Universidade Estadual de Campinas
Instituto de Computação

Luiz Fellipe Machi Pereira

Classificação de histórias e coerência textual: Uma abordagem com inclusão de estrutura retórica e sintática em modelos de linguagem

CAMPINAS
2025

Luiz Fellipe Machi Pereira

Classificação de histórias e coerência textual: Uma abordagem com inclusão de estrutura retórica e sintática em modelos de linguagem

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Orientadora: Profa. Dra. Sandra Eliza Fontes de Avila
Coorientadoras: Profa. Dra. Nádia Félix Felipe da Silva e Dra. Helena de Almeida Maia

Este exemplar corresponde à versão final da Dissertação defendida por Luiz Fellipe Machi Pereira e orientada pela Profa. Dra. Sandra Eliza Fontes de Avila.

CAMPINAS
2025

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

Pereira, Luiz Fellipe Machi, 1998-
P414c Classificação de histórias e coerência textual : uma abordagem com inclusão de estrutura retórica e sintática em modelos de linguagem / Luiz Fellipe Machi Pereira. – Campinas, SP : [s.n.], 2025.

Orientador: Sandra Eliza Fontes de Avila.
Coorientadores: Nádia Félix Felipe da Silva, Helena de Almeida Maia.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Computação.

1. Coesão (Linguística). 2. Classificação textual (Aprendizado de máquina). 3. Manipulação textual (Aprendizado de máquina). 4. Filtro adversarial (Aprendizado de máquina). 5. Classificação de coerência (Inteligência artificial). 6. Classificação de história (Aprendizado de máquina). 7. Processamento de linguagem natural (Computação). 8. Inteligência artificial generativa. I. Avila, Sandra Eliza Fontes de, 1982-. II. Silva, Nádia Félix Felipe da, 1983-. III. Maia, Helena de Almeida, 1992-. IV. Universidade Estadual de Campinas (UNICAMP). Instituto de Computação. V. Título.

Informações complementares

Título em outro idioma: Story classification and textual coherence : an approach with inclusion of rhetorical and syntactic structure in language models
Palavras-chave em inglês:
Cohesion (Linguistics)
Textual classification (Machine learning)
Textual manipulation (Machine learning)
Adversarial filter (Machine learning)
Coherence classification (Artificial intelligence)
Story classification (Machine learning)
Natural language processing (Computing)
Generative artificial intelligence
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Sandra Eliza Fontes de Avila [Orientador]
Fabíola Souza Fernandes Pereira
Marcos Medeiros Raimundo
Data de defesa: 26-02-2025
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/0000-0002-8372-472X
- Currículo Lattes do autor: http://lattes.cnpq.br/1110118717040783

- Profa. Dra. Sandra Eliza Fontes de Avila
  IC/UNICAMP
- Profa. Dra. Fabíola Souza Fernandes Pereira
  FACOM/UFU
- Prof. Dr. Marcos Medeiros Raimundo
  IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Agradecimentos

A todos que de alguma forma contribuíram com este trabalho, em especial a minhas orientadoras, os membros da meta 5 do H.IAAC, aos membros do Recod.ai, ao IC e a UNICAMP.

Este trabalho está inserido no Hub de Inteligência Artificial e Arquiteturas Cognitivas (H.IAAC), apoiado pelo Ministério da Ciência, Tecnologia e Inovações (MCTI), com recursos da Lei Número 8.248, de 23 de outubro de 1991, no âmbito do PPI-SOFTEX, coordenado pela Softex.

Resumo

O surgimento de modelos de linguagem mais sofisticados, como GPT-3, BERT e seus derivados, revolucionou as interações de sistemas computacionais e humanos. Com o tempo, sistemas com modelos maiores, com respostas melhores e interfaces amigáveis, como ChatGPT e Copilot, os tornaram ainda mais populares. Esses modelos são amplamente utilizados em aplicações que vão desde assistentes virtuais até geração automatizada de conteúdo, oferecendo respostas fluidas e contextualizadas. No entanto, um desafio persistente reside na capacidade de garantir que os textos gerados sejam não apenas gramaticalmente corretos, mas também semanticamente coerentes. A incoerência textual — como contradições internas, quebras na progressão temática ou falhas na estrutura lógica — pode comprometer a utilidade e a confiabilidade desses sistemas, especialmente em cenários críticos, como atendimento ao cliente, educação ou divulgação de informações.

Identificar incoerências em textos gerados antes de disponibilizá-los aos usuários é um problema complexo. A fluência superficial dos modelos de linguagem muitas vezes mascara deficiências estruturais, criando a ilusão de qualidade em narrativas que, na realidade, carecem de lógica ou coesão. Essa limitação torna-se ainda mais relevante quando consideramos aplicações que demandam precisão narrativa, como a geração de textos com temática jornalística, roteiros ou materiais educativos. Além disso, a escassez de bases de dados anotadas com informações sobre coerência textual dificulta o treinamento e a avaliação de sistemas automatizados para essa tarefa. Anotar manualmente textos quanto à sua coerência exige perícia linguística e tempo, uma vez que a coerência envolve múltiplas camadas, como a organização de argumentos, definição de temática e contexto de mundo, aspectos que não são trivialmente quantificáveis.

Diante desse cenário, este estudo propõe uma metodologia para realizar a classificação de histórias coerentes usando modelos de linguagem e comparar seu desempenho ao de um modelo em que é feita a integração de informações sintáticas e retóricas. A abordagem central baseia-se na incorporação de símbolos especiais derivados de conhecimentos advindos de teorias da linguística. Para validar a proposta, construímos um *corpus* de histórias, denominado H.IAAC CommonStories, anotado automaticamente com relações retóricas e categorias sintáticas, com narrativas coerentes e versões incoerentes delas. Esse *corpus* foi utilizado para treinar e avaliar um modelo de linguagem adaptado, cuja robustez foi impulsionada ao estender o conhecimento do modelo.

Além da avaliação no *corpus* desenvolvido, realizamos testes *zero-shot* em uma base de dados brasileira de desinformação (FakeTrue.BR), visando explorar a hipótese de que a coerência textual pode servir como indicador indireto para detecção de desinformação em cenários offline. Os resultados preliminares foram satisfatórios, sugerindo que textos incoerentes ou com estruturas retóricas fragmentadas tendem a correlacionar-se com conteúdo potencialmente enganoso, especialmente em contextos onde a verificação externa de fatos é limitada.

Abstract

The emergence of more sophisticated language models, such as GPT-3, BERT, and their derivatives, has revolutionized the interactions between computer systems and humans. Over time, systems with larger models, better responses, and user-friendly interfaces, such as ChatGPT and Copilot, have made them even more popular. These models are widely used in applications ranging from virtual assistants to automated content generation, providing fluid and contextualized responses. However, a persistent challenge lies in the ability to ensure that the generated texts are not only grammatically correct but also semantically coherent. Textual incoherence — such as internal contradictions, breaks in thematic progression, or flaws in logical structure — can compromise the usefulness and reliability of these systems, especially in critical scenarios such as customer service, education, or information dissemination.

Identifying incoherence in generated texts before making them available to users is a complex problem. The superficial fluency of language models often masks structural deficiencies, creating the illusion of quality in narratives that, in reality, lack logic or cohesion. This limitation becomes even more relevant when we consider applications that demand narrative precision, such as the generation of journalistic texts, scripts, or educational materials. Furthermore, the scarcity of annotated databases with information on textual coherence makes it difficult to train and evaluate automated systems for this task. Manually annotating texts for their coherence requires linguistic expertise and time since coherence involves multiple layers, such as the organization of arguments, the definition of theme, and world context, aspects that are not trivially quantifiable.

In view of this scenario, this study proposes a methodology to classify coherent stories using language models and compare their performance to that of a model that integrates syntactic and rhetorical information. The central approach is based on the incorporation of special symbols derived from knowledge from linguistic theories. We built a *corpus* of stories to validate the proposal, called H.IAAC CommonStories, automatically annotated with rhetorical relations and syntactic categories, with coherent narratives and incoherent versions of them. We used this *corpus* to train and evaluate an adapted language model, whose robustness was boosted by extending the model's knowledge.

In addition to the evaluation of the developed *corpus*, we performed *zero-shot* tests on a Brazilian disinformation dataset (FakeTrue.BR), aiming to explore the hypothesis that textual coherence can serve as an indirect indicator for detecting disinformation in offline scenarios. The preliminary results were satisfactory, suggesting that incoherent texts or texts with fragmented rhetorical structures tend to correlate with potentially misleading content, especially in contexts where external fact-checking is limited.

Lista de Figuras

Lista de Tabelas

Lista de Abreviações e Siglas

AF	Filtro Adversário
EDU	Unidades Elementares do Discurso
EN	Inglês
GPU	Unidade de Processamento Gráfico
$H.IAAC$	Hub de Inteligência Artifical e Arquitesturas Cognitivas
IA	Inteligência Artificial
IC	Instituto de Computação
LLM	Large Language Model
LM	Language Model
LR	Learning Rate
$MCTI$	Ministério da Ciência, Tecnologia e Inovações
$NLTK$	Natural Language Toolkit
PLN	Processamento de Linguagem Natural
POS	Part-Of-Speech
PT	Português
RST	Teoria da Estrutura Retórica
$UNICAMP$	Universidade Estadual de Campinas
$UPOS$	Universal POS tags
$WBCE$	Entropia Cruzada Binária Ponderada
WER	Word Error Rate

Sumário

Capítulo 1

Introdução

Na última década, os modelos de linguagem tornaram-se protagonistas na transformação das interações entre humanos e máquinas, partindo de modelos mais simples, como o `BERT` [36], a grandes arquiteturas e coleções de dados, como o **GPT-3** [21]. Impulsionados por modelos com interfaces de comunicação intuitivas, como ChatGPT[1] e Copilot[2], esses sistemas, treinados em tarefas de predição de cadeias de caracteres [14], facilitam a simulação de conversas, a coleta de textos escritos por humanos, e promovem a criação de textos automática sobre várias áreas de conhecimento com facilidade, apesar de nem sempre cientificamente corretos [134]. Contudo, à medida que sua popularidade cresce e suas aplicações se diversificam — abrangendo desde a redação assistida [75] até o jornalismo automatizado —, emerge o desafio crítico de garantir que os textos gerados não só imitem a linguagem humana, mas também mantenham uma coerência lógica e semântica consistente.

A coerência textual — a construção de uma narrativa logicamente conectada e tematicamente consistente — é um pilar fundamental da comunicação eficaz [128, 143]. Em aplicações práticas, como a geração de respostas em *chatbots* ou a produção de relatórios técnicos, falhas de coerência podem levar a mal-entendidos, disseminação de informações equivocadas ou até danos reputacionais [134]. Embora métricas tradicionais, quem medem corretude gramatical, sejam bem estabelecidas [34, 114, 117], a avaliação da coerência permanece um desafio, especialmente em idiomas como o português, onde recursos anotados são escassos e discursos com linguagem informal são mais comuns.

A geração automática de textos, embora não seja o foco principal, fornece a base para esta pesquisa. Os textos poderiam ser gerados em diferentes níveis, por exemplo, ao nível de palavra, frase e parágrafo [116]. No contexto de geração de textos, o desafio começa na escolha do formato de entrada, que variam de palavras-chave ou um título, até frases que servem como início ou fim do texto [134]. Ainda que textos possam ser gerados de tantas maneiras, o problema é abrangente e pode ser dividido em diferentes tarefas. Porém, Fatima et al. [44] argumentam que para todas elas, um texto é considerado coerente quando regras gramaticais e sintáticas da linguagem base são respeitadas.

Abordamos a coerência com bases de histórias, uma vez que, a apresentação de acontecimentos reais ou fictícios por meio de uma narrativa provou-se eficaz como meio de

[1] `https://openai.com/index/chatgpt`
[2] `https://news.microsoft.com/september-2023-event`

comunicação entre pessoas e transmissão de conhecimentos. A geração de histórias, que é uma sub tarefa da geração de textos, pode exigir que o texto possua um contexto de local, um ou mais personagens, interação com objetos e/ou outros personagens, entre outros aspectos, a depender do conceito de história utilizado [5, 7]. Contudo, desde o primeiro sistema de geração automática de histórias por Inteligência Artificial [130], apresentado no início da década de 1960, mesmo com uma definição ampla de história, o trabalho já sinalizava aspectos desafiadores a serem trabalhos no futuro, tais como: estrutura de textos gerados muito repetitiva e histórias consideradas pouco criativas.

Essas limitações motivaram a criação do *corpus* H.IAAC CommonStories, composto por amostras de bases comumente utilizadas em tarefas de geração de história, extração de *plot* e avaliação da qualidade de narrativas. Além de reunir essas histórias, adicionamos histórias modificadas automaticamente para introduzir incoerências controladas e fazer a avaliação de coerência delas. Cada texto foi anotado automaticamente com relações (e.g., Causa, Contraste) da Teoria da Estrutura Retórica — do inglês, *Rhetorical Structure Theory* (RST) [103] — e anotações de classes gramaticais (e.g., verbos, substantivos), comumente endereçada como *Part-Of-Speech* (POS), por meio de *parsers* especializados.

Com o *corpus* pronto, investigamos aspectos de coerência textual, em especial na classificação de textos. Treinamos classificadores de textos, que julgam um texto como coerente ou incoerente, para servirem de ferramentas que separam textos considerados de baixa qualidade ou que possuem alucinações daqueles considerados de boa qualidade. Para conferir credibilidade as predições, em nossas investigações buscamos enriquecer o conhecimento linguístico do modelo. A estratégia escolhida foi a adição de conhecimento por meio da expansão de vocabulário dos modelos. Essa escolha foi movida pela facilidade de implementação, compatibilidade com modelos pré-treinados e resultados relevantes para as tarefas de classificação e geração de texto, mas que não deixa de apresentar desafios, tais como: (1) integrar as informações estruturais sem prejudicar a capacidade interpretativa do modelo em textos originais, (2) garantir que os ruídos, inerentes de *parsers* automáticos, não gerem resultados piores que um modelo sem a sua adição [71, 154, 158, 175].

1.1 Descrição do Problema

A crescente adoção de modelos de linguagem tem transformado como sistemas computacionais interagem com seres humanos [85, 168], permitindo a geração de textos fluidos e com mais contexto. No entanto, essa capacidade impressionante esbarra em uma limitação crítica: a dificuldade de garantir que textos produzidos sejam semanticamente coerentes [144, 174]. A incoerência textual — caracterizada por contradições internas, desvios abruptos de tema ou falhas na estrutura lógica — compromete a utilidade desses sistemas em aplicações sensíveis, como atendimento ao cliente, produção de conteúdo jornalístico ou educação, onde a clareza e a consistência são fundamentais [44, 128, 143] .

Diante desse cenário, este trabalho propõe uma abordagem que combina teorias linguísticas consolidadas — como a RST [103], que analisa relações retóricas entre segmentos

textuais, e classes gramaticais (POS) — com técnicas de aprendizado de máquina. O objetivo é desenvolver um método para aprimorar a detecção de incoerências em narrativas geradas automaticamente, par que não sejam enviadas ao usuário. Para validar essa proposta, foi necessária a construção do *corpus* H.IAAC CommonStories, composto por pares de histórias coerentes e versões modificadas para introduzir incoerências, anotadas automaticamente com informações de RST e POS. Esse recurso não apenas serve como base para treinamento de modelos, mas também permite investigar como elementos estruturais influenciam a percepção de qualidade textual.

Além disso, explora-se a hipótese de que a incoerência textual pode ser um indicador indireto de desinformação, especialmente em ambientes offline onde a verificação externa de fatos é limitada. Para testar essa ideia, o modelo adaptado foi avaliado no FakeTrue.BR, uma base de dados brasileira de desinformação, com resultados preliminares que sugerem uma correlação entre estruturas retóricas fragmentadas e conteúdo enganoso.

1.2 Motivações e Desafios

Um dos principais desafios para abordar a classificação de coerência reside na escassez de recursos anotados que capturem a complexidade e diversidade de gêneros textuais, em especial as narrativas. Enquanto métricas para avaliar traduções [13, 27, 97] e legendas [12, 59, 79, 151] são amplamente estudadas, a avaliação da coerência, que considera critérios que vão desde organização retórica até a alinhamento com o conhecimento de mundo, tem avançado a passos lentos. A anotação manual de textos para esses fins é um processo demorado e subjetivo, dependente de especialistas linguísticos, o que limita a criação de bases de dados em escala.

O uso de ferramentas para anotação automática de informações linguísticas, apesar de possível e utilizada neste trabalho, enfrenta desafios como a generalização de modelos para contextos culturais específicos e a validação das anotações geradas, principalmente em idiomas com recursos anotados limitados [40, 74, 95, 96]. Superar essas barreiras não só avançaria o estado da arte em processamento de narrativas, mas também contribuiria para aplicações práticas, como a verificação de conteúdo em comunidades com acesso restrito a tecnologias digitais.

Diante da complexidade da tarefa, buscamos conhecimentos e teorias específicas da linguística para auxiliar a exploração do problema ao invés de se apoiar apenas em modelos de linguagem. A RST [103] e o uso de classes gramaticais (POS) foram pré-selecionadas dada a sua capacidade de representar, respectivamente, a organização discursiva e a função sintática das palavras. Enquanto a RST captura conexões entre eventos (e.g., Motivação, Resolução), POS identifica padrões gramaticais associados à clareza narrativa (e.g., proporção entre verbos de ação e substantivos concretos).

1.3 Objetivos

A presente dissertação visa explorar a relação entre a expansão do vocabulário de modelos de linguagem com símbolos especiais, baseados na RST e POS, e a identificação de

histórias coerentes. Para alcançar esse objetivo geral, formulamos os seguintes objetivos específicos:

- Verificar a disponibilidade de bases de dados de qualidade para a tarefa;
- Caso não existam bases adequadas, criar um *corpus* utilizando histórias, com temas variados, escritas por humanos;
- Extrair e anotar automaticamente, informações estruturais e linguísticas dos textos disponíveis, caso não existam, que possam auxiliar no entendimento de características do gênero textual;
- Treinar modelos de linguagem, sem realizar alterações no fluxo padrão e expandindo o vocabulário desses modelos com a informação extraída previamente.
- Avaliar os resultados dos modelos treinados para a tarefa de classificação de coerência com as diferentes abordagens, pontuando fatores como acurácia e robustez.
- Avaliar o desempenho dos modelos treinados no passo anterior em uma base brasileira de desinformação para determinar se podem ser utilizados como *proxy* para classificação de desinformação.

1.4 Perguntas de Pesquisa

Essa dissertação explora duas hipóteses que serviram de base para definir as perguntas de pesquisa. A primeira hipótese, e ponto central da metodologia desenvolvida, é de que a adição de novos símbolos a modelos de linguagem poderia servir de guia para identificação de textos coerentes, que apresentariam uma estrutura sintática e semântica bem definida. A segunda hipótese é de que textos incoerentes aparecem com menor frequência em gêneros que exigem, ou tem como padrão, uma escrita com termos mais formais, portanto, explorar essa correlação pode ser uma estratégia válida para identificar desinformação em contextos em que o acesso a fontes digitais é limitado ou inexistente. Dessa forma, as perguntas de pesquisa que este trabalho se propôs a responder foram:

- A expansão do vocabulário de modelos de linguagem com símbolos especiais, baseados em RST e POS, auxiliam na identificação de histórias coerentes?
- Como utilizar a classificação de textos coerentes como um *proxy* para classificação de desinformação no cenário *offline*?

1.5 Contribuições

Essa dissertação propõe as seguintes contribuições para a área de processamento de linguagem natural e análise de conteúdo:

- **Modelo de geração de texto a partir de unidades de *plot* (Plot2Text).** Este modelo possibilita a criação de narrativas a partir das unidades fundamentais de uma história, as unidades de *plot*, que possuem grande flexibilidade em sua construção.

- ***Corpus* de histórias incoerentes com anotação de POS e RST.** A construção desta base de dados, que contém histórias marcadas com informações de sintaxe (POS) e estrutura retórica (RST), oferece uma ferramenta valiosa para a pesquisa de coerência textual. Esta base serve como um recurso para treinar e testar modelos de detecção de incoerências narrativas, além de possibilitar uma análise mais precisa dos fatores que influenciam a construção de textos coerentes, contribuindo também para a detecção de desinformação.

1.6 Organização do Texto

O restante desta dissertação foi organizado da forma a seguir. O Capítulo 2 revisa os conceitos relacionados a histórias, coerência, teorias da linguística e geração de texto. O Capítulo 3 apresenta uma visão geral da literatura sobre melhoria e classificação de coerência, adição de conhecimento externo a modelos e as bases de dados filtradas nesta pesquisa. O Capítulo 4 descreve a metodologia para atingir os objetivos desta pesquisa. O Capítulo 5 descreve a metodologia de construção do *corpus* H.IAAC CommonStories. O Capítulo 6 apresenta e discute os resultados alcançados. O Capítulo 7 apresenta as principais conclusões, limitações e trabalhos futuros desta dissertação. O código desenvolvido durante esta dissertação, bem como demais artefatos podem ser encontrado no link: `https://github.com/LFMP/cohereclassifier`.

Capítulo 2

Conceitos Fundamentais

Neste capítulo, apresentamos os conceitos relacionados ao desenvolvimento desta dissertação. Especificamente, vamos abordar conceitos utilizados pela literatura no escopo em que esse trabalho se insere (Seção 2.1), definições básicas para o entendimento de trabalhos que utilizam a Teoria da Estrutura Retórica (Seção 2.2) e de *Part-of-Speech* (POS) *Tagging* (Seção 2.3). Por fim, apresentamos conceitos que costumam aparecer ao lidar com geração de texto e que são necessários para o entendimento dessa dissertação em sua totalidade (Seção 2.4).

2.1 Convenções da Literatura

História/narrativa. Os termos história e narrativa são comumente confundidos, mas são conceitualmente diferentes. A diferença entre eventos e sua representação é a diferença entre *história* (o evento ou sequência de eventos) e *narrativa* (como a história é transmitida). A história pode levar um dia, um minuto, uma vida inteira, ou outra medida de tempo. Uma história tem sua própria extensão de tempo e uma ordem de eventos que prossegue cronologicamente do mais antigo ao mais recente. Pode ser verdadeira ou falsa, histórica ou ficcional. A ordem dos eventos e sua extensão são muitas vezes diferentes do tempo e da ordem dos eventos no discurso narrativo. Além disso, toda história possui dois componentes: os *eventos* e as *entidades*, ou personagens, envolvidas nos eventos [1]. Esse conceito de história e narrativa foi crucial para determinar quais seriam as bases de dados escolhidas e quais seriam os trabalhos listados no Capítulo 3 (Trabalhos Relacionados).

Um *evento* de uma história possui momento e lugar definidos e transforma o mundo de um estado para outro. A sequência de eventos e as causas que afetam esses eventos são chamadas de *enredo*, ou *plot*. Esses elementos, juntamente com cenários, adereços e qualquer coisa presente física ou abstratamente no espaço da narrativa, compõem um *espaço* da história. A estrutura que acomoda todos os aspectos da história também é conhecida como *discurso* [7, 99, 108].

Unidade de *plot*. Cada um dos eventos pertencentes ao enredo é chamado de *unidade de plot* ou *eventos de plot*. Nesse trabalho, uma unidade de *plot* também é usada como uma estrutura controlada que pode ser usada para (re)construção de histórias. Usar

uma estrutura controlada nos permite fazer alterações e construções de histórias com maior facilidade se comparada à geração a partir de palavras-chave dada a quantidade de informações. Essa estrutura é usada para criação do Corpus *H.IAAC CommonStories*.

Apesar do termo *unidade de plot* já ser conhecido da literatura [11, 42, 49, 51], não há uma definição de estrutura que acomode todos os trabalhos. Nesse trabalho, definimos *unidade de plot* como um conjunto de palavras-chave ou segmento de uma sentença que contenha sujeito, verbo, objeto e complemento, que fornece as principais informações para sua compreensão e reconstrução do discurso a partir de um conjunto dessas unidades. Essa definição também compreende aspectos como ações tomadas por personagens e cenários da narrativa. Na Tabela 5.2, é possível ver exemplos de textos e as unidades de *plot* extraídas.

Coerência. A coerência textual é um aspecto essencial da comunicação escrita que se refere à organização lógica e semântica das ideias em um texto, permitindo que os leitores compreendam e interpretem a mensagem consistentemente [57]. No contexto da produção textual, a coerência é alcançada por meio da conexão lógica entre as informações apresentadas, garantindo uma progressão suave de ideias e a manutenção de um foco central ao longo do texto [61]. Embora a coerência seja uma característica amplamente reconhecida da qualidade textual, é importante reconhecer que ela pode se manifestar de diferentes maneiras. Dentre os diversos tipos de coerência identificados na literatura, destacam-se a coerência referencial, que envolve a consistência na referência de elementos ao longo do texto; a coerência temporal, relacionada à ordenação lógica dos eventos ou ideias; a coerência causal, que estabelece relações de causa e efeito entre as partes do texto; e a coerência temática, que se concentra na manutenção de um tópico central ao longo da narrativa [61].

Coesão. A coesão é um conceito fundamental na produção textual, referindo-se à forma como os elementos linguísticos estão interligados para criar uma estrutura coesa e organizada [57]. A manutenção da coesão é crucial para garantir a clareza e a fluidez da expressão escrita em diversos contextos. Ao contrário da coerência, que se concentra na relação lógica e semântica entre as ideias, a coesão diz respeito à conexão gramatical e lexical entre as partes do texto [57, 129]. Enquanto a coerência garante a compreensibilidade global do texto, a coesão contribui para a fluidez e a clareza na transmissão da informação. Um texto pode perder sua coesão de várias maneiras, incluindo a falta de referência apropriada entre pronomes e seus antecedentes, a ausência de conectores adequados para indicar relações entre sentenças e a repetição excessiva de palavras ou estruturas sintáticas sem variação [57, 129]. Essas falhas na coesão podem prejudicar a compreensão do texto pelo leitor, resultando em uma comunicação menos eficaz.

Parser. Na Ciência da Computação, um *parser*, ou *analisador*, é uma ferramenta usada para analisar e interpretar a sintaxe de um texto ou programa para extrair informações relevantes. A tarefa de analisar sintaticamente — do inglês, *parsing* — envolve a quebra de um conjunto complexo de estruturas de dados ou código em componentes menores e mais gerenciáveis que podem ser analisados e compreendidos [38, 56].

História adversária/adversarial. Uma história ou texto adversário é essencialmente um exemplo cuidadosamente projetado que, embora mantenha a naturalidade e a gramaticalidade geral, inclui modificações deliberadas destinadas a enganar um modelo de linguagem ou uma métrica de avaliação. Na geração de histórias de domínio aberto, esse conceito é usado para criar um conjunto mais desafiador de exemplos negativos, manipulando enredos de histórias em vez de depender apenas de métodos heurísticos [49].

2.2 Teoria da Estrutura Retórica

A definição de relações que ocorrem entre partes de um texto/discurso facilita a identificação de incoerências e contradições e cria uma estrutura/esquema que não deve ser violada. A área de atuação que considera esses aspectos é comumente chamada de Linguística do Texto ou Linguística Funcional [47].

Um importante trabalho desta área é a Teoria da Estrutura Retórica — do inglês, *Rhetorical Structure Theory* (RST) [103], uma teoria descritiva que enuncia que, além do conteúdo proposicional explícito contido nas orações de um texto, há proposições implícitas, chamadas proposições relacionais, que surgem das relações que se estabelecem entre porções do texto. Skoufaki [137] demonstra que é possível identificar quebras de coerência quando um texto é analisado seguindo as definições e diagramas da RST. Para Mann and Thompson [103], as proposições relacionais resultam de combinações de orações ou partes do texto, que além de seu conteúdo proposicional apresentam um conteúdo implícito, responsável por ressaltar ou restringir algum aspecto em uma sentença. Desta forma, não estão ligadas somente à organização do texto como também exibem relações estabelecidas na temática do texto, ou seja, a aplicação dessa teoria na geração de texto poderia guiar tanto a estrutura do texto quanto a escolha das próximas sentenças.

2.2.1 Unidades do discurso

Para utilização da RST, a primeira etapa é separar o texto em Unidades Elementares do Discurso — do inglês, *Elementary Discourse Unit* (EDU) por serem as folhas da árvore de discurso. O tipo mais simples de EDU é formado por uma oração, porém uma EDU pode conter mais de uma oração a depender do conteúdo. No manual de referência de marcação de discurso, Carlson and Marcu [23] discutem os fenômenos linguísticos que causam a junção de orações para formação de EDUs e quando não devem ocorrer. Como em última instância, cada oração é uma EDU, o número máximo de EDUs possíveis é igual ao número de orações para aquele texto e o número mínimo é um, caso o texto inteiro seja uma única oração. Na Figura 2.1, temos exemplos de EDUs que são iguais a uma sentença (EDUs 1 e 4) e EDUs que são menores que uma sentença (EDUs 2, 3, 5, 6, 7. 8, 9 e 10), formando 10 EDUs e somente 5 sentenças.

2.2.2 Relações de coerência

Na RST, uma relação identifica o relacionamento entre duas ou mais porções de texto não sobrepostas, que podem ser núcleos ou um satélite. Em uma relação mononuclear existe

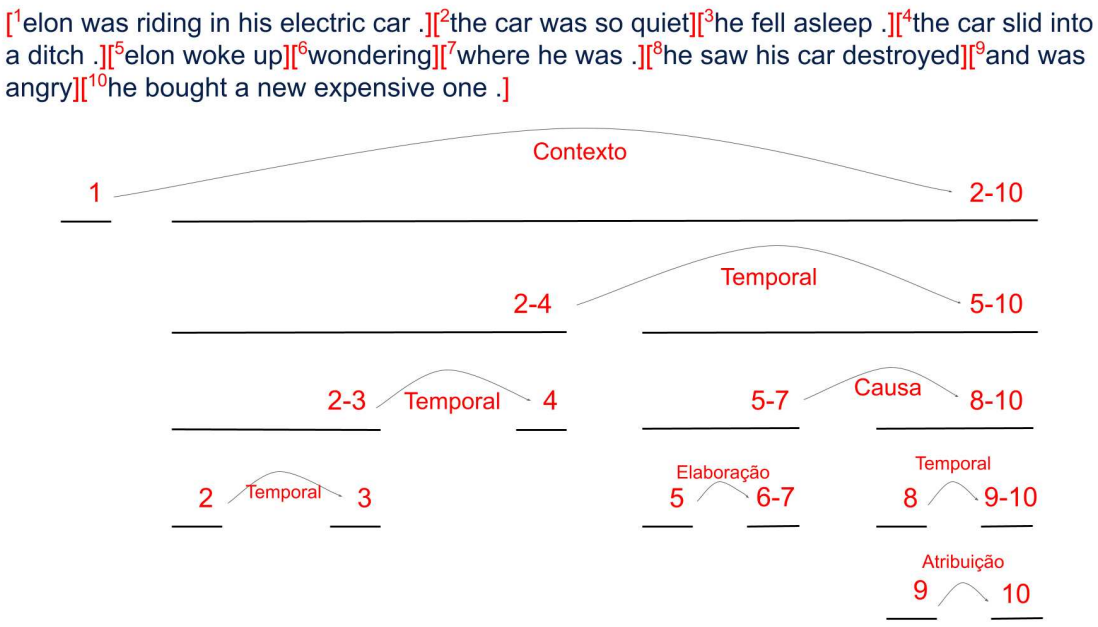

Figura 2.1: Exemplo de aplicação da RST. Na imagem, os arcos representem relações, nomeadas acima ou abaixo do arco, entre EDUs, limitadas por colchetes e numeradas no texto.

apenas um núcleo e um satélite, enquanto em relações multinucleares há somente núcleos. Nas relações, núcleos representam a parte mais notável ou a informação essencial na relação, enquanto o satélite apresenta uma informação de apoio ou antecedente. Usamos 19 relações de coerência nesse trabalho, uma lista pode ser vista na Tabela 2.1. Porém, é possível criar novas relações ao definir: o efeito, as restrições para o núcleo e satélite da relação, e restrições sobre as partes da relação.

2.2.3 Esquemas

A RST pressupõe uma homogeneidade na organização textual, o que significa que, em cada nível da hierarquia do texto, existe um conjunto de padrões estruturais disponíveis para organizar o discurso. Esses padrões são denominados esquemas da RST.

Os esquemas descrevem a organização dos constituintes textuais, funcionando como modelos abstratos que envolvem três componentes principais: (1) um número reduzido de segmentos textuais, (2) a especificação das relações entre esses segmentos e (3) a definição de como certos segmentos, chamados de núcleos, se relacionam com os demais. Em essência, os esquemas atuam de maneira análoga a regras gramaticais, embora com maior flexibilidade [103].

Baseados nas relações retóricas, os esquemas indicam como diferentes partes do texto podem coexistir. Por meio de condições específicas, eles determinam as possíveis estruturas que um texto pode assumir dentro da RST. A teoria reconhece cinco tipos de esquemas, que podem ser representados graficamente, como mostra a Figura 2.2. Nesses diagramas, arcos simbolizam as relações retóricas, com setas que partem do satélite em

Tabela 2.1: Cada parte de uma relação RST tem uma função. Esta tabela detalha os efeitos no Núcleo e no Satélite para o tipo. Tabela construída com base nas definições clássicas da RST por Mann and Thompson [102].

Relação	Núcleo	Satélite
Atribuição	O efeito	O fator ao qual pode ser atribuído
Contexto	Texto cuja compreensão está sendo facilitada	Texto cuja compreensão está sendo facilitada
Causa	Ação/situação	Fator que resultou na ocorrência da ação/situação
Comparação	Uma comparação entre dois ou mais sujeitos/objetos	Uma comparação entre dois ou mais sujeitos/objetos
Condição	Ação/situação resultante da ocorrência de uma situação condicionante	Situação condicionante
Contraste	Uma alternativa	A outra alternativa
Elaboração	Informação básica	Informação adicional
Facilitação	Uma ação/evento facilitado por um fator	O fator
Avaliação	Uma situação	Um comentário avaliativo sobre a situação
Explicação	Uma declaração	A declaração de suporte para explicar a declaração
Conjunto	Uma lista ou disjunção	NA
Modo-Meio	A ação (sendo) executada	O modo ou meio pelo qual a ação foi realizada/atingida
Comentário do Tópico	Uma declaração como uma pergunta, tópico ou afirmação	Uma declaração emparelhada, como uma resposta / comentário do tópico ou resposta
Resumo	Uma declaração	Uma reafirmação, que é mais curta
Temporal	Uma declaração com dependência temporal	Fator do qual depende
Mudança de Tópico	Uma mudança deste tópico A para	Uma mudança para este tópico B
Mesma Unidade	Usado para vincular partes do discurso separadas por uma relação discursiva embutida	NA
Organização Textual	Usado para vincular partes do discurso separadas por uma relação discursiva embutida	NA
Extensão	Não classificado	NA

direção ao núcleo, enquanto linhas retas demarcam o núcleo da relação. Em esquemas multinucleares, como os que representam relações de Contraste, União ou Sequência, há múltiplos núcleos conectados por linhas retas. Nesses casos, a relação ocorre entre duas ou mais partes do texto, e os esquemas definem padrões que permitem analisar uma parte do texto em função de outras [103].

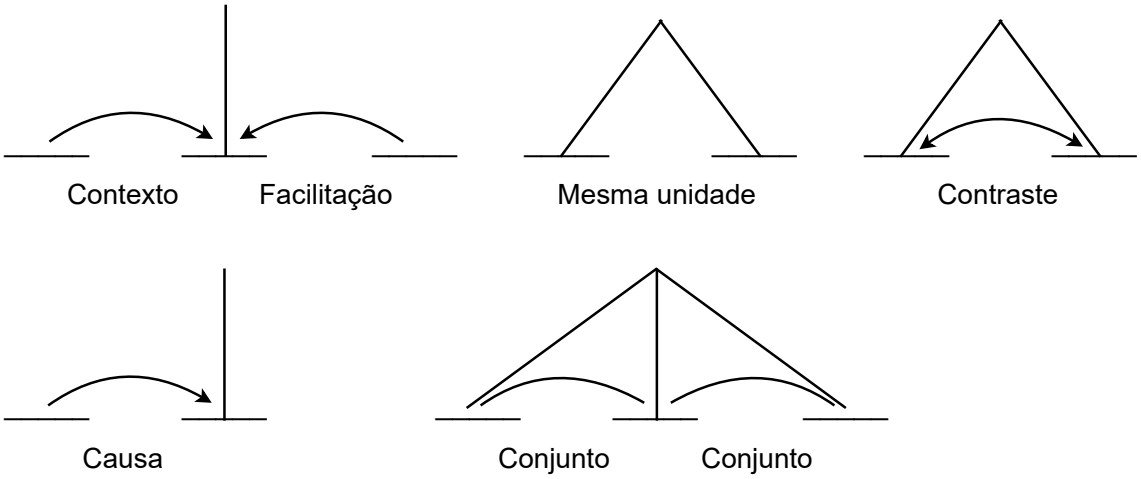

Figura 2.2: Possíveis esquemas para representação de relações na RST de acordo com Mann and Thompson [103]. Imagem feita pelo Autor.

As convenções de aplicação dos esquemas estabelecem como eles podem ser exemplificados, oferecendo uma flexibilidade que vai além de uma reprodução literal. A estrutura global de um texto é compreendida como a combinação desses esquemas aplicados em diferentes níveis. Como os esquemas podem ser utilizados em qualquer nível da hierarquia textual, e uma vez que é o analista quem interpreta a estrutura RST com base na intenção presumida do autor, é possível realizar diversas análises, considerando os diferentes níveis hierárquicos presentes no texto [23, 103].

2.2.4 *Parsers*

Os *parsers* são essenciais para automatizar análises computacionais e geração de dados para tarefas usando a RST, tais como sumarização, identificação de desinformação e geração de texto. Ao identificar EDUs e suas hierarquias relacionais (e.g., Contraste, Causa, Elaboração), os analisadores sintáticos RST transformam texto não estruturado em árvores de discurso interpretáveis, espelhando a compreensão humana. Sem análise sintática automatizada, a anotação manual de estruturas RST se torna demorada e impraticável para aplicações em larga escala ou em tempo real.

Levantamos *parsers* utilizados na literatura de RST, os pontos principais de cada um podem ser vistos na Tabela 2.2. O DMRST Parser [96] surge como a ferramenta mais versátil, combinando suporte multilíngue (inglês, alemão, espanhol e português brasileiro) com uma arquitetura neural baseada em BERT. Ele aborda os desafios do texto em português brasileiro, por meio de análise sintática em nível de documento e rotulagem

conjunta de segmentação-relação EDU. Seu código disponível publicamente[1] e modelos pré-treinados contrastam fortemente com o DiZer 2.0 [105], um dos primeiros analisadores de português brasileiro limitado a gêneros formais como notícias. O design do DiZer, embora pioneiro, carece de adaptabilidade à comunicação informal moderna e sofre com a indisponibilidade do código, apesar de possuir uma ferramenta online[2]. O *parser* ToNy [112][3], visando narrativas em alemão e inglês, emprega métodos baseados em regras para analisar estruturas de texto, mas exclui o português e aplicações de discurso mais amplas. O Cross-Rep [95], um analisador neural multilíngue, que apesar de estar disponível para o português, carece de análise ao nível de documento e não possui código disponível. O EusDisParser [70], projetado para o basco, destaca a aplicabilidade do RST às línguas minoritárias, mas não oferece suporte a linguagens como o inglês e português.

Tabela 2.2: Pontos principais sobre os *parsers* de RST investigados para adoção neste trabalho.

Parser	Suporte de idioma	Ano	Metodologia	Disponibilidade	Limitação
DMRST Parser [96]	Inglês, Alemão, Espanhol, Português-BR	2021	Neural (BERT)	Disponível	Corpus de treinamento PT-BR menor
DiZer 2.0 [105]	Português-BR	2011	Rule-based/SVM	Indisponível	Desatualizado; não paralelizável
ToNy [112]	Alemão, Inglês	2019	Rule-based	Disponível	Foco em textos pequenos; sem PT-BR
Cross-Rep [95]	Inglês, Alemão, Espanhol, Português-BR	2020	Neural (mBERT)	Indisponível	Foco em textos pequenos; sem código
EusDisParser [70]	Basco	2019	Rule-based	Indisponível	Somente basco; sem recursos multilíngues; sem código

Ao final, escolhemos o DMRST Parser devido a dois principais pontos: 1) disponibilidade de código: um código disponível e atualizado nos permitiu gerar artefatos da RST em grande quantidade; 2) idiomas suportados: como o trabalho tem foco no inglês e português, precisávamos de um parser com capacidade de processar entradas em diferentes idiomas e com saída consistente para ambos.

2.3 *Part-of-Speech* (POS) *Tagging*

A marcação de partes do discurso (POS *tagging*) é uma tarefa fundamental no PLN, que envolve a atribuição de etiquetas de classe gramatical a cada *token* em um texto. Essas etiquetas identificam a função sintática de cada palavra em uma sentença, permitindo

[1] `https://github.com/seq-to-mind/DMRST_Parser`
[2] `http://www.nilc.icmc.usp.br/dizer2/`
[3] `https://gitlab.inria.fr/andiamo/tony`

uma análise mais profunda da estrutura gramatical e do significado do texto [123]. A tarefa de POS *tagging* é frequentemente realizada por meio de modelos de aprendizado supervisionado, nos quais cada palavra é associada à sua classe gramatical correspondente. Esses modelos aprendem padrões linguísticos a partir dos dados de treinamento e os aplicam para prever as etiquetas de POS para novos textos não etiquetados [123].

Existem diversas ferramentas disponíveis para a extração automática de POS *tagging*. Entre elas, destacam-se o `NLTK` (*Natural Language Toolkit*), uma biblioteca em `Python` amplamente utilizada para PLN, que oferece diversas funcionalidades, incluindo modelos pré-treinados para POS *tagging* em diferentes idiomas [17]. Outra ferramenta popular é o `spaCy`, uma biblioteca de PLN em `Python` que fornece modelos de POS *tagging* de alta precisão e eficiência, além de oferecer recursos adicionais, como reconhecimento de entidades nomeadas e análise de dependência sintática [62]. Além dessas, outras ferramentas como `Stanford NLP` [125], `Gensim` [179] e `CoreNLP` [104] também são amplamente utilizadas na comunidade de PLN para a execução da tarefa de POS *tagging* de maneira automatizada e eficaz.

Essas ferramentas desempenham um papel crucial no processamento eficiente de grandes volumes de texto, possibilitando uma análise linguística detalhada e a extração de informações relevantes para uma ampla gama de aplicações, incluindo tradução automática, sumarização de texto, análise de sentimento e muito mais. Nesta dissertação, extraímos a classe gramatical de cada *token* e os usamos no protocolo de fusão com o texto, descrito na Seção 4.2.3, usando o `spaCy`, que possui uma interface python amigável, com suporte a diversos idiomas e execução em GPU com modelos Transformers.

2.4 Geração de Texto

Os modelos de geração de texto representam uma classe importante de modelos de linguagem natural que têm como principal objetivo a criação automatizada de sequências de texto coesas e relevantes. Surgindo da evolução de técnicas de Inteligência Artificial e Aprendizado de Máquina, esses modelos têm ganhado destaque significativo nos últimos cinco anos [35], tais como `BERT` [36], `XLM-RoBerta` [94], `GPT-3` [21].

Os usos mais comuns desses modelos incluem a produção de conteúdo para redes sociais, blogs, descrições de produtos, resumos automáticos, assistentes virtuais e até mesmo obras de ficção [6, 9, 16, 39, 146, 147]. Esses modelos aceitam uma variedade de tipos de entradas, desde *prompts* simples até condições de controle mais complexas. Os *prompts* podem ser uma frase inicial, uma pergunta ou uma frase incompleta que orienta o modelo na produção de texto relevante.

As tarefas de geração de texto podem ser vistas como variações de quatro principais, são elas: 1) continuar um texto, 2) preencher lacunas, 3) produzir texto baseado em imagem e 4) gerar texto a partir de texto. Nesta dissertação, utilizamos uma variação da última, fazendo a geração de texto a partir de unidades de *plots* (*plot2text*), apresentadas como texto, para gerar um corpus de histórias e histórias adversárias.

Já as condições de controle podem especificar características específicas que o texto gerado deve possuir, como tom de voz, estilo de escrita ou até mesmo a inclusão de de-

terminadas informações. A qualidade das respostas produzidas por esses modelos varia significativamente, sendo influenciada por diversos fatores, como a arquitetura do modelo, o tamanho e a qualidade do conjunto de dados de treinamento e a precisão das entradas fornecidas pelo usuário. Além disso, o refinamento contínuo dos algoritmos de geração de texto e o aumento da capacidade computacional têm contribuído para melhorias constantes na qualidade das respostas geradas por esses modelos.

Um exemplo desse refinamento é a exploração de métodos de decodificação, que impactam diretamente na qualidade dos textos gerados por esses modelos, tais como: *Beam search* [80, 118], *Speculative decoding* [86] e *Contrastive search* [140], sendo o último utilizado nesta pesquisa. Há também propostas que buscaram melhorar a qualidade da geração de texto por alteração na forma de aprendizado, como por reforço e interativo, e modificações na arquitetura [91], em geral, com adição de camadas, como adição de atenção global [93], adição de *caching* [35] e enriquecimento de informações, sejam elas sintáticas, semânticas ou factuais [153, 160, 164].

Embora avanços significativos ocorreram na área, desafio como maior fluidez [28, 107, 176], repetição de termos e ideias [26, 98, 133]. Mesmo com o surgimento de modelos que produzem respostas mais precisas e contextualmente relevantes, a custo de aumento no tamanho e complexidade do modelo, com a captura mais eficaz de nuances e sutilezas linguísticas, esses problemas ainda permanecem em modelos mais avançados como o `GPT-3` [21]. Isso ocorre, pois os modelos, principalmente aqueles que possuem uma janela de contexto — quantidade de tokens que o modelo tem acesso durante treinamento e geração de textos — muito restrita, ao tentarem prever a sequência mais provável de palavras, frequentemente se prendem a padrões de repetição, e ao lidarem com trechos mais longos, os modelos tendem a perder o foco no tema principal e gerar conteúdo que parece desarticulado ou descontextualizado. Outro problema significativo é o fenômeno das alucinações [66, 162], que acontece quando o modelo gera informações factualmente incorretas ou inteiramente inventadas. Como os modelos de linguagem não possuem uma compreensão real do mundo, eles se baseiam puramente em padrões de texto observados durante o treinamento. Isso os leva a produzir dados que podem parecer plausíveis, mas que não têm base na realidade [15, 18].

Prompt. No âmbito dos modelos de linguagem, um *prompt* é uma entrada textual projetada para orientar o modelo na geração de texto ou na resposta a uma determinada tarefa [21]. Esta entrada pode assumir diversas formas, desde uma simples frase inicial até uma série de instruções específicas. As estratégias mais comuns para a construção de *prompts* incluem formular perguntas claras e específicas, a seleção cuidadosa de palavras-chave relevantes e a adaptação do formato do *prompt* conforme a natureza da tarefa a ser realizada [87]. Em tarefas de geração de texto criativo, por exemplo, *prompts* abertos e inspiradores podem ser utilizados para incentivar a produção de narrativas envolventes e originais. Já em tarefas de respostas curtas, *prompts* mais diretos e específicos são empregados para solicitar respostas precisas e concisas. Além disso, os *prompts* podem ser empregados em uma ampla gama de aplicações, desde a geração de texto automatizada até a tradução automática, a resolução de problemas de linguagem natural e a criação de respostas automáticas em sistemas de conversação [21, 127]. Eles também são

utilizados na formulação de consultas em mecanismos de busca baseados em linguagem natural, demonstrando a versatilidade e a importância desses elementos no contexto dos modelos de linguagem. Assim, a construção cuidadosa de *prompts* desempenha um papel fundamental na obtenção de resultados coerentes e relevantes por parte desses modelos em uma variedade de tarefas de processamento de linguagem natural. Nesta dissertação, utilizamos *prompts* na etapa de extração de unidades de *plots*, em que passamos como entrada para o modelo um *prompt* com instruções e uma parcela da história ao modelo e esperamos um texto no formato, como nas instruções, como saída.

Capítulo 3

Trabalhos Relacionados

A inserção de algum tipo de conhecimento externo, especialmente o linguístico, em um modelo pode ocorrer de diferentes maneiras. A escolha de como esse conhecimento será incluído varia dependendo do aspecto da linguística que será incluído e da tarefa pretendida. Modelos que incluem incorporação de camadas ou modificação da arquitetura do Transformer, em geral, utilizam redes de grafos para modelar a teoria linguística, como a estrutura sintática [88]. Outra forma menos comum, e abordada nesta dissertação, é o pré-processamento da entrada com a inclusão de informações linguísticas para o modelo de geração e classificação. Neste capítulo, focamos em analisar as principais contribuições da literatura envolvendo a RST (Seção 3.1), POS (Seção 3.2), coerência textual (Seção 3.3) e as bases utilizadas (Seção 3.4) para tal. A Tabela 3.1 apresenta informações extraídas dos trabalhos analisados para a construção deste capítulo.

3.1 Teoria da Estrutura Retórica

Uma vez que a Teoria da Estrutura Retórica — do inglês, *Rhetorical Structure Theory* (RST) [103] (Seção 2.2) é formada por componentes, diferentes abordagens podem ser estabelecidas usando um ou mais componentes. As separações de Unidades Elementares do Discurso — do inglês, *Elementary Discourse Unit* (EDU) podem ser consideradas quebras de discurso e/ou indicações de conteúdos relevantes. Relações carregam informações importantes sobre o sentimento expressado em cada trecho do texto, seu relacionamento com demais partes do texto e a importância entre eles quando acompanhado da nuclearidade. Esquemas revelam padrões de escrita de autores, tipos de documentos e veracidade de fatos. Identificar e extrair essas informações dos textos é uma tarefa feita geralmente em uma etapa separada da tarefa alvo e depois incorporada a outros modelos. Os trabalhos a seguir apresentam diferentes formas de lidar com esses componentes para diferentes tarefas.

Xiao et al. [157] propõem o uso de uma árvore binária para representar as divisões de EDUs de um documento, com as nuclearidades inclusas, e um novo mecanismo de autoatenção capaz de codificar seu conteúdo para uma camada de *embedding*. A camada de *embeddings* correspondente é incluída entre o codificador e decodificador de um modelo `BERT` pré-treinado. O *fine-tuning* do modelo é feito usando *corpus* de notícias,

CNNDailyMail [113], tendo como alvo a tarefa de sumarização. Como o modelo `BERT` utilizado tem uma janela de contexto de 512 *tokens*, os textos foram truncados antes de passarem por um *parser*. Para obter as informações de separação do discurso, os autores utilizam o *parser* monolíngue proposto por Wang et al. [152], baseado em um modelo `LSTM` e treinado no *corpus* RST-DT [24]. O fato de utilizarem um *parser* sem suporte a outros idiomas limita o escopo do trabalho e dificulta a replicação para outros idiomas.

O método proposto por Zhang et al. [171] utiliza dos mesmos componentes que o anterior, incluindo a codificação da árvore binária, a inclusão no modelo `BERT` [36] e a utilização da mesma base, contudo, os autores utilizam o *parser* disponível na biblioteca `StanfordNLP` [125]. Além disso, os autores aumentam o tamanho do contexto de 512 para 1024 *tokens* para diminuir a quantidade de cortes que ocorrem nos textos da base, problema enfrentado por Xiao et al. [157]. Apesar do resultado superior nas tarefas escolhidas, ambos os trabalhos apresentam fatores limitantes em relação ao tamanho do texto, a latência da rede e dependência do *parser* para construção da camada de *embeddings*.

O uso de RST para produção de sumários também é estudado por Pu et al. [124], que propõe um novo mecanismo de autoatenção que incorpora as informações de separação de EDUs e as relações entre elas, chamado de RST-Attention. Esse mecanismo de atenção substitui o padrão do modelo `Longformer` utilizado como base para criação do modelo proposto, chamado de `RSTFormer`, inicializado com pesos de um modelo `RoBERTa` [94]. Ao usar um modelo com uma grande janela de contexto, os autores conseguem utilizar textos maiores para *fine-tuning* do modelo, diferentemente dos trabalhos citados anteriormente, feito com os *corpora* BookSum Chapter [81], eLife [52] e Multi-LexSum [135]. Os autores, assim como esta dissertação, utilizam o *parser* DMRST [96] para fazer a extração das informações da RST, apesar de discutirem resultados somente para textos em inglês.

Os componentes da RST também dão suporte a tarefas de classificação. Chernyavskiy and Ilvovsky [31] modelam as marcações de EDUs como *embeddings* em uma rede `LSTM` ao passo que modela as relações como *embeddings* em outra rede `LSTM`. Os *embeddings* das duas redes são concatenados e passam por uma cabeça de classificação. Os autores mostraram a eficiência da rede em classificação em diferentes cenários, sendo eles: classificação de tipo de argumentação, identificação de textos contendo mentiras e classificação de sentimentos. Para esses cenários, usaram os *corpora* Internet Argument Corpus (IAC), LIAR, Movies Dataset, respectivamente, em conjunto com o *parser* `ALT Document-level Discourse Parser` [72] para extração das informações de RST. Guz et al. [55] também utilizam redes `LSTM` para modelar a tarefa de classificação, dessa vez para classificar a coerência de textos. A modelagem é semelhante a de Chernyavskiy and Ilvovsky [31], utilizando a concatenação de *embeddings* de duas redes `LSTM` o autor cria um *embedding* de coerência. O treinamento e avaliação da rede são feitos usando a base de dados GCDT [119], contendo textos somente em inglês. Apesar desta base de dados possuir informações sobre separações de EDUs, os autores escolheram recorrer ao *parser* `CODRA` para extrair as relações e EDUs. Apesar dos resultados promissores desses trabalhos, o modelo utiliza redes `LSTM`, conhecidas por problemas de latência, e com desempenho, em geral, inferior a redes Transformers, já populares quando os artigos foram publicados.

Dada a forte relação da RST com coerência textual e a dificuldade em modelos de geração de texto de garantir a produção de textos com qualidade e sem contradições, pesquisadores buscaram utilizar esse ferramental teórico para tentar solucionar o problema. Chernyavskiy et al. [32] utilizam da RST em uma etapa de pós-geração de texto a fim de corrigir o texto criado, selecionando os melhores candidatos para sua completude. Os autores comparam a árvore de discurso criada por cada texto candidato, gerado por um modelo `GPT-2` pré-treinado, e as compara usando a métrica RSTRecNN [32], aquele com melhor pontuação é selecionado. Para avaliação do *framework*, utilizaram o *corpus* IMDB [100], e devido às particularidades das bases, como presença de textos curtos, os autores ignoram as relações Elaboração, Conjunto, Mesma Unidade, Modo-Meio, Organização Textual e Mudança de Tópico (Tabela 2.1), produzidas pelo `ALT Parser`, sendo as três primeiras, devida à alta ocorrência nos texto e as três últimas por sua rara aparição.

Adewoyin et al. [4], por sua vez, investem na adoção da RST ao adaptar divisões de EDUS, nuclearidade e relações como componentes a serem inseridos na arquitetura `Transformer`. Além da inclusão desses componentes na arquitetura `BART`, com suporte a até 4096 *tokens*, os autores também propõem um *framework* neuro-simbólico para controle da geração de texto também baseado na RST. Tais alterações na arquitetura visam criar textos mais coerentes no cenário argumentativo e de histórias, para isso, o *fine-tuning* do modelo é feito na base WritingPrompts [41] e nas bases criadas por web mining de tópicos do Reddit, são eles: DebateReligion, RelationshipAdvice, Politics e ChangeMyView. Apesar de uma proposta inovadora e com bons resultados, o autor utiliza um *parser* de RST mais antigo, proposto por Feng and Hirst [46], e com acurácia menor que os demais trabalhos citados anteriormente.

Tabela 3.1: Sumarização das informações extraídas de trabalhos relacionados. A coluna *Conteúdo* se refere ao gênero textual das amostras da base, em que: *V* indica o uso de base de dados com conteúdo variado, *H* bases de histórias, *N* bases de notícias, *A* bases de incoerência ou textos adversários. Na coluna *Fase Foco*, as siglas AN, PREP, TR e PP significam, respectivamente, Análise, Pré-Processamento, Treinamento e Pós-Processamento. Para a coluna *Contexto Grande*, consideramos como grandes, os modelos com suporte a uma janela de contexto maior do que 10 mil *tokens*. Em qualquer coluna, a presença da sigla *NA* significa Não se Aplica.

Nome	Bases	Conteúdo	Fase foco	Modelo base	Pré-treinado	Contexto Grande	Altera Transformers	Tags	Expande Voc.	Parser POS	Parser RST	Multilingual	Multimodal
[172]$_{2021}$	RST-DT	N	PREP	MPNet	✔	✗	✗	✔	✗	NA	Próprio	✗	✗
[169]$_{2021}$	RST-DT, CDTB	B, N	PREP	LSGAN	✗	✗	✗	✗	✗	NA	Próprio	✔	✗
[122]$_{2024}$	GUM Corpus	V	AN	NA	✗	✔	✗	✗	✗	NA	NA	✗	✗
[171]$_{2023}$	CNN/ DailyMail	H, N	TR	BERT	✔	✗	✔	✗	✗	NA	Stanford NLP, SpanExt	✗	✗
[48]$_{2020}$	Amazon, Yelp, IMDB, MR, MPQA, Subj, TREC	V, A	PREP	BERT-MLM	✔	✗	✗	✗	✗	NA	NA	✗	✗
[150]$_{2015}$	VG, COCO, Flickr30k	V, A	PREP	NA	✔	✗	✗	✗	✔	spaCy	NA	✗	✔
[101]$_{2024}$	RST-DT, Instr-DT, GUM Corpus	N	PREP	Llama 2 + QLora	✗	✗	✗	✗	✗	NA	Próprio	✔	✗
[10]$_{2020}$	CST Bank	V	PREP	KNN	✗	✗	✗	✔	✔	NA	NA	✗	✗
[8]$_{2020}$	Conceptual Captions, Open Images	V	PREP	BERT + ResNet	✔	✗	✗	✗	✗	NA	NA	✗	✔
[32]$_{2021}$	IMDB	V, A	PP	GPT-2	✔	✗	✗	✗	✔	NA	ALT parser	✗	✗
[157]$_{2020}$	CNN/ DailyMail	H, N	TR	BERT	✔	✗	✔	✗	✗	NA		✗	✗
[178]$_{2020}$	NA	V	AN	NA	✗	✗	✗	✗	✗	NA	NA	✗	✗
[119]$_{2022}$	GCDT, GUM	V, A	AN	NA	✗	✗	✗	✗	✗	NA	DMRST	✔	✗
[161]$_{2021}$	NA	V	AN	NA	✗	✔	✗	✗	✗	NA	NA	✗	✗
[90]$_{2022}$	ChnSenti, weibo100k, THUCNews, Ontonotes 4.0, MSRA-NER, LCQMC, BQ, XNLI, ERNIE, DRCD, CMRC2018	V, H	TR	BERT + GCN	✔	✗	✗	✗	✗	NA	NA	✗	✗
[45]$_{2023}$	Urdu News, "A Million News Headlines" (ABC)	N	PP	GPT-2	✔	✗	✗	✗	✗	Stanza	NA	✔	✗
[124]$_{2023}$	BookSum Chapter, eLife, Multi-LexSum	H, V	TR	Longformer	✔	✔	✔	✗	✗	NA	DMRST	✗	✗
[136]$_{2021}$	IWSLT	V	TR	Roberta	✔	✗	✔	✗	✗	Flair	NA	✗	✔
[173]$_{2019}$	Inspect, Krapivin, NUS, SemEval, KP20k	V	TR	RNN	✗	✗	✔	✗	✗	Stanford-NLP	NA	✗	✗

Tabela 3.1: Sumarização das informações extraídas de trabalhos relacionados. A coluna *Conteúdo* se refere ao gênero textual das amostras da base, em que: *V* indica o uso de base de dados com conteúdo variado, *H* bases de histórias, *N* bases de notícias, *A* bases de incoerência ou textos adversários. Na coluna *Fase Foco*, as siglas AN, PREP, TR e PP significam, respectivamente, Análise, Pré-Processamento, Treinamento e Pós-Processamento. Para a coluna *Contexto Grande*, consideramos como grandes, os modelos com suporte a uma janela de contexto maior do que 10 mil *tokens*. Em qualquer coluna, a presença da sigla *NA* significa Não se Aplica.

Nome	**Bases**	**Conteúdo**	**Fase foco**	**Modelo base**	**Pré-treinado**	**Contexto Grande**	**Altera Transformers**	**Tags**	**Expande Voc.**	**Parser POS**	**Parser RST**	**Multilingual**	**Multimodal**
[54]$_{2021}$	ROC, WP, BookCorpus	H	TR	BART	✔	✗	✔	✗	✔	NA	NA	✗	✗
[55]$_{2020}$	GCDC	V, A	TR	LSTM	✗	✗	✗	✗	✗	NA	CODRA	✗	✗
[31]$_{2020}$	Internet Argument Corpus (IAC), LIAR, Movies Dataset	V, N, A	TR	LSTM	✗	✗	✗	✗	✗	NA	ALT parser	✗	✗
[65]$_{2020}$	Reddit ChangeMyView, New York Times (NYT) corpus	V, N, A	TR	BART	✔	✔	✗	✗	✔	NA	NA	✗	✗
[4]$_{2022}$	Reddit ChangeMyView, WP	V, H	TR	BART	✔	✗	✔	✗	✗	NA		✗	✗
[76]$_{2020}$	Yelp Dataset Challenge 2019, Stanford Sentiment Treebank (SST), Movie Review (MR), IMDB, Yelp-2/5	V	TR	Roberta	✔	✗	✔	✗	✗	Stanford-NLP	NA	✗	✗
[3]$_{2022}$	Wall Street Journal (WSJ), GCDC, Recognizing Textual Entailment (RTE)	V, N	TR	Roberta Longformer	✔	✗	✔	✗	✗	NA	NA	✗	✗

3.2 *Part-of-Speech* (POS) *Tagging*

POS esteve presente em muitos trabalhos relacionados a tarefas de PLN, de tal forma que seria impossível falar sobre todos nesta seção. Portanto, destacamos os trabalhos dos últimos quatro anos que usam informações semânticas e sintáticas junto a um modelo de linguagem. Para uma visão mais ampla da literatura, recomendamos [33, 73].

Fatima et al. [45], assim como Chernyavskiy et al. [32], se atém a melhora da qualidade de texto gerado pelo modelo `GPT-2` após a sua geração. O *framework* proposto pelos autores atua como mecanismo de revisão de texto, dando preferência a textos que seguem um padrão ouro de POS, semelhantes aos Schemas da RST. Para a tarefa de POS *tagging*, recorrem à biblioteca `Stanza` [126], dado o seu suporte ao idioma Urdu, considerada uma linguagem de poucos recursos computacionais disponíveis. A avaliação do *framework*, tanto por humanos quanto por métricas computacionais, foi feita sobre os *corpora* de notícias Urdu News [67] e A Million News Headlines (ABC) [82].

Como palavras podem carregar diferentes sentimentos e papéis semânticos em diferentes contextos, Ke et al. [77] propõem a combinação dessas informações com a arquitetura Transformers do modelo `RoBERTa` por meio da inclusão de camadas de *embeddings* que as representam. Os autores optaram pela utilização da rede proposta, `SentiLare`, para a extração das informações de POS e sentimentos, aplicando-a as bases de dados Yelp Dataset Challenge 2019[1], Stanford Sentiment Treebank (SST) [138], Movie Review (MR) [115], IMDB [100] e Yelp-2/5 [170]. O modelo foi então treinado para tarefa de classificação de sentimentos, atingindo o estado da arte para todos os *corpus* em questão para a tarefa. Assim como os trabalhos que representam a RST como uma camada de *embedding*, os dados precisam ser pré-processados para inclusão das informações externas e a entrada de dados para o modelo precisa ser estruturada, limitando o contexto em que o modelo pode ser aplicado.

3.3 Melhoria e Classificação de Coerência

Nos trabalhos investigados, observamos duas formas de melhoria de coerência: alteração dos dados ou alteração da arquitetura Transformer.

Usando um protocolo de anotação desenvolvido especificamente para capturar relações de coerência entre imagem e legenda, Alikhani et al. [8] anotaram 10.000 pares imagem-legenda disponíveis publicamente. A criação desse *corpus*, apelidado de CLUE, foi feita para explorar a tarefa de previsão de relações de coerência e também treinar modelos de geração de legendas de imagens controláveis e com controle de coerência. O trabalho não apresenta informações sobre a proficiência dos anotadores, dado que seria importante para determinar a confiabilidade das anotações disponibilizadas.

Guan et al. [54] por outro lado, apesar de adicionar *tokens* para auxiliar a tarefa de geração de textos longos e coerentes, foca na criação de novos *encoders* e *decoders*, chamado Hint. Para diferenciar incoerências, os autores manipulam a ordem de sentenças de textos para treinar o modelo, semelhante ao processo de adição de incoerências que

[1] `https://www.yelp.com/dataset`

abordamos nesse trabalho. Ainda no treinamento de modelos, Abhishek et al. [3] apresentam diferentes modelos para trabalhar a coerência textual em diferentes tarefas, sendo elas: classificação binária de coerência, classificação em três vias de coerência, pontuação de coerência e ordenação de sentenças. As tarefas de classificação são feitas sem alteração em arquitetura em base de dados ou arquitetura do modelo pré-treinado, nesse caso um `Longformer`. Para a tarefa de ordenação de sentenças e pontuação de coerência, os autores treinam uma rede siamesa, em que um gêmeo da rede é alimentado com textos coerentes e outro com textos incoerentes, mudando apenas a cabeça do modelo.

3.4 Bases de Dados

Esta dissertação introduz um *corpus* de histórias adversárias baseado nos *corpora* ROCStories [111] e WritingPrompts [41]. Contudo, os trabalhos apresentados anteriormente dificilmente exploram tais bases por falta de disponibilidade de suas versões com os dados da RST e de POS. Nesta seção, apresentamos as principais características das bases de histórias e anotações de RST utilizadas. As demais bases citadas na Tabela 3.1, e que não foram detalhadas nesta seção, não envolvem histórias, notícias ou coerência textual.

3.4.1 Corpus de Histórias

Na Tabela 3.1, na coluna "Conteúdo", as linhas que contêm um H utilizam uma ou mais bases de histórias, sendo necessária a presença de uma das bases a seguir: ROCStories [110], WritingPrompts [41], CNNDaily Mail [113], BookSum [81] e BookCorpus [177]. Esses trabalhos apresentam histórias, histórias publicadas em jornal, capítulos de livro ou resumos de livros e, por isso, compõem essa categoria.

A ROCStories (ROC) [110] é uma coleção de contos de senso comum e eventos cotidianos. O *corpus* consiste em histórias de cinco frases e um título. Cada história contém tópicos do cotidiano criados por escritores via *Amazon Mechanical Turk*. Por outro lado, o *corpus* WritingPrompts (WP) possui histórias escritas por usuários do fórum Reddit, desde o título até o texto correspondente. As histórias são independentes, possuem temas diversos e tamanhos diferentes, uma vez que não há restrição sobre esses parâmetros no fórum. Semelhantemente, não existem restrições em relação ao estilo de escrita, tempos verbais ou até mesmo ao uso correto das regras gramaticais.

O *corpus* CNNDaily Mail [113] é composto por resumos a partir de notícias nos sites da CNN e Daily Mail no formato de perguntas e respostas, sendo considerado um *corpus* de notícias, pelo veículo de publicação, e de histórias, pelo conteúdo e formato das respostas. Dado o seu escopo, o estilo de escrita é bem definido e os textos possuem tamanhos semelhantes.

O BookCorpus [81] concentra-se em alinhar livros com filmes correspondentes para fornecer explicações descritivas abrangentes sobre o conteúdo visual. Este conjunto de dados oferece detalhes refinados e semântica de alto nível, semelhante à encontrada em livros. Já o BookSum [177] abrange resumos de textos literários, como romances, peças de teatro e contos. Os resumos são escritos por humanos com grau de abstração em três

níveis de granularidade de dificuldade crescente: nível de parágrafo, capítulo e livro. O domínio deste conjunto de dados é composto por documentos longos, dependências causais e temporais não triviais e estruturas de discurso ricas.

Apresentado por Ghazarian et al. [49], a base de dados ManPlts reúne subconjuntos das bases ROCStories e WritingPrompts, além de histórias adversárias geradas a partir de modificações de *plots* dos dois conjuntos. Contudo, apesar de inspirar a criação do *corpus* H.IAAC CommonStories, não consideramos a base adequada para tarefas envolvendo histórias, uma vez que a qualidade dos textos gerados, e que compõem a base, são de baixa qualidade e de baixa fidelidade, ou seja, as histórias geradas e que deveriam ser semelhantes as originais, apresentam alucinações e se distanciam, de forma nítida, das originais tanto em quantidade de caracteres quanto em contexto da história.

3.4.2 Bases com RST

Apesar de existirem trabalhos que fazem a extração da RST de outras bases, muitos desses trabalhos não disponibilizam o *corpus* processado para download, diminuindo a reprodutibilidade, disseminação e melhoria da qualidade de trabalhos com RST. Os *corpora* apresentados nesta seção possuem tais dados extraídos por *parsers* automáticos ou por anotadores humanos.

RST Discourse Treebank (RST-DT) [24] é um dos conjuntos com anotação da RST feita por especialistas. Os 385 textos são publicações e resumos de publicações do *Wall Street Journal*, possuindo estilo de escrita semelhante ao conjunto CNNDailyMail, dado o tipo de veículo original do texto. Outro *corpus* anotado por especialistas é o Instructional-DT (Instr-DT) [141], dessa vez contendo somente 176 documentos, porém longos, sobre instruções de conserto doméstico.

O conjunto GUM [166] também é anotado manualmente, mas por estudantes de cursos de linguística. Contudo, é uma base de dados que além de anotações de RST também apresenta anotações de POS, segmentações, construções gramaticais, entre outros dados linguísticos. A base de dados é periodicamente atualizada com textos de notícias, blogs, Reddit, tutoriais, coletados por alunos que fazem a disciplina. Outro grupo de pesquisadores adotou o protocolo usado na criação do GUM para criar uma versão em Mandarin Chinese, denominada de Georgetown Chinese Discourse Treebank (GCDT) [119]. Este conjunto aborda apenas cinco gêneros literários: artigos acadêmicos, biografias, entrevistas, notícias e tutoriais. Diferentemente do GUM, a separação de EDUs e anotação das relações é feita usando o *parser* `DMRST`, também utilizado nesta dissertação.

3.5 Considerações

A análise da literatura revela um cenário diversificado na integração de conhecimento linguístico em modelos de linguagem, com ênfase em três eixos: (1) estratégias de incorporação (pré-processamento, modificação de arquiteturas ou pós-processamento), (2) dependência de recursos especializados, como *parsers* de RST e POS, e (3) desafios de escalabilidade e latência. Contudo, lacunas persistem, especialmente em aplicações multilíngues e na generalização de estruturas narrativas.

Assim como Xiao et al. [157] e Zhang et al. [171], que codificam estruturas de RST como camadas de *embedding* em modelos `BERT`, nossa dissertação utiliza informações retóricas para enriquecer a representação textual. Contudo, contrariamente a esses trabalhos, que dependem de *parsers* monolíngues (e.g., `StanfordNLP`) e truncam textos longos, adotamos o *parser* DMRST [96] — com suporte a múltiplos idiomas — e modelos com janelas ampliadas de contexto (e.g., `Longformer`), mitigando perdas de informação. Além disso, seguindo Adewoyin et al. [4], integramos nuclearidade e relações retóricas como *tokens* especiais na entrada do modelo, sem alterar a arquitetura do *Transformer*, uma escolha que simplifica a replicação e reduz a latência comparada a redes `LSTM` [31].

No âmbito do POS, diferentemente de Fatima et al. [45], que utiliza *tags* como filtro pós-geração, inserimos classes gramaticais diretamente como *tokens* na etapa de pré-processamento, alinhando-nos à estratégia de Ke et al. [77] para controle semântico. Essa decisão permite maior flexibilidade na modelagem de dependências sintáticas durante a geração e classificação de histórias.

A maioria dos trabalhos analisados (e.g., Guz et al. [55], Pu et al. [124]) limita-se a domínios estruturados (notícias, resenhas) ou idiomas com alta quantidade de dados anotados (inglês), ignorando outros gêneros narrativos. Nossa contribuição preenche essa lacuna ao utilizar os *corpora* ROCStories e WritingPrompts, que abrangem temas cotidianos e fantásticos em textos curtos e longos. Além disso, seguindo Alikhani et al. [8], adotamos anotações de coerência baseadas em relações causais e temporais, mas diferentemente deles, evitamos a criação de um novo protocolo de anotação, ao invés disso utilizamos anotações de relações da RST e simulamos incoerências com manipulações das unidades de *plot*.

A dependência de *parsers* automáticos — crítica em Chernyavskiy et al. [32] e Guz et al. [55] — persiste como um gargalo, especialmente para idiomas com poucos recursos. Quanto à escalabilidade, contrariamente a Abhishek et al. [3], que utiliza redes siamesas para pontuação de coerência, optamos por um classificador único baseado em *softmax*, reduzindo a complexidade computacional.

Em resumo, nossa dissertação oferece avanços na literatura ao integrar RST e POS como *tokens* especiais em modelos de linguagem, sem modificações na arquitetura do *Transformer*; e ao utilizar manipulações de unidades de *plots* para simular incoerências, contornando a escassez de dados anotados.

Capítulo 4

Materiais e Métodos

Este capítulo descreve os recursos, ferramentas e metodologias empregados no desenvolvimento e avaliação do classificador de coerência textual proposto. A seleção dos materiais (Seção 4.1) buscou equilibrar robustez teórica, viabilidade técnica, priorizando soluções reprodutíveis, e alinhadas ao estado da arte na área de extração de informações de RST e POS (Seção 4.1.1), modelos pré-treinados (Seção 4.1.2) e bases de história (Seção 4.1.3). Neste capítulo, incluímos também a metodologia para criação de um classificador de textos coerentes (Seção 4.2), enquanto dedicamos um capítulo somente para criação do *Corpus* H.IAAC CommonStories (Capítulo 5).

4.1 Materiais

Essa seção descreve os materiais utilizados para a realização dos experimentos com o classificador de textos coerentes, como os *corpus* de textos, os *parsers* de RST e POS e o modelo pré-treinado, bem como o motivo de sua escolha.

4.1.1 *Parsers* de RST e POS

RST. O *parser* automático `DMRST` (*Document-Level Multilingual* RST) [96] foi utilizado para anotação da RST do *corpus*, tanto em inglês quanto em português. O *parser* foi treinado com uma coleção multilíngue de bancos de árvores de discurso RST e suporta nativamente seis idiomas: inglês, português, espanhol, alemão, holandês e basco. Dentre os analisadores e segmentadores multilíngua (Seção 2.2.4), o que reportou um melhor desempenho para vários idiomas e possuía código disponível foi o `DMRST`, com resultados superiores a 0,62 para classificação do tipo da relação, 0,74 para classificação de nuclearidade e 0,87 para segmentação de EDUs, na métrica F1. Além disso, uma versão adaptada do `DMRST` foi utilizada para criação do Georgetown Chinese Discourse Treebank (GCDT) [119], um banco de árvores de discurso hierárquico para o chinês mandarim.

A escolha de um analisador multilíngua, o `DMRST`, ao invés de dois analisadores de um único idioma, um para inglês e outro para português, ocorreu devido aos seguintes fatores: (1) poucas ferramentas para a tarefa com o idioma português; (2) baixo desempenho e indisponibilidade dos analisadores propostos para o português; (3) poucas bases de dados

para o idioma; e (4) a diversidade de formatos de entrada e saída para as diferentes ferramentas.

POS. O *parser* de *Part of Speech* (POS) utilizado foi o `spaCy` [62]. O `spaCy` é uma biblioteca de Processamento de Linguagem Natural de código aberto projetada para construir aplicativos de Processamento de Linguagem Natural. O *parser* tem suporte para vários idiomas, incluindo português e inglês, e modelos com diferentes acurácias. O `spaCy` foi escolhido por ser uma ferramenta de fácil uso e ampla documentação.

4.1.2 Modelos pré-treinados

Para manipulação dos *plots* foi necessário usar os modelos pré-treinados do COMET [19], disponíveis para download via link no repositório do projeto[1]. Os demais modelos utilizados estão todos hospedados na plataforma Hugging Face[2]. Para geração das histórias a partir de *plots*, fizemos o *fine-tuning* de um modelo mLongT5 com janela de contexto de 16K *tokens*[3]. O modelo XLM-Roberta Longform com contexto de 16K *tokens*[4] foi utilizado para o *fine-tuning* do modelo de classificação.

4.1.3 *Corpus* de textos

Grammarly Corpus of Discourse Coherence (GCDC). O corpus GCDC [84] contém textos, em inglês, de diferentes fontes e de diferentes gêneros textuais, como e-mails e análises de empresas. Foram utilizadas quatro *corpora* de textos: publicações do fórum do Yahoo, análises de empresas feitas no site Yelp e dois *corpora* de e-mail, um de e-mails do escritório de Hillary Clinton e outro de e-mails enviados por funcionários da Enron. Para reduzir a quantidade de textos a serem anotados, os autores removeram textos usando os seguintes critérios: tamanho do texto, presença de URLs e quantidade de quebras de linha. Foram mantidos os textos que possuíam entre 100 e 300 palavras, sem URLs e com poucas quebras de linha, até que fossem obtidos 1200 textos, 1000 para treino e 200 para teste. Os textos foram anotados por especialistas com experiência em anotar textos e usando anotadores inexperientes contratados via plataforma Amazon Mechanical Turk. Cada anotador, três especialistas e cinco inexperientes, classificou os textos em diferentes níveis de coerência, sendo eles: baixa coerência, coerência média e alta coerência. O rótulo de coerência foi atribuído a cada texto segundo o rótulo mais frequente entre os anotadores especialistas, sendo que, em caso de discordância entre os anotadores, o texto era anotado com o rótulo de coerência média. Dentro da base de treino, em média, 1 a cada 20 textos foram rotulados com o nível mais baixo de coerência. Como alguns desses *corpora* estão disponíveis somente via solicitação, utilizamos uma versão já processada e disponibilizada por Abhishek et al. [3], que possui apenas os textos e as anotações dos especialistas.

[1] `https://github.com/atcbosselut/comet-commonsense`
[2] `http://huggingface.co/`
[3] `https://huggingface.co/agemagician/mlong-t5-tglobal-base`
[4] `https://huggingface.co/severinsimmler/xlm-roberta-longformer-base-16384`

Além de funcionar como um auxílio para análise textual, as informações apresentadas pelo parser da RST ainda apresentam dados que podem ser utilizados para caracterizar os textos de uma base de dados. No caso do *corpus* GCDC, a distribuição da quantidade de EDUs, conforme mostra a Figura 4.1, varia de 5 a 55 unidades, concentrando-se no intervalo de 13 a 18 unidades. Essa grande variação mostra que controlar apenas a quantidade de palavras não garante uma homogeneidade em relação à densidade e qualidade de conteúdo. Além da análise em relação à quantidade de EDUs, podemos ainda verificar a distribuição de relações da RST (Figura 4.2) e identificar padrões textuais da fonte do texto, como, por exemplo: a predominância de relações do tipo Conjunto e Elaboração pode indicar textos com encadeamento de contextos e listagem de tarefas, tipo de texto comum em e-mails, enquanto a baixa quantidade de relações de Resumo indica que a base não é focada em resumos de artigos, revistas, séries, etc.

Em relação à utilização do *corpus*, apesar de estar presente em tarefas de classificação em três classes, como em [3, 43, 92, 159], utilizamos uma versão binária do *corpus*, onde os textos com alta coerência foram considerados coerentes, e aqueles com baixa coerência, incoerentes. Desconsideramos as amostras de coerência média. A escolha por uma versão binária do *corpus* se deu pela dificuldade em determinar, conceitualmente, o que seria um texto com coerência média. Segundo nosso entendimento de coerência, a presença de um tipo de incoerência, como a presença de contradições, já é o suficiente para classificar um texto como incoerente, não havendo um meio-termo ou algo que possa ser considerado coerência média.

No Capítulo 5, introduzimos um novo *corpus* similar ao GCDC, construído para classificação de coerência. Diferente do GCDC, o novo *corpus*, chamado H.IAAC CommonStories, é criado a partir da manipulação de histórias coerentes, e não através da avaliação de usuários. Essa estratégia traz mais controle sobre quais aspectos tornam os textos incoerentes. Além disso, possui maior escalabilidade, uma vez que a mesma técnica pode ser aplicada a outros *corpora* de textos coerentes.

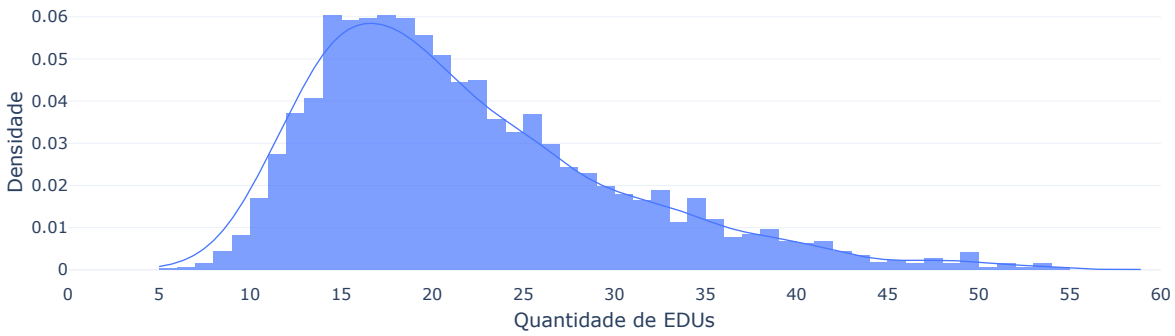

Figura 4.1: Distribuição de quantidade de EDUs no *corpus* GCDC separada em intervalos de 5 EDUs.

FakeTrue.BR. O *corpus* FakeTrue.BR [29] é uma coleção de cerca de 3500 textos em português-BR, projetada para a análise de desinformação e verificação de fatos,

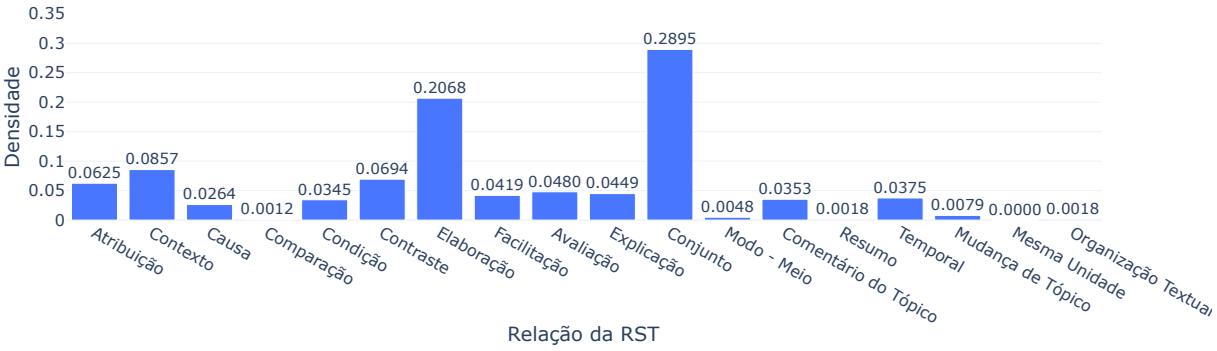

Figura 4.2: Distribuição de relações da RST nos textos do *corpus* GCDC.

incluindo artigos de notícias, postagens em redes sociais e outros formatos de conteúdo digital coletados por meio de um rastreador de conteúdo. Ele se destaca por possuir textos pareados, ou seja, apresenta correspondência entre a desinformação e uma notícia, aumentando sua relevância no contexto atual, onde a disseminação de desinformações se tornou uma preocupação crescente, especialmente em redes sociais e plataformas digitais, acarretando até mesmo na criação de sites especializados em verificar notícias [78, 109, 145].

Assim como no Corpus GCDC, a distribuição de quantidade de EDUs nesse Corpus (Figura 4.3) se estende por uma grande faixa, variando de 1 até 200 EDUs. Neste Corpus, muitas desinformações são feitas em forma de mensagem de texto para aplicativos, geralmente curtas, com poucas informações ou explicações sobre os fatos apresentados, o que gera uma baixa quantidade de EDUs. Em contrapartida, os textos pareados a elas são, em muitos casos, artigos de notícias ou postagens de meios jornalísticos, que cumprem o papel de esclarecer os fatos aos leitores, apresentando informações pertinentes ao contexto, o que os torna maiores e, portanto, acarretam numa quantidade maior de EDUs. Ao comparar as Figuras 4.2 e 4.4, é fácil notar que para o*corpus* FakeTrue.BR, a quantidade de relações Temporais é muito maior, característica esperada uma vez que marcações temporais são muito mais comuns em notícias do que respostas de perguntas em fóruns.

Este *corpus* foi escolhido para investigação do cenário *zero-shot*, não sendo utilizado em etapas de treinamento ou validação dos modelos. A ideia de avaliar nosso método em textos de notícias, mais especificamente em notícias em português-BR, tem como base nossa hipótese de que modelos treinados para classificação de textos coerentes podem apresentar resultados satisfatórios em tarefas de identificação de desinformação, dado a presença de contradições e linguagem informal nesses textos e em textos classificados como incoerentes. Em trabalhos fora do cenário *zero-shot*, mesmo com modelos menores, a identificação de desinformação nessa base poderia ser considerada resolvida, uma vez que experimentos mostram mais de 98% de acurácia nessa tarefa para esse conjunto [121].

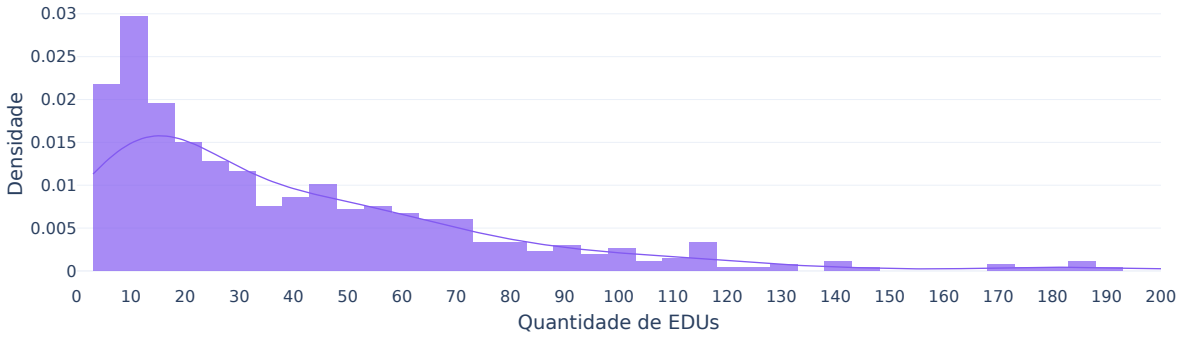

Figura 4.3: Distribuição de quantidade de EDUs no *corpus* FakeTrue.BR separada em intervalos de 10 EDUs.

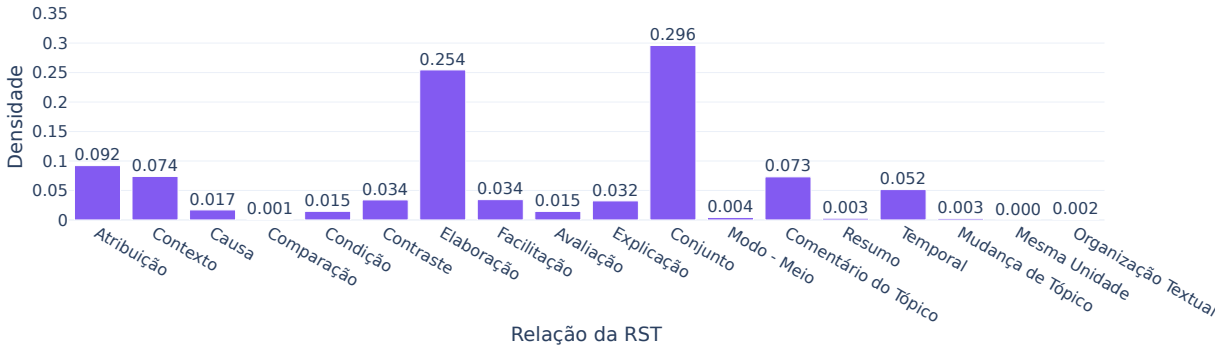

Figura 4.4: Distribuição de relações da RST nos textos do *corpus* FakeTrue.BR.

4.2 Classificador de Textos Coerentes

4.2.1 Metodologia proposta

A metodologia proposta para classificação de textos coerentes, ilustrada na Figura 4.5, foi dividida nas seguintes etapas: 1) extração de informações de RST e POS dos textos; 2) aplicação das técnicas de RSTMix e POSMix; 3) alteração de tokenizador, modelo e função de perda; 4) codificação dos textos em vetores; e 5) treinamento do classificador de textos em coerentes e incoerentes.

Para a etapa 4, a codificação dos textos em vetores foi feita com base no tokenizador `XLM-Roberta Longform`. Para a etapa 5, o treinamento do classificador dos textos foi feito com base em um modelo `XLM-Roberta Longform` com uma camada de classificação de saída. O modelo recebe como entrada os vetores codificados dos textos e retorna a probabilidade de serem coerentes ou incoerentes. Para o treinamento do modelo separamos o *corpus* em treino, validação e teste, seguindo a proporção 70%, 15% e 15%, respectivamente. Entendemos que as etapas 1 a 3 são específicas deste trabalho e por isso necessitam de uma explicação mais detalhada, que será feita nas próximas seções. As etapas 4 e 5 são comuns a qualquer classificador de texto e por isso não necessitam de uma explicação detalhada. Decidimos detalhar a etapa de validação posteriormente, por

ser uma etapa crucial para a avaliação do classificador.

Para avaliar o impacto da aplicação das técnicas RSTMix e POSMix, criamos 3 fluxos de execução (*pipelines*) experimental, sendo eles: (i) *Pipeline Vanilla*: não são executadas as etapas 1, 2 e 3; (ii) *Pipeline RSTMix*: na etapa 1 extraímos somente as informações da RST, na etapa 2 somente a RSTMix é aplicada, as seguintes etapas ocorrem normalmente; (iii) *Pipeline POSMix*: na etapa 1 extraímos somente as informações de POS, aplicamos o POSMix na segunda etapa e usamos as etapas seguintes, semelhante ao que foi feito no *pipeline* anterior.

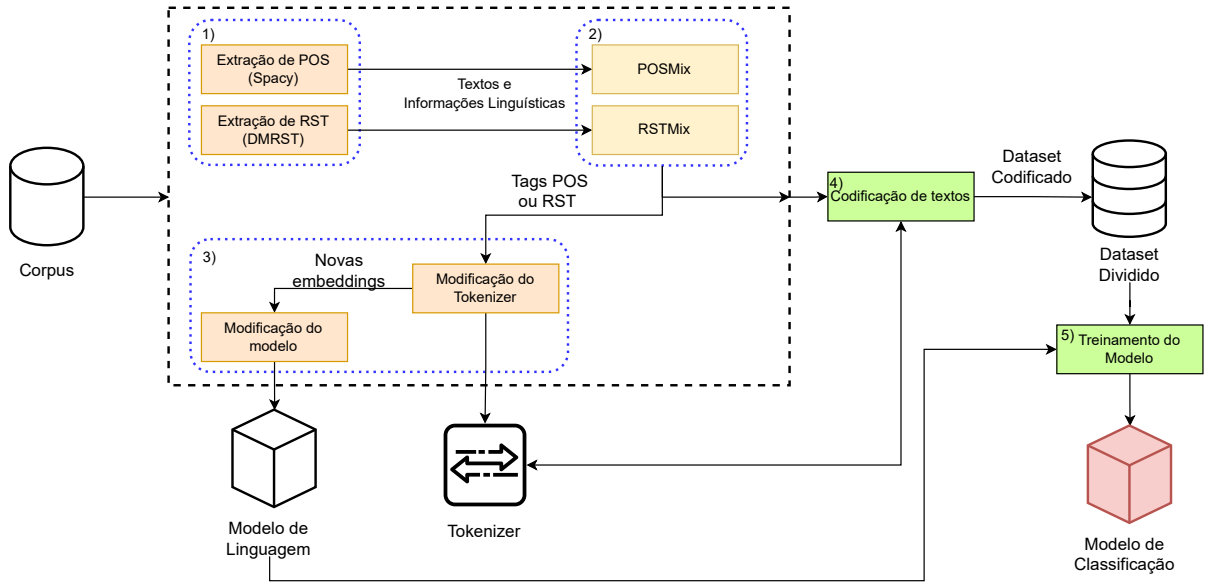

Figura 4.5: Fluxo do classificador de textos coerentes. As caixas tracejadas em azul indicam que os procedimentos contidos nela são necessários para adição de informações da RST ou POS-tagging e estão numeradas segundo as etapas definidas na Seção 4.2.1. A caixa tracejada em preto indica as alterações em um *Pipeline* padrão de treinamento de modelos de classificação para texto. Componentes na cor laranja representam ferramentas ou processos já conhecidos da literatura, enquanto os componentes em amarelo são contribuições deste trabalho.

4.2.2 Extração de informações de RST e POS

Extração de informações de RST. Para extração de informações de RST, utilizamos o *parser* `DMRST` (Seção 4.1.1). O *parser* recebe um texto, separa em *tokens* e retorna as quebras de EDUs e as relações da RST com nuclearidade, sendo necessário um processamento adicional para obter as EDUs. Na Figura 4.6, é mostrado um exemplo de entrada e saída do *parser*.

Extração de informações de POS. Para extração de informações de POS, utilizamos o *parser* `spaCy` (Seção 4.1.1). O *parser* foi utilizado para anotar as palavras dos textos com suas respectivas classes gramaticais, utilizadas para etapa de POSMix. Na Figura 4.7, é mostrado um exemplo de entrada e saída do *parser*.

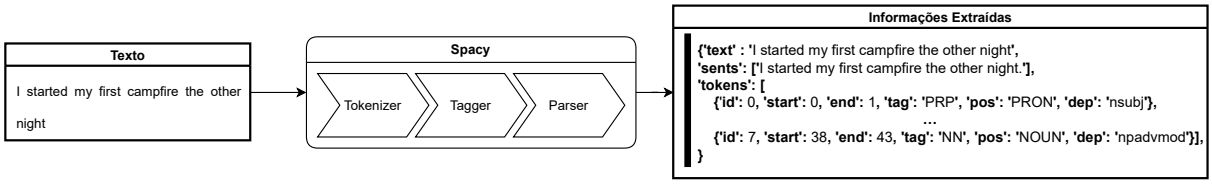

Figura 4.6: Fluxo de extração de informações para extração da RST com o *parser* `DMRST`. Um exemplo dos dados obtidos após a extração pode ser vista no componente mais a direita.

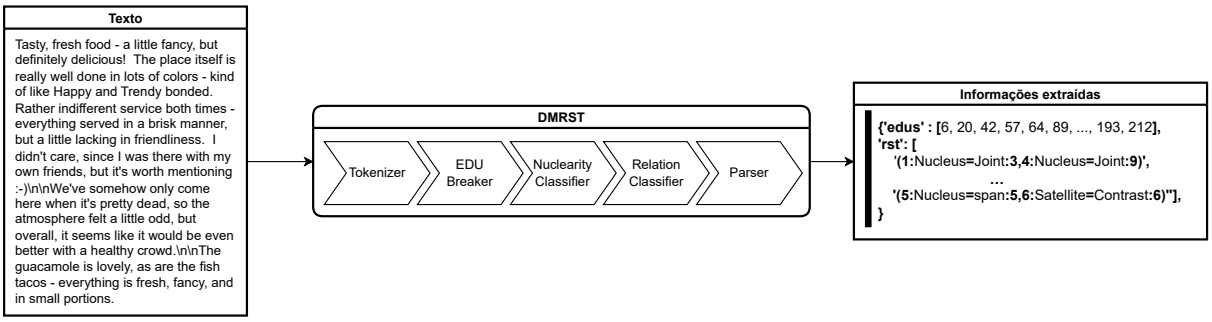

Figura 4.7: Fluxo de extração de informações de POS com a biblioteca `spaCy`. Um exemplo dos dados obtidos após a extração pode ser vista no componente mais a direita.

4.2.3 Aplicação das técnicas de RSTMix e POSMix

Apesar de serem técnicas diferentes, RSTMix e POSMix possuem um funcionamento similar. Ambas as técnicas consistem em adicionar informações pré-processadas ao texto original, para melhorar a classificação de coerência. A diferença entre as técnicas está na forma como as informações são adicionadas ao texto original. Como a informação da RST se dá ao nível de EDUs [5] (os dados são obtidos quando relacionamos duas ou mais EDUs), a quantidade de *tokens* adicionados ao texto original é menor do que a quantidade de *tokens* adicionados com informações de POS, dada ao nível de palavra.

RSTMix

Nosso método para enriquecer o conhecimento de modelos de linguagem com informações estruturais, extraídas da RST, e nomeado de RSTMix, consiste no pré-processamento de textos e a adição de novos símbolos a modelos de linguagem. Para isso, são criados *tokens* especiais que representam as relações da RST, a nuclearidade e a quebra de EDUs. Cada *token* é composto de nuclearidade e relação, separados por dois pontos (:), e envolto por símbolos de maior e menor (<>). A criação desses *tokens* e o seu posicionamento foram inspirados pela adição de *tokens* especiais ao modelo Phi-3 [2] (Figura 4.8 [6]) para lidar com trechos de textos que possuissem algum tipo de formatação ou estrutura pré-definida. Como o *parser* de RST possui 19 categorias de relações e 2 categorias de nuclearidade, seriam gerados 38 *tokens* especiais, contudo, em relações multinucleares (Conjunto, Mesma

[5] Similar a informação ao nível de sentença, com diferença de que EDUs podem ser compostas de mais de uma sentença

[6] `https://huggingface.co/spaces/Xenova/the-tokenizer-playground?tokenizer=microsoft%2FPhi-3-mini-128k-instruct`

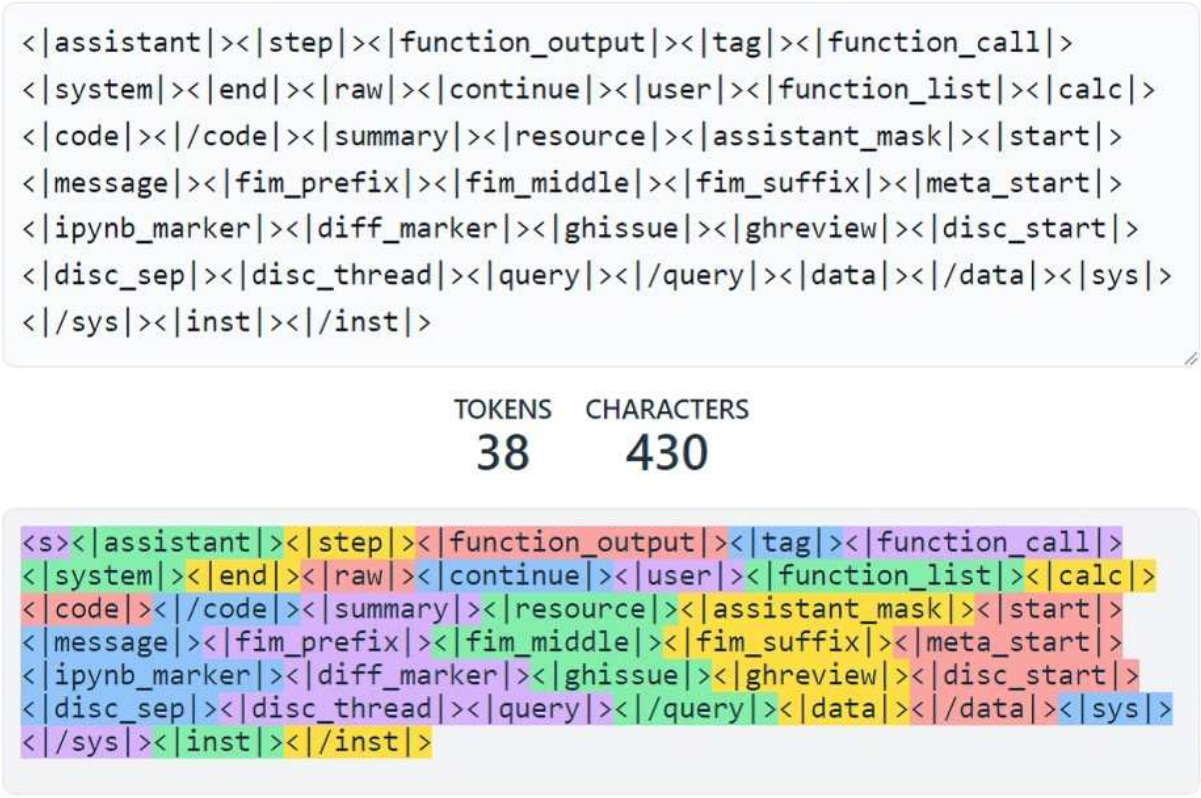

Figura 4.8: *Tokens* especiais do modelo Phi-3. Imagem gerada com Tokenizer Playground em 23/04/2024.

Unidade e Organização Textual) a nuclearidade é a mesma para todos os participantes, considerados como núcleo, dessa forma, geramos apenas 35 *tokens* para representar todas as separações de EDUs, relações entre elas e suas nuclearidades. Uma lista completa dos *tokens* pode ser vista na Tabela 4.1. Para representar a divisão de EDUs, os *tokens* especiais são adicionados no início e no final de cada EDU consoante a relação da RST. Como a ordem de adição dos *tokens* especiais pode influenciar a classificação, e não há uma ordem padrão, utilizamos a ordem retornada pelo *parser*. O processo de adição desses *tokens* pode ser visto na Tabela 4.2.

Tabela 4.1: Associação das *tags* geradas para relação da RST.

Relação	***Tags* geradas**
Mudança de Tópico	<N:Topic-Change>, <S:Topic-Change>
Contexto	<N:Background>, <S:Background>
Contraste	<N:Contrast>, <S:Contrast>
Explicação	<N:Explanation>, <S:Explanation>
Comparação	<N:Comparison>, <S:Comparison>
Temporal	<N:Temporal>, <S:Temporal>
Facilitação	<N:Enablement>, <S:Enablement>
Causa	<N:Cause>, <S:Cause>
Comentário do Tópico	<N:Topic-Comment>, <S:Topic-Comment>
Modo-Meio	<N:Manner-Means>, <S:Manner-Means>
Atribuição	<N:Attribution>, <S:Attribution>
Avaliação	<N:Evaluation>, <S:Evaluation>
Condição	<N:Condition>, <S:Condition>
Resumo	<N:Summary>, <S:Summary>
Elaboração	<N:Elaboration>, <S:Elaboration>
Span	<N:span>, <S:span>
Conjunto	<N:Joint>
Mesma Unidade	<N:Same-Unit>
Organização Textual	<N:TextualOrganization>

Tabela 4.2: Exemplos de textos retirados do *corpus* GCDC que passaram pelo processo de extração da RST, o texto utilizado e o texto resultante após a aplicação do método RSTMix.

Texto Original	RST extraída	RSTMix
I chose to stay here because it's so close to the convention center for my son's bball tourny. The check in process was sooo slow. Even if I requested for a double bedroom a while back it was not given to us and then we have to change rooms the next day and the room they gave us smells so bad of cigarette smoke and it's supposed to be a non-smoking room! I complained about it over the phone and they said they cannot help us. We try to bear with it but it just smells so bad! We end up going to the front desk and talk to the manager and they did help us change rooms but they were not happy about it! Will not come back to this hotel ever!	(1:Nucleus=span:19,20:Satellite=Evaluation:20) (1:Nucleus=Temporal:15,16:Nucleus=Temporal:19) (1:Nucleus=span:2,3:Satellite=Elaboration:15) (1:Nucleus=span:1,2:Satellite=Explanation:2) (3:Nucleus=Joint:3,4:Nucleus=Joint:15) (4:Nucleus=Joint:10,11:Nucleus=Joint:15) (4:Nucleus=Joint:5,6:Nucleus=Joint:10) (4:Satellite=Contrast:4,5:Nucleus=span:5) (6:Nucleus=Joint:6,7:Nucleus=Joint:10) (7:Nucleus=Joint:9,10:Nucleus=Joint:10) (7:Nucleus=Same-Unit:8,9:Nucleus=Same-Unit:9) (7:Nucleus=span:7,8:Satellite=Elaboration:8) (11:Nucleus=Temporal:13,14:Nucleus=Temporal:15) (11:Nucleus=Joint:11,12:Nucleus=Joint:13) (12:Satellite=Attribution:12,13:Nucleus=span:13) (14:Satellite=Contrast:14,15:Nucleus=span:15) (16:Nucleus=Temporal:16,17:Nucleus=Temporal:19) (17:Nucleus=Temporal:17,18:Nucleus=Temporal:19) (18:Nucleus=span:18,19:Satellite=Contrast:19)	<N:span><N:span><N:Temporal><N:span>I chose to stay here<N:span><S:Explanation>because its so close to the convention center for my son's bball tourny.<N:span><S:Explanation><N:Joint><S:Elaboration> The check in process was sooo slow.<N:Joint><S:Contrast><N:Joint><N:Joint><N:Joint>Even if I requested for a double bedroom a while back<S:Contrast><N:span>it was not given to us<N:Joint><N:span><N:Joint><N:Joint>and then we have to change rooms the next day<N:Joint><N:span><N:Same-Unit><N:Joint><N:Joint>and the room<N:span><S:Elaboration>they gave us<N:Same-Unit><S:Elaboration><N:Same-Unit>smells so bad of cigarette smoke<N:Joint><N:Same-Unit><N:Joint>and it's supposed to be a non-smoking room!<N:Joint><N:Joint><N:Joint><N:Joint><N:Joint><N:Temporal><N:Joint>I complained about it over the phone<N:Joint><S:Attribution><N:Joint>and they said<S:Attribution><N:span>they cannot help us.<N:Temporal><N:Joint><N:span><S:Contrast><N:Temporal>We try to bear with it<S:Contrast><N:span>but it just smells so bad! <N:Temporal><S:Elaboration><N:Joint><N:Joint><N:Temporal><N:span><N:Temporal><N:Temporal> We end up going to the front desk<N:Temporal><N:Temporal><N:Temporal>and talk to the manager<N:Temporal><N:span><N:Temporal>and they did help us change rooms<N:span><S:Contrast>but they were not happy about it!<N:span><N:Temporal><N:Temporal><N:Temporal><S:Contrast><S:Evaluation>Will not come back to this hotel ever!<S:Evaluation>
Second visit this evening and they rocked again! Fresh and amazing shrimp Louie salad with a twist and Blackened Mahi with "Nola" remoulade which had to be explained to me but I loved! This is after the first visit when we had the shrimp and jalapeño grits and a scallop special which were both TO DIE FOR! I\'ve had shrimp and grits in the low country and this surpassed both in presentation and innovation. We are pretty fussy and love the food, the ambiance and the service which all exceeded expectations again. It\'s a central Phoenix sleeper and we hope they thrive!	(1:Nucleus=span:5,6:Satellite=Elaboration:15) (1:Nucleus=span:2,3:Satellite=Elaboration:5) (1:Satellite=Background:1,2:Nucleus=span:2) (3:Nucleus=span:3,4:Satellite=Elaboration:5) (4:Satellite=Contrast:4,5:Nucleus=span:5) (6:Nucleus=span:10,11:Satellite=Evaluation:15) (6:Satellite=Elaboration:8,9:Nucleus=span:10) (6:Nucleus=span:6,7:Satellite=Elaboration:8) (7:Nucleus=span:7,8:Satellite=Elaboration:8) (9:Satellite=Background:9,10:Nucleus=span:10) (11:Nucleus=Joint:13,14:Nucleus=Joint:15) (11:Nucleus=Joint:11,12:Nucleus=Joint:13) (12:Nucleus=span:12,13:Satellite=Elaboration:13) (14:Nucleus=Joint:14,15:Nucleus=Joint:15)	<S:Background><N:span><N:span>Second visit this evening<S:Background><N:span>and they rocked again!<N:span><N:span><N:span><S:Elaboration>Fresh and amazing shrimp Louie salad with a twist and Blackened Mahi with "Nola" remoulade <N:span><S:Contrast><S:Elaboration>which had to be explained to me<S:Contrast><N:span>but I loved! <N:span><S:Elaboration><S:Elaboration><N:span><N:span><S:Elaboration> <N:span><S:Elaboration> This is after the first visit<N:span><N:span><S:Elaboration>when we had the shrimp and jalapeño grits and a scallop special<N:span><S:Elaboration>which were both TO DIE FOR!<S:Elaboration><S:Elaboration><S:Elaboration><S:Background><N:span>I\'ve had shrimp and grits in the low country<S:Background><N:span>and this surpassed both in presentation and innovation.<N:span><N:span><N:span><N:Joint><N:Joint><S:Evaluation>We are pretty fussy<N:Joint><N:span><N:Joint>and love the food, the ambiance and the service<N:span><S:Elaboration>which all exceeded expectations again.<N:Joint><N:Joint><S:Elaboration><N:Joint><N:Joint>It\'s a central Phoenix sleeper<N:Joint><N:Joint>and we hope they thrive!<S:Elaboration><S:Evaluation><N:Joint><N:Joint>

POSMix

Nosso método de adição de informações ao nível de palavras, chamado de POSMix, é semelhante ao método RSTMix e também consiste na adição de *tokens* especiais ao texto original, mas nesse caso, os *tokens* representam apenas as classes gramaticais das palavras. No momento da criação desse método, tendo já desenvolvido o RSTMix, a anotação das classes gramaticais foi inspirada pelo pré-processamento apresentado por Wang et al. [149], que funde essas informações com o texto, e adaptada ao nosso contexto. Cada *token* é formado por um subtraço (_) seguido de uma das 17 classes gramaticais retornadas pelo *parser* de POS. As classes gramaticais usadas correspondem pelo *parser* são as mesmas definidas no conjunto universal de tags — do inglês Universal POS tags (UPOS) — para anotações de banco de árvores no cenário multilíngue [120]. Uma lista completa dos tokens pode ser vista na Tabela 4.3. Os *tokens* são adicionados após cada palavra do texto original, sem espaço entre eles. Na Tabela 4.4, é mostrado um exemplo de adição dos *tokens* especiais.

Tabela 4.3: Relação de símbolos de POS gerados pela biblioteca `spaCy`, seu significado e a *tag* gerada.

UPOS	**Significado**	***Tag* gerada**
SCONJ	Conjunção subordinada	_SCONJ
NUM	Numeral	_NUM
X	Outros/ palavras desconhecidas	_X
NOUN	Substantivo	_NOUN
PUNCT	Pontuação	_PUNCT
AUX	Auxiliar	_AUX
ADP	Adposição (preposições e posposições)	_ADP
DET	Determinante	_DET
VERB	Verbo	_VERB
SYM	Símbolo	_SYM
PROPN	Substantivo próprio	_PROPN
ADV	Advérbio	_ADV
ADJ	Adjetivo	_ADJ
PRON	Pronome	_PRON
CCONJ	Conjunção coordenativa	_CCONJ
PART	Partícula	_PART
INTJ	Interjeição	_INTJ
SPACE	Espaço entre palavras ou pontuação	Não gerado

Tabela 4.4: Exemplos de textos retirados do *corpus* GCDC que passaram pelo processo de extração da POS, o texto utilizado e o texto resultante após a aplicação do método POSMix.

Texto Original	POSMix
I chose to stay here because it's so close to the convention center for my son's bball tourny. The check in process was sooo slow. Even if I requested for a double bedroom a while back it was not given to us and then we have to change rooms the next day and the room they gave us smells so bad of cigarette smoke and it's supposed to be a non-smoking room! I complained about it over the phone and they said they cannot help us. We try to bear with it but it just smells so bad! We end up going to the front desk and talk to the manager and they did help us change rooms but they were not happy about it! Will not come back to this hotel ever!	I_PRON chose_VERB to_PART stay_VERB here_ADV because_SCONJ it_PRON 's_AUX so_ADV close_ADJ to_ADP the_DET convention_NOUN center_NOUN for_ADP my_PRON son_NOUN 's_PART bball_NOUN tourny_NOUN ._PUNCT The_DET check_NOUN in_ADP process_NOUN was_AUX sooo_ADV slow_ADJ ._PUNCT Even_ADV if_SCONJ I_PRON requested_VERB for_ADP a_DET double_ADJ bedroom_NOUN a_DET while_NOUN back_ADV it_PRON was_AUX not_PART given_VERB to_ADP us_PRON and_CCONJ then_ADV we_PRON have_VERB to_PART change_VERB rooms_NOUN the_DET next_ADJ day_NOUN and_CCONJ the_DET room_NOUN they_PRON gave_VERB us_PRON smells_VERB so_ADV bad_ADJ of_ADP cigarette_NOUN smoke_NOUN and_CCONJ it_PRON 's_AUX supposed_VERB to_PART be_AUX a_DET non_ADJ -_ADJ smoking_ADJ room_NOUN !_PUNCT I_PRON complained_VERB about_ADP it_PRON over_ADP the_DET phone_NOUN and_CCONJ they_PRON said_VERB they_PRON can_AUX not_PART help_VERB us_PRON ._PUNCT We_PRON try_VERB to_PART bear_VERB with_ADP it_PRON but_CCONJ it_PRON just_ADV smells_VERB so_ADV bad_ADJ !_PUNCT We_PRON end_VERB up_ADP going_VERB to_ADP the_DET front_ADJ desk_NOUN and_CCONJ talk_VERB to_ADP the_DET manager_NOUN and_CCONJ they_PRON did_AUX help_VERB us_PRON change_VERB rooms_NOUN but_CCONJ they_PRON were_AUX not_PART happy_ADJ about_ADP it_PRON !_PUNCT Will_AUX not_PART come_VERB back_ADV to_ADP this_DET hotel_NOUN ever_ADV !_PUNCT
Second visit this evening and they rocked again! Fresh and amazing shrimp Louie salad with a twist and Blackened Mahi with "Nola" remoulade which had to be explained to me but I loved! This is after the first visit when we had the shrimp and jalapeño grits and a scallop special which were both TO DIE FOR! I\'ve had shrimp and grits in the low country and this surpassed both in presentation and innovation. We are pretty fussy and love the food, the ambiance and the service which all exceeded expectations again. It\'s a central Phoenix sleeper and we hope they thrive!	Second_ADJ visit_NOUN this_DET evening_NOUN and_CCONJ they_PRON rocked_VERB again_ADV !_PUNCT Fresh_ADJ and_CCONJ amazing_ADJ shrimp_NOUN Louie_PROPN salad_NOUN with_ADP a_DET twist_NOUN and_CCONJ Blackened_PROPN Mahi_PROPN with_ADP "_PUNCT Nola_PROPN "_PUNCT remoulade_NOUN which_PRON had_VERB to_PART be_AUX explained_VERB to_ADP me_PRON but_CCONJ I_PRON loved_VERB !_PUNCT This_PRON is_AUX after_ADP the_DET first_ADJ visit_NOUN when_SCONJ we_PRON had_VERB the_DET shrimp_NOUN and_CCONJ jalapeño_NOUN grits_NOUN and_CCONJ a_DET scallop_NOUN special_NOUN which_PRON were_AUX both_PRON TO_PART DIE_VERB FOR_ADP !_PUNCT I_PRON \'ve_AUX had_VERB shrimp_NOUN and_CCONJ grits_NOUN in_ADP the_DET low_ADJ country_NOUN and_CCONJ this_PRON surpassed_VERB both_PRON in_ADP presentation_NOUN and_CCONJ innovation_NOUN ._PUNCT We_PRON are_AUX pretty_ADV fussy_ADJ and_CCONJ love_VERB the_DET food_NOUN ,_PUNCT the_DET ambiance_NOUN and_CCONJ the_DET service_NOUN which_PRON all_PRON exceeded_VERB expectations_NOUN again_ADV ._PUNCT It_PRON \'s_AUX a_DET central_ADJ Phoenix_PROPN sleeper_NOUN and_CCONJ we_PRON hope_VERB they_PRON thrive_VERB !_PUNCT

4.2.4 Alteração de tokenizador, modelo e função de perda

Tokenizador. Para suportar os *tokens* especiais dos métodos RSTMix e POSMix, foi necessário fazer a inclusão dos *tokens* necessários, a depender do método escolhido, no vocabulário do tokenizador. A escolha de tratar os *tokens* adicionais como especiais permite que sejam ignorados durante a etapa de decodificação, não sofram divisão em subpalavras e auxiliem no entendimento dos resultados gerados por um modelo. A não divisão em subpalavras é particularmente importante para os métodos propostos, uma vez que a divisão desses *tokens* pode alterar seu significado e prejudicar a classificação de textos longos, uma vez que adicionamos mais *tokens* aos textos, podendo ultrapassar o limite suportado pelo modelo.

Modelo. A alteração no tokenizador faz com que o modelo precise de alteração, uma vez que a matriz de *embeddings* de *tokens* do modelo deve conter a mesma quantidade de linhas que o vocabulário do tokenizador. Os vetores adicionados ao final da matriz de *embeddings* são inicializados com valores aleatórios, prejudicando a convergência do modelo. Para aliviar esse problema e permitir que o modelo aprenda os vetores dos *tokens* adicionais mais rapidamente, os pesos desses vetores foram substituídos por valores correspondentes a média dos pesos de outros vetores da matriz de *embeddings*, prática comum quando é feita uma extensão do vocabulário [60].

Função de perda. A função de perda comumente utilizada para problemas de classificação binária é a entropia cruzada binária. No entanto, utilizamos uma função de perda que considera o desbalanceamento das classes, a entropia cruzada binária ponderada (Equação 4.1). A entropia cruzada binária ponderada é uma extensão da entropia cruzada binária que atribui pesos diferentes para cada classe, de forma que a classe minoritária tenha um peso maior que a classe majoritária. A escolha por essa função de perda se deu pelo desbalanceamento das classes no *corpus* GCDC, onde a classe coerente é majoritária e a classe incoerente é minoritária.

$$\text{WBCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[w_1 y_i \log(p_i) + w_0(1-y_i)\log(1-p_i)\right], \tag{4.1}$$

onde w_1 refere-se ao peso para a classe positiva (e.g., casos com incoerência), w_0 é o peso para a classe negativa (e.g., textos coerentes), y_i é o rótulo binário do exemplo i ($y_i \in \{0, 1\}$), p_i é a probabilidade prevista pelo modelo para a classe positiva, e N é o número total de exemplos.

4.2.5 Validação e avaliação dos resultados

Para avaliação dos resultados de cada experimento, e definição da melhor versão do modelo, utilizamos como métrica a acurácia balanceada, relatada junto da média e o desvio padrão de 5 execuções. Para compreender melhor os resultados fornecidos, analisamos as probabilidades entregues pelos modelos treinados, olhando para a sua distribuição quando o modelo acertava a predição e também quando errava. Separamos os resultados segundo

o *corpus* utilizado e os subconjuntos que os compõe. Além disso, avaliamos a confiança dos modelos ao fazer as predições. Para isso, usamos o Brier Score Loss (ou "Pontuação de Brier") [20], uma métrica proposta para avaliar a qualidade de previsões probabilísticas, especialmente em problemas de classificação. Sua principal função é medir quão bem as probabilidades previstas por um modelo se alinham com os resultados reais observados. Essa métrica varia de 0 (previsões perfeitas) a 1 (pior cenário possível) e é considerada uma "regra de pontuação própria", o que significa que é minimizada apenas quando as probabilidades previstas refletem as verdadeiras probabilidades subjacentes aos dados.

No Capítulo 6, apresentaremos os resultados das métricas para cada subgrupo de textos, conforme o *corpus* de origem, para entender se o modelo se comporta de forma similar para todos eles.

Capítulo 5

Corpus H.IAAC CommonStories

Para o problema de classificação de histórias coerentes e incoerentes, optamos pela criação de um novo *corpus* com histórias adversárias após uma tentativa de uso da base ManPlts [49]. Tal escolha foi feita após analisar os métodos de extração utilizados e a qualidade dos textos gerados. Na Tabela 5.1, mostramos exemplos de como há perda de informação nas histórias geradas, já que diversas frases não tiveram informações extraídas corretamente, e um distanciamento do conteúdo original, uma vez que o modelo de geração usado não recebeu amostras adequadas para seu treinamento. Não considerados as amostras como adequadas, devido ao processo de extração de *plots*, que remove partes importantes da história coerente, e a história incoerente reconstruída pelo gerador, após as manipulações dos *plots*, fica muito diferente da original. Como consequência, as duas classes se tornam mais separáveis, não pela incoerência, como era esperado, mas pela incompletude das histórias incoerentes. Por esse motivo, em nossa visão, ManPlts deixa de ser uma base para avaliação de coerência, sendo uma base para reconhecimento de textos gerados por modelos de linguagem. Desta forma, a criação de um novo *corpus* de histórias foi necessária para possibilitar o treinamento do modelo de classificação. A criação desse *corpus* foi inspirado no trabalho de Ghazarian et al. [49] e um diagrama com as etapas necessárias pode ser visto na Figura 5.1. Os principais pontos de divergência ocorrem na extração de unidades de *plots*, na qualidade do modelo usado para geração e classificação, no treinamento do filtro adversário e na proporção de histórias coerentes e incoerentes usadas para o treinamento do modelo final. Ghazarian et al. [49] usam extrações de unidades de *plots* diferentes para cada subconjunto, enquanto nós usamos um único método, baseado em grandes modelos de linguagem, para todos os subconjuntos. Consideramos os modelos usados por nós como sendo de maior qualidade tanto pelos dados utilizados quanto pela janela de contexto muito maior que a usada por Ghazarian et al. [49]. A divergência no uso do filtro adversário ocorre, pois o reimplementamos e retreinamos, inserindo melhorias, tais como: suporte a múltiplos modelos de linguagem, adição de critérios de avaliação e adição de critérios de seleção.

As etapas principais do processo de criação do *corpus* H.IAAC CommonStories, que dão nome as seções a seguir, são: a extração das unidades de *plot* (Seção 5.1), a sua alteração para geração das histórias adversárias (Seção 5.2), o treinamento do modelo para geração de histórias a partir de *plots* e a inferência (Seção 5.3), e a seleção das histórias adversárias (Seção 5.4).

Tabela 5.1: A extração de unidades de *plot* feita por Ghazarian et al. [49] apresenta grande perda de informação, representado na figura por "Plot ManPlts", se comparado a extração de unidades de *plots* apresentada neste trabalho ("Plot WizardLM" no diagrama).

Texto	So many times have I walked on ruins , the remainings of places that I loved and got used to.. At first I was scared , each time I could feel my city , my current generation collapse , break into the black hole that thrives within it , I could feel humanity , the way I 'm able to feel my body.. After a few hundred years , the pattern became obvious , no longer the war and damage that would devastate me over and over again in the far past was effecting me so dominantly . <newline>It 's funny , but I felt as if after gaining what I desired so long , what I have lived for my entire life , only then , when I achieved immortality I started truly aging . <newline><newline>5 world wars have passed , and now they feel like a simple sickeness that would pass by every so often , I could no longer evaluate the individual human as a being of its own , the importance of mortals is merely the same as the importance of my skin cells ; They are a part of a mechanism so much more advanced , a mechanism that is so dear to my fallen heart a mechanism that I have seen fall and rise so many times , a mechanism that when lost all of which it had , had me loosing my will to live , for the first time in all of my thousands years of existence . <newline><newline>Acceptance , something so important . a skill that has proved itself worthy dozens of times , an ability that looks so easy to achieve , a gift , that I was n't able to aquire in all my years , until now . When the ashes on the ground flew into the now empty air upon humanity 's fall , I felt as if all of it 's weight was crushing me . Ignorance took over and I searched years for a hope , a sign of the very same patterns that I used to watch reappear every hundred years , the very core of my will to exist that was now no more that I so strongly wish was . <newline><newline>If you have ever wondered if silence can drive people crazy , it can.. <newline>I ca n't feel my legs , I have walked for days , just to hear the sound of gravel , crushed bones , crushed buildings and crushed civilizations under my steps to keep my sanity.. until I remembered , the day in my far past . The day of my rebirth , I took out of my pocket a small plastic box , with nine buttons and a small glass window . I could n't believe this was our past , I could n't believe how far we have been able to progress and yet , be destroyed by our own violence . <newline>I slowly dialed the number I was given , exactly 1729 years ago . <newline><newline>I dropped a tear , a tear that was too slow to hit the ground as I got sucked into the darkness that emerged around me . <newline><newline>A chill went through my spine as I saw my destiny rise above me , I could see the white teeth under the dark cloack ... <newline><newline>" You have finally arrived " He projected into my mind , with the most chilling cold and unhuman voice . <newline><newline>" I 'm ready to obey " I answered . I knew who was sitting infront of me , and it was time for me to obey him , after all these years of playing god , even I came to it . <newline><newline>Funny is n't it ? Even by achieving immortality , death , is inescapable .
Plot ManPlts	walked places loved got used to the black hole thrives feel humanity body the patternbecameobvious the war and damage devastate </s>desired lived achieved started truly aging </s>feel like a simple remembered </s></s>believe was our past been able to progressdestroyed our own violence </s>number given</s>hit the groundthe darkness emerged </s>A chill went arrived </s>projected my mind </s>I obey </s>answered </s>who sitting of me obey playing I came
Plot WizardLM	|| I walked on ruins || I loved and used to || I was scared || I could feel city collapse || I could feel humanity || I could feel my body || pattern became obvious || no longer war and damage effecting me |||| It's funny || I felt as if after gaining what I desired so long || what I lived my entire life || only then when I achieved immortality || I started truly aging |||| 5 world wars passed || now feel simple sickness || I could no longer evaluate individual human || importance of mortals same as skin cells || They part of mechanism advanced || mechanism dear fallen heart || mechanism fall rise times || mechanism lost all had me lose will live |||| Acceptance || something so important || a skill || proved itself worthy dozens times || an ability || looks so easy to achieve || a gift || I wasn't able acquire all my years || until now || When ashes on ground flew empty air upon humanity's fall || I felt weight crushing me || Ignorance took over || I searched years hope || sign same patterns || core my will exist || no more || I strongly wish || If silence can drive people crazy || it can |||| I can't feel legs || I walked days || hear sound gravel crushed bones crushed buildings crushed civilizations || I remembered day far past || I took out pocket small plastic box nine buttons small glass window || I couldn't believe past || I couldn't believe progress || I couldn't believe destroyed violence || I dialed number exactly 1729 years ago || I dropped tear tear slow hit ground || I got sucked into darkness emerged around me || A chill went spine || I saw destiny rise above me || I could see white teeth under dark cloak ... || "You have finally arrived| He projected mind || with most chilling cold unhuman voice |||| I ready obey || I knew who sitting front of me || it time for me obey him || after all these years playing god || even I came it || funny isn't it || even achieving immortality death inescapable

5.1 Extração de *Plot*

Na tarefa de extração de *plot*, visamos identificar e recuperar estruturas textuais, as unidades de *plot*, que possibilitem a reconstrução do texto posteriormente. Vários trabalhos implementam o próprio extrator de *plot* [11, 30, 37, 53, 106, 142], pois a própria definição

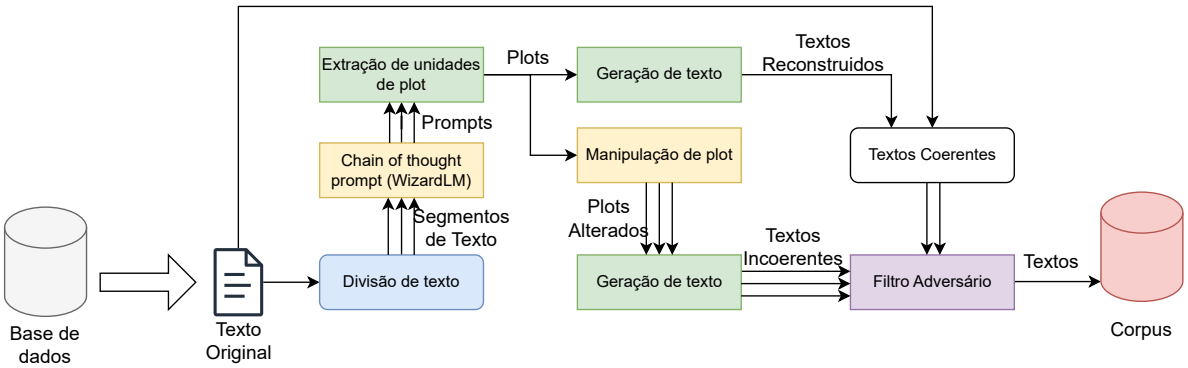

Figura 5.1: Diagrama representando os processos necessários para criação da base de histórias adversárias. Na figura, elementos da cor branca representam artefatos (bases de dados e textos), na cor azul *scripts* o segmentador de textos, em amarelo estão técnicas da literatura, em verde os componentes que utilizam inferência em modelos de linguagem e em lilás o componente que foi reformulado neste trabalho, o Filtro Adversário.

de *plot* e a escolha de uma estrutura para representá-la computacionalmente, varia de autor para autor. Para aqueles que se debruçam no uso de informações sintáticas para construção de uma unidade de *plot*, é comum que se defina uma estrutura computacional (tuplas, objetos, tabelas, etc.) e regras para o seu preenchimento conforme as informações fornecidas por modelos de linguagem treinados para rotulação de classes gramaticais. Há também a possibilidade de usar modelos de linguagem para identificação de palavras-chave ou de unidades de *plots*, nesses casos, uma parcela das bases de dados devem ser destinada ao *fine-tuning* do modelo [7, 89, 131, 156].

Neste trabalho, aproveitamos o poder de grandes modelos de linguagem para fazer a extração sem a necessidade de uma etapa de *fine-tuning* usando *prompt engineering*. Bueno et al. [22][1] propuseram a extração de unidades de *plots* por meio da estratégia de *chain-of-though* com *few-shot learning*, isto é, o *prompt* criado contém instruções da tarefa e exemplos de entrada e saída. Ao usar um único método para extração das unidades de *plots*, criamos uma base mais padronizada em relação à base de Ghazarian et al. [49]. Entretanto, como o modelo usado possui limitações em relação ao tamanho do contexto que pode ser utilizado, as histórias foram divididas em sentenças e incluídas nos *prompts*, de forma que o total de *tokens* na entrada não ultrapassasse 800. Para a inferência ocorrer de maneira mais eficiente, utilizamos o *framework* `VLLM` [83] para inferência em *batches*. Um diagrama com o processo de extração para textos grandes pode ser visto na Figura 5.2.

5.2 Alteração de *Plots*

Nesta dissertação, temos o objetivo de criar um classificador binário de coerência em histórias. Para isso, era necessário histórias incoerentes para o treinamento e avaliação. Para criação dessas histórias, selecionamos aleatoriamente a quantidade de alterações e sua ordem de aplicação. As manipulações feitas seguem as propostas na ManPlts [49], sendo elas: inserção de contradição, inserção de repetição, ordenação ilógica de unidades

[1] Todos os autores são membros do H.IAAC.

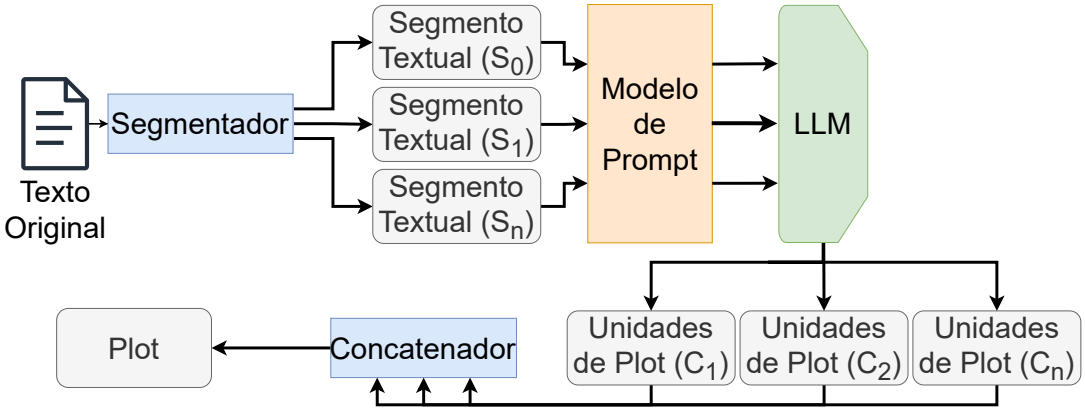

Figura 5.2: Diagrama representando o processo de extração de unidades de plot de textos grandes utilizando um LLM para inferência. Na figura, elementos de cor branca representam artefatos (textos e segmentos de texto), na cor azul *scripts* auxiliares para segmentação e concatenação de texto, em laranja modelo de prompt para inferência definidos por Bueno et al. [22].

de *plots* e substituição aleatória. Na Tabela 5.2, mostramos exemplos de *plots* alterados e os textos resultantes. A seguir, apresentamos uma breve explicação de cada técnica utilizada.

1 – Inserção de contradição. Unidades de *plots* consecutivas são selecionadas aleatoriamente. Para cada unidade de *plot*, um termo aleatório, como um verbo ou substantivo, é extraído e seu antônimo é identificado usando a `ConceptNet` [139], o antônimo é incluído em uma das unidades de *plots* adjacentes. Essa estratégia compromete diretamente a coerência referencial (consistência na referência de elementos) e a coerência causal (relações lógicas de causa e efeito), ao introduzir inconsistências semânticas e quebras abruptas na progressão lógica das ideias.

2 – Inserção de repetições. Textos de unidades de *plots* são inseridos em outras unidades de *plots* não consecutivas ao longo do enredo. Essa prática afeta a coerência temática (manutenção do tópico central) e a progressão lógica, pois repetições descontextualizadas fragmentam o foco narrativo e criam descontinuidades na estrutura do texto, simulando falhas de coesão típicas cometidas por modelos de linguagem.

3 – Ordernação ilógica. Verbos aleatórios são escolhidos das unidades de *plots* e entregues ao modelo `COMET` [19], que prevê um próximo evento ou um pré-requisito para a ação. Os termos que compõem a unidade de plot são então trocados sem adição de um conectivo ou com um conectivo aleatório. Essa manipulação viola a coerência temporal (sequência cronológica de eventos) e a coerência causal, já que a ausência de conectivos adequados ou a escolha aleatória rompe a relação lógica entre ações, gerando uma narrativa desconexa e cronologicamente inconsistente.

4 – Substituição aleatória. Uma unidade de *plot* de outra história é selecionada aleatoriamente para substituir, também de forma aleatória, uma unidade de *plot* da história

Tabela 5.2: Exemplo de um texto original, as respectivas unidades de *plot* extraídas, as manipulações (em ordem e numeradas consoante a Seção 5.2) realizadas sob as unidades de *plots*, o resultado obtido com a manipulação, e o texto gerado a partir de inferência em um modelo treinado para realizar a reconstrução do texto.

Texto original	Unidades de *plot*	Manipulações	*Plot* manipulado	Texto gerado
I started my first campfire the other night. First I gathered wood and kindling. Next I arranged the wood and kindling in the fire pit. I then lit the fire. Then we had an amazing campfire!	|| started first campfire || gathered wood kindling || arranged wood kindling fire pit || lit fire || had amazing campfire	2, 1, 4	|| started first campfire || gathered wood kindling lit fire|| arranged wood kindling fire pit || lit fire || had amazing campfire	A few years ago I started my first campfire. I gathered the wood kindling and lit a fire. I arranged the wood kindling to fire the pit. I lit the fire and had an amazing campfire
Nancy had an uncle who was very sick. She went to visit him to say her final goodbye. With a tear in her eye she told him how much she cared. Later on he passed away peacefully. Nancy was sad but knew he was no longer in pain.	|| Nancy had uncle || uncle was sick || Nancy went visit || She cried || She told him || He passed away peacefully || Nancy was sad	1, 2, 3	|| Nancy had uncle || uncle was sick || When truck arrived, I walked to front || She cried He passed away peacefully|| She told him || He passed away peacefully || Nancy was sad He passed away peacefully	Nancy had an uncle. Her uncle was sick. When the truck arrived, I walked to the front. She cried He passed away peacefully. She told him He passed a peacefully. Nancy was sadly.
Nina had just turned sixteen and wanted her license. She decided to go to driver's ed lessons after school. The lessons were hard, though, and Nina worried. She feared she would fail the road test after all. But in five weeks, Nina was granted a driver's license!	|| Nina turned sixteen || wanted license || decided go driver's ed lessons || after school || lessons hard || Nina worried || feared fail road test || But five weeks || Nina granted driver's license	4, 2, 3	|| Nina turned sixteen || wanted license || decided go driver's ed lessons || after school || lessons hard || Nina worried || feared fail road test || But five weeks || doctor said John's blood levels were alarming	Nina turned sixteen. She wanted an entry to the driver's ed lessons. After school, she was able to learn a hard lesson. Nina wasn't nervous, feared to fail the road test. But five weeks, the doctor said John's blood levels were alarming.

que estamos manipulando. Essa técnica perturba a coerência temática (tópico central unificado) e a coerência referencial, ao introduzir elementos narrativos externos sem ligação semântica ou contextual com o enredo original, resultando em uma estrutura híbrida e desintegrada.

5.3 Geração de Histórias

A geração de histórias a partir de unidades de *plots* é uma tarefa feita por um modelo de linguagem treinado visando a reconstrução do texto, de onde foram extraídas as unidades. Para ser feita, precisamos treinar um modelo com suporte a tarefas *text2text*, ou seja, que tenham textos como entrada e saída, em nosso caso, unidades de *plots* e histórias, respectivamente. Durante o treinamento do modelo, utilizamos somente textos coerentes e suas respectivas unidades de *plots*. A escolha de não usar os textos incoerentes está relacionada a não utilização de uma função de perda mais sofisticada durante o treinamento, não sendo capaz de lidar com amostras negativas.

O modelo escolhido para essa tarefa foi um `mLongT5` pré-treinado, isto é, um modelo `T5`

modificado para suportar entradas de até 16K *tokens* e treinado com *corpora* multilíngue. A escolha desse modelo se deu por sua capacidade de gerar textos longos, característica essencial ao lidar com histórias, e por já possuir suporte à língua portuguesa, uma vez que temos como objetivo gerar histórias nesse idioma.

O treinamento do modelo é feito com parte do *corpus* de histórias e unidades de *plots*, onde a entrada é a unidade de *plot*, somente aquelas sem alteração, e a saída é a história correspondente. Da base resultante da extração de unidades de *plot*, com cerca de 320 mil amostras, que contém textos das bases ROCStories (ROC) [111] e Writing Prompts (WP) [41], reservamos 8% — aproximadamente 25 mil pares de *plot* e história — para treinamento do modelo e 1% para validação — aproximadamente 3 mil pares — e 1% para teste. Para lidar com os separadores de unidades de *plots*, o duplo pipe (||), alteramos o tokenizador do modelo, adicionando-o como um *token* especial para não ser separado em *tokens* menores. Ao fazer esta alteração, precisamos aumentar o tamanho da camada de *embeddings* do modelo em 1, para que o tokenizador consiga lidar com o novo *token*. Após isso, o treinamento ocorreu normalmente.

Durante a etapa de validação, o modelo precisa gerar histórias a partir de unidades de *plots* que não foram vistas durante o treinamento antes de passarem por uma métrica de avaliação. Na etapa de geração, as configurações relativas à estratégia de geração impactam diretamente no tempo de execução e na qualidade das histórias geradas. Como estamos utilizando um modelo com suporte a uma muitos *tokens*, faz sentido escolher uma estratégia que tenha como capacidade de gerar textos longos com uma maior qualidade. Esta é a premissa da estratégia de decodificação de busca contrastiva [140], gerar textos menos repetitivos, problema enfrentado por estratégias determinísticas, e com maior coerência, problema que ocorre com estratégias estocásticas. A busca contrastiva utiliza dois parâmetros, a penalidade por degeneração e o número de top-k *tokens* a serem considerados, o primeiro auxiliando na diminuição de repetições e o segundo reduzindo a geração de textos incoerentes.

Para escolher o melhor *checkpoint* do modelo, utilizamos a métrica *Word Error Rate* (WER) [155], que indica o número médio de erros por palavra de referência. Apesar de ser uma métrica comumente utilizada para avaliação de sistemas de reconhecimento automático de fala, ela é útil em nosso contexto, uma vez que nosso objetivo é que a história gerada seja mais próxima possível da história original. O *checkpoint* com menor WER é utilizado para gerar as histórias para o restante do *corpus*, incluindo para as unidades de *plots* alteradas. O *corpus* poderia ser finalizado após essa etapa, porém como nosso objetivo é que a distinção de histórias coerentes e incoerentes sejam as mais difíceis possíveis, os textos gerados são submetidos a uma etapa de filtragem, onde são mantidas as H histórias mais difíceis para cada conjunto de história e suas modificações.

5.4 Seleção de Histórias

Para construção do *corpus* fizemos a seleção das histórias mais difíceis. No contexto deste trabalho, a dificuldade está relacionada a identificação de incoerência e senso comum. Quanto menor a confiança dos modelos de linguagem na avaliação da coerência de uma

história, maior a dificuldade de uma história.

Ao montar um *corpus* com as histórias mais difíceis, encorajamos a criação de métricas, classificadores e métodos de geração que consigam identificar incoerências textuais e de senso comum, mesmo sem uma ferramenta externa de checagem de fatos.

Para selecionar quais amostras são as mais difíceis, nos baseamos no Filtro Adversário (ou *Adversarial Filter*, AF [167]). AF é um paradigma de seleção de dados em que discriminadores selecionam iterativamente um conjunto adversário de textos, utilizado originalmente para seleção de textos gerados como resposta para questões. Para a seleção ocorrer, é necessário fornecer o texto correto, os textos adversários e o tamanho do conjunto filtrado resultante. Para a criação desse *corpus*, consideramos suficiente a seleção de três histórias adversárias para cada história original.

Uma vez que o código disponibilizado por Zellers et al. [167] possui problemas de desempenho e dá suporte somente a um modelo `BERT`, com uma janela de apenas 512 *tokens*, decidimos por implementar uma nova versão do AF. A versão proposta nesse trabalho é agnóstica a modelo, sendo possível utilizar qualquer modelo de texto disponível na plataforma Hugging Face, e por isso tem uma janela de *tokens* variável, a depender da escolha do modelo. Além disso, a versão proposta aceita mais de uma alternativa correta para cada enunciado, possibilitando a geração de um *corpus* mais balanceado em relação a classes. Para selecionar a versão mais adversária do *corpus*, utilizamos as probabilidades dadas pelos classificadores, treinados a cada nova iteração, para selecionar os textos alterados com maior probabilidade para a classe de texto original. A atualização de uma versão para outra em cada iteração é condicionada a melhora em um critério, sendo possível escolher entre: soma das probabilidades, mediana das probabilidades e mediana da métrica METEOR (*Metric for Evaluation of Translation with Explicit ORdering*, ou métrica para avaliação de tradução com ordenação explícita) [13]. A métrica METEOR foi desenvolvida para avaliar a qualidade de traduções automáticas, mas é constantemente usada para avaliação de outras tarefas. Ela combina elementos de precisão e *recall* (revocação), considerando não apenas a exatidão de palavras, mas também relações semânticas e a ordem das palavras. o METEOR permite flexibilidade ao identificar sinônimos, variações morfológicas (como verbos em tempos diferentes) e até mesmo alinhamentos parciais entre palavras, usando bases linguísticas, ou seja, possui grande potencial para medir a semelhança entre o texto original e um texto reconstruído/gerado.

5.5 Estatísticas do *Corpus*

A versão inicial do Corpus, usada nos experimentos descritos no Capítulo 4 e com resultados mostrados no Capítulo 6, não utiliza todas as amostras que tiveram as unidades de *plot*, já extraídas, durante as etapas de geração e seleção de histórias. Isso ocorre pelo alto custo atrelado a geração de textos em modelos que usam grandes janelas de contextos e a quantidade de amostras que deveriam passar por esse processo. Para a versão inicial, usamos 20% — cerca de 60 mil amostras — dos *plots* extraídos durante as etapas de manipulação de unidades de *plots*. Durante essa etapa, cada amostra resultou na geração de 12 outras com manipulações. Para inferência no modelo *Plot2Text*, usamos tanto o *plot*

original quanto as 12 versões manipuladas para cada amostra, ou seja, 13 textos foram gerados a cada 1 texto original. Na etapa de seleção de histórias, escolhemos apenas 3 das 12 versões manipuladas, a versão original e a versão reconstruída para compor o *corpus*. Ao final dessas etapas, a versão inicial contou com 120 mil amostras de textos coerentes e 360 mil amostras de textos incoerentes. Todos esses procedimentos para criação do *corpus* foram executados com textos em inglês (H.IAAC CommonStories (EN)) e então traduzidos para o português (H.IAAC CommonStories (PT)) por limitações de recursos.

Capítulo 6

Experimentos e Resultados

Neste capítulo, detalhamos a metodologia experimental e analisamos os resultados obtidos em três eixos principais: a construção do *corpus* multilingue H.IAAC CommonStories com textos adversarialmente manipulados (Seção 6.1) e a avaliação de modelos de classificação de coerência textual em diferentes cenários (Seção 6.2).

A primeira seção descreve o processo de geração adversarial do *corpus*, abordando desde a manipulação sistemática de unidades de (*plots*) até o refinamento por filtragem adversarial. Apresentamos estatísticas detalhadas sobre a distribuição de manipulações e análises de consistência estrutural entre as versões em português e inglês.

Na sequência, focamos nos experimentos com classificadores de coerência baseados no modelo `XLM-Roberta`, comparando três abordagens distintas: *pipeline vanilla*, RSTMix (integração de características RST) e POSMix (incorporação de informações morfossintáticas). Avaliamos o desempenho em diferentes *corpora* (GCDC, H.IAAC CommonStories e FakeTrue.BR), considerando aspectos como: confiança das predições, robustez cross-lingua e generalização em cenários *zero-shot*.

6.1 Geração do *Corpus* H.IAAC CommonStories

Alteração de *plots*. A alteração de *plots* é a principal ferramenta para criação de textos incoerentes no que se refere a geração de textos a partir de unidades de *plot*. A qualidade dos textos gerados, neste caso a semelhança com um texto original, depende não só do modelo utilizado para geração, mas também dos tipos de incoerência inseridos e a sua quantidade. Em nossos experimentos, utilizamos 4 tipos de manipulação, aplicadas de maneira cumulativa as unidades de *plot*. A quantidade de manipulações, variando de 1 a 4, e a ordem foram decididas de forma aleatória, contudo, garantimos que fossem atribuídas com a mesma probabilidade para que a quantidade de cada manipulação não fosse discrepante, conforme mostram as Tabelas 6.1 e 6.2 com as estatísticas de atribuição de manipulações.

Geração de texto. Para o treinamento do modelo *Plot2Text*, utilizamos 8% dos textos originais e os respectivos *plots* extraídos, sendo 90% das amostras para o conjunto de treino, 5% para o conjunto de validação e 5% para teste. As amostras utilizadas para o

Tabela 6.1: Porcentagem de amostras com cada tipo de manipulação de unidades de *plots*. A soma das porcentagens é maior que 100% porque uma amostra pode conter mais de um tipo de manipulação. A variação na quantidade é menor que 1,7 pontos percentuais para quaisquer tipo de manipulação tomada 2 a 2.

Tipo de manipulação	**Amostras com a manipulação (%)**
Inserção de contradição	24,8
Inserção de repetições	25,2
Ordenação ilógica	25,0
Substituição aleatória	25,2

Tabela 6.2: Porcentagem de amostras, apenas alteradas, com determinada quantidade de manipulações, podendo variar de 1 a 4.

Quantidade de manipulações	**Porcentagem das amostras (%)**
1	25,2
2	25,1
3	24,9
4	24,7

treinamento do modelo são descartadas após o treinamento do modelo e não participam das outras etapas. Para realizar o treinamento adequado do modelo, variamos o *scheduler* de *learning rate* e seus parâmetros, a quantidade de passos utilizados para o *warmup* do modelo e quantidade de épocas para treinamento. Ao final dos testes, a melhor combinação de parâmetros com os valores mais baixos de *WER*, com média de $0,586 \pm 0,0003$ para 3 experimentos, possuía os seguintes valores: *scheduler* de cosseno com 8 reinícios e *warmup* de 5% do conjunto de treino, uma *learning rate* igual a 5^{-4}, 10 épocas, tamanho de *batch* igual a 4, top-k igual a 5 e penalidade alfa igual a $0,85$.

Para auxiliar na análise dos textos gerados, incluímos estatísticas, como quantidade de EDUs e quantidade de relações, extraídas com o auxílio da RST. O primeiro ponto a se notar está relacionado a distribuição das relações na base em português e inglês, Figuras 6.1 e 6.2, que foi igual para ambos os idiomas, corroborando com os fundamentos da RST e mostrando a consistência do `DMRST` *parser* entre idiomas.

Dentre as relações, as que apareceram com mais frequência nesse conjunto de dados foram as mono-nucleares, com exceção da relação *Joint*, frequente no uso de listagens, somando quase 20% do total de relações. Vale lembrar, que esse tipo de relação esteve presente em abundância nos *corpus* GCDC e FakeTrue.BR, e pode, na verdade, indicar uma falha do *parser* em identificar a relação correta [32]. Essa suspeita é reforçada quando comparamos as distribuições contendo somente os textos coerentes e somente os textos não coerentes, pelos seguintes motivos: nos textos incoerentes, a quantidade da relação *Joint* é 3 pontos percentuais maiores que para a mesma relação em textos coerentes.

Filtro adversário. Várias iterações foram feitas durante o processo de AF para criação do *corpus* H.IAAC CommonStories em inglês. Os textos selecionados na etapa

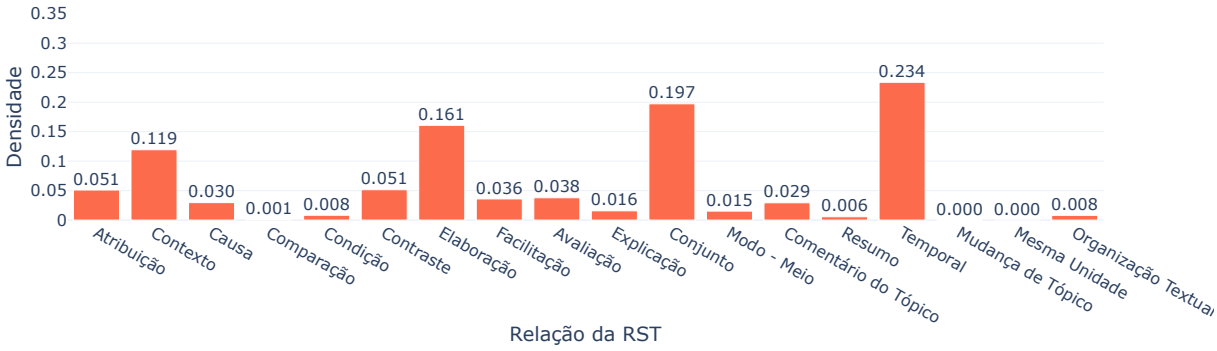

Figura 6.1: Distribuição de relações da RST que aparecem nos textos da base H.IAAC CommonStories em português.

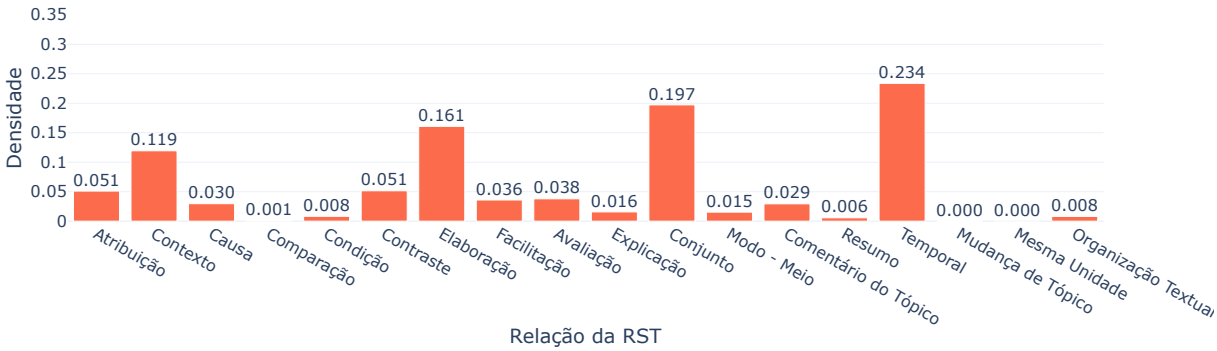

Figura 6.2: Distribuição de relações da RST que aparecem nos textos da base H.IAAC CommonStories em inglês.

final compõem o *corpus* atual, sendo divididos em treino, validação e teste. Investigamos estes textos com o intuito de verificar se o modelo havia privilegiado textos com maior ou menos quantidade de manipulações, ou ainda se havia selecionado mais textos com determinado tipo de manipulação. Como mostra a Tabela 6.2, houve pouca variação, menos de 0,06 pontos percentuais, na quantidade de cada classe (quantidade de manipulações). Acreditamos que essa pequena diferença não seja suficiente para entendermos os vieses adquiridos pelo modelo ao longo do treinamento. Semelhantemente, não identificamos sequências que se sobressaíssem, significativamente, dentro de cada quantidade de manipulações.

6.2 Resultados para Classificação de Coerência

O treinamento do classificador de coerência, tanto para o *corpus* GCDC quanto o H.IAAC CommonStories, foi feito com o modelo `XLM-Roberta`. Para carregar o modelo, por meio da interface `AutoModelForSequenceClassification`, utilizamos a biblioteca Hugging Face. Para cada *corpus*, executamos as 3 variações de *pipeline* propostos, com 5 execuções para cada versão, apresentados na Seção 4.2, e fizemos experimentos alterando

a quantidade de ciclos usados pelo *scheduler* e a *learning rate*. Com as observações referentes ao comportamento do *scheduler* e da *learning rate* (LR) com o *Pipeline vanilla* no *corpus* GCDC, diminuímos a quantidade de experimentos com os demais *pipelines* e outras bases. Após finalizar os experimentos com a versão em inglês, executamos os experimentos com a base em português usando as mesmas configurações do melhor resultado obtido para a outra versão.

Adicionalmente, para o *corpus* GCDC, incluímos os resultados obtidos com a utilização de um mecanismo voltado para diminuição do tempo de treinamento de modelos de linguagem, os *Adapters* [63]. Em resumo, *Adapters* são métodos baseados na adição de camadas, com parâmetros treináveis, após as camadas de atenção e camadas totalmente conectadas de um modelo pré-treinado, em que houve o congelamento de camadas, para reduzir o uso de memória e acelerar o treinamento. O método varia dependendo do *Adapter*, podendo ser simplesmente uma camada extra adicionada ou envolver passos mais complexos, como atualizações de peso, decomposição de matrizes de peso, entre outros métodos. Apesar da simplicidade, os *Adapters* são tipicamente pequenos, mas já demonstraram desempenho comparável a um modelo totalmente ajustado e permitem o treinamento de modelos maiores com menos recursos [64, 132]. O *Adapter* LoKr (*Low-Rank Kronecker Product*) [68, 163], uma variante do LoRA [165] que faz aproximação da matriz de peso com duas matrizes de *low-rank* e as combina com o produto de Kronecker [58], foi escolhido pelo seu desempenho superior, e baixo custo, comparado a outros *Adapters* disponíveis na data de levantamento [163].

6.2.1 GCDC

A Tabela 6.3 apresenta um sumário dos resultados para o *corpus* GCDC. Os experimentos foram iniciados pelo *pipeline vanilla* sem adição de *Adapters*. Primeiro, variamos o parâmetro de quantidade de ciclos de reinício da *LR*, mantendo a *LR* em 5×10^{-5}. Os primeiros resultados para a métrica de acurácia balanceada, com 10 ciclos, eram superiores a $0,7$, resultado superior ao apresentado para métrica de acurácia, que tende a ser maior que a balanceada, por Abhishek et al. [3]. Após esses resultados, passamos a diminuir a quantidade de ciclos, até ser igual a 1, para observar o comportamento do classificador, que continuou a apresentar resultados melhores. Por fim, diminuímos a *LR*, assim como recomendado por Godbole et al. [50], como último experimento para o *pipeline vanilla*. O resultado obtido nessa configuração atingiu o maior valor médio para a acurácia balanceada. Repetimos o experimento, mas dessa vez com a adição do *Adapter Lokr*. Porém, o modelo não apresentou bons resultados para a métrica, apresentando o pior resultado para o *pipeline vanilla*.

A adição de *Adapter* ocasionou o pior resultado para o *pipeline vanilla* e *pipeline* POSMix. Contudo, para o *pipeline* RSTMix, o uso de *Adapter* não representou um grande impacto na acurácia balanceada do modelo (0,004 na média). Ainda que o resultado geral, que pode ser visto na Tabela 6.3, não seja favorável aos modelos que usaram a RST e a POS, separamos os resultados de acordo com cada *pipeline* e subconjunto para análise.

Ao inspecionar as probabilidades de predição para cada *pipeline*, com o auxílio da Tabela 6.4 e das Figuras 6.3 e 6.4, podemos observar alguns pontos:

Tabela 6.3: Resumo dos principais experimentos para o *corpus* GCDC (EN). Apresentamos os valores de acurácia balanceada média e o desvio padrão para 5 execuções. Os valores de acurácia balanceada usados para média são sempre os melhores para validação da execução.

Pipeline	***Adapter***	**Cycles**	***Learning Rate***	**Acurácia Balanceada ↑**
Vanilla	–	10	5×10^{-5}	$0{,}712 \pm 0{,}027$
		8	5×10^{-5}	$0{,}720 \pm 0{,}002$
		5	5×10^{-5}	$0{,}717 \pm 0{,}002$
		3	5×10^{-5}	$0{,}715 \pm 0{,}003$
		2	5×10^{-5}	$0{,}717 \pm 0{,}002$
		1	5×10^{-5}	$0{,}711 \pm 0{,}004$
		1	1×10^{-5}	$\underline{\mathbf{0{,}724}} \pm 0{,}001$
	Lokr	1	1×10^{-5}	$0{,}703 \pm 0{,}001$
RSTMix	–	2	5×10^{-5}	$0{,}671 \pm 0{,}003$
		1	5×10^{-5}	$0{,}672 \pm 0{,}002$
		1	1×10^{-5}	$\underline{0{,}690} \pm 0{,}002$
	Lokr	1	1×10^{-5}	$0{,}686 \pm 0{,}001$
POSMix	–	1	1×10^{-5}	$\underline{0{,}681} \pm 0{,}004$
	Lokr	1	1×10^{-5}	$0{,}590 \pm 0{,}004$

- O *pipeline* com POSMix apresenta pouca variação nos valores de probabilidades, permanecendo, majoritariamente, na faixa de [0,5, 0,6]. Ainda assim, o *pipeline* possui um Brier Score *loss* menor para o *corpus* em relação aos demais. Contudo, ao errar a predição, o modelo também o faz com menor confiança que os demais *pipelines*.
- O *pipeline* com RSTMix, por outro lado, tem as probabilidades distribuídas ao longo da faixa de [0,5, 1], concentradas em 0,6 e 0,8. Com a concentração marcada em dois valores para um mesmo tamanho de texto, temos como hipótese que o modelo tenha se adaptado melhor a classificar textos de domínios específicos ou com determinados estilos de escrita. Para validarmos essa hipótese, separamos as probabilidades por subconjuntos (Seção 6.2.1).
- Apesar de obter melhor desempenho para a métrica de acurácia balanceada, é fato que o *pipeline vanilla* também teve a maioria das probabilidades das predições próximas a 1 mesmo para os casos em que errava a predição, ou seja, o classificador apresenta alta confiança mesmo quando erra, o que diminui o valor para métrica Brier Score *loss*.

Fonte dos dados

Para confirmar a hipótese de que o modelo se ajustou melhor a determinados tipos de texto, separamos os textos e probabilidades segundo o subconjunto, computamos a média

Tabela 6.4: Síntese do Brier Score *loss* para os *corpus* GCDC e H.IAAC CommonStories. A métrica foi calculada de forma separada para os subconjuntos de cada *Corpus* e de forma global, isso é, usando todos o *corpus*. Para essa métrica, quanto menor o valor melhor. Valores sublinhados representam o melhor resultado para o pipeline no subconjunto de cada *corpus*.

Pipeline	***Corpus***	**Subconjunto**	**Brier Score *loss* ↓**
Vanilla	GCDC	Clinton	0,300
		Yelp	<u>0,296</u>
		Enron	0,303
		Yahoo	0,422
		Todos	0,331
RSTMix	GCDC	Clinton	0,272
		Yelp	0,268
		Enron	<u>0,263</u>
		Yahoo	0,368
		Todos	0,294
POSMix	GCDC	Clinton	0,236
		Yelp	<u>0,234</u>
		Enron	0,239
		Yahoo	0,260
		Todos	0,243
Vanilla	H.IAAC CommonStories (EN)	ROC	0,442
		WP	<u>0,437</u>
		Todos	0,440
RSTMix	H.IAAC CommonStories (EN)	ROC	0,426
		WP	<u>0,419</u>
		Todos	0,423
POSMix	H.IAAC CommonStories (EN)	ROC	0,406
		WP	<u>0,402</u>
		Todos	0,404
Vanilla	H.IAAC CommonStories (PT)	ROC	0,441
		WP	<u>0,434</u>
		Todos	0,438
RSTMix	H.IAAC CommonStories (PT)	ROC	0,426
		WP	<u>0,418</u>
		Todos	0,423
POSMix	H.IAAC CommonStories (PT)	ROC	0,397
		WP	<u>0,395</u>
		Todos	0,396

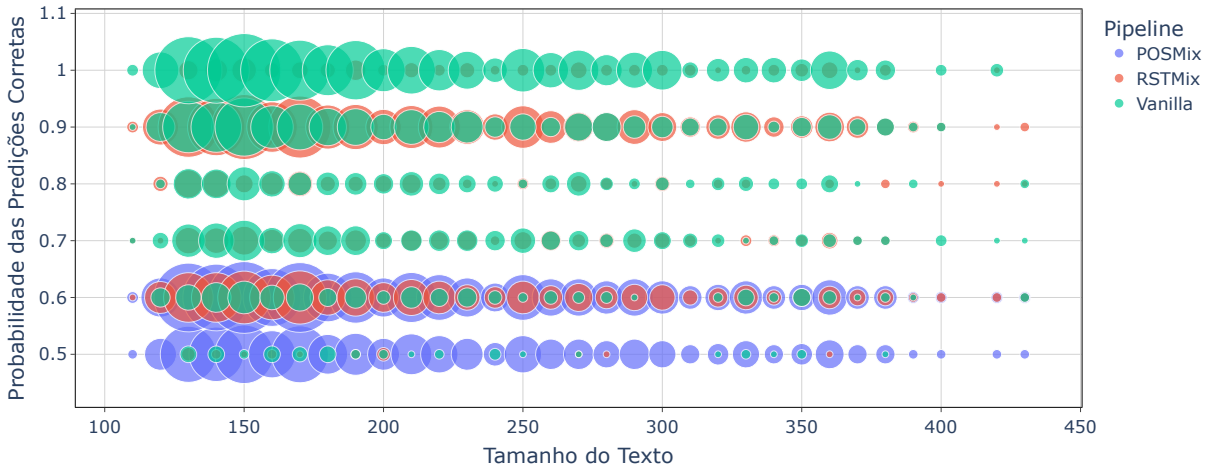

Figura 6.3: Distribuição de probabilidade das predições corretas, no *corpus* GCDC, para os *Pipelines Vanilla, RSTMix* e *POSMix*, representado pelas cores verde, vermelho e azul, respectivamente. O tamanho dos círculos é determinado pela quantidade de textos com aquele tamanho e probabilidade.

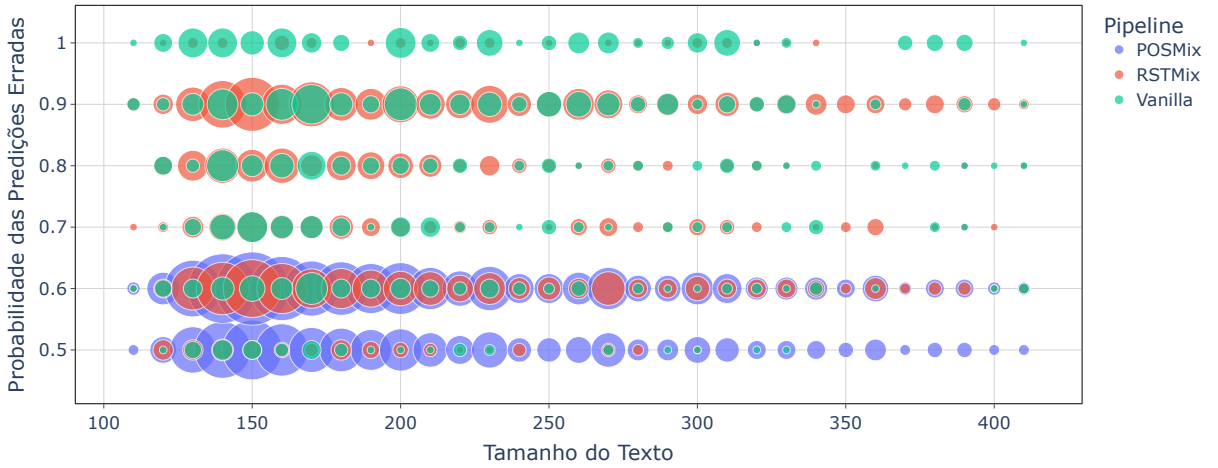

Figura 6.4: Distribuição de probabilidade das predições erradas, no *corpus* GCDC, para os *Pipelines Vanilla*, RSTMix e POSMix, representado pelas cores verde, vermelho e azul, respectivamente. O tamanho dos círculos é determinado pela quantidade de textos com aquele tamanho e probabilidade.

e geramos um gráfico violino, que pode ser visto na Figura 6.5, com a distribuição das probabilidades. Além disso, calculamos o Brier Score *loss* em cada subconjunto do *corpus* (Tabela 6.4).

No *pipeline* RSTMix, observamos que a média das probabilidades, de predições corretas, dos subconjuntos Clinton e Yahoo são maiores que os demais e que para este *pipeline* o modelo erra com maior confiança as predições do subconjunto Yelp. Para o subconjunto Enron, o modelo acerta os resultados com maior confiança e erra com menor confiança que os demais subconjuntos. Esse comportamento nos indica um melhor desempenho e adaptação do modelo a estes subgrupos.

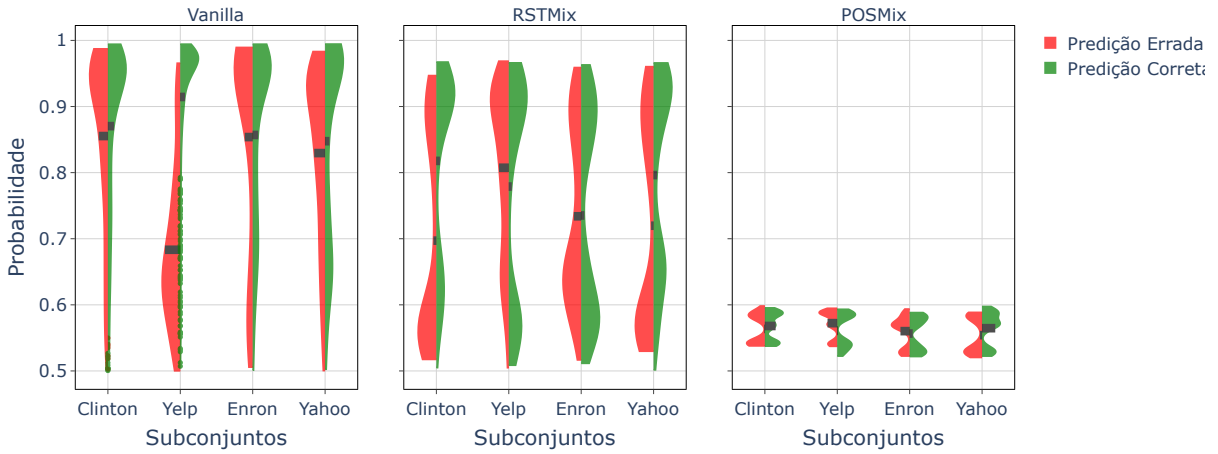

Figura 6.5: Distribuição de probabilidades para cada *pipeline* (*vanilla*, RSTMix e POSMix) conforme a fonte/subconjunto de dados no *corpus* GCDC. As probabilidades para previsões corretas são exibidas na cor verde, enquanto na cor vermelha são exibidas as probabilidades para previsões errôneas.

Esta mesma adaptação não acontece para o *pipeline vanilla*, em que em nenhum dos subconjuntos o modelo erra com baixa confiança, com exceção do subconjunto Yelp, em que apresenta uma confiança menor que os demais, e, portanto, um melhor Brier Score *loss*. Essa diferença pode ser causada pela natureza da anotação de coerência deste subconjunto, que tende a rotular texto com escrita coloquial como incoerentes e textos que seguem a norma padrão como coerentes.

Assim como ressaltado anteriormente, o *pipeline* de POSMix apresenta um comportamento bem distinto, acumulando as probabilidades, tanto corretas como incorretas, na faixa de [0,5, 0,6]. Porém, tal acúmulo não nos impede de observar que, contrastivamente ao que ocorre no *pipeline vanilla*, o classificador erra com maior confiança do que acerta as previsões para o subconjunto Yelp. Por outro lado, o modelo apresenta o mesmo comportamento que os treinados nos demais *pipelines* para o subconjunto Yahoo, acertando, em média, com maior confiança do que quando erra.

Para corroborar com a análise feita em relação às probabilidades e confiança dos modelos treinados em cada *pipeline*, calculamos a acurácia balanceada para cada subconjunto e as organizamos na Tabela 6.5. Perceba que, assim como esperado pela análise anterior, o desempenho para o subconjunto Yelp foi o pior dentre os demais para todos os *pipelines*, apresentando a menor acurácia balanceada neles. A acurácia balanceada por subconjunto ainda mostra que, apesar do desempenho geral superior do método RSTMix em relação ao POSMix, para estilos específicos de texto, como do subconjunto Yahoo, o mesmo comportamento não se aplica.

6.2.2 H.IAAC CommonStories

Assim como para o *corpus* GCDC, a acurácia balanceada do *pipeline vanilla* foi superior, em média, aos demais para o *corpus* H.IAAC CommonStories (Tabela 6.6). Contudo,

Tabela 6.5: Acurácia balanceada para cada *pipeline* calculada consoante a cada subconjunto do *corpus* GCDC. Os valores em destaque representam o melhor resultado obtido e os valores sublinhados o segundo melhor resultado para métrica escolhida.

Pipeline	**Clinton**	**Enron**	**Yahoo**	**Yelp**
Vanilla	**0,731** ± 0,002	**0,724** ± 0,002	**0,741** ± 0,002	**0,645** ± 0,004
RSTMix	0,646 ± 0,005	0,706 ± 0,002	0,705 ± 0,005	0,598 ± 0,003
POSMix	0,677 ± 0,002	0,704 ± 0,001	0,727 ± 0,003	0,591 ± 0,001

a diferença no desempenho dos modelos foi menor que para o *corpus* GCDC (EN) — $0,003$ contra $0,034$, no melhor caso. O desempenho dos modelos treinados foi superior, na métrica acurácia balanceada, para os textos em inglês, idioma original dos textos, do que em português, com textos traduzidos. A degradação do desempenho com textos traduzidos reforça as descobertas de Intrator et al. [69], que mostra que o uso de pré-tradução dos dados impacta negativamente no desempenho dos modelos.

Não apresentaremos os resultados com *Adapters*, pois, assim como para o *corpus* anterior, o seu uso para este *corpus* não ocasionou melhoras no desempenho dos *pipelines*. Para este *corpus*, além da acurácia balanceada por subconjunto em cada idioma, incluímos os resultados por número de manipulações de *plots* e por tipo de manipulação. Para análises referentes à confiança dos *pipelines*, usaremos o Brier Score *loss*, uma vez que a grande quantidade de textos dificulta a visualização das distribuições como foi feito para o *corpus* GCDC.

Tabela 6.6: Resumo dos principais experimentos para o *corpus* H.IAAC CommonStories. Apresentamos os valores de acurácia balanceada média e o desvio padrão para 5 execuções. Os valores de acurácia balanceada usados para média são sempre os melhores para validação da execução.

Pipeline	**Cycles**	***Learning Rate***	**Acurácia Balanceada**
Vanilla (EN)	1	1×10^{-5}	**0,794** ± 0,001
RSTMix (EN)	1	1×10^{-5}	0,791 ± 0,002
POSMix (EN)	1	1×10^{-5}	0,784 ± 0,001
Vanilla (PT)	1	1×10^{-5}	**0,779** ± 0,001
RSTMix (PT)	1	1×10^{-5}	0,774 ± 0,002
POSMix (PT)	1	1×10^{-5}	0,713 ± 0,001

Fonte de dados

Assim como para o *corpus* GCDC, os conjuntos de dados que compõem o H.IAAC CommonStories possuem estilos de escrita distintos e uma grande diversidade de tamanhos de texto, ainda que o número de subconjuntos que o compõem seja menor, o que nos levou a investigar o resultado para cada subconjunto.

Diferentemente do *corpus* anterior, os resultados para o *pipeline* com RSTMix (Tabela 6.7), em inglês, apresentaram resultados melhores para um dos subconjuntos, o conjunto de textos ROC. Contudo, o mesmo não acontece para as versões dos dados em português, no qual o *pipeline vanilla* apresenta o melhor resultado para os conjuntos ROC e WP.

Apesar da vantagem na classificação dos textos do subconjunto WP (EN), tanto para os modelos treinados com textos em português quanto em inglês, a diferença, na média, entre os subconjuntos é pequena, menor que 1,5 pontos percentuais, $0,790 \pm 0,001$ na ROC (EN) com RSTMix contra $0,803 \pm 0,002$ com *pipeline vanilla* na WP (EN). Esse mesmo comportamento é percebido analisando o Brier Score *loss*, que mostra uma confiança maior do modelo nas predições para esse subconjunto.

Tabela 6.7: Acurácia balanceada para cada *pipeline* calculada consoante a cada subconjunto do *corpus* H.IAAC CommonStories. Os valores em destaque representam o melhor resultado obtido e os valores sublinhados o segundo melhor resultado para métrica escolhida.

Pipeline	**ROC**	**WP**
Vanilla (EN)	<u>0,788</u> ± 0,000	**0,803** ± 0,002
RSTMix (EN)	**0,790** ± 0,001	<u>0,797</u> ± 0,004
POSMix (EN)	0,783 ± 0,002	0,786 ± 0,004
Vanilla (PT)	**0,777** ± 0,000	**0,782** ± 0,002
RSTMix (PT)	<u>0,775</u> ± 0,001	<u>0,771</u> ± 0,004
POSMix (PT)	0,715 ± 0,002	0,711 ± 0,004

Quantidade de manipulações

Ainda que todos os textos com manipulações sejam considerados pertencente a uma mesma classe — a dos textos incoerentes — para tentar descobrir possíveis vieses, separamos os textos por quantidade de manipulações e calculamos a acurácia como se cada uma fosse uma classe diferente. Esse cálculo foi feito para o modelo com melhores resultados de acurácia, o modelo treinado com o *corpus* H.IAAC CommonStories (EN). Para os *pipelines* RSTMix e POSMix, a diferença entre as classes foi mais significativa do que para o *Pipeline Vanilla*.

Nossa hipótese inicial era de que textos com quantidades maiores de manipulações seriam identificados mais facilmente como incoerentes, ou seja, a acurácia cresceria de forma proporcional a quantidade de manipulações. Contudo, conforme apresentado na Tabela 6.8, essa hipótese foi refutada pelos resultados obtidos, que mostram uma acurácia maior para textos com 2 manipulações, no *Pipeline Vanilla*, e 3 manipulações nos demais *pipelines*, sem um padrão de crescimento ou decrescimento com o aumento e diminuição da quantidade de manipulações.

Tabela 6.8: Sumarização dos resultados obtidos para a métrica Acurácia, divididos por quantidade de manipulações e Pipelines.

Pipeline	***Corpus***	**Manipulações**	**Acurácia**
Vanilla	H.IAAC CommonStories (EN)	1	0,792
		2	0,793
		3	0,792
		4	0,792
RSTMix	H.IAAC CommonStories (EN)	1	0,731
		2	0,719
		3	0,735
		4	0,722
POSMix	H.IAAC CommonStories (EN)	1	0,761
		2	0,759
		3	0,762
		4	0,758

6.2.3 FakeTrue.BR

Os experimentos realizados para o *corpus* FakeTrue.BR, nesta dissertação, ocorreram somente no cenário *zero-shot*, fazendo predições com os modelos treinados com um dos *corpus*: GCDC, H.IAAC CommonStories (EN) e H.IAAC CommonStories (PT). Para comparar os resultados, calculamos a acurácia balanceada (Tabela 6.9). Contudo, como se trata de um conjunto de dados com classes balanceadas, na prática, poderíamos ter usado a acurácia, métrica reportada no artigo que apresenta o conjunto.

Apesar de não se sobressair nos experimentos com os *corpora* em que foi treinado, o *pipeline* POSMix registrou desempenho superior ao demais para este cenário. Seu maior destaque foi quando treinado no *corpus* H.IAAC CommonStories (PT), em que obteve a melhor acurácia entre todos os *pipelines*, com $0,757 \pm 0,002$. Para o conjunto H.IAAC CommonStories (EN), seu desempenho foi de $0,663 \pm 0,005$, o que é inferior ao desempenho do RSTMix, mas demonstra uma robustez considerável do modelo, uma vez que além de ser treinado em outro conjunto, a língua alvo também é diferente daquela vista na etapa de treinamento.

De forma geral, o *pipeline* RSTMix apresentou uma maior estabilidade, com menor diferença na acurácia entre os conjuntos utilizados para treino. Enquanto isso, o *Pipeline Vanilla* apresenta o melhor resultado quando o *corpus* GCDC é utilizado para treinamento, enquanto os *pipelines RSTMix* e o *POSMix* apresentaram melhores resultados nos *corpus* H.IAAC CommonStories, especialmente em português, sendo o último, o melhor resultado para a tarefa. Ainda que distante da acurácia média apresentada no artigo original, $0,937 \pm 0,005$, em nosso melhor caso, conseguimos acertar a classe de mais de 3/4 dos textos, treinando modelos em outras tarefas e com outros conjuntos.

Tabela 6.9: Sumarização dos resultados obtidos com a métrica acurácia balanceada para o *corpus* FakeTrue.BR em um cenário *zero-shot*.

Pipeline	**Training Corpus**	**Acurácia Balanceada**
Vanilla	GCDC	0,731 ± 0,008
	H.IAAC CommonStories (EN)	0,648 ± 0,002
	H.IAAC CommonStories (PT)	0,624 ± 0,002
RSTMix	GCDC	0,615 ± 0,005
	H.IAAC CommonStories (EN)	0,696 ± 0,004
	H.IAAC CommonStories (PT)	0,676 ± 0,004
POSMix	GCDC	0,595 ± 0,006
	H.IAAC CommonStories (EN)	0,663 ± 0,005
	H.IAAC CommonStories (PT)	**0,757** ± 0,002

Capítulo 7

Conclusões

Esta dissertação investigou a classificação de coerência textual em narrativas e a integração de estruturas retóricas (RST) e informações sintáticas (POS) em modelos de linguagem. Propusemos uma metodologia que combina teorias linguísticas com técnicas de aprendizado de máquina, demonstrando que a expansão do vocabulário de modelos com símbolos baseados em RST e POS pode melhorar, em determinados cenários, a identificação de incoerências em histórias. A criação do Corpus H.IAAC CommonStories, composto por pares de narrativas coerentes e modificadas, serviu de base para treinar e avaliar classificadores, além de revelar possibilitar a análise de padrões estruturais em trabalhos futuros.

Os experimentos realizados mostraram que modelos não servem apenas para detecção de incoerências, mas também apresentaram resultados promissores em testes *zero-shot* na base de dados FakeTrue.BR, uma base brasileira de desinformação. Apesar dos avanços, a dependência de anotações automáticas de RST e POS introduz possíveis ruídos, e homogeneidade de gêneros textuais e temáticas no *corpus* H.IAAC CommonStories limita a aplicação em gêneros narrativos além de histórias. Além disso, a correlação entre coerência e desinformação, embora estatisticamente relevante, demanda investigação mais aprofundada para distinguir entre textos intencionalmente enganosos e aqueles simplesmente mal estruturados.

Nas seções desse capítulo, respondemos às questões de pesquisa (Seção 7.1) e discutimos sobre as limitações enfrentadas no desenvolvimento desta dissertação, sejam elas motivadas por recursos computacionais, materiais ou da metodologia empregada (Seção 7.2). Além disso, mapeamos os possíveis próximos passos que decorrem dos resultados e limitações dessa pesquisa, sejam eles para melhorar os resultados obtidos ou interpretabilidade deles, superar limitações encontradas durante o desenvolvimento dessa dissertação ou incentivar a exploração de abordagens híbridas (linguística + IA) (Seção 7.3).

7.1 Respostas às Questões de Pesquisa

As perguntas de pesquisa foram projetadas para orientar o processo de pesquisa e definir o conjunto de experimentos. Seguindo os resultados do experimento, respondemos a cada pergunta de acordo.

A expansão do vocabulário de modelos de linguagem com símbolos especiais, baseados em RST e POS, auxiliam na identificação de histórias coerentes? Apenas para determinados subconjuntos e somente no Corpus H.IAAC CommonStories. Como mostrado na Figura 6.7, a classificação utilizando o *Pipeline RSTMix* apresentou resultados melhores, na métrica acurácia balanceada, para o subconjunto com textos da ROCStories, quando treinado e avaliado com textos em inglês.

Como utilizar a classificação de textos coerentes como um *proxy* para classificação de desinformação no cenário *offline*? Modelos com expensão do vocabulário com símbolos especiais baseados em POS, utilizando textos em português, apresentaram desempenho satisfatório, acertando mais de 3\4 das previsões (Figura 6.9). Para seu treinamento e utilização como um *proxy* para classificação de desinformação no cenário *offline*, o modelo deve ser treinado com textos de cunho narrativo, com amostras de textos coerentes e incoerentes, o último criado por um processo de manipulação de unidades de *plots*, na tarefa de classificação de histórias coerentes seguindo *Pipeline POSMix*.

7.2 Limitações

Dados anotados. Apesar do caráter multilinguagem dos modelos treinados, o *DMRST Parser*, utilizado para extração de informações de RST, tem suporte a somente cinco idiomas, sendo eles: inglês, português, espanhol, alemão, holandês e basco. Essa limitação tem origem principalmente pela escolha do extrator de RST, mas também pela disponibilidade de bases de dados. Além da dificuldade em encontrar especialistas capazes de fazer a anotação correta da RST, anotadores podem criar diferentes anotações para o mesmo texto, seja por descuido ou interpretação diferente de determinada cláusula. Apesar da existência de um manual de anotação em inglês, regras e marcadores de discurso não podem ser diretamente traduzidos para outro idioma, por exemplo: a ordem dos elementos que compõem uma frase em português difere da ordem utilizada no inglês. Portanto, um marcador de quebra de discurso em português não representará uma quebra de discurso no inglês. Tais fatores, dificultam a criação de novas bases de dados e de novos extratores.

Tradução de textos. A versão em português do *corpus* H.IAAC CommonStories foi produzida por meio da tradução dos textos já selecionados da versão em inglês. Apesar de todo o *pipeline* dar suporte ao uso de textos em português, seria necessário fazer a tradução de todos os textos, bem como o treinamento dos modelos de extração de *plots*, alteração de *plots* e geração de texto. Diversos modelos de tradução automática foram testados para o andamento desta e de outras pesquisas do H.IAAC. Contudo, na ocasião não foram encontrados modelos que produzissem traduções para textos longos com qualidade ou com qualidade superior à tradução feita pela ferramenta do Google (*Google Translate*). Desta forma, decidimos prosseguir com a tradução de maior qualidade, feita pela ferramenta citada, mas somente no conjunto final, uma vez que a quantidade de traduções feitas gratuitamente é limitada.

7.2.1 Recursos computacionais

Ainda que a metodologia proposta fora executada com modelos de linguagem com suporte a grande quantidade de *tokens*, o *corpus* gerado e os experimentos realizados não são isentos dos limites computacionais disponíveis. Durante a fase de extração de *plots*, o texto é particionado para a inferência no modelo. Tal operação é executada porque ao incluir os textos completos, ultrapassávamos a quantidade máximo de *tokens* que o modelo conseguia cobrir sem se esquecer da tarefa alvo. Ao solucionar o problema com o particionamento do texto, permitimos que outro tipo de problema surgisse, a possível quebra de contexto ao fazer a separação dos textos e, portanto, uma possível degradação na qualidade das unidades de *plot* extraídas, fruto da falta de um contexto presente apenas no texto completo.

No treinamento do modelo *Plot2Text*, a quantidade de memória de GPU disponível limita os textos que podem ser utilizados. Para o treinamento de variações de modelos *Seq2Seq*, cada amostra do *corpus* é transformada em uma sequência de *tokens*. Desta vez não só a entrada, no nosso caso as unidades de *plot*, mas também o alvo, i.e., o texto a ser gerado. Durante o treinamento, a soma da quantidade de *tokens* das duas partes não deve superar a quantidade de *tokens* suportada pelo modelo e não ocupar mais memória do que a disponível quando é iterada para o treinamento do modelo. Esse mesmo problema ocorre durante o treinamento do modelo para o Filtro Adversário, no qual as unidades de *plots* e as variações de textos são utilizadas para compor as amostras que farão parte da seleção.

Além de afetar cada etapa singularmente, o uso de textos grandes e modelos que os suportem, todo o *pipeline* sofre de limitações, tais como: o tamanho de *batch* que pode ser utilizado, fator que impacta a qualidade dos modelos produzidos; e o tempo de treinamento de um modelo, reduzindo a quantidade de experimentos que podem ser realizados.

7.2.2 Coerência textual e argumentos

Em nosso contexto, nos referimos a discurso quando lidamos com sentenças estruturadas, e usamos a palavra coerência aqui para nos referir à relação entre sentenças que formam discursos reais diferentes de apenas montagens aleatórias de sentenças. Para considerar um discurso como coerente, consideramos a coerência local, a coerência global e coerência tópica, isto é, frases coerentes com tópicos próximos geralmente são sobre o mesmo tópico e usam o mesmo vocabulário ou vocabulário semelhante para discutir esses tópicos, sem contradição entre as ideias. Ainda que um discurso seja coerente, os argumentos utilizados podem não corresponder com a realidade ou ainda reproduzir ideias consideradas, por determinado viés, inocentes, preconceituosas, criminosas, etc. Contudo, mesmo que nosso objetivo não seja a identificação destes textos, incluímos exemplos de problemas encontrados nas bases utilizadas que devem ser considerados em trabalhos futuros.

Textos do *corpus* GCDC

Na Tabela 7.1, apresentamos exemplos de textos extraídos do *corpus* GCDC, que discutiremos nessa seção, e suas anotações. No texto da primeira linha, um discurso anti-

imigrantes mexicanos é apresentado, incluindo ainda uma fala que incentiva o ataque aos imigrantes ilegais. O autor do texto apresenta como argumento o uso de imigrantes ilegais como força de trabalho, os benefícios que recebem e as condições de trabalho submetidas. Apesar da construção textual não apresentar erros de coerência global ou local, o argumento utilizado não é baseado em dados e distorcido da realidade, conforme dados expostos por Castaneda et al. [25].

Outro problema encontrado está ligado a rotulação dos dados deste *corpus*. Nas linhas 1 e 3, o rótulo final (utilizado para classificação) não representa a opinião da maioria dos anotadores, sendo que em caso de discordância entre os anotadores, com a presença somente de anotações iguais a 1 e 2, os autores preferiram considerar o texto como tendo nível de coerência 2; enquanto para os casos em que há discordância completa, como na linha 5 da Tabela 7.1, os autores escolheram por atribuir o rótulo 1. Reforçamos ainda que, segundo nosso entendimento, não existe uma escala de coerência tal qual apresentada ou um texto com nível médio de coerência.

7.3 Trabalhos Futuros

Ainda que propomos um *corpus*, uma nova versão para o Filtro Adversarial, uma nova forma de inserir informações linguísticas em um modelo Transformer, e um classificador de histórias coerentes, ainda há muitos problemas e melhorias que podem ser explorados no futuro, que incluem:

Adicionar novas fontes de dados ao H.IAAC CommonStories. No momento, nosso *corpus* é composto de apenas duas fontes: as bases de dados ROCStories (ROC) e Writing Prompts (WP), ambas nativamente em inglês. Nosso objetivo é expandir esse *corpus* com a adição de outras bases tanto em inglês quanto em português, tais como: CNN/Daily Mail [113], BookSum [81] e BookCorpus [177].

Filtragem de relações da RST. Assim como no estudo desenvolvido por Chernyavskiy et al. [32], identificamos relações que ocorrem com alta e baixa frequência nas extrações realizadas, que devem ser verificadas, para confirmar se resultado da classificação de relações está enviesada pelo *parser* `DMRST`. Como próximos estudos, propomos validar essa hipótese e, em caso de confirmação, avaliar junto a especialistas do domínio amostras de texto e classificação de relações e prosseguir com o procedimento realizado por Chernyavskiy et al. [32], retirando as relações inconsistentes.

Binarizar relações da RST. Uma abordagem computacional comum para trabalhar com a RST é a criação da árvore binária de relações da RST, a RST Tree [46, 124, 148]. A utilização da RST neste formato pode diminuir a quantidade de *tags* adicionadas ao texto, uma vez que cada EDU contará com uma quantidade fixa e reduzida de *tags*.

Novas manipulações de *plots*. As manipulações atuais visam explorar alguns pontos fracos na geração de textos com modelos de linguagem menores, tais como repetição de

Tabela 7.1: Exemplos de textos extraídos do *corpus* GCDC. Os rótulos originais foram ajustados para o intervalo [0,2] (originalmente [1,3]) mas a escala foi mantida, isto é, o rótulo 0 passa a representar o menor nível de coerência enquanto 2 representa o maior nível de coerência.

Texto	Anotações	Rótulo Final	Fonte
I strongly agree with everyone here except indigo. Illegals should not be allowed here because this is our country. Not only do illegals come him because of freedom, but they come here and take all of our jobs, reap all the benefits, have multiple babies that they can't afford and the government sits back, supports them, and we americans get nothing. Illegals also get paid under table and ship food and clothes back home to mexico. I don't understand why we are not fighting back and putting a stop to this. This is USA, the home of the red, white, and blue. I don't see any signs on our flag that reads "Mexico"anywhere. We should ship them back and if they cross our borders and are illegal, attack them all...	1, 1, 2	2	Yahoo
To tell you in person today what an honor and a pleasure it has been to work with you over these last four years as Feed the Future has evolved. Today's event was a terrific pulling together of threads, like country driven strategies, that we have been working on since literally the first day of the Administration. You and the Secretary are both rock stars. Over this term, you have both done an incredible amount to help hundreds of millions of the world's most vulnerable people, including most especially young children. As you begin to think about your next adventure, I hope that I get a chance to say goodbye and to assure you that there are many career folks at State who want to see this effort continue on into the future With highest regards, Bill Craft.	2, 1, 2	2	Clinton
Best Thai food I have found in the Las Vegas area. The prices are great and the food is well prepared. I especially love the Galanga Curry, chicken satay, coconut soup, crab rangoon, ginger chicken, etc. I could probably keep listing, but the items listed above are my "must-haves."I do not like how stingy they are with the rice and if you ask for more than the single scoop, they charge $1-2 extra. The service is extremely slow, as they only have one server at a time and she is there EVERY SINGLE day. This is a Mom & Pop shop, itś close to my house and easy to get to-go food from. I will keep coming back as long as they are serving food! Which I was concerned about when I first started partonizing, but as of late, there are more people in the restaurant. I tell everyone I know. I want to keep this place in business!	2, 1, 1	2	Yelp
This is a very thoughtful piece that identifies many of the same factors we are in our diplomatic chapter (and that is not just because the author – the former Bangladesh ambassador to the US—cites me). What is most striking is that it affirms that our analysis in the QDDR offers a set of questions that foreign ministries all over the world are grappling with. I had hoped to have time to do some traveling to broadcast the QDDR to key allies and partners around the world after the QDDR is in, but even if that can't happen, we will devise an outreach strategy re sending the report out to foreign ministers around the world and to our ambassadors that will greatly amplify its impact. AM	1, 1, 1	1	Clinton
Because the Palestinians are killed with weapons that US is sending to Israel.US sends billions of dollars to the Israeli government.(we can only imagine how those money are spend).As for the UN,the UN is just a political tool for US...US vetoes whatever doesn't like.So Israel has nothing to worry about.If UN wants to condemn Israel for something,US is going to "protect"it by vetoing the condemnation... The Talibans have become terrorists the moment they turned away from US.Before they were (for US) legitimate resistance fighters ,fighting against the occupation of the Soviet Union and they were fully supported by US...	1, 2, 0	1	Yahoo
My hubbie and I LOVE this venue. It's one of the only places in the valley to see independent films and the location is fantastic, as there are so many great restaurants, not to mention the Fashion Square Mall, nearby. Once a month, this place is a Date Night staple. Only issue I've had over the years is the seating. It's "old school"auditorium, not stadium seating, so–as I'm a "shorty--I often end up having to shift seats to find a clear view of the screen. Sometimes I'm tempted to get a child booster seat, but I'm not sure my butt would fit (kind of embarrassing if it got stuck, huh? LOL)	0, 0, 2	0	Yelp
Sounds like you guys are turning back into regular yankees! I'm sure your poor kids are freezing their butts off. Tell them spring is just around the corner–and down the street a block or two. We won't move until summer, but I'm down at Planet Houston 3 days a week on average. I'm pretty flexible so let me know if you can make it to OMA or HOU and feel free to rack out with me in either place. The OMA joint has more room and more chow in the frig, but I've got a rockin' bachelor pad apt. in downtown hou that ain't bad either. Give me a couple of days notice and I'll get 'Stros tickets at Enron Field. We get paid in baseball tickets now since the stock options are all worthless. (but don't ask Egert or Carmany about that). Good luck on the new gig–you always rag being 3d guy on 727s, but it sounds better than being co-driver on the Airbus (i.e., the "Renault of the skies"). All you have to do is make the friendly announcement about how long the flight will be and then sit around listening to your 8 tracks. Adios dude. df	0, 0, 0	0	Enron
Renting a car in Italy is very cheap. About 40 Euro a day for a comfortable car. It's only about a 1 hour drive from Rome on easy to drive AutoStrada. Drive to Napoli (lots of signs) Once outside ofnapoli look for sign to Pompei. It's real easy, even if you don't know alot of Italian. the trains are a pain. Lots and Lots of smelly people, very cramped, and it's real easy to miss your destination. I lived in Italy (Sicily) for three years. I drove to Pompei, Rome, Milan, Firenze, etc, etc. It's much more of an adventure, plus you can stop and grab breaks to sightsee on the way. There is alot see in the countryside along the way. Have fun no matter how you go.	0, 0, 0	0	Yahoo

tokens e adição de palavras imprevisíveis, em termos de previsão de próximos *tokens*. Temos como objetivo adicionar novas manipulações, como: substituição de verbos ou objetos, alteração ou adição de sujeitos nas histórias, adição e alteração de marcadores temporais aleatórios, adição de resumos não relacionados ao texto, entre outros. Essas novas manipulações afetam diretamente a identificação de relações da RST e inserem mais diversidade aos tipos de textos considerados incoerentes.

Gerar textos a partir de dados de atividades físicas. Um dos objetivos de pesquisa compartilhado entre os pesquisadores do H.IAAC é trabalhar dados de sensores, transformando-as em informações e apresentando-as a usuários de *smartphones* de uma forma que o entendimento seja facilitado, como em forma de textos, imagens e gráficos. Com os modelos desenvolvidos nessa dissertação, é necessário que os dados de sensores passem por classificadores, sejam transformados em rótulos e então sejam construídas unidades de *plots*, que por sua vez podem virar textos com os modelos *Plot2Text*. Contudo, os textos gerados dessa forma podem parecer pouco naturais ou repetitivos, além de ficarem muito grandes se o tempo de coleta for grande e a quantidade de atividades realizadas for alta. Dessa forma, em trabalhos futuros, além de explorar a alteração do formato de entrada, podemos incluir etapas de sumarização ao *pipeline* final.

Gerar textos com LLMs e em português. Apesar de treinarmos modelos para classificação de coerência tanto em inglês quanto em português, os resultados atuais para textos em português foram obtidos com a tradução da base em inglês, sem passar pelas etapas de extração de unidades de *plots*, manipulação de *plots* e geração de texto. Além disso, os textos foram gerados com base em um modelo `XLM-Roberta` e por maior que sejam os esforços em melhorar o desempenho da geração de textos, a qualidade ainda é inferior a de textos produzidos por LLMs. Como próximos passos, planejamos avaliar o impacto do uso desses modelos quando usados para geração de texto a partir de *plots* em nosso *pipeline*.

Incluir RST e POS na arquitetura Transformers. Assim como mostramos no Capítulo 3, outros trabalhos abordam a inclusão de elementos de teorias da linguística e outras estruturas por meio da adição de camadas de *embeddings* ou até mesmo outras redes, como GNNs ou CNNs, e apresentam resultados superiores em tarefas como geração e classificação de texto. Planejamos reproduzir essas formas de inclusão, em especial da RST e POS, e avaliar o desempenho do novo *pipeline* com elas.

Métrica de coerência comparada a avaliação humana. Ainda que seja possível utilizar o modelo de classificação de coerência e usar os *logits* para quantificar a coerência de um texto, o ideal é que uma avaliação humana, em escala, de uma porção da base seja feita e sua correlação com os resultados do modelo e outras métricas computado. Para isso, coletaremos essas avaliações em textos em português e inglês e compararemos o resultado com a versão atual e com futuras modificações.

Comparar resultados com classificação *zero-shot* em LLMs. A classificação de coerência de textos não é uma tarefa que aparece com frequência, visto a revisão da literatura no Capítulo 3, e a comparação dos resultados não é abrangente por esse motivo. Ao executar a classificação *zero-shot* em LLMs, conseguiremos comparar os resultados de nossos *pipelines* com modelos mais poderosos que vem sendo empregados como base em outros trabalhos.

Interpretação de incoerência com RST. As relações que fazem parte da RST são uma ferramenta importante para análise de coerência textual. Ao mapear trocas de relações que ocorrem em textos incoerentes, podemos reportá-las como forma de auxílio na identificação de trechos que tornam o texto incoerente, identificando possíveis pontos de melhoria no *pipeline* e fornecendo ideias de novas reformulações no *pipeline*.

Referências Bibliográficas

[1] H Porter Abbott. *The Cambridge introduction to narrative*. Cambridge University Press, 2021. 19

[2] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 44

[3] Tushar Abhishek, Daksh Rawat, Manish Gupta, and Vasudeva Varma. Transformer models for text coherence assessment. *arXiv preprint arXiv:2109.02176*, 2021. 33, 35, 37, 39, 40, 63

[4] Rilwan A Adewoyin, Ritabrata Dutta, and Yulan He. Rstgen: imbuing fine-grained interpretable control into long-formtext generators. *arXiv preprint arXiv:2205.12590*, 2022. 31, 33, 37

[5] Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. Automatic story generation: Challenges and attempts. *arXiv preprint arXiv:2102.12634*, 2021. 15

[6] Kholoud Aldous, Joni Salminen, Ali Farooq, Soon-gyo Jung, and Bernard Jansen. Using chatgpt in content marketing: enhancing users' social media engagement in cross-platform content creation through generative ai. In *Proceedings of the 35th ACM Conference on Hypertext and Social Media*, pages 376–383, 2024. 26

[7] Arwa I Alhussain and Aqil M Azmi. Automatic story generation: a survey of approaches. *ACM Computing Surveys (CSUR)*, 54(5):1–38, 2021. 15, 19, 54

[8] Malihe Alikhani, Piyush Sharma, Shengjie Li, Radu Soricut, and Matthew Stone. Clue: Cross-modal coherence modeling for caption generation. *arXiv preprint arXiv:2005.00908*, 2020. 32, 34, 37

[9] Suha AlQaruty, Reema Al Qaruty, Khawlah M Al-Tkhayneh, Samer Abdel Hadi, and Ziyad Kamel Ellala. The role of artificial intelligence in the media content industry (chat gpt as a model). In *2024 International Conference on Multimedia Computing, Networking and Applications (MCNA)*, pages 50–56. IEEE, 2024. 26

[10] Nouf Ibrahim Altmami and Mohamed El Bachir Menai. Cast: A cross-article structure theory for multi-article summarization. *IEEE Access*, 8:100194–100211, 2020. 32

[11] Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O Riedl. Automated storytelling via causal, commonsense plot ordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5859–5867, 2021. 20, 53

[12] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer, 2016. 16

[13] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 16, 58

[14] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198, 2020. 14

[15] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021. 27

[16] Karim Benharrak, Tim Zindulka, Florian Lehmann, Hendrik Heuer, and Daniel Buschek. Writer-defined ai personas for on-demand feedback generation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024. 26

[17] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.", 2009. 26

[18] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 27

[19] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, 2019. 39, 55

[20] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. 51

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 14, 26, 27

[22] Pedro H Bueno, Sandra E F Avila, Luiz F M Pereira, and Helena A Maia. Engenharia de Prompts para Extração de Eventos e Unidades de Plot. Technical Report IC-PFG-23-33, Institute of Computing, University of Campinas, February 2024. 54, 55

[23] Lynn Carlson and Daniel Marcu. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, 54(2001):56, 2001. 21, 24

[24] Carlson, Lynn, Marcu, Daniel, and Okurowski, Mary Ellen . Rst discourse treebank, 2002. URL `https://catalog.ldc.upenn.edu/LDC2002T07`. 30, 36

[25] Ernesto Castaneda, Edgar Aguilar, and Natalie Turkington. Migration as a driver of economic growth: Increasing productivity and filling labor gaps. *SSRN Electronic Journal*, 2024. ISSN 1556-5068. doi: 10.2139/ssrn.4740925. URL `http://dx.doi.org/10.2139/ssrn.4740925`. 75

[26] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*, 2020. 27

[27] Yee Seng Chan and Hwee Tou Ng. Maxsim: A maximum similarity metric for machine translation evaluation. In *Proceedings of ACL-08: HLT*, pages 55–62, 2008. 16

[28] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024. 27

[29] Juan Pablo Chavarro, Jonata Tyska Carvalho, Tarlis Tortelli Portela, and Jonathan Cardoso Silva. Faketruebr: Um corpus brasileiro de notícias falsas. In *Escola Regional de Banco de Dados (ERBD)*, pages 108–117. SBC, 2023. 40

[30] Gang Chen, Yang Liu, Huanbo Luan, Meng Zhang, Qun Liu, and Maosong Sun. Learning to generate explainable plots for neural story generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:585–593, 2020. 53

[31] Alexander Chernyavskiy and Dmitry Ilvovsky. Recursive neural text classification using discourse tree structure for argumentation mining and sentiment analysis tasks. In *Foundations of Intelligent Systems: 25th International Symposium, ISMIS 2020, Graz, Austria, September 23–25, 2020, Proceedings*, pages 90–101. Springer, 2020. 30, 33, 37

[32] Alexander Chernyavskiy, Dmitry Ilvovsky, and Boris Galitsky. Correcting texts generated by transformers using discourse features and web mining. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 36–43, 2021. 31, 32, 34, 37, 61, 75

[33] Alebachew Chiche and Betselot Yitagesu. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9 (1):10, 2022. 34

[34] Leshem Choshen and Omri Abend. Automatic metric validation for grammatical error correction. *arXiv preprint arXiv:1804.11225*, 2018. 14

[35] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 26, 27

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arxiv. *arXiv preprint arXiv:1810.04805*, 2019. 14, 26, 30

[37] Belén Díaz-Agudo, Pablo Gervás, and Federico Peinado. A case based reasoning approach to story plot generation. In *European Conference on Case-Based Reasoning*, pages 142–156. Springer, 2004. 53

[38] David R Dowty, Lauri Karttunen, and Arnold M Zwicky. *Natural language parsing: Psychological, computational, and theoretical perspectives*. Cambridge University Press, 1985. 20

[39] Fiona Draxler, Anna Werner, Florian Lehmann, Matthias Hoppe, Albrecht Schmidt, Daniel Buschek, and Robin Welsch. The ai ghostwriter effect: When users do not perceive ownership of ai-generated text but self-declare as authors. *ACM Transactions on Computer-Human Interaction*, 31(2):1–40, 2024. 26

[40] Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. What can we get from 1000 tokens? a case study of multilingual pos tagging for resource-poor languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 886–897, 2014. 16

[41] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, 2018. 31, 35, 57

[42] Angela Fan, Mike Lewis, and Yann Dauphin. Strategies for Structuring Story Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660. Association for Computational Linguistics, 2019. 20

[43] Youmna Farag and Helen Yannakoudakis. Multi-task learning for coherence modeling. *arXiv preprint arXiv:1907.02427*, 2019. 40

[44] Noureen Fatima, Ali Shariq Imran, Zenun Kastrati, Sher Muhammad Daudpota, Abdullah Soomro, and Sarang Shaikh. A systematic literature review on text generation using deep neural network models. *IEEE Access*, 2022. 14, 15

[45] Noureen Fatima, Sher Muhammad Daudpota, Zenun Kastrati, Ali Shariq Imran, Saif Hassan, and Nouh Sabri Elmitwally. Improving news headline text generation quality through frequent pos-tag patterns analysis. *Engineering Applications of Artificial Intelligence*, 125:106718, 2023. 32, 34, 37

[46] Vanessa Wei Feng and Graeme Hirst. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 60–68, 2012. 31, 75

[47] Joaquim Fonseca. Linguística e texto/discurso: teoria, descrição, aplicação. *Open Repository of University of Porto*, 1992. 21

[48] Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020. 32

[49] Sarik Ghazarian, Zixi Liu, SM Akash, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. Plot-guided Adversarial Example Construction for Evaluating Open-domain Story Generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4334–4344, 2021. 9, 20, 21, 36, 52, 53, 54

[50] Varun Godbole, George E. Dahl, Justin Gilmer, Christopher J. Shallue, and Zachary Nado. Deep learning tuning playbook, 2023. URL `http://github.com/google-research/tuning_playbook`. Version 1.0. 63

[51] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. Content planning for neural story generation with aristotelian rescoring. *arXiv preprint arXiv:2009.09870*, 2020. 20

[52] Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. Making science simple: Corpora for the lay summarisation of scientific literature. *arXiv preprint arXiv:2210.09932*, 2022. 30

[53] Amit Goyal, Ellen Riloff, and Hal Daumé Iii. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 77–86, 2010. 53

[54] Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. Long text generation by modeling sentence-level and discourse-level coherence. *arXiv preprint arXiv:2105.08963*, 2021. 33, 34

[55] Grigorii Guz, Peyman Bateni, Darius Muglich, and Giuseppe Carenini. Neural rst-based evaluation of discourse coherence. *arXiv preprint arXiv:2009.14463*, 2020. 30, 33, 37

[56] Ned Halley. *Dictionary of Modern English Grammar: Grammar, Syntax and Style for the 21st Century*. Wordsworth Editions, 2005. 20

[57] M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Routledge, 1976. 20

[58] Harold V Henderson, Friedrich Pukelsheim, and Shayle R Searle. On the history of the kronecker product. *Linear and Multilinear Algebra*, 14(2):113–120, 1983. 63

[59] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 16

[60] John Hewitt. Initializing new word embeddings for pretrained language models. https://nlp.stanford.edu/ johnhew/vocab-expansion.html, 2021. 50

[61] Michael Hoey. Another perspective on coherence and cohesive harmony. *Functional and systemic linguistics: Approaches and uses*, pages 385–414, 1991. 20

[62] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420, 2017. 26, 39

[63] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 63

[64] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023. 63

[65] Xinyu Hua and Lu Wang. Pair: Planning and iterative refinement in pre-trained transformers for long text generation. *arXiv preprint arXiv:2010.02301*, 2020. 33

[66] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. 27

[67] Khalid Hussain, Nimra Mughal, Irfan Ali, Saif Hassan, and Sher Muhammad Daudpota. Urdu news dataset 1m. *Mendeley Data*, 3, 2021. 34

[68] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv:2108.06098*, 2021. 63

[69] Yotam Intrator, Matan Halfon, Roman Goldenberg, Reut Tsarfaty, Matan Eyal, Ehud Rivlin, Yossi Matias, and Natalia Aizenberg. Breaking the language barrier: Can direct inference outperform pre-translation in multilingual llm applications? *arXiv preprint arXiv:2403.04792*, 2024. 68

[70] Mikel Iruskieta and Chloé Braud. Eusdisparser: improving an under-resourced discourse parser with cross-lingual data. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 62–71, 2019. 25

[71] Maliheh Izadi, Roberta Gismondi, and Georgios Gousios. Codefill: Multi-token code completion by jointly learning from structure and naming sequences. In *Proceedings of the 44th International Conference on Software Engineering*, pages 401–412, 2022. 15

[72] Shafiq Joty, Giuseppe Carenini, and Raymond Ng. A novel discriminative framework for sentence-level discourse analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 904–915, 2012. 30

[73] Suvarna G Kanakaraddi and Suvarna S Nandyal. Survey on parts of speech tagger techniques. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–6. IEEE, 2018. 34

[74] Katharina Kann, Ophélie Lacroix, and Anders Søgaard. Weakly supervised pos taggers perform poorly on truly low-resource languages. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8066–8073, 2020. 16

[75] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023. 14

[76] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. Sentilare: Sentiment-aware language representation learning with linguistic knowledge. *arXiv preprint arXiv:1911.02493*, 2019. 33

[77] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.567. URL `https://aclanthology.org/2020.emnlp-main.567/`. 34, 37

[78] Hadas Emma Kedar. Fake news in media art: fake news as a media art practice vs. fake news in politics. *Postdigital Science and Education*, 2(1):132–146, 2020. 41

[79] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. *arXiv preprint arXiv:1612.07600*, 2016. 16

[80] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*, 2017. 27

[81] Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*, 2021. 30, 35, 75

[82] Rohit Kulkarni. A million news headlines. *V6. Cambridge: Harvard Dataverse. Available online: https://dataverse.harvard.edu/dataset.xhtml*, 2018. 34

[83] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 54

[84] Alice Lai and Joel Tetreault. Discourse coherence in the wild: A dataset, evaluation and methods. *arXiv preprint arXiv:1805.04993*, 2018. 39

[85] Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, et al. Evaluating human-language model interaction. *arXiv preprint arXiv:2212.09746*, 2022. 15

[86] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023. 27

[87] Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang, et al. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *arXiv preprint arXiv:2109.08306*, 2021. 27

[88] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2201.05273*, 2022. 29

[89] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. A survey of pretrained language models based text generation. *arXiv preprint arXiv:2201.05273*, 2022. 54

[90] Yanzeng Li, Jiangxia Cao, Xin Cong, Zhenyu Zhang, Bowen Yu, Hongsong Zhu, and Tingwen Liu. Enhancing chinese pre-trained language model via heterogeneous linguistics graph. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1986–1996, 2022. 32

[91] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI open*, 3:111–132, 2022. 27

[92] Wei Liu, Xiyan Fu, and Michael Strube. Modeling structural similarities between documents for coherence assessment with graph convolutional networks. *arXiv preprint arXiv:2306.06472*, 2023. 40

[93] Yichao Liu, Zongru Shao, and Nico Hoffmann. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv preprint arXiv:2112.05561*, 2021. 27

[94] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019. 26, 30

[95] Zhengyuan Liu, Ke Shi, and Nancy Chen. Multilingual neural rst discourse parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6730–6738, 2020. 16, 25

[96] Zhengyuan Liu, Ke Shi, and Nancy Chen. DMRST: A Joint Framework for Document-Level Multilingual RST Discourse Segmentation and Parsing. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 154–164, 2021. 16, 24, 25, 30, 37, 38

[97] Chi-kiu Lo and Dekai Wu. Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 220–229, 2011. 16

[98] Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. Neural text generation: Past, present and beyond. *arXiv preprint arXiv:1803.07133*, 2018. 27

[99] John Lyons. *Natural Language and Universal Grammar: Volume 1: Essays in Linguistic Theory*, volume 1. Cambridge University Press, 1991. 19

[100] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 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, 2011. 31, 34

[101] Aru Maekawa, Tsutomu Hirao, Hidetaka Kamigaito, and Manabu Okumura. Can we obtain significant success in rst discourse parsing by using large language models? *arXiv preprint arXiv:2403.05065*, 2024. 32

[102] William C Mann and Sandra A Thompson. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles, 1987. 23

[103] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988. 8, 15, 16, 21, 22, 24, 29

[104] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL `http://www.aclweb.org/anthology/P/P14/P14-5010`. 26

[105] E Maziero, Thiago Alexandre Salgueiro Pardo, Iria da Cunha, Juan-Manuel Torres-Moreno, and Eric SanJuan. Dizer 2.0-an adaptable on-line discourse parser. In *Proceedings of the III RST Meeting (8th Brazilian Symposium in Information and Human Language Technology)*, pages 50–57, 2011. 25

[106] Neil McIntyre and Mirella Lapata. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1562–1572, 2010. 53

[107] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024. 27

[108] Ruslan Mitkov. Discourse processing. *The handbook of computational linguistics and natural language processing*, pages 599–629, 2010. 19

[109] Michela Montesi. Understanding fake news during the covid-19 health crisis from the perspective of information behaviour: The case of spain. *Journal of Librarianship and Information Science*, 53(3):454–465, 2021. 41

[110] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, 2016. 35

[111] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. *CoRR*, abs/1604.01696, 2016. URL `http://arxiv.org/abs/1604.01696`. 35, 57

[112] Philippe Muller, Chloé Braud, and Mathieu Morey. Tony: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 115–124, 2019. 25

[113] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023, 2016. URL `http://arxiv.org/abs/1602.06023`. 30, 35, 75

[114] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, 2015. 14

[115] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005. 34

[116] David Paper. *Automated Text Generation*, pages 183–202. Apress, Berkeley, CA, 2021. ISBN 978-1-4842-6649-6. doi: 10.1007/978-1-4842-6649-6_8. URL https://doi.org/10.1007/978-1-4842-6649-6_8. 14

[117] Chanjun Park, Yeongwook Yang, Chanhee Lee, and Heuiseok Lim. Comparison of the evaluation metrics for neural grammatical error correction with overcorrection. *IEEE Access*, 8:106264–106272, 2020. 14

[118] R Paulus. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017. 27

[119] Siyao Peng, Yang Janet Liu, and Amir Zeldes. GCDT: A Chinese RST Treebank for Multigenre and Multilingual Discourse Parsing. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 382–391, 2022. 30, 32, 36, 38

[120] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*, 2011. 48

[121] Vinícius Baião Pires, Daniel Guerreiro, et al. Portuguese fake news classification with bert models. In *Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, pages 834–845. SBC, 2024. 41

[122] Andrew Potter. An algorithmic approach to analyzing rhetorical structures. In *Proceedings of the 5th Workshop on Computational Approaches to Discourse (CODI 2024)*, pages 1–11, 2024. 32

[123] Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H Martin, and Dan Jurafsky. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 581–588, 2005. 26

[124] Dongqi Pu, Yifan Wang, and Vera Demberg. Incorporating distributions of discourse structure for long document abstractive summarization. *arXiv preprint arXiv:2305.16784*, 2023. 30, 32, 37, 75

[125] Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL https://nlp.stanford.edu/pubs/qi2018universal.pdf. 26, 30

[126] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. URL https://nlp.stanford.edu/pubs/qi2020stanza.pdf. 34

[127] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 27

[128] Gisela Redeker. Coherence and structure in text and discourse. *Abduction, belief and context in dialogue*, 233(263), 2000. 14, 15

[129] Jan Renkema. *Discourse studies: An introductory textbook*. Benjamins, 1993. 20

[130] James Ryan. Grimes' fairy tales: a 1960s story generator. In *International Conference on Interactive Digital Storytelling*, pages 89–103. Springer, 2017. 15

[131] Brenda Santana, Ricardo Campos, Evelin Amorim, Alípio Jorge, Purificação Silvano, and Sérgio Nunes. A survey on narrative extraction from textual data. *Artificial Intelligence Review*, 56(8):8393–8435, 2023. 54

[132] Gabriel Oliveira dos Santos, Diego AB Moreira, Alef Iury Ferreira, Jhessica Silva, Luiz Pereira, Pedro Bueno, Thiago Sousa, Helena Maia, Nádia Da Silva, Esther Colombini, et al. Capivara: Cost-efficient approach for improving multilingual clip performance on low-resource languages. *arXiv preprint arXiv:2310.13683*, 2023. 63

[133] Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. Long and diverse text generation with planning-based hierarchical variational model. *arXiv preprint arXiv:1908.06605*, 2019. 27

[134] Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. *arXiv preprint arXiv:2304.08979*, 2023. 14

[135] Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities. *Advances in Neural Information Processing Systems*, 35:13158–13173, 2022. 30

[136] Ning Shi, Wei Wang, Boxin Wang, Jinfeng Li, Xiangyu Liu, and Zhouhan Lin. Incorporating external pos tagger for punctuation restoration. *arXiv preprint arXiv:2106.06731*, 2021. 32

[137] Sophia Skoufaki. Rhetorical structure theory and coherence break identification. *Text & Talk*, 40(1):99–124, 2020. 21

[138] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013. 34

[139] Robyn Speer and Catherine Havasi. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686. European Language Resources Association (ELRA), 2012. 55

[140] Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561, 2022. 27, 57

[141] Rajen Subba and Barbara Di Eugenio. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574, 2009. 36

[142] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Aligning plot synopses to videos for story-based retrieval. *International Journal of Multimedia Information Retrieval*, 4:3–16, 2015. 53

[143] Isabelle Thompson. Readability beyond the sentence: Global coherence and ease of comprehension. *Journal of Technical Writing and Communication*, 16(1):131–140, 1986. 14, 15

[144] Svitlana Vakulenko, Maarten de Rijke, Michael Cochez, Vadim Savenkov, and Axel Polleres. Measuring semantic coherence of a conversation. In *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part I 17*, pages 634–651. Springer, 2018. 15

[145] Paulo R Vasconcellos-Silva and Luis David Castiel. Covid-19, fake news, and the sleep of communicative reason producing monsters: the narrative of risks and the risks of narratives. *Cadernos de Saúde Pública*, 36:e00101920, 2020. 41

[146] Akash Verma, Arun Kumar Yadav, Mohit Kumar, and Divakar Yadav. Automatic image caption generation using deep learning. *Multimedia Tools and Applications*, 83(2):5309–5325, 2024. 26

[147] Minh Duc Vu, Han Wang, Zhuang Li, Jieshan Chen, Shengdong Zhao, Zhenchang Xing, and Chunyang Chen. Gptvoicetasker: Llm-powered virtual assistant for smartphone. *arXiv preprint arXiv:2401.14268*, 2024. 26

[148] Shujun Wan, Tino Kutschbach, Anke Lüdeling, and Manfred Stede. Rst-tace a tool for automatic comparison and evaluation of rst trees. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 88–96, 2019. 75

[149] Fei Wang, Liang Ding, Jun Rao, Ye Liu, Li Shen, and Changxing Ding. Can linguistic knowledge improve multimodal alignment in vision-language pretraining? *arXiv preprint arXiv:2308.12898*, 2023. 48

[150] Fei Wang, Liang Ding, Jun Rao, Ye Liu, Li Shen, and Changxing Ding. Can linguistic knowledge improve multimodal alignment in vision-language pretraining? *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(12):1–22, 2024. 32

[151] Qingzhong Wang, Jia Wan, and Antoni B Chan. On diversity in image captioning: Metrics and methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1035–1049, 2020. 16

[152] Yizhong Wang, Sujian Li, and Jingfeng Yang. Toward fast and accurate neural discourse segmentation. *arXiv preprint arXiv:1808.09147*, 2018. 30

[153] Yuxin Wang, Jieru Lin, Zhiwei Yu, Wei Hu, and Börje F Karlsson. Open-world story generation with structured knowledge enhancement: A comprehensive survey. *Neurocomputing*, page 126792, 2023. 27

[154] Gideon Maillette de Buy Wenniger, Thomas van Dongen, Eleri Aedmaa, Herbert Teun Kruitbosch, Edwin A Valentijn, and Lambert Schomaker. Structure-tags improve text classification for scholarly document quality prediction. *arXiv preprint arXiv:2005.00129*, 2020. 15

[155] J.P. Woodard and J.T. Nelson. An information theoretic measure of speech recognition performance. In *Workshop on standardisation for speech I/O technology, Naval Air Development Center, Warminster, PA*, 1982. 57

[156] Wei Xiang and Bang Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019. 54

[157] Wen Xiao, Patrick Huber, and Giuseppe Carenini. Do we really need that many parameters in transformer for extractive summarization? discourse can help! *arXiv preprint arXiv:2012.02144*, 2020. 29, 30, 32, 37

[158] Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*, 2023. 15

[159] Naganand Yadati. Gainer: Graph machine learning with node-specific radius for classification of short texts and documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 609–626, 2024. 40

[160] Jian Yang, Xinyu Hu, Gang Xiao, and Yulong Shen. A survey of knowledge enhanced pre-trained models. *arXiv preprint arXiv:2110.00269*, 2021. 27

[161] Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 32

[162] Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*, 2023. 27

[163] Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2023. 63

[164] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38, 2022. 27

[165] Yu Yu, Chao-Han Huck Yang, Jari Kolehmainen, Prashanth G Shivakumar, Yile Gu, Sungho Ryu Roger Ren, Qi Luo, Aditya Gourav, I-Fan Chen, Yi-Chieh Liu, et al. Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023. 63

[166] Amir Zeldes. The gum corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612, 2017. 36

[167] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019. 58

[168] Ceng Zhang, Junxin Chen, Jiatong Li, Yanhong Peng, and Zebing Mao. Large language models for human-robot interaction: A review. *Biomimetic Intelligence and Robotics*, page 100131, 2023. 15

[169] Longyin Zhang, Fang Kong, and Guodong Zhou. Adversarial learning for discourse rhetorical structure parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3946–3957, 2021. 32

[170] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015. 34

[171] Xin Zhang, Qiyi Wei, Qing Song, and Pengzhou Zhang. An extractive text summarization model based on rhetorical structure theory. In *2023 26th ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter)*, pages 74–78. IEEE, 2023. 30, 32, 37

[172] Ying Zhang, Hidetaka Kamigaito, and Manabu Okumura. A language model-based generative classifier for sentence-level discourse parsing. In *Proceedings of the 2021*

Conference on Empirical Methods in Natural Language Processing, pages 2432–2446, 2021. 32

[173] Jing Zhao and Yuxiang Zhang. Incorporating linguistic constraints into keyphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5224–5233, 2019. 32

[174] Qingjuan Zhao, Jianwei Niu, Xuefeng Liu, Wenbo He, and Shaojie Tang. Generation of coherent multi-sentence texts with a coherence mechanism. *Computer Speech & Language*, 78:101457, 2023. 15

[175] Qinyu Zhao, Ming Xu, Kartik Gupta, Akshay Asthana, Liang Zheng, and Stephen Gould. The first to know: How token distributions reveal hidden knowledge in large vision-language models? *arXiv preprint arXiv:2403.09037*, 2024. 15

[176] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 27

[177] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and readi ng books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 35, 75

[178] Zining Zhu, Chuer Pan, Mohamed Abdalla, and Frank Rudzicz. Examining the rhetorical capacities of neural language models. *arXiv preprint arXiv:2010.00153*, 2020. 32

[179] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 45–50, Valetta, MT, May 2010. University of Malta. URL `http://is.muni.cz/publication/884893/en`. 26