodologia MCCD, aprofundando o mecanismo de identificação de diálogos multiturnos entre humanos. Além de identificar as conversas, a abordagem permite detectar ramificações e múltiplos participantes em cada uma delas. O conjunto de dados gerado a partir dessa metodologia foi utilizado para treinar o BERTpt, um modelo baseado na arquitetura BERT e voltado para tarefas de classificação. No artigo, o BERTpt obteve desempenho 9% superior ao do BERTimbau na classificação de mensagens em categorias e subcategorias [Souza et al., 2020] [Sanches et al., 2023]. No Capítulo 4, os ganhos de acurácia foram de 4 pontos percentuais em categoria e 3 pontos percentuais em subcategoria, equivalentes a ganhos relativos de 5,1% e 3,8%. O valor de 9% corresponde ao arredondamento da soma desses ganhos relativos (8,9%), não a uma métrica agregada independente.

### 10.2.2 Trabalhos em Conferências

- **EFIS - Ecological Fuel-consumption Intelligent System:** Este trabalho propôs um sistema inteligente para reduzir a emissão de $CO_2$ em rodovias por meio da diminuição do consumo de combustível, considerando apenas a velocidade instantânea, a aceleração, a inclinação atual e a inclinação futura para definir, a cada instante, a aceleração ideal. O principal objetivo foi explorar a inércia do veículo

e as características da via para otimizar as decisões de condução. Duas abordagens principais foram comparadas. A primeira utiliza um controlador com lógica fuzzy e regras bem definidas. A segunda abordagem realiza o treinamento de uma rede neural por meio de um algoritmo genético. Para validar os resultados obtidos, diferentes cenários foram simulados com o uso do Simulation of Urban MObility (SUMO) [Lopez et al., 2018], e os dados foram comparados ao mecanismo padrão de definição de velocidade do simulador. Os resultados indicaram redução de 11% e 9% no consumo para o controlador fuzzy e para a rede neural, respectivamente. Em contrapartida, foram observados aumentos no tempo de viagem de 19% e 7%, respectivamente. [Sanches et al., 2021]

- **MCCD: Generating Human Natural Language Conversational Datasets:** Este trabalho propôs uma metodologia para a geração de conjuntos de dados conversacionais humano-humano a partir de fóruns existentes. A metodologia contempla as etapas de aquisição, anonimização, limpeza, identificação de conversas com base em referências e criação de fluxos conversacionais. A ferramenta Miner-XenForo [Ferraroni Sanches, 2021b], que implementa a metodologia proposta, foi desenvolvida e utilizada para gerar dois conjuntos de dados conversacionais em português brasileiro [Ferraroni Sanches, 2021a], contendo milhões de mensagens e conversas. [Sanches et al., 2022b]

- **Textual Datasets For Portuguese-Brazilian Language Models:** Este trabalho descreve e compartilha quatro conjuntos de dados para o português brasileiro. Esses conjuntos incluem dois gerados com a ferramenta Miner-XenForo, um conjunto anotado traduzido do inglês e um conjunto de grande volume filtrado para remover conteúdos de baixa qualidade ou ofensivos. O trabalho apresenta a metodologia empregada em cada processo de geração e demonstra que o modelo treinado com a arquitetura BERT [Devlin et al., 2019] obtém melhores resultados em tarefas específicas do que o BERTimbau [Souza et al., 2020], modelo considerado estado da arte. [Sanches et al., 2022a]

### 10.2.3 Cooperação em Outro Trabalho

- **Curricular Transfer Learning for Sentence Encoded Tasks:** Este trabalho propôs o uso de Curricular Transfer Learning para o refinamento de modelos de linguagem. O foco principal foi o aprimoramento de sistemas de diálogo orientados a tarefas, reduzindo o desvio entre o pré-treinamento em linguagem geral e o ajuste fino em um domínio específico com poucos recursos disponíveis. O conjunto de dados MultiWOZ [Budzianowski et al., 2020] foi utilizado como benchmark para modelos baseados no GPT-2. Os resultados obtidos indicaram melhor convergência, menor perda durante o treinamento e ganhos consistentes nas métricas BLEU, INFORM, SUCCESS e COMBINED em comparação com estratégias tradicionais de pré-treinamento. [de Sá et al., 2023]

