

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

Arthur de Sousa Costa

**Extração de Informação de Textos Literários em Língua Portuguesa usando Modelos de
Linguagem visando integração em Grafo de Conhecimento**

Florianópolis
2026

Arthur de Sousa Costa

**Extração de Informação de Textos Literários em Língua Portuguesa
usando Modelos de Linguagem visando integração em Grafo de
Conhecimento**

Trabalho de Conclusão de Curso submetido ao
Curso de Graduação em Ciência da Compu-
tação do Centro Tecnológico da Universidade
Federal de Santa Catarina como requisito para
obtenção do título de Bacharel em Ciência da
Computação.

Orientador: Prof. Renato Fileto, Dr.

Florianópolis

2026

Arthur de Sousa Costa

**Extração de Informação de Textos Literários em Língua Portuguesa usando Modelos de
Linguagem visando integração em Grafo de Conhecimento**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 2026.

Banca Examinadora:

Prof. Renato Fileto, Dr.
Orientador
Universidade Federal de Santa Catarina

Prof^ª. Carina Friedrich Dorneles, Dr^ª.
Avaliadora
Universidade Federal de Santa Catarina

Prof. Ronaldo dos Santos Mello, Dr.
Avaliador
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Dr. Renato Fileto, pelos conselhos, pela orientação e pelo auxílio prestado ao longo do desenvolvimento deste trabalho.

À Universidade Federal de Santa Catarina, por todo o conhecimento proporcionado e pela infraestrutura disponibilizada durante minha formação acadêmica.

À minha esposa, aos meus pais e aos meus irmãos, por todo o carinho, apoio e incentivo ao longo desta trajetória.

"Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit." (William Pollard)

RESUMO

Atualmente, uma grande quantidade de informações está disponível na forma de textos em documentos digitais, como livros, reportagens de jornais e revistas, páginas da Internet e publicações em redes sociais. O volume crescente desses dados torna inviável o processamento manual e demanda soluções de Processamento de Linguagem Natural (PLN). No domínio literário, a interpretação automatizada ainda representa um desafio relevante, especialmente quando envolve a identificação de personagens, suas características e as relações estabelecidas entre eles ao longo da narrativa. Este trabalho investiga a extração de informação (IE) em obras literárias em português do Brasil, com foco na estruturação de elementos narrativos por meio de Modelos de Linguagem de Grande Escala (LLMs) *open-source*, usando *prompts* que orientam a geração de dados estruturados em formato adequado para a integração em um grafo de conhecimento (KG). A metodologia proposta utiliza LangChain para orquestrar um *pipeline* composto por três cadeias principais de processamento: simplificação de sentenças, resolução de correferência e extração estruturada de personagens, características e relações. Para lidar com obras extensas, o texto é segmentado em sentenças e organizado em *chunks*, preservando metadados e evidências textuais associadas às extrações. Embora seja utilizado um único modelo de linguagem, são conduzidos três experimentos distintos, variando a estratégia de *prompting*: um experimento com *prompt* básico, um experimento com *chain-of-thought prompting* e um experimento com *in-context learning*. Essa configuração permite comparar o impacto de diferentes formas de instrução sobre o mesmo modelo. As informações extraídas são organizadas em formatos estruturados e posteriormente integradas em um KG, no qual personagens são modelados como nós e suas relações como arestas. Os resultados indicam que a abordagem contribui para automatizar a análise de elementos narrativos, oferecendo um recurso para pesquisadores e leitores explorarem personagens, atributos e interações de modo mais sistemático, auditável e reproduzível.

Palavras-chave: Extração de Informação. Modelo de Linguagem de Grande Escala. Processamento de Linguagem Natural. Grafos de Conhecimento. Textos Literários. Extração de Relações.

ABSTRACT

Currently, a large amount of information is available as text in digital documents, such as books, newspaper and magazine articles, web pages, and social media posts. The growing volume of these data makes manual processing impractical and demands Natural Language Processing (PLN) solutions. In the literary domain, automated semantic interpretation remains a relevant challenge, especially when it involves identifying characters, their characteristics, and the relationships established among them throughout the narrative. This work investigates information extraction (IE) in literary works written in Brazilian Portuguese, focusing on the structuring of narrative elements through *open-source* Large Language Models (LLMs), using *prompts* that guide the generation of structured data in a format suitable for integration into a knowledge graph (KG). The proposed methodology uses LangChain to orchestrate a *pipeline* composed of three main processing chains: sentence simplification, coreference resolution, and structured extraction of characters, characteristics, and relationships. To handle long literary works, the text is segmented into sentences and organized into *chunks*, preserving metadata and textual evidence associated with the extractions. Although a single language model is used, three distinct experiments are conducted by varying the *prompting* strategy: one experiment with a basic *prompt*, one experiment with *chain-of-thought prompting*, and one experiment with *in-context learning*. This configuration makes it possible to compare the impact of different instruction strategies on the same model. The extracted information is organized into structured formats and integrated in a KG, in which characters are modeled as nodes and their relationships as edges. The results indicate that the approach contributes to automating the analysis of narrative elements, offering a resource for researchers and readers to explore characters, attributes, and interactions in a more systematic, auditable, and reproducible way.

Keywords: Information Extraction. Large Language Models. Natural Language Processing. Knowledge Graphs. Literary Texts. Relation Extraction.

LISTA DE FIGURAS

Figura 1 – Exemplo de identificação de NER	17
Figura 2 – Exemplo de JSON com RE	18
Figura 3 – Exemplo de grafo com três triplas do mesmo sujeito	21
Figura 4 – Exemplo conceitual de visualização em grafo narrativo	22
Figura 5 – Pipeline metodológico proposto para IE em obras literárias em português do Brasil	29
Figura 6 – Segmentação em sentenças e formação de <i>chunks</i>	33
Figura 7 – Resolução de correferência com base em contexto recente	35
Figura 8 – Exemplo de inspeção tabular dos metadados em CSV	38
Figura 9 – Exemplo de visualização em grafo HTML gerada pelo pipeline	39
Figura 10 – Grafos gerados pelos experimentos para o primeiro exemplo de <i>Dom Casmurro</i>	49
Figura 11 – Grafos gerados pelos experimentos para o segundo exemplo de <i>Senhora</i> . .	51
Figura 12 – Informações extraídas pelos experimentos para o terceiro exemplo de <i>As Vítimas Algozes</i>	52
Figura 13 – Grafo de conhecimento gerado a partir das relações extraídas de <i>O Cortiço</i> com <i>prompt</i> básico	54
Figura 14 – Grafo de conhecimento gerado a partir das relações extraídas de <i>As Vítimas Algozes</i> com <i>chain-of-thought prompting</i>	55
Figura 15 – Grafo de conhecimento gerado a partir das relações extraídas de <i>Memórias Póstumas de Brás Cubas</i> com <i>in-context learning</i>	56

LISTA DE TABELAS

Tabela 1 – Tokenização ilustrativa com descrição	16
Tabela 2 – Quadro comparativo dos trabalhos relacionados	27
Tabela 3 – Obras literárias utilizadas no corpus experimental	30
Tabela 4 – Esquemas estruturados utilizados na extração	31
Tabela 5 – Objetivos e restrições do prompt de simplificação	34
Tabela 6 – Categorias de características utilizadas na extração	37
Tabela 7 – Quadro-resumo dos prompts utilizados no pipeline	37
Tabela 8 – Parâmetros experimentais principais do pipeline	40
Tabela 9 – Configuração do ambiente experimental	43
Tabela 10 – Estatísticas do corpus	44
Tabela 11 – Estatísticas dos trechos selecionados para os experimentos	44
Tabela 12 – Resultados gerais por estratégia experimental nos trechos processados . . .	48
Tabela 13 – Resultados quantitativos por obra e estratégia experimental nos trechos processados	48
Tabela 14 – Características extraídas pelos experimentos no segundo exemplo de <i>Senhora</i>	50
Tabela 15 – Comparação dos grafos de conhecimento apresentados	53

LISTA DE ABREVIATURAS E SIGLAS

AESOP	<i>Approximate Entity Set Overlap</i>
BLPL	Biblioteca Digital de Literatura de Países Lusófonos
CSV	<i>Comma-Separated Values</i>
DBScan	<i>Density-Based Spatial Clustering</i>
IE	<i>Information Extraction</i>
JSON	<i>JavaScript Object Notation</i>
KG	<i>Knowledge Graph</i>
KnowFi	<i>Knowledge Extraction from Long Fictional Texts</i>
LLM	<i>Large Language Models</i>
MuSEE	<i>Multistage Structured Entity Extraction</i>
NER	<i>Named Entity Recognition</i>
NLTK	Natural Language Toolkit
OWL	<i>Web Ontology Language</i>
PLN	Processamento de Linguagem Natural
PNE	<i>Proper Nominal Entities</i>
RDF	<i>Resource Description Framework</i>
RE	<i>Relation Extraction</i>
RLHF	<i>Reinforcement Learning from Human Feedback</i>
SBBD	Simpósio Brasileiro de Banco de Dados
SMOTE	<i>Synthetic Minority Oversampling Technique</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	DESCRIÇÃO DO PROBLEMA	12
1.2	OBJETIVOS	13
1.3	METODOLOGIA	13
1.4	ESTRUTURA DO TRABALHO	14
2	FUNDAMENTOS	15
2.1	PROCESSAMENTO DE LINGUAGEM NATURAL	15
2.2	MODELOS DE LINGUAGEM DE GRANDE ESCALA	19
2.3	REPRESENTAÇÃO ESTRUTURADA E GRAFOS DE CONHECIMENTO	20
3	TRABALHOS RELACIONADOS	24
3.1	TABELA COMPARATIVA	26
4	PROPOSTA E PROCESSO GERAL	28
4.1	VISÃO GERAL DA PROPOSTA	28
4.2	CORPUS E ORGANIZAÇÃO DOS DADOS	30
4.3	ARQUITETURA LÓGICA DO PIPELINE	30
4.4	PRÉ-PROCESSAMENTO TEXTUAL	32
4.5	ETAPA DE SIMPLIFICAÇÃO DE SENTENÇAS	33
4.6	RESOLUÇÃO DE CORREFERÊNCIA COM CONTEXTO RECENTE . .	34
4.7	EXTRAÇÃO ESTRUTURADA DE PERSONAGENS, CARACTERÍSTI- CAS E RELAÇÕES	36
4.8	PROCESSAMENTO PRINCIPAL E GERAÇÃO DE REGISTROS	37
4.9	AGREGAÇÃO E CONSOLIDAÇÃO GLOBAL	38
4.10	EXPORTAÇÃO DOS RESULTADOS E REPRESENTAÇÃO FINAL	39
4.11	REPRODUTIBILIDADE E PARÂMETROS EXPERIMENTAIS	40
4.12	SÍNTESE DO PROCESSO GERAL	40
5	EXPERIMENTOS E RESULTADOS	42
5.1	CONSIDERAÇÕES INICIAIS	42
5.2	IMPLEMENTAÇÃO E AMBIENTE EXPERIMENTAL	42
5.3	CONJUNTO DE DADOS	43
5.4	DESENHO EXPERIMENTAL	45
5.4.1	Experimento 1: Prompt Básico	45
5.4.2	Experimento 2: Chain-of-Thought Prompting	45
5.4.3	Experimento 3: In-Context Learning	46
5.5	CRITÉRIOS DE AVALIAÇÃO	46

5.6	RESULTADOS OBTIDOS	47
5.6.1	Resultados Quantitativos	47
5.6.2	Resultados Qualitativos	49
5.6.3	Grafos de Conhecimento Gerados	52
5.7	DISCUSSÃO	56
6	CONCLUSÕES E TRABALHOS FUTUROS	58
	REFERÊNCIAS	59
	APÊNDICE A – <i>PROMPTS</i> DO EXPERIMENTO 1	60
	APÊNDICE B – <i>PROMPTS</i> DO EXPERIMENTO 2	62
	APÊNDICE C – <i>PROMPTS</i> DO EXPERIMENTO 3	65
	APÊNDICE D – ARTIGO DO TCC	69
	APÊNDICE E – CÓDIGO FONTE	79

1 INTRODUÇÃO

O volume crescente de textos digitais como livros, reportagens, páginas Web e redes sociais, torna inviável a análise exclusivamente manual e exige soluções de Processamento de Linguagem Natural (PLN). No domínio literário, a extração de informação sobre personagens é especialmente valiosa para estudos narratológicos e análises de redes sociais de ficção, pois permite identificar sujeitos, suas características e os vínculos estabelecidos ao longo da obra (SRINIVASAN; POWER, 2022). Este trabalho investiga a extração de informação em obras literárias em português do Brasil (PT-BR), com foco na identificação de personagens, suas características e na classificação de relações (e.g. amizade, parentesco, conflito), contribuindo para leituras mais sistemáticas e reprodutíveis.

Do ponto de vista técnico, propomos uma metodologia baseada em Modelos de Linguagem de Grande Escala (LLMs) *open-source* integrada via LangChain¹, combinando (i) divisão recursiva do texto em *chunks* para lidar com textos extensos, (ii) *prompting* controlado para extração estruturada de personagens, características e relações, e (iii) serialização dos resultados em JSONL para armazenamento incremental e rastreável das extrações. A partir desses registros estruturados, os personagens identificados são consolidados como nós de um grafo de conhecimento, enquanto as relações extraídas são representadas como arestas direcionadas, acompanhadas de rótulos e evidências textuais. Essa integração permite organizar os elementos narrativos em uma estrutura explorável, favorecendo a visualização de interações entre personagens e a identificação de núcleos relacionais. A qualidade dos resultados é avaliada por meio de uma inspeção quantitativa e qualitativa por amostragem.

1.1 DESCRIÇÃO DO PROBLEMA

O problema investigado neste trabalho é a extração automática de informações em textos literários escritos em português do Brasil, com foco na identificação de entidades do tipo personagem, na caracterização dessas entidades por meio de atributos, papéis sociais, estados ou traços descritivos, e na extração de relações estabelecidas entre elas ao longo da narrativa. A fonte primária e gratuita que compreende obras em domínio público é a Biblioteca Digital de Literatura de Países Lusófonos (BLPL)². A motivação decorre do volume e da extensão das obras literárias, que tornam inviável a análise exclusivamente manual e demandam métodos de PLN capazes de operar de forma reprodutível e escalável em documentos longos.

A literatura apresenta desafios específicos para a extração de informação: linguagem figurada, arcaísmos, variação ortográfica, ambiguidade de nomes, uso de apelidos, hipocorísticos ou títulos, além de dependências de longo alcance, como a correferência entre sentenças, parágrafos ou capítulos. Um exemplo ilustrativo aparece em *O Cortiço*, de Aluísio Azevedo, quando o personagem João Romão é apresentado da seguinte forma:

¹ <https://github.com/langchain-ai/langchain>

² <https://www.literaturabrasileira.ufsc.br/>

1. “*João Romão foi, dos treze aos vinte e cinco anos^a, empregado^b de um vendeiro^c que enriqueceu entre as quatro paredes de uma suja e obscura taverna nos refolhos do bairro do Botafogo; (...)*”

Nesse trecho, um sistema de extração de informação deve reconhecer *João Romão* como personagem central da sentença. Além disso, deve identificar que a expressão *dos treze aos vinte e cinco anos*^a representa uma informação temporal associada à trajetória do personagem, indicando o intervalo de idade em que exerceu determinada condição. A palavra *empregado*^b caracteriza seu papel social ou ocupacional naquele período, enquanto a expressão *de um vendeiro*^c estabelece uma relação de trabalho ou subordinação entre João Romão e outro participante humano da narrativa.

Esse exemplo evidencia que a extração de informação em textos literários não se limita ao reconhecimento de nomes próprios. É necessário também associar corretamente atributos e papéis ao personagem mencionado, distinguir descrições de personagens de descrições de lugares ou objetos, e representar relações de maneira estruturada. No caso citado, expressões como *suja e obscura taverna* descrevem o ambiente, e não João Romão, o que ilustra a necessidade de separar características de personagens de características do cenário. Assim, a tarefa exige abordagens de PLN capazes de combinar identificação de personagens, resolução de contexto, classificação de atributos e extração de relações com preservação da evidência textual.

1.2 OBJETIVOS

O objetivo geral deste trabalho é extrair informação de obras literárias em português do Brasil por meio de LLMs *open-source*, visando identificar personagens, seus atributos e as relações entre eles de forma reprodutível e escalável. Para alcançá-lo, é necessário atingir os objetivos específicos abaixo relacionados.

1. Estudar, selecionar e dominar técnicas e ferramentas de PLN necessárias ao problema, com ênfase em LLMs *open-source*, engenharia de *prompts*, *chunking* de longos documentos, serialização em JSONL e integração da informação extraída em grafos de conhecimento.
2. Desenvolver e avaliar um *pipeline* de extração para entidades, atributos e relações em textos literários.
3. Analisar e disponibilizar os resultados para exploração e reuso, incluindo grafos de personagens e relações, além de materiais que garantam reprodutibilidade (código, parâmetros e dados de saída).

1.3 METODOLOGIA

Inicialmente, foi realizado um estudo do estado da arte em PLN aplicado à extração de informação em textos literários, cobrindo conceitos, tarefas e ferramentas pertinentes. A partir

desse levantamento, foi definido um *pipeline* reproduzível para obras em português do Brasil: (i) aquisição e preparo dos dados (BLPL), (ii) divisão recursiva em *chunks* para lidar com livros extensos, (iii) extração orientada a *prompts* de entidades (personagens), suas características e relações, (iv) serialização das informações extraídas em estruturas JSONL, preservando evidências textuais, identificadores de sentença e identificadores de *chunk* e (v) integração das informações extraídas em KG. O *pipeline* foi implementado com LangChain para orquestração, além de utilitários de apoio para PLN e construção de grafos de conhecimento.

Adotamos o modelo qwen/qwen3-32b³ por combinar acesso aberto e baixo custo computacional com boas propriedades para extração de informação via *prompting*. Trata-se de um modelo *instruct-tuned open-source*, o que favorece experimentos mais transparentes e reduz a dependência de modelos proprietários fechados. Sua capacidade de seguir instruções favorece a produção de saídas estruturadas em JSON, requisito para a representação dos resultados e para a geração de grafos de conhecimento. Embora não seja específico para o português, apresenta generalização satisfatória para o idioma, que é reforçada por *prompts* em português. Além disso, sua integração com LangChain por meio da interface ChatGroq viabiliza um *pipeline* eficiente, escalável e transparente.

Por fim, os artefatos de reprodutibilidade estão disponíveis em repositório público⁴, incluindo código-fonte, arquivos de entrada, parâmetros utilizados e arquivos de saída gerados pelos experimentos.

1.4 ESTRUTURA DO TRABALHO

O restante deste trabalho está estruturado como se segue. O Capítulo 2 descreve os fundamentos utilizados no trabalho e necessários ao seu entendimento. O Capítulo 3 discute os trabalhos relacionados e compara as características do trabalho aqui proposto com o estado da arte. O Capítulo 4 descreve a proposta para extrair informações em textos literários, por meio de LLMs *open-source*. O Capítulo 5 relata sobre os experimentos realizados e discute seus resultados. Finalmente, o Capítulo 6 apresenta as conclusões obtidas e descreve temas para trabalhos futuros.

³ <https://huggingface.co/Qwen/Qwen3-32B>

⁴ https://github.com/ArthurSousaCosta/ie_obras_literarias_ptbr

2 FUNDAMENTOS

Este capítulo apresenta os conceitos e bases técnicas que sustentam a proposta deste trabalho. Iniciamos com uma visão de PLN aplicada a textos literários (Seção 2.1), destacando tarefas de extração de informação (entidades, características e relações), particularidades do português do Brasil nesse domínio e procedimentos de segmentação para documentos extensos (divisão em *chunks*).

Na sequência, discutimos LLMs e seu papel na extração orientada por *prompts* (Seção 2.2), com ênfase no modelo *qwen/qwen3-32b* adotado neste estudo, bem como práticas para geração estruturada. Por fim, detalhamos a representação estruturada dos resultados (Seção 2.3), abrangendo o uso de triplas, vocabulários e JSONL para interoperabilidade, além de noções sobre KGs.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural (PLN) fornece o arcabouço conceitual e algorítmico para transformar texto bruto em estruturas analisáveis, desde pré-processamento até tarefas de extração de informação (IE) e avaliação (JURAFSKY; MARTIN, 2024). Em literatura, a IE aparece frequentemente associada à construção de redes de personagens a partir de menções e coocorrências, como em trabalhos clássicos que derivam grafos sociais de romances para permitir análise computacional da narrativa (ELSON; DAMES; MCKEOWN, 2010). Ao mesmo tempo, estudos comparativos mostram que Reconhecimento de Entidades Nomeadas (NER) genérico pode degradar em prosa literária, demandando ajustes metodológicos e validação cuidadosa (DEKKER; KUHN; ERP, 2019). Recentemente, LLMs ampliaram o leque de abordagens ao viabilizar extração estruturada via *prompting* e esquemas JSON, aproximando IE de um paradigma mais generativo (XU et al., 2024).

Atualmente, há diversos kits consolidados de ferramentas para PLN e engenharia com LLMs. Este trabalho apoia-se em: NLTK (*Natural Language Toolkit*)¹, Groq API (carregamento do modelo)², LangChain (orquestração e *prompting*), além de utilitários como *Pandas*³, *NetworkX*⁴ e *PyVis*⁵ (grafos e visualização) e serialização em JSONL. As tarefas consideradas incluem tokenização, segmentação em sentenças, divisão em *chunks* para documentos longos, NER (personagens), extração de relações e agregação inter-*chunks*. As subseções a seguir descrevem esses componentes em maior detalhe.

¹ <https://www.nltk.org/>

² <https://console.groq.com/home>

³ <https://pandas.pydata.org/>

⁴ <https://networkx.org/>

⁵ <https://pyvis.readthedocs.io/en/latest/>

Tokenização

Tokenização é o processo de segmentar o texto em unidades mínimas (tokens), como palavras, números e pontuação. Em PLN, ela é a primeira etapa que alimenta praticamente todas as demais tarefas e, portanto, suas escolhas impactam diretamente a qualidade da extração (JURAFSKY; MARTIN, 2024). Em textos literários em português, a tokenização deve lidar com contrações, hifenização, títulos e abreviações, nomes com apóstrofo e sinais de diálogo.

Exemplo: “— *D. Glória viu Capitu na casa-de-campo, às margens d’Ávila.*”

Tabela 1 – Tokenização ilustrativa com descrição

Token	Descrição
-	Pontuação (travessão)
D.	Título abreviado de "Dona", importante para NER de <i>D. Glória</i>
Glória	Nome próprio, parte do nome canônico “D. Glória”
viu	Verbo, útil para detectar relações
Capitu	Nome próprio, alvo principal de NER
na	Contração <i>em + a</i>
casa-de-campo	Substantivo composto (hifenizado)
,	Pontuação (vírgula)
às	Preposição contraída <i>a + as</i>
margens	Substantivo comum
d’Ávila	Nome próprio com apóstrofo
.	Pontuação final, limite de sentença

Fonte: o autor.

Na Tabela 1, observa-se uma tokenização ilustrativa da sentença apresentada, destacando como elementos como travessão, títulos abreviados, nomes próprios, contrações e pontuação são tratados como unidades relevantes para o processamento textual.

Segmentação em sentenças

Segmentação em sentenças é a tarefa de dividir o texto contínuo em unidades maiores que os tokens, delimitadas por sinais de pontuação e/ou padrões sintáticos. Ela organiza o fluxo narrativo em “janelas” semânticas que servem de base para contagem, coocorrência, extração de relações e avaliação. Em português literário, a segmentação precisa lidar com abreviações e títulos (e.g. *Sr., Sra., D.*), diálogos marcados por travessão, citações entre aspas, reticências e pontuação expressiva. Ferramentas como o NLTK oferecem segmentadores baseados em regras e modelos estatísticos.

Divisão em *chunks*

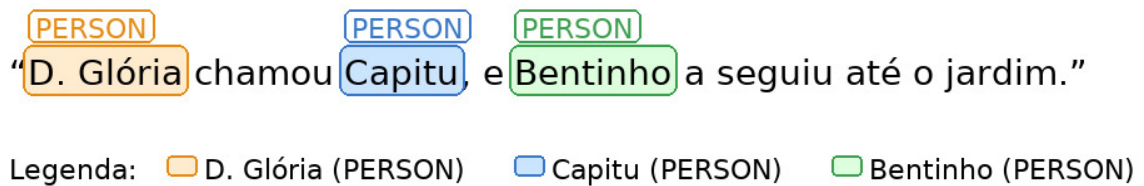
A divisão em *chunks* consiste em particionar textos longos em blocos menores e manejáveis para respeitar limites de contexto dos modelos e permitir processamento incremental, sem perder a continuidade semântica. Na prática, prioriza-se cortar nos limites de sentença ou

parágrafo, adotando um tamanho-alvo (e.g. 500–1 500 tokens) e mantendo o contexto de *chunks* anteriores, para que menções e relações não se percam. Esse contexto melhora a *recall* de entidades e relações. *Chunks* muito grandes podem degradar coerência e respostas do modelo, enquanto *chunks* muito pequenos fragmentam evidências. Na implementação, salvamos metadados para rastrear a origem de cada extração e depois agregamos os resultados inter-*chunks*, consolidando relações para reconstruir a visão global da obra.

Reconhecimento de entidades nomeadas (NER)

NER é a tarefa de identificar no texto as *menções* a entidades de tipos pré-definidos. Neste trabalho, focamos principalmente em identificar pessoas, mas também é possível extrair locais e organizações quando relevantes. As menções a personagens podem aparecer como nome próprio (*Capitu*), título e nome (*D. Glória*), alcunha (*Bentinho* para *Bento*), ou mesmo descrições nominais (*a menina de olhos de ressaca*). O NER fornece a lista dessas menções por sentença e *chunk*, para consolidar cada personagem em um único nó no grafo narrativo.

Figura 1 – Exemplo de identificação de NER



Fonte: o autor.

A Figura 1 ilustra um exemplo de reconhecimento de entidades nomeadas, destacando como menções a personagens podem ser identificadas no texto literário e posteriormente utilizadas nas etapas de extração de características e relações.

Extração de relações (RE)

A tarefa de extrair relações (RE) é a que identifica e rotula vínculos semânticos entre entidades detectadas pelo NER, normalmente representados como triplas (*sujeito, predicado, objeto*). Esses vínculos podem ser explícitos (e.g. “*Dona Glória é mãe de Bentinho*”) ou implícitos e variam de relações sociais a papéis narrativos. Para textos longos, é comum extrair por *chunks* e, depois, consolidar globalmente, mantendo evidências como trecho, índice do *chunk* e sentença.

Na Figura 2, a relação identifica (*D. Firmina Mascarenhas, acompanhava, Aurélia Camargo*), inferida a partir da evidência “*D. Firmina Mascarenhas sempre acompanhava Aurélia na sociedade.*”, encontrada no *chunk* *senhora_chunk_001*, sentença *senhora_sent_0010*.

Figura 2 – Exemplo de JSON com RE

```

{
  "metadata": { ...
  },
  "characters": [
    {
      "canonical_name": "Aurélia Camargo",
      "first_mention_text": "Aurélia Camargo",
      "first_evidence_text": "Ninguém se recorda da Aurélia Camargo",
      "first_sentence_id": "senhora_sent_0006",
      "first_chunk_id": "senhora_chunk_001"
    },
    {
      "canonical_name": "Firmina",
      "first_mention_text": "D. Firmina Mascarenhas",
      "first_evidence_text": "Parenta se chamava D. Firmina Mascarenhas",
      "first_sentence_id": "senhora_sent_0010",
      "first_chunk_id": "senhora_chunk_001"
    }
  ],
  "characteristics": [ ...
  ],
  "relationships": [
    { ...
    },
    {
      "source_character": "D. Firmina Mascarenhas",
      "relation": "acompanhava",
      "target_character": "Aurélia Camargo",
      "evidence_text": "D. Firmina Mascarenhas sempre acompanhava Aurélia na sociedade.",
      "sentence_id": "senhora_sent_0010",
      "chunk_id": "senhora_chunk_001"
    }
  ]
}

```

Fonte: o autor.

Agregação inter-chunks

Agregação inter-chunks é o processo de consolidar, após a extração local, as entidades e relações produzidas em diferentes blocos do livro para reconstruir uma visão global e coerente. Na prática, agrupamos menções por nome canônico, fundimos aliases e traços, e unificamos relações iguais, acumulando evidências e combinando confiança.

Particularidades do português na literatura

A prosa literária em português do Brasil traz fenômenos que afetam diretamente tokenização, NER e extração de relações, tais como contrações e clíticos, títulos e tratamentos

abreviados, hipocorísticos e epítetos, nomes compostos, hifenização e apóstrofes, variação ortográfica e arcaísmos, entre outros. Em romances, referências indiretas, metáforas e anáforas de longo alcance elevam a ambiguidade de menções e degradam modelos genéricos de NER, exigindo regras e validações adicionais (DEKKER; KUHN; ERP, 2019). Na extração, *prompts* em português com exemplos literários reduzem alucinações e induzem JSON estável (*few-shot*).

2.2 MODELOS DE LINGUAGEM DE GRANDE ESCALA

Modelos de Linguagem de Grande Escala (LLMs) são redes neurais baseadas em Transformers⁶ treinadas em grandes coleções de texto para prever tokens subsequentes, aprendendo regularidades linguísticas e conhecimento distribuído. As próximas subseções revisam arquitetura e treinamento, modelos com ênfase em propriedades que impactam extração e *prompting* para IE, apresentando padrões para induzir saídas estruturadas (esquemas JSON) e cuidados para reduzir alucinações.

Arquitetura e treinamento

A arquitetura *Transformer* pode ser vista como um “leitor” que olha para todas as palavras ao mesmo tempo e decide quais são mais importantes para prever a próxima. Isso é feito pelo mecanismo de autoatenção (*self-attention*), que calcula pesos de relevância entre tokens. Como os modelos não trabalham com palavras inteiras, o texto é quebrado em *subtokens*, e cada posição recebe um sinal posicional para marcar a ordem. O pré-treinamento é auto-regressivo: o modelo aprende a prever o próximo token em grandes coleções de texto. Depois, um ajuste por instrução (*instruction tuning*) ensina o modelo a seguir pedidos do usuário e técnicas como RLHF (preferência humana) refinam o comportamento.

Modelos

Um *Large Language Model* (LLM) é uma rede neural treinada para prever o próximo token de um texto, após ler enormes quantidades de exemplos. Ao repetir essa tarefa bilhões de vezes, o modelo aprende padrões de vocabulário, gramática, estilo e associações semânticas, conseguindo continuar frases, responder perguntas e seguir instruções. Na prática, você fornece uma entrada (*prompt*) e o LLM gera a continuação mais provável, token a token. Essa capacidade, combinada com ajustes por instrução (*instruction tuning*), permite solicitar saídas em formatos específicos (como JSON) e aplicar o modelo a tarefas de extração de informação, resumo, reescrita e raciocínio leve. Para extração de informação, alguns fatores simples ajudam a escolher um modelo:

⁶ <https://huggingface.co/docs/transformers/index>

1. **Janela de contexto:** define quantos tokens o modelo “enxerga” de uma vez; textos longos pedem *chunking*.
2. **Versão *instruct*:** modelos ajustados por instrução tendem a *seguir formato* (e.g. JSON) com menos esforço.
3. **Estabilidade de geração:** usar temperatura baixa e *stop tokens* reduz respostas “criativas” demais e facilita validação.
4. **Custo e disponibilidade:** modelos médios equilibram qualidade e viabilidade.

Um modelo *instruct* de porte médio, combinado com *prompts* claros, *chunks* bem definidos e pós-processamento, pode ser uma ferramenta muito boa para se obter entidades e relações úteis em obras literárias.

Prompting para IE

Um *prompt* é uma instrução ou comando em linguagem natural, que é fornecido a um sistema de inteligência artificial para que uma tarefa seja executada, como escrever um texto, criar uma imagem ou gerar um pedaço de código. O *prompt* funciona como uma ponte entre o usuário e a máquina, guiando-a para produzir a resposta desejada. A qualidade da resposta depende diretamente da clareza e especificidade do *prompt*.

Neste trabalho, o objetivo principal do nosso *prompt* é induzir saída estruturada e estável: (i) atribuição de papel (*role prompting*); (ii) definição da tarefa a ser realizada de forma objetiva; (iii) uso de contexto recente e regras de resposta; (iv) especificação do formato de saída (campos obrigatórios e tipos). Em textos longos, combinamos *chunks* a fim de agregar entidades e relações.

2.3 REPRESENTAÇÃO ESTRUTURADA E GRAFOS DE CONHECIMENTO

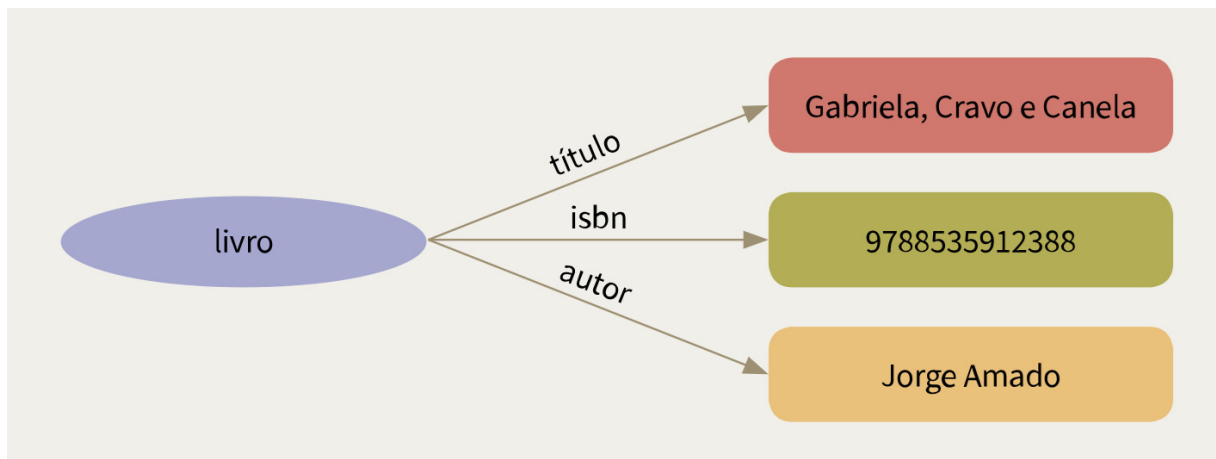
Representação estruturada é a prática de organizar informações em formatos formais, com campos e relações bem definidos, de modo que possam ser validados, combinados e consultados por máquinas. Grafos de conhecimento (KGs) são uma forma especialmente útil dessa representação, eles modelam o mundo como **nós** (entidades) e **arestas** (relações), rotuladas por um vocabulário. Em termos básicos, cada afirmação é uma **tripla** (*sujeito, predicado, objeto*). Essa estrutura facilita integração de fontes, inferências simples (e.g. propagação de relações) e consultas flexíveis (buscas por vizinhança, padrões e caminhos).

Neste projeto, extraímos de cada *chunk* literário um conjunto de entidades (personagens) e relações (amizade, parentesco, interação, etc.) e serializamos o resultado em JSONL. Cada relação armazena evidências e confiança, permitindo reconstruir o raciocínio e repetir a avaliação. Para análise exploratória, materializamos um grafo por meio da biblioteca NetworkX, produzindo visualizações de redes de personagens. Nas próximas subseções, detalhamos triplas e ontologias, JSONL e análise de grafos.

Triplas e ontologias

Em grafos de conhecimento, fatos sobre o mundo são representados como triplas (*sujeito, predicado, objeto*), onde o sujeito e o objeto são entidades (nós) e o predicado é um relacionamento (aresta). Esta formalização simples, porém expressiva, facilita integração, consulta e inferência sobre dados textuais estruturados a partir de IE. Para garantir consistência semântica, emprega-se uma ontologia (ou vocabulário), isto é, um conjunto de classes (e.g. *Person*, *Place*) e propriedades. A ontologia atua como um contrato que orienta a extração, a validação e a interoperabilidade entre sistemas. No fluxo deste trabalho, as triplas derivam de saídas estruturadas (JSON) produzidas por LLMs e são serializadas em formatos compatíveis para posterior agregação e análise de redes. Essa abordagem conecta a IE generativa ao ecossistema de *knowledge graphs* (KGs), permitindo tanto inspeção humana quanto operações algébricas sobre o grafo (XU et al., 2024).

Figura 3 – Exemplo de grafo com três triplas do mesmo sujeito



Fonte: Centro de Estudos sobre Tecnologias Web (<https://ceweb.br/>)

A Figura 3 exemplifica a representação de fatos em formato de grafo, seguindo a estrutura de triplas.

JSONL

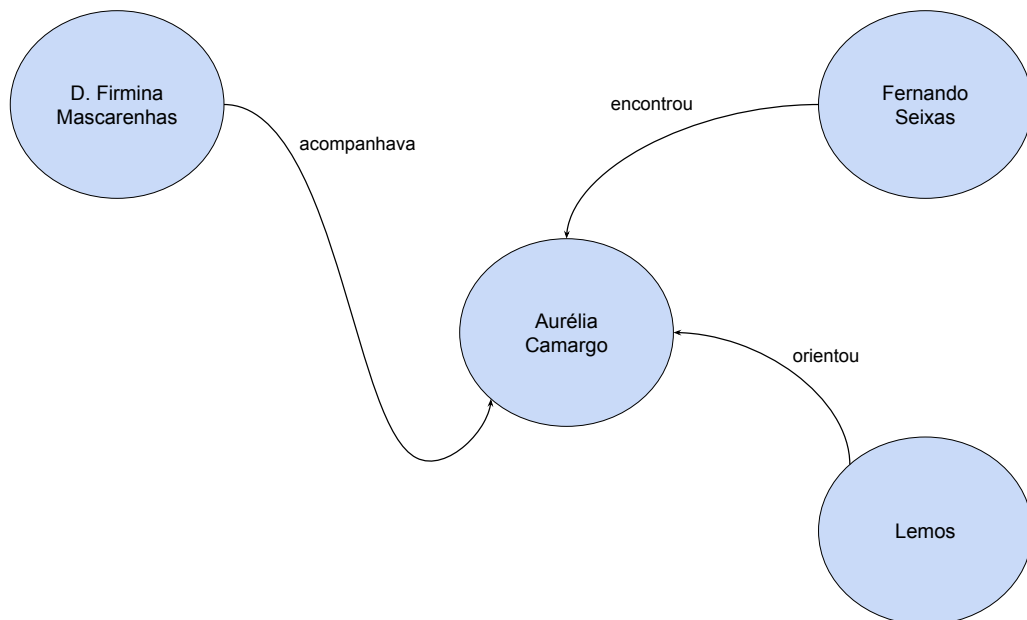
JSONL (*JSON Lines*) é um formato de serialização no qual cada linha de um arquivo corresponde a um objeto JSON independente. Diferentemente de um arquivo JSON tradicional, que normalmente armazena todos os registros em uma única estrutura hierárquica, o JSONL organiza os dados como uma sequência de objetos separados por linhas. Essa característica facilita o processamento incremental, a leitura parcial de arquivos grandes e a adição de novos registros sem a necessidade de reescrever toda a estrutura do arquivo.

No escopo deste trabalho, o formato JSONL é utilizado para armazenar os registros produzidos pelo *pipeline*, incluindo personagens identificados, características extraídas, relações entre personagens e metadados de proveniência, como obra, sentença, *chunk* e evidência textual. Essa organização favorece a rastreabilidade das extrações, pois cada registro pode ser associado ao trecho de texto que o originou. Além disso, JSONL é um formato adequado para extrair informação a ser integrada em KG, compatível com RDF e OWL, facilitando a posterior manipulação dos resultados com ferramentas de análise de dados, a conversão para formatos tabulares e a construção dos grafos de conhecimento utilizados na visualização das relações narrativas.

Grafos de Conhecimento

Grafos de conhecimento (KGs) são estruturas que representam informações por meio de entidades e relações, permitindo organizar dados de maneira conectada, consultável e semanticamente interpretável. Segundo a abordagem apresentada em *Knowledge Graphs*⁷, um KG pode ser compreendido como uma abstração baseada em grafos para integrar, enriquecer e explorar conhecimento proveniente de diferentes fontes. Nesse tipo de representação, os nós correspondem a entidades ou conceitos, enquanto as arestas representam relações rotuladas entre esses nós.

Figura 4 – Exemplo conceitual de visualização em grafo narrativo



Fonte: o autor.

⁷ <https://kgbook.org/>

No contexto deste trabalho, os grafos de conhecimento são utilizados para representar elementos narrativos extraídos de obras literárias. Os personagens identificados são modelados como nós, enquanto as relações extraídas entre personagens são representadas como arestas direcionadas e rotuladas. Assim, uma relação como (*D. Firmina Mascarenhas, acompanhava, Aurélia Camargo*) pode ser representada como uma tripla, permitindo que a informação textual seja convertida em uma estrutura visual e navegável, como é possível observar na Figura 4.

Essa representação é particularmente útil para textos literários porque permite observar padrões de interação entre personagens, identificar núcleos relacionais da narrativa e visualizar a centralidade de determinados personagens dentro dos trechos processados. Em vez de analisar apenas listas de personagens ou relações isoladas, o grafo permite observar a rede formada pelas extrações, favorecendo uma leitura exploratória da estrutura narrativa.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão exploratória da literatura sobre extração de informação aplicada a textos literários e temas correlatos. O objetivo não foi realizar uma revisão sistemática exaustiva, mas identificar trabalhos representativos que auxiliassem na contextualização da proposta, especialmente em relação a extração de personagens, reconhecimento de entidades nomeadas, extração de relações, uso de LLMs, segmentação de textos longos e representação estruturada dos resultados. A busca foi realizada entre janeiro de 2023 e novembro de 2025, contemplando bases como *ACL Anthology*¹, *arXiv*, *IEEE Xplore Digital Library*² e *Google Scholar*³. As expressões de busca (em inglês e português) combinaram termos como: “information extraction”, “named entity recognition”, “relation extraction”, “literary”, “fiction”, “character network”, “knowledge graph”, “prompting”, “large language models”, “Portuguese”, “Brazilian Portuguese”, além de variantes. A busca teve como prioridade trabalhos mais recentes, a fim de acompanhar o avanço tecnológico exponencial. As publicações encontradas foram selecionadas de acordo com os seguintes critérios de inclusão: trabalhos completos (periódicos, conferências ou *preprints* relevantes), em inglês ou português, com foco em extração de informação (IE), reconhecimento de entidades nomeadas (NER), extração de relacionamentos (RE) e grafos aplicados a textos literários ou passíveis de adaptação, publicados a partir de 2021. Após triagem por título, resumo e leitura seletiva, foram priorizados quatro estudos centrais discutidos a seguir.

O trabalho *KnowFi* de Chu, Razniewski e Weikum (2021) propõe uma estrutura de extração de conhecimento voltada a textos ficcionais longos, com ênfase em lidar com dependências de longo alcance típicas de romances. A abordagem organiza o processamento por segmentos (capítulos e trechos) e depois realiza agregação global, combinando componentes clássicos de IE e NER, resolução de correferência e extração de relações com regras para aumentar cobertura sem exigir anotação intensiva. As entidades são consolidadas por normalização e agrupamento de menções, incluindo variantes nominais, e as relações são acumuladas com evidências textuais e filtros de consistência, reduzindo falsos positivos ocasionados por contexto local. A avaliação indica melhorias de precisão quando a consolidação inter-capítulos e critérios de evidência múltipla são aplicados, ainda que persistam desafios em metáforas, alusões implícitas e desambiguação de nomes pouco frequentes. Como conclusão, o *KnowFi* demonstra que estratégias de segmentação somadas com estratégias de agregação são cruciais para transformar narrativas extensas em grafos utilizáveis, oferecendo princípios que auxiliam *pipelines* modernos de IE em literatura.

Srinivasan e Power (2022) investigam duas tarefas encadeadas em narrativas: (i) extração de personagens a partir de obras sumarizadas e (ii) identificação de tipos de personagem (e.g. protagonista, antagonista, coadjuvante). Foram obtidas obras sumarizadas de 20 histórias

¹ <https://aclanthology.org/>

² <https://ieeexplore.ieee.org/Xplore/home.jsp>

³ <https://scholar.google.com.br/>

fictícias, com um total de 218 personagens. Para tal, foram utilizados *scripts* de *webscraping*, conseguindo armazenar detalhes extraídos em arquivos CSV. O tamanho das obras sumariizadas varia entre 3000 a 8000 caracteres, enquanto o número de sentenças varia entre 25 a 74. Para extração de personagens das obras, foi feita uma abordagem de cinco passos: (i) pré-processamento de dados, (ii) identificação de PNEs detectadas no texto, (iii) confirmação de gênero por meio de regras heurísticas, (iv) resolução de alias (combinação de nomes e sobrenomes, variantes abreviadas) e (v) eliminação de entidades não-personagens por meio de um conjunto final de regras heurísticas. A classificação de tipos de personagem é explorada tanto por aprendizado supervisionado com avaliação por *precision*, *recall* e F1, quanto por algoritmos não-supervisionados (e.g. *K-Means*, *Agglomerative Clustering*, DBScan), incluindo experimentos com SMOTE para mitigar desbalanceamento. Os resultados mostram ganhos na identificação de personagens e padrões coerentes de agrupamento por papéis, embora persistam dificuldades em casos ambíguos e com menções raras. O estudo conclui que regras de alias, correferência e atributos derivados potenciam a robustez da extração em resumos de histórias, e que a tipificação beneficia-se de sinais distribuídos ao longo do enredo, ainda que dependente da qualidade da consolidação inicial.

Pires et al. (2023) apresentam a família **Sabiá**, uma linha de LLMs treinados especificamente para o português do Brasil, cobrindo pré-treinamento auto-regressivo e citando *instruction tuning* como uma possível técnica complementar para trabalhos futuros. A metodologia envolve (i) curadoria de dados em larga escala com fontes abertas em língua portuguesa (e.g., coleções web, enciclopédias, notícias), aplicando filtragem e deduplicação, (ii) tokenização treinada para o idioma, reduzindo fragmentação morfológica e melhorando a cobertura lexical, (iii) pré-treinamento com objetivo de modelagem de linguagem para aprender regularidades sintático-semânticas do português e (iv) ajuste supervisionado mínimo para seguir formatos de saída e tarefas de alto nível. O estudo reporta que o modelo 7B foi treinado com 1 trilhão de tokens, enquanto o modelo 65B foi treinado com 1,4 trilhões de tokens. Foram apresentados ganhos consistentes em tarefas em português quando comparados a modelos multilíngues de porte similar, destacando vantagens de vocabulário e cobertura cultural do idioma. Além disso, são disponibilizados artefatos para reprodutibilidade e adoção prática pela comunidade.

Wu et al. (2024) reformulam a extração de informação para um paradigma **entidade-cêntrico**, no qual o objetivo primário é produzir um conjunto estruturado de entidades com atributos a partir de texto, em vez de focar exclusivamente em triplas. Para avaliar esse cenário, os autores introduzem a métrica *Approximate Entity Set Overlap* (AESOP), que mede a sobreposição aproximada entre o conjunto de entidades extraídas e um conjunto de referência, permitindo tolerâncias a variações de conteúdo. Em termos de método, apresentam o *Multistage Structured Entity Extraction* (MuSEE), um *pipeline* de múltiplas etapas que decompõe a tarefa em estágios sucessivos (tipicamente um rascunho inicial gerado pelo modelo, seguido de verificação, correção e consolidação), o que melhora tanto a eficácia (qualidade de campos e consistência estrutural) quanto a eficiência. O estudo compara abordagens e reporta ganhos por métricas quantitativas e avaliação humana lado a lado, além de disponibilizar código e da-

dos para reprodutibilidade. Para este trabalho, a contribuição é especialmente relevante porque alinha *prompting* estruturado e JSON/JSONL com um processo de validação incremental e avaliação apropriada ao foco em entidades (em lugar de apenas relações), facilitando a posterior agregação inter-*chunks* e a integração a grafos de conhecimento.

3.1 TABELA COMPARATIVA

A Tabela 2 apresenta um resumo comparativo dos trabalhos selecionados na revisão exploratória, organizados em ordem cronológica nas linhas da tabela. As colunas refletem os aspectos que consideramos mais relevantes para análise de extração de informação em textos literários. A primeira coluna (**Trabalho**) contém o nome dos autores do trabalho avaliado e o ano de publicação. A segunda coluna (**Corpus**) refere-se ao tipo e origem dos textos processados. A terceira coluna (**Objetivos**) apresenta as metas principais de cada proposta. A quarta coluna (**Abordagem**) refere-se às técnicas e componentes empregados, como NER, correferência, RE, *prompting* e agregação. Na quinta coluna (**Saída**), é descrita a forma de representação dos resultados (e.g. listas de entidades, relações, grafos ou JSON). Por fim, a última coluna (**Idioma**) refere-se ao foco linguístico do estudo.

Observa-se que os trabalhos analisados diferem em escopo, idioma e forma de representação. Chu, Razniewski e Weikum (2021) prioriza um *pipeline* clássico, com NER, correferência e RE, utilizando segmentação para ficção longa em inglês. Srinivasan e Power (2022) trabalha sobre resumos, com pré-processamento, regras de alias e tipificação de personagens. Pires et al. (2023) foca na construção e avaliação de LLMs para PT-BR, sem focar em IE literária. Por fim, Wu et al. (2024) introduz IE entidade-cêntrica com *prompting* estruturado e métrica específica, porém em textos gerais e majoritariamente em inglês. Em contraste, nossa proposta mira obras literárias em português do Brasil e integra LLM *instruct* com LangChain e agregação via *chunking*, por fim realizando uma serialização em JSONL e visualização via *knowledge graph*. Assim, além de tratar o desafio de longos textos de romances em português, buscamos produzir uma representação apta a alimentar grafos e análises subsequentes.

Tabela 2 – Quadro comparativo dos trabalhos relacionados

Trabalho	Corpus	Objetivos	Abordagem	Saída	Idioma
Chu, Raz-niewski e Weikum (2021)	Romances longos	Extração de conhecimento em ficção longa	Segmentação por trechos e agregação global; NER; correferência; RE	Entidades e relações com evidências; grafos narrativos	EN (ficção)
Srinivasan e Power (2022)	Resumos de histórias	Extração de personagens e papéis	Pré-processamento; PNE; confirmação de gênero; regras de alias; eliminação de não-personagens	Lista de personagens; tipos; grafo preliminar	EN (resumos)
Pires et al. (2023)	Português em larga escala	Avaliar LLMs para PT-BR	Pré-treinamento; tokenizador PT	Métricas de avaliação de LLMs	PT-BR
Wu et al. (2024)	Textos gerais	IE entidade cêntrica; avaliação robusta	<i>Prompting</i> estruturado; <i>pipeline</i> MuSEE; métrica AESOP	Conjunto de entidades estruturadas JSON; métricas	EN
Nossa proposta	Obras literárias em PT-BR	IE de personagens, atributos e relações	LLM <i>instruct</i>; LangChain; <i>chunking</i>; agregação <i>inter-chunks</i>	JSONL; JSON CSV; grafos (KG)	PT-BR

Fonte: o autor.

4 PROPOSTA E PROCESSO GERAL

Este capítulo descreve a proposta metodológica e o processo geral adotados neste trabalho para a extração automática de personagens, características e relações em textos literários em português do Brasil. A proposta foi concebida para atender simultaneamente a quatro requisitos centrais do problema investigado: (i) operar sobre obras longas, nas quais a análise integral em uma única passagem é inviável; (ii) preservar evidências textuais e metadados de proveniência, de modo a favorecer auditoria e reprodutibilidade; (iii) restringir a extração ao recorte específico de personagens, seus atributos e suas interações, reduzindo ruído de entidades irrelevantes; e (iv) produzir saídas estruturadas que permitam posterior consolidação, análise tabular e materialização em grafos de conhecimento.

A arquitetura adotada parte do princípio de que textos literários longos demandam processamento incremental. Em vez de submeter a obra inteira ao modelo de linguagem, o texto é primeiramente segmentado em sentenças e, em seguida, particionado em blocos menores (*chunks*) compatíveis com o processamento iterativo. Sobre cada sentença, o *pipeline* executa três etapas encadeadas: simplificação sintático-semântica, resolução de correferência com base em contexto recente e extração estruturada propriamente dita. A saída de cada sentença é armazenada com identificadores, contexto utilizado, evidências textuais e estrutura validada, permitindo posterior agregação intraobra. Ao final, os resultados são consolidados em uma representação unificada de personagens, características e relações, exportados em formatos intermediários para inspeção e, adicionalmente, projetados em um grafo direcionado de personagens.

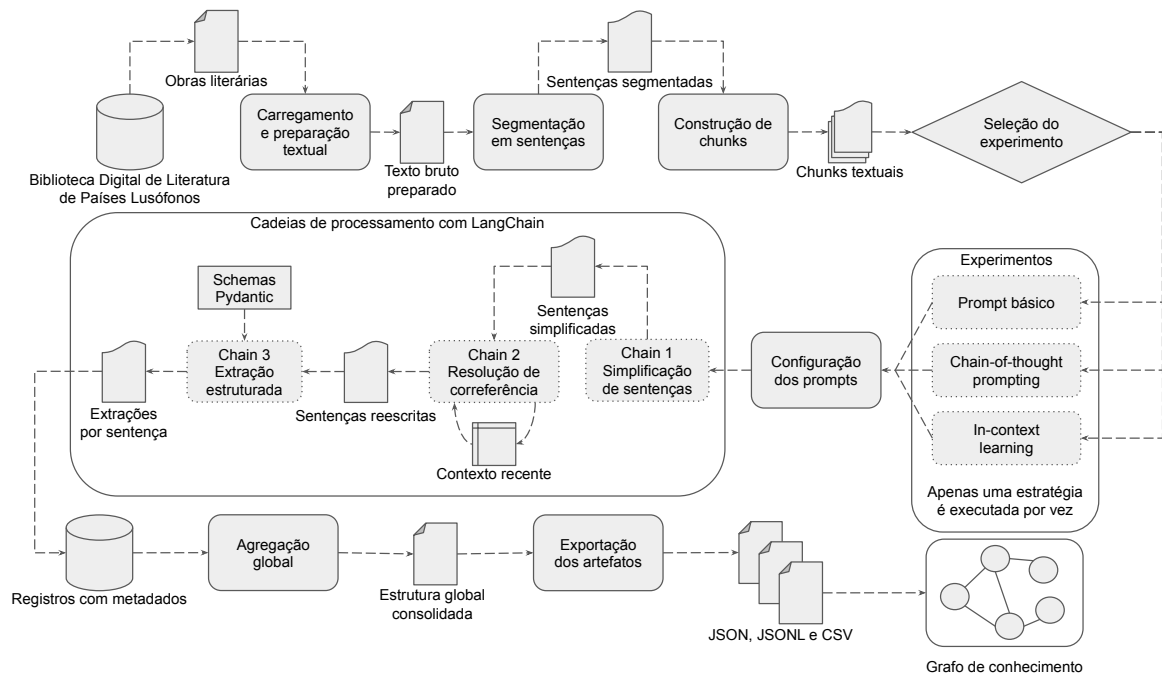
4.1 VISÃO GERAL DA PROPOSTA

A proposta deste trabalho pode ser entendida como um *pipeline* entidade-cêntrico orientado por LLM, no qual o objeto principal de extração não é qualquer entidade mencionada no texto, mas especificamente o personagem literário. A partir desse núcleo, o sistema busca recuperar dois tipos de informação complementares: (i) características explicitamente associadas a personagens, tais como traços físicos, estados, papéis sociais ou qualificações psicológicas; e (ii) relações textualmente sustentadas entre pares de personagens. Essa escolha metodológica delimita o escopo da tarefa e reduz a tendência de sistemas generativos a introduzir objetos, lugares ou abstrações como itens centrais da estrutura extraída.

Do ponto de vista operacional, o processo geral pode ser resumido da seguinte forma. Inicialmente, a obra literária é adquirida em formato textual e carregada para o ambiente experimental. Em seguida, o texto é segmentado em sentenças, preservando a granularidade mínima adequada para localização da evidência. As sentenças são agrupadas em *chunks* de tamanho máximo pré-definido, de modo a permitir processamento incremental sem extrapolar o contexto utilizável pelo modelo. Sobre cada sentença, aplica-se uma cadeia de três chamadas estruturadas ao LLM: a primeira simplifica a sentença em proposições mais simples; a segunda tenta

resolver pronomes e referências indiretas com apoio de uma janela de contexto recente; a terceira extrai personagens, características e relações em um esquema validado por *schema*. Os registros produzidos são então consolidados em uma estrutura global por documento, da qual derivam artefatos tabulares, arquivos JSON e JSONL e o grafo final de personagens. Um dos tutoriais do Simpósio Brasileiro de Banco de Dados (SBBD)¹ de 2024 foi utilizado como auxílio para o processo geral.

Figura 5 – Pipeline metodológico proposto para IE em obras literárias em português do Brasil



Fonte: o autor.

A Figura 5 apresenta o fluxo metodológico geral adotado neste trabalho. A partir de obras literárias obtidas da BLPL, o texto é preparado, segmentado em sentenças e organizado em chunks. Em seguida, seleciona-se uma das três estratégias experimentais de *prompting*: *prompt* básico, *chain-of-thought prompting* ou *in-context learning*. A estratégia escolhida parametriza as três cadeias de processamento com LangChain: simplificação de sentenças, resolução de correferência com contexto recente e extração estruturada de personagens, características e relações. As extrações são armazenadas com metadados e evidências textuais, consolidadas globalmente e exportadas em formatos estruturados. Por fim, os personagens e suas relações são representados em um grafo de conhecimento, permitindo análise exploratória da rede narrativa.

¹ <https://sprout-handspring-d7a.notion.site/SBBD-2024-Tutorial-Methods-and-Tools-for-Generating-Knowledge-Graphs-from-Natural-Language-Text-10d92fe574a080bc8526f1aeb9892fb2>

4.2 CORPUS E ORGANIZAÇÃO DOS DADOS

O corpus deste trabalho é composto por obras literárias em português do Brasil, preferencialmente em domínio público, obtidas a partir da Biblioteca Digital de Literatura de Países Lusófonos (BLPL). A escolha dessa fonte atende aos critérios de acesso aberto, relevância literária e disponibilidade textual, além de favorecer a reprodutibilidade dos experimentos. As obras são utilizadas em formato textual simples, o que simplifica o carregamento no ambiente computacional e reduz interferências de marcações editoriais.

Para compor o corpus experimental, foram selecionadas cinco obras literárias em português do Brasil, listadas na Tabela 3. Essas obras serviram como fonte textual para os experimentos, mas não foram necessariamente processadas integralmente. A partir delas, foram selecionados trechos representativos contendo menções a personagens, descrições, características e relações narrativas, de modo a permitir a avaliação exploratória do *pipeline*.

Tabela 3 – Obras literárias utilizadas no corpus experimental

Título	Autor	Ano	Temática
As Vítimas Algozes	Joaquim Manuel de Macedo	1869	Romance abolicionista
Dom Casmurro	Machado de Assis	1899	Romance impressionista.
Memórias póstumas de Brás Cubas	Machado de Assis	1881	Romance impressionista
O Cortiço	Aluísio Azevedo	1890	Romance naturalista
Senhora	José de Alencar	1875	Romance urbano

Fonte: o autor.

4.3 ARQUITETURA LÓGICA DO PIPELINE

A implementação foi organizada como um *pipeline* modular, com componentes claramente separados para configuração experimental, definição de esquemas, pré-processamento textual, *prompting* estruturado, processamento principal, consolidação de resultados e visualização do grafo. Essa modularização tem duas vantagens principais. A primeira é facilitar a compreensão do fluxo completo, uma vez que cada componente responde a uma função específica. A segunda é tornar o processo reprodutível e extensível, permitindo a substituição de partes do sistema sem alterar a lógica global da proposta.

No nível de configuração experimental, são definidos os parâmetros que controlam a execução do processo. Nesta implementação, os principais parâmetros incluem o nome do modelo, a temperatura de geração, o tamanho máximo dos *chunks*, a janela de contexto recente para correferência e o identificador padrão do documento. O modelo adotado é o qwen/qwen3-32b, acionado por meio de ChatGroq, com temperatura igual a 0.0.

A temperatura é um parâmetro de inferência do LLM que controla a aleatoriedade da geração. Temperaturas mais baixas tornam a saída mais determinística e consistente, enquanto valores mais altos aumentam a diversidade, mas também a variabilidade das respostas. Neste trabalho, a temperatura foi mantida fixa em todos os experimentos para preservar a comparabilidade entre as estratégias de *prompting* e favorecer saídas estruturadas mais estáveis.

A janela de contexto recente para correferência foi definida em 5 sentenças anteriormente identificadas e o tamanho-alvo dos *chunks* foi definido em aproximadamente 1200 caracteres. Esse valor foi adotado empiricamente, buscando equilibrar a preservação de contexto local, a redução do custo computacional e o processamento incremental das obras. Na implementação, esse valor funciona como uma referência para a formação dos *chunks*, e não como uma regra de corte rígido no meio das sentenças. Assim, quando uma sentença não cabe integralmente no *chunk* atual, ela é movida para o próximo *chunk*, preservando sua unidade textual. Caso uma sentença isolada ultrapasse esse tamanho-alvo, ela é mantida integralmente em um único *chunk*, ainda que o bloco exceda o valor previsto. Essa decisão evita a fragmentação de períodos literários longos, que poderia prejudicar a interpretação de personagens, características e relações.

Tabela 4 – Esquemas estruturados utilizados na extração

Schema	Campos principais	Finalidade
SimplifiedSentences	sentences	Armazenar as proposições simplificadas derivadas da sentença original.
CoreferenceResolution	resolved_sentence	Representar a sentença após resolução de pronomes e referências indiretas.
CharacterItem	canonical_name, mention_text, evidence_text	Registrar a menção de um personagem e sua evidência local.
Characteristic	character, category, description, evidence_text	Representar atributos explicitamente associados a um personagem.
Relationship	source_character, relation, target_character, evidence_text	Representar relações entre dois personagens.
SentenceExtraction	listas dos três itens acima	Estruturar a saída final da extração por sentença.

Fonte: o autor.

No nível estrutural, a saída esperada do modelo é formalizada por classes `Pydantic`². Essa decisão é importante porque transforma a geração textual livre em uma geração condi-

² <https://pydantic.dev/docs/>

onada a um esquema explícito, reduzindo ambiguidades e facilitando a validação sintática dos resultados. No total, foram definidos seis esquemas estruturados, apresentados na Tabela 4. Esses esquemas não correspondem a seis etapas independentes do *pipeline*, mas a diferentes estruturas de dados utilizadas ao longo do processamento.

Os esquemas `SimplifiedSentences` e `CoreferenceResolution` representam, respectivamente, as saídas das etapas de simplificação de sentenças e resolução de correferência. Já os esquemas `CharacterItem`, `Characteristic` e `Relationship` descrevem os três tipos de informação extraídos na etapa central: personagens, características e relações. Por fim, o esquema `SentenceExtraction` organiza a extração completa de uma sentença, reunindo listas desses três tipos de elementos.

Em termos metodológicos, essa etapa é crucial, pois fixa o contrato entre o LLM e o restante do *pipeline*. Em vez de permitir que o modelo devolva qualquer descrição textual, a proposta impõe uma estrutura explícita e auditável, coerente com a necessidade de posterior consolidação e análise dos resultados.

4.4 PRÉ-PROCESSAMENTO TEXTUAL

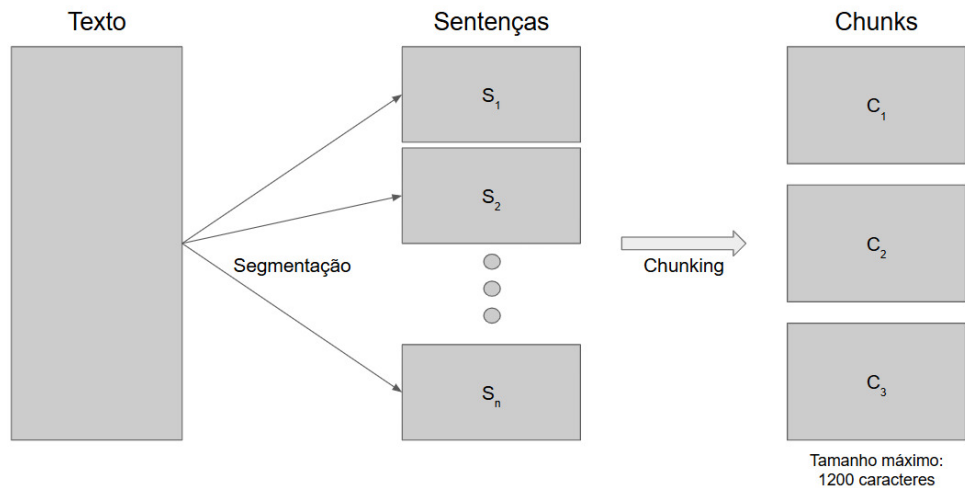
O pré-processamento textual é a primeira etapa operacional do *pipeline*. Seu objetivo é converter a obra em uma sequência ordenada de sentenças, que servirá como unidade básica de processamento e rastreamento. A implementação adota uma função de carregamento simples para leitura do arquivo em UTF-8, seguida de segmentação em sentenças com o uso do `sent_tokenize` do NLTK configurado para português. O segmentador foi utilizado como uma aproximação operacional, e não como uma solução perfeita para todos os fenômenos da prosa literária. Em textos do século XIX, pontuação de diálogos, reticências, abreviações, enumerações e períodos longos podem dificultar a segmentação automática. Durante a inspeção dos trechos processados, não se observou quebra sistemática de diálogos ao meio que comprometesse os experimentos, embora erros pontuais possam ocorrer. Por esse motivo, sentenças longas foram preservadas como unidades textuais e tratadas posteriormente pela etapa de simplificação de sentenças. A opção por sentenças como unidade elementar de análise atende a dois objetivos: primeiro, preservar proximidade entre a evidência textual e o item extraído; segundo, permitir a geração de identificadores finos, úteis para auditoria e para comparação manual posterior.

É importante observar que, em textos literários, sentenças nem sempre correspondem a unidades triviais de interpretação. Diálogos, enumerações, aposições, períodos longos e construções subordinadas podem concentrar múltiplas ações ou múltiplas descrições em uma única sentença ortográfica. Por essa razão, a segmentação em sentenças, embora essencial, não é suficiente por si só para tornar a informação imediatamente extraível. É justamente essa limitação que motiva a etapa posterior de simplificação.

Após a segmentação, as sentenças são agrupadas em *chunks* com base em um limite máximo de caracteres, como é possível observar na Figura 6. A implementação percorre sequencialmente a lista de sentenças e acumula novas sentenças no *chunk* corrente até que a

adição seguinte ultrapasse o tamanho máximo estipulado. Quando isso ocorre, o *chunk* atual é fechado e um novo bloco é iniciado. Essa estratégia preserva as fronteiras de sentença e, ao mesmo tempo, produz blocos de tamanho controlado, adequados ao processamento iterativo.

Figura 6 – Segmentação em sentenças e formação de *chunks*



Fonte: o autor.

4.5 ETAPA DE SIMPLIFICAÇÃO DE SENTENÇAS

A primeira chamada ao modelo em nível de sentença tem como função simplificar a entrada em proposições mais simples, preservando o sentido original. Essa etapa é necessária porque textos literários frequentemente condensam várias ações, descrições e relações em um mesmo período. Quando o modelo recebe uma sentença longa e estruturalmente complexa, aumenta o risco de perda de itens relevantes, fusão indevida de eventos ou extrações parcialmente inconsistentes. A simplificação atua, portanto, como um mecanismo de normalização leve antes da extração final.

Do ponto de vista do *prompting*, a instrução de simplificação estabelece que cada sentença final deve conter apenas uma ação principal, preservar o sujeito explícito sempre que possível, não inventar informações e manter nomes próprios e relações relevantes. A saída é solicitada no formato determinado pelo esquema *SimplifiedSentences*. Esse desenho do *prompt* procura equilibrar dois objetivos potencialmente conflitantes: decompor períodos complexos sem descaracterizar o conteúdo narrativo local.

Prompt de simplificação (resumo):

“Você é um especialista em análise linguística de textos literários em português. Sua tarefa é transformar a entrada em sentenças simples, preservando o sentido original. Regras: cada sentença final deve conter apenas uma ação principal;

mantenha o sujeito explícito sempre que possível; não invente informações; preserve nomes próprios e relações relevantes; retorne no formato solicitado.”

Tabela 5 – Objetivos e restrições do prompt de simplificação

Aspecto	Descrição
Objetivo	Decompor sentenças complexas em proposições mais simples.
Unidade-alvo	Sentença original.
Restrição central	Apenas uma ação principal por sentença simplificada.
Cuidados	Não introduzir fatos novos; preservar nomes próprios, sujeitos e relações.
Saída esperada	Lista validada de sentenças simplificadas.

Fonte: o autor.

A Tabela 5 sintetiza os principais objetivos e restrições do *prompt* utilizado na etapa de simplificação de sentenças, destacando a necessidade de preservar o sentido original e evitar a perda de informações relevantes.

Caso o modelo falhe nessa etapa, a implementação preserva a sentença original como alternativa de segurança. Esse comportamento é metodologicamente relevante, pois evita perda de cobertura e assegura que o *pipeline* continue operando mesmo quando uma etapa intermediária não gera saída válida.

4.6 RESOLUÇÃO DE CORREFERÊNCIA COM CONTEXTO RECENTE

Após a simplificação, cada sentença resultante passa por uma etapa de resolução de correferência. O objetivo aqui não é reconstruir toda a cadeia anafórica do romance, o que exigiria mecanismos mais amplos e intercapitulares, mas sim realizar uma resolução local e pragmática, suficiente para reduzir ambiguidades imediatas antes da extração. Em termos práticos, o sistema mantém uma fila limitada de sentenças recentemente identificadas. Essa fila funciona como memória de curto alcance e é convertida em um contexto textual enviado ao LLM.

O *prompt* de correferência determina que a sentença seja reescrita resolvendo pronomes e referências indiretas somente quando houver base razoável no contexto recente. Em caso de ambiguidade forte, a orientação é manter a formulação original. Essa escolha evita substituições excessivamente agressivas, que poderiam inserir falsos positivos ou deslocar o referente correto. Em outras palavras, a correferência aqui é tratada como uma etapa auxiliar de desambiguação conservadora, não como um objetivo autônomo do sistema.

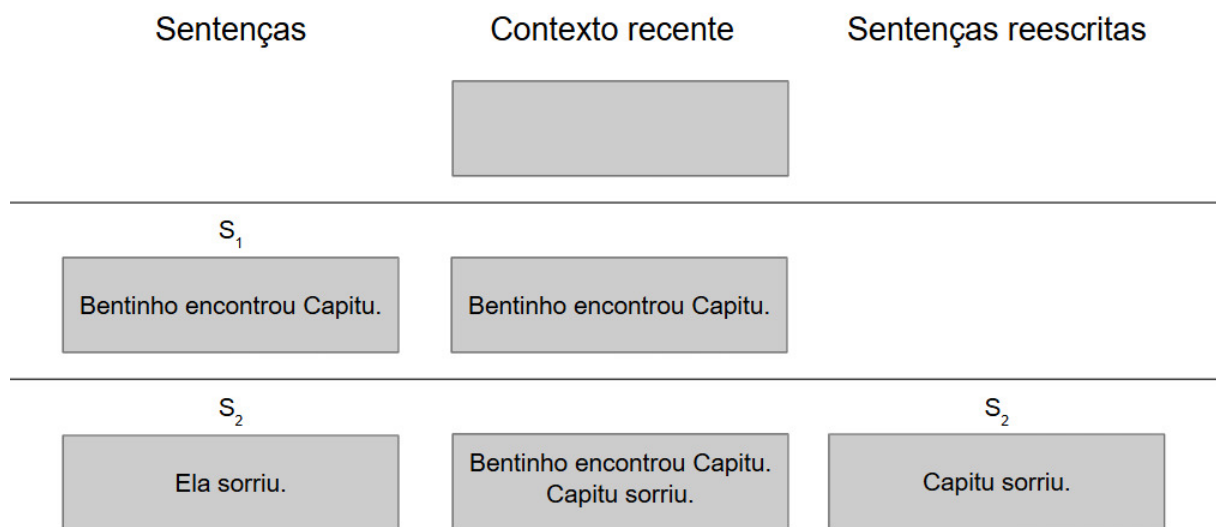
Prompt de correferência (resumo):

“Você é um especialista em resolução de correferência em textos literários em português. Sua tarefa é reescrever a sentença resolvendo pronomes e referências indiretas usando somente o contexto recente informado. Regras: só substitua quando

houver base razoável no contexto; se houver ambiguidade forte, mantenha a sentença original; não acrescente fatos novos; retorne no formato solicitado.”

No código, o contexto recente é derivado das últimas sentenças extraídas pelo LLM, armazenadas em uma estrutura do tipo deque com limite fixo de 5 entradas. A janela é mantida ao longo do processamento sequencial das sentenças, independentemente da divisão em *chunks*. Dessa forma, ao iniciar o processamento de um novo *chunk*, as últimas sentenças processadas no *chunk* anterior continuam disponíveis como contexto para as primeiras sentenças do bloco corrente, reduzindo a perda de continuidade narrativa causada pela divisão do texto. Essa solução é propositalmente simples e transparente. Em vez de depender de um módulo externo especializado de correferência, o sistema utiliza a própria capacidade do modelo de linguagem de interpretar pronomes e expressões nominais em contexto curto. Embora essa estratégia não resolva todos os casos de anáfora literária, ela se mostra adequada ao objetivo do trabalho, que é apoiar a extração estruturada local preservando rastreabilidade.

Figura 7 – Resolução de correferência com base em contexto recente



Fonte: o autor.

A Figura 7 exemplifica a estratégia adotada para a resolução de correferência com contexto recente. Antes do processamento da primeira sentença, a memória do contexto ainda não contém elementos. Após a análise de S_1 , a sentença extraída é incorporada ao contexto recente. Esse contexto é então utilizado na interpretação da sentença seguinte, S_2 , permitindo