### 10.2.4 Premiação

- **MCCD: Generating Human Natural Language Conversational Datasets [Sanches et al., 2022b]:** Best Paper Award

### 10.2.5 Metodologias Propostas

- **Methodology for Creating Conversational Datasets**: Metodologia desenvolvida para utilizar estruturas de referência entre mensagens para identificar diálogos multiturnos e multiparticipantes.
- **Mechanical Dialogue Creator**: Metodologia para a criação, por meio de *crowdsourcing*, de conjuntos de dados conversacionais.
- **Metodologia de Coleta Web Controlada**: Metodologia desenvolvida para explorar e obter dados textuais da internet a partir de termos e pontos de partida direcionados ao domínio de interesse.
- **Substituição Semântica Rotativa em Aprendizado Federado**: Mecanismo de transformação local dos nove campos pessoais protegidos em valores sintéticos do mesmo tipo e formato, com consistência durante o treinamento local e renovação a cada rodada federada.

### 10.2.6 Ferramentas Criadas

**Ferramentas compartilhadas publicamente**

- **Miner-XenForo**: Ferramenta desenvolvida para a criação de conjuntos de dados conversacionais compostos por diálogos multiturnos e multiparticipantes.
- **MDC:** Ferramenta completa para a criação, por meio de *crowdsourcing*, de conjuntos de dados conversacionais com múltiplos participantes.
- **WackyWacky**: Ferramenta desenvolvida para explorar URLs *seed* de maneira recursiva para criar conjuntos de dados de larga escala.

**Pipelines experimentais**

- **Pipeline de Pré-Treinamento Continuado e Avaliação:** Ferramenta reproduzível para preparação de corpora, pré-treinamento continuado e avaliação de modelos de linguagem em português brasileiro. Integra configurações versionadas, rastreabilidade dos dados, execução retomável e exportação de modelos com verificação de integridade.
- **Pipeline de Treinamento Federado e Avaliação de Privacidade**: Implementação reproduzível para geração de dados pessoais sintéticos, treinamento federado, aplicação de substituição semântica rotativa e DP-AdamW, auditoria de extração e avaliação de utilidade sintética. Integra configurações versionadas, execução retomável e separação lógica entre clientes, servidor e avaliador confiável.

### 10.2.7 Conjuntos de Dados Compartilhados

- **Adrenaline**: Conjunto de dados criado a partir de fóruns de tecnologia on-line.
- **BrWaC-CLEAN**: Conjunto de dados composto por uma versão filtrada do BrWaC [Wagner Filho et al., 2018], da qual foram removidos conteúdos classificados como de baixa qualidade ou potencialmente ofensivos.
- **MultiWOZ-PTBR**: Conjunto de dados originalmente criado para o treinamento de assistentes virtuais, com grande relevância por ser completamente anotado. Por essa razão, foi desenvolvida uma versão traduzida para o português, na qual as anotações foram preservadas para atender aos mesmos objetivos.
- ***Nutritious Food***: Conjunto de dados de exemplo que simula conversas de pessoas em busca de informações nutricionais sobre alimentos.
- **OuterSpace**: Conjunto de dados criado a partir de fóruns on-line de diversos temas.
- **WackyWacky PTBR Geral**: Conjunto de dados de grande volume, abrangendo diversos domínios.

### 10.2.8 Modelos Treinados

- **Tucano2-CPT-General (0.6B):**

  Refinado com BrWaC-CLEAN, GigaVerbo, WackyWacky e MultiWOZ-PTBR, utilizando BrWaC-CLEAN adicional em substituição ao volume destinado aos corpora de fórum.

- **Tucano2-CPT-Forum (0.6B):**

  Refinado com BrWaC-CLEAN, GigaVerbo, WackyWacky e MultiWOZ-PTBR, substituindo o BrWaC-CLEAN adicional por Adrenaline e OuterSpace, sem alterar o orçamento total de tokens.

## 10.3 Trabalhos Futuros

Os resultados apresentados reforçam a importância de articular criação, adaptação e curadoria de dados com o refinamento e a avaliação sistemática de LLMs em português brasileiro. Os riscos associados ao treinamento federado também indicam a necessidade de aprofundar as técnicas de mitigação de vazamento, considerando seus efeitos sobre a privacidade e o desempenho dos modelos.

A continuidade desta pesquisa deve explorar novos domínios, escalas de modelos e condições de aplicação das abordagens desenvolvidas. Os trabalhos futuros estão organizados nas quatro linhas a seguir.

- Criação de Novos Conjuntos de Dados

Ampliar os conjuntos de dados em português brasileiro, especialmente em domínios com menor disponibilidade de recursos, considerando mudanças linguísticas e temáticas ao longo do tempo. Essa expansão deve preservar a rastreabilidade das fontes e os controles de qualidade, permitindo avaliar a contribuição dos novos dados para o desempenho dos modelos.

- Treinamento de Modelos de Diferentes Escalas

  Investigar o treinamento de modelos com diferentes números de parâmetros, utilizando os conjuntos de dados preparados neste trabalho, para compreender a relação entre capacidade do modelo, volume de dados e custo computacional. Outra direção é explorar o treinamento do zero, a partir de inicialização aleatória, e comparar essa estratégia com o pré-treinamento continuado de modelos existentes, considerando os recursos necessários e o desempenho obtido em português brasileiro.

- Defesa de LLMs com Treinamento Federado

  Ampliar a avaliação para sementes independentes das utilizadas no desenvolvimento, diferentes participações do cliente auxiliar na agregação, distribuições de dados e modelos de linguagem. Investigar separadamente os efeitos da substituição do nome, da renovação dos valores e da exposição local, além de utilizar novos formatos de conversa e tarefas externas para avaliar a utilidade. Essa ampliação deve permitir caracterizar os limites da defesa e a generalização dos resultados observados.

- Identificação e Substituição Semântica em Dispositivos com Recursos Limitados

  Desenvolver métodos leves para identificar informações sensíveis e gerar automaticamente substituições semanticamente adequadas, preservando o contexto e a utilidade dos dados. Investigar sua execução local em dispositivos com restrições de memória, processamento e energia, buscando conciliar precisão na identificação, qualidade das substituições e baixo custo computacional.

# Referências Bibliográficas

[Abadi et al., 2016] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, page 308–318. ACM.

[Abadji et al., 2022] Abadji, J., Ortiz Suarez, P., Romary, L., and Sagot, B. (2022). Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

[Bagdasaryan et al., 2020] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020). How to backdoor federated learning. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR.

[Baroni and Bernardini, 2004] Baroni, M. and Bernardini, S. (2004). BootCaT: Bootstrapping corpora and terms from the web. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., and Silva, R., editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

[Baroni et al., 2009] Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

[Bick et al., 2024] Bick, A., Blandin, A., and Deming, D. J. (2024). The rapid adoption of generative AI. Working Paper 32966, National Bureau of Economic Research.

[Bommasani et al., 2021] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R. B., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L. E., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri,

P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., et al. (2021). On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

[Brasil, 2018] Brasil (2018). Lei geral de proteção de dados pessoais (lei nº 13.709, de 14 de agosto de 2018). https://www.gov.br/pt-br/lgpd/lei-geral-de-protecao-de-dados-pessoais-lei-no-13-709-2018. Accessed: 2026-04-11.

[Budzianowski et al., 2020] Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2020). MultiWOZ – a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. arXiv preprint arXiv:1810.00278.

[Carlini et al., 2019] Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. (2019). The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, Santa Clara, CA. USENIX Association.

[Carlini et al., 2021] Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. (2021). Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

[Chakrabarti et al., 1999] Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11):1623–1640.

[Chatterji et al., 2025] Chatterji, A., Cunningham, T., Deming, D. J., Hitzig, Z., Ong, C., Shan, C. Y., and Wadman, K. (2025). How people use ChatGPT. Working Paper 34255, National Bureau of Economic Research.

[Corrêa et al., 2025] Corrêa, N. K., Sen, A., Falk, S., and Fatimah, S. (2025). Tucano: Advancing neural text generation for portuguese. *Patterns*, 6(11):101325.

[Corrêa et al., 2026] Corrêa, N. K., Sen, A., Fatimah, S., Falk, S., Landgraf, L., Kastner, J., and Flek, L. (2026). Tucano 2 cool: Better open source LLMs for Portuguese. arXiv preprint arXiv:2603.03543.

[Crespo et al., 2023] Crespo, M. C. R. M., de Souza Jeannine Rocha, M. L., Sturzeneker, M. L., Serras, F. R., de Mello, G. L., Costa, A. S., Palma, M. F., Mesquita, R. M., de Paula Guets, R., da Silva, M. M., Finger, M., de Sousa, M. C. P., Namiuti, C., and do Monte, V. M. (2023). Carolina: a general corpus of contemporary Brazilian Portuguese with provenance, typology and versioning information. arXiv preprint arXiv:2303.16098.

[Crowston, 2012] Crowston, K. (2012). Amazon mechanical turk: A research tool for organizations and information systems scholars. In Bhattacherjee, A. and Fitzgerald, B., editors, *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221, Berlin, Heidelberg. Springer Berlin Heidelberg.

[de Sá et al., 2023] de Sá, J. M. C., Sanches, M. F., de Souza, R. R., dos Reis, J. C., and Villas, L. A. (2023). Curricular transfer learning for sentence encoded tasks. arXiv preprint arXiv:2308.01849.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Ferraroni Sanches, 2021a] Ferraroni Sanches, M. (2021a). datasets-from-minerxenforo. `https://github.com/MatheusFerraroni/datasets_from_minerxenforo`. Accessed: 2026-01-15.

[Ferraroni Sanches, 2021b] Ferraroni Sanches, M. (2021b). miner-xenforo. `https://github.com/MatheusFerraroni/miner-xenforo`. Accessed: 2026-01-15.

[Galke and Scherp, 2022] Galke, L. and Scherp, A. (2022). Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide MLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4038–4051, Dublin, Ireland. Association for Computational Linguistics.

[Geiping et al., 2020] Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients - how easy is it to break privacy in federated learning? *CoRR*, abs/2003.14053.

[GitHub, Inc., 2026] GitHub, Inc. (2026). Github. `https://github.com`. Accessed: 2026-02-05.

[Gonçalves, 2022] Gonçalves, L. (2022). Imdb pt-br. `https://www.kaggle.com/datasets/luisfredgs/imdb-ptbr`. Accessed: 2022-05-25.

[Google Scholar, 2026] Google Scholar (2026). View article: The brWaC corpus: A new open resource for Brazilian Portuguese. `https://scholar.google.com/citations?view_op=view_citation&citation_for_view=1XDuwHwAAAAJ:O3NaXMp0MMsC`. Accessed: 2026-04-28.

[Hu et al., 2025] Hu, Y., Zhang, Z., Zhang, J., Wang, J., Wang, Q., Qu, L., and Xu, Z. (2025). Simple yet effective: Extracting private data across clients in federated fine-tuning of large language models. In Inui, K., Sakti, S., Wang, H., Wong, D. F., Bhattacharyya, P., Banerjee, B., Ekbal, A., Chakraborty, T., and Singh, D. P., editors, *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1808–1827, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

[Hugging Face, 2026] Hugging Face (2026). Hugging face - the ai community building the future. `https://huggingface.co/`. Accessed: 2026-03.

[Kairouz and McMahan, 2021] Kairouz, P. and McMahan, H. B. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210.

[Kaplan et al., 2020] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

[Kelley, 1984] Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Trans. Inf. Syst.*, 2(1):26–41.

[Klimt and Yang, 2004] Klimt, B. and Yang, Y. (2004). Introducing the Enron corpus. In *International Conference on Email and Anti-Spam*.

[Kousha and Thelwall, 2026] Kousha, K. and Thelwall, M. (2026). How much are LLMs changing the language of academic papers after ChatGPT? a multi-database and full text analysis. *Scientometrics*.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[Lopez et al., 2018] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using SUMO. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE.

[Lowe et al., 2015] Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In Koller, A., Skantze, G., Jurcicek, F., Araki, M., and Rose, C. P., editors, *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

[Maas et al., 2011] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

[McMahan et al., 2016] McMahan, H. B., Moore, E., Ramage, D., and Agüera y Arcas, B. (2016). Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629.

[McMahan et al., 2017] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial*

*Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

[Min et al., 2023] Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2).

[Mireshghallah et al., 2022] Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., and Berg-Kirkpatrick, T. (2022). An empirical analysis of memorization in fine-tuned autoregressive language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Pacheco and Guaranha, 2022] Pacheco, W. E. A. and Guaranha, M. F. (2022). Parser sintático para o português brasileiro: desafios e soluções. *Texto Livre*, 15:e37569.

[Piccialli et al., 2025] Piccialli, F., Chiaro, D., Qi, P., Bellandi, V., and Damiani, E. (2025). Federated and edge learning for large language models. *Information Fusion*, 117:102840.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Technical report, OpenAI.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI*. Accessed: 2024-11-15.

[Rivest, 1992] Rivest, R. L. (1992). The MD5 message-digest algorithm. Technical Report RFC 1321, MIT Laboratory for Computer Science.

[Sanches et al., 2023] Sanches, M. F., de Sá, J. M. C., de Souza, R. R., de Souza, A. M., Dos Reis, J. C., and Villas, L. A. (2023). Automatic extraction of conversation flows from human dialogues: Understanding their impact to refine NLP models. *SN Comput. Sci.*, 4(6).

[Sanches et al., 2022a] Sanches, M. F., de Sá, J. M. C., Foerste, H. T. S., Souza, R. R., Dos Reis, J. C., and Villas, L. A. (2022a). Textual datasets for Portuguese-Brazilian language models. In *Anais do IV Dataset Showcase Workshop (DSW 2022)*, pages 1–12. Sociedade Brasileira de Computação.

[Sanches et al., 2022b] Sanches, M. F., de Sá, J. M. C., de Souza, A. M., Silva, D., de Souza, R. R., Dos Reis, J. C., and Villas, L. A. (2022b). MCCD: Generating human natural language conversational datasets. In *Proceedings of the 24th International Conference on Enterprise Information Systems*. SCITEPRESS - Science and Technology Publications.

[Sanches et al., 2021] Sanches, M. F., Oliveira, M. V. R., Ciceri, O. J., Ladeira, L. Z., Garcia, I. C., Da Fonseca, N. L. S., and Villas, L. A. (2021). Efis - ecological fuel-consumption intelligent system. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 117–123.

[Shokri et al., 2017] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

[Sousa and Kern, 2023] Sousa, S. and Kern, R. (2023). How to keep text private? a systematic review of deep learning methods for privacy-preserving natural language processing. *Artif. Intell. Rev.*, 56(2):1427–1492.

[Souza et al., 2020] Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pretrained BERT models for Brazilian Portuguese. In *Intelligent Systems*, Lecture Notes in Computer Science, pages 403–417. Springer International Publishing, Cham.

[Szep et al., 2026] Szep, M., Marin Ruiz, J., Kaissis, G., Seidl, P., von Eisenhart-Rothe, R., Hinterwimmer, F., and Rueckert, D. (2026). Unintended memorization of sensitive information in fine-tuned language models. In Demberg, V., Inui, K., and Marquez, L., editors, *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6461–6480, Rabat, Morocco. Association for Computational Linguistics.

[Sánchez and Batet, 2017] Sánchez, D. and Batet, M. (2017). Toward sensitive document release with privacy guarantees. *Engineering Applications of Artificial Intelligence*, 59:23–34.

[Université Toulouse Capitole, 2026] Université Toulouse Capitole (2026). Ut1 blacklists. Accessed: 2026-04-06.

[Urbanek and Ringshia, 2024] Urbanek, J. and Ringshia, P. (2024). Mephisto: A framework for portable, reproducible, and iterative crowdsourcing. arXiv preprint arXiv:2301.05154.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

[Wagner Filho et al., 2018] Wagner Filho, J. A., Wilkens, R., Idiart, M., and Villavicencio, A. (2018). The brWaC corpus: A new open resource for Brazilian Portuguese. In Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[XenForo Ltd., 2026] XenForo Ltd. (2026). Xenforo - compelling community forum platform — xenforo.com. `https://xenforo.com/`. Accessed: 2026-01-29.

[Xia et al., 2025] Xia, Z., Zhu, L., Li, B., Chen, F., Li, Q., Liao, C., Wang, F., and Liu, H. (2025). Analyzing 16,193 LLM papers for fun and profits. arXiv preprint arXiv:2504.08619.

[Xie et al., 2020] Xie, C., Huang, K., Chen, P.-Y., and Li, B. (2020). DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*.

[Yousefpour et al., 2021] Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., and Mironov, I. (2021). Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*.

[Zanchetta, 2026] Zanchetta, E. (2026). BootCaT — bootcat.dipintra.it. `https://bootcat.dipintra.it/`. Accessed: 2026-03-27.

[Zang et al., 2020] Zang, X., Rastogi, A., Sunkara, S., Gupta, R., Zhang, J., and Chen, J. (2020). MultiWOZ 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

[Zeng et al., 2024] Zeng, S., Li, Y., Ren, J., Liu, Y., Xu, H., He, P., Xing, Y., Wang, S., Tang, J., and Yin, D. (2024). Exploring memorization in fine-tuned language models. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3917–3948, Bangkok, Thailand. Association for Computational Linguistics.

[Zhang et al., 2025] Zhang, J., Tian, Z., Zhu, M., Song, Y., Sheng, T., Yang, S., Du, Q., Liu, X., Huang, M., and Li, D. (2025). DYNTEXT: Semantic-aware dynamic text sanitization for privacy-preserving LLM inference. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20243–20255, Vienna, Austria. Association for Computational Linguistics.

[Zhu et al., 2019] Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Zhu et al., 2025] Zhu, Y., Zhong, H., Lin, Q., Wei, H., Sun, X., Yu, Z., Liu, M., Zheng, Z., and Chen, L. (2025). What matters in LLM-generated data: Diversity and its effect on model fine-tuning. arXiv preprint arXiv:2506.19262.

# Apêndice A

# WackyWacky PTBR Geral - Termos Semente

git, sql, linux, nosql, devops, docker, esports, netcode, shaders, big data, anti-cheat, kubernetes, renda fixa, cultura pop, geopolítica, ia em jogos, programação, ux em jogos, venv python, apis restful, blog pessoal, level design, métricas esg, brics análise, crédito rural, cultura gamer, esg no brasil, saúde pública, c documentação, edital público, github actions, jogos digitais, manual técnico, poluição do ar, c# documentação, crise econômica, design de jogos, educação básica, ensino superior, física em jogos, go documentação, observabilidade, análise de dados, análise de jogos, análise de série, análise política, c++ documentação, crítica de filme, crítica de jogos, cultura de memes, empreendedorismo, estudo acadêmico, futebol notícias, gem documentação, lua documentação, npm documentação, php documentação, pip documentação, reforma política, regex referência, turismo destinos, violência urbana, .net documentação, artigo de opinião, economia de jogos, java documentação, jvm especificação, marketing digital, mobilidade social, notícias recentes, perl documentação, relatório técnico, ruby documentação, rust documentação, sdk e bibliotecas, análise de mercado, cargo documentação, comentário crítico, design de sistemas, eleições no brasil, gestão empresarial, godot documentação, http especificação, indústria de games, inflação no brasil, json especificação, julia documentação, literatura análise, maven documentação, modelos preditivos, narrativa em jogos, política ambiental, políticas de saúde, políticas públicas, scala documentação, streaming de jogos, swift documentação, testes de software, unity documentação, webgl documentação, yaml especificação, campeonato nacional, computação em nuvem, comércio eletrônico, documentação de api, engenharia de dados, ferrovias no brasil, governança de dados, gradle documentação, história do esporte, indicadores sociais, kotlin documentação, mercado de trabalho, negociação coletiva, opengl documentação, pesquisa científica, python documentação, servidores de jogos, tecnologia no campo, violência de gênero, vulkan documentação, agricultura familiar, alimentação saudável, biografia de artista, cidades inteligentes, descrição de produto, directx documentação, documentação oficial, educação a distância, game design document, graphql documentação, inovação tecnológica, lançamentos de jogos, monetização em jogos, movimentos culturais, node.js documentação, patrimônio histórico, polarização política, processo legislativo, página institucional, qa e testes em jogos, raça e desigualdades, segurança energética, transição energética, composer documentação, demografia

brasileira, desempenho de atletas, diplomacia brasileira, direitos trabalhistas, divulgação científica, eficiência energética, lançamentos do cinema, legislação atualizada, openapi especificação, procedural generation, programação funcional, psicologia do jogador, relato de experiência, rodovias e concessões, transformação digital, tutorial de linguagem, visualização de dados, aprendizado de máquina, direito constitucional, energia solar e eólica, governança corporativa, história contemporânea, latência e matchmaking, multiplayer networking, política internacional, acessibilidade em jogos, arquitetura de software, bastidores da televisão, dados abertos no brasil, desenvolvimento pessoal, exportações industriais, gastronomia tradicional, go modules documentação, gráficos 3d fundamentos, história dos videogames, indústria 4.0 no brasil, informalidade no brasil, javascript documentação, licenciamento ambiental, política de privacidade, referência de linguagem, reforma do ensino médio, salário mínimo impactos, segurança da informação, steamworks documentação, typescript documentação, artigo científico resumo, compliance e integridade, desemprego e indicadores, desmatamento na amazônia, engenharia de requisitos, gerenciamento de pacotes, influenciadores digitais, investimento sustentável, petróleo e gás no brasil, produtividade e salários, python biblioteca padrão, qualidade de vida urbana, r linguagem documentação, redes sociais tendências, resenha de álbum musical, saúde mental no trabalho, setor automotivo análise, tendências de tecnologia, agenda do governo federal, biocombustíveis no brasil, biodiversidade brasileira, controle de gastos mensal, documentação de linguagem, inovação em biotecnologia, mercados globais panorama, paradigmas de programação, qualificação profissional, termos e condições de uso, tratamento para ansiedade, ética em ciência de dados, aeroportos e aviação civil, campanha eleitoral análise, ciência política conceitos, competitividade industrial, crime organizado no brasil, desindustrialização debate, economia gig e plataformas, educação financeira básica, energia elétrica no brasil, exportações do agronegócio, logística e infraestrutura, migração interna no brasil, policiamento e estratégias, política agrícola nacional, portos e comércio exterior, relato de sustentabilidade, relatório econômico mensal, renderização em tempo real, risco climático e finanças, trabalho remoto tendências, unreal engine documentação, bancos de dados relacionais, cenário econômico do brasil, desenvolvimento de software, economia criativa no brasil, energia renovável no brasil, governança e gestão pública, legislação penal atualizada, organizações internacionais, pobreza e políticas sociais, prevenção de doenças comuns, reforma trabalhista análise, regulação do setor elétrico, saneamento básico no brasil, segurança pública no brasil, acordos comerciais e impacto, cadeias produtivas no brasil, commodities agrícolas preços, comércio exterior brasileiro, direitos humanos e segurança, matriz energética brasileira, privacidade em redes sociais, sistema prisional brasileiro, desigualdade social no brasil, estatísticas de criminalidade, indústria de base e mineração, investimentos para iniciantes, mudanças climáticas no brasil, política industrial no brasil, políticas educacionais atuais, resíduos sólidos e reciclagem, segurança alimentar no brasil, sustentabilidade no cotidiano, compiladores e interpretadores, decisões do congresso nacional, defensivos agrícolas regulação, imposto de renda pessoa física, inovação em modelos de negócio, juventude e políticas públicas, obras públicas e investimentos, partidos políticos brasileiros, políticas de segurança pública, preços de combustíveis análise, qualidade de dados e validação, sanções internacionais efeitos, streaming e indústria cultural, transparência e accountability, agricultura brasileira panorama,

automação industrial tendências, descobertas recentes na ciência, impactos climáticos na produção, indústria de alimentos panorama, planejamento financeiro pessoal, programação orientada a objetos, violência doméstica e políticas, algoritmos e estruturas de dados, coalizões e articulação política, conflitos internacionais análise, habitação e déficit habitacional, inteligência artificial aplicada, migração internacional e refúgio, métodos de aprendizagem eficazes, crise hídrica e recursos hídricos, desinformação e checagem de fatos, indicadores econômicos principais, comunidades online e comportamento, conservação e unidades de proteção, corrupção e controle institucional, processamento de linguagem natural, diversidade e inclusão nas empresas, metodologia de pesquisa estatística, publicidade em plataformas digitais, responsabilidade social corporativa, envelhecimento populacional no brasil, cadeia de fornecedores e due diligence, opinião pública e pesquisas eleitorais, transporte público e mobilidade urbana.

## ANEXO. DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA

Eu, **Matheus Ferraroni Sanches**, autor do trabalho intitulado **"Métodos e Recursos de Dados para Modelos de Linguagem em Português Brasileiro: da Coleta e Adaptação à Privacidade em Aprendizado Federado"**, apresentado ao **Programa de Pós-Graduação em Ciência da Computação** da Universidade Estadual de Campinas (Unicamp), declaro, para os devidos fins de direito e de integridade acadêmica, ter utilizado ferramenta de inteligência artificial generativa (IAG) de forma estritamente auxiliar no desenvolvimento da presente pesquisa e/ou na redação deste trabalho.

As seguintes atividades específicas foram realizadas com o apoio das ferramentas de IAG, sob plena supervisão humana, validação crítica e responsabilidade do autor:

- [X] Revisão gramatical e edição textual;
- [ ] Auxílio na descrição, estruturação ou execução de procedimentos estatísticos;
- [X] Suporte na geração ou otimização de códigos de programação;
- [ ] Outras atividades (especificar):

  _____.

A ferramenta de inteligência artificial generativa utilizada foi:

**ChatGPT, da OpenAI** (versão 5.5), utilizado para revisão gramatical e edição textual, e para suporte na geração ou otimização de códigos de programação.

Em conformidade com a Deliberação CONSU-A-005/2026, declaro estar ciente de que a responsabilidade pelo conteúdo final, veracidade das informações e conclusões deste trabalho é integralmente minha. As ferramentas de inteligência artificial generativa utilizadas não são reconhecidas como coautoras e não possuem qualquer responsabilidade sobre os resultados, dados e conteúdos apresentados.

Campinas, SP, 03/11/2026.

**Matheus Ferraroni Sanches**

**Prof. Dr. Leandro Aparecido Villas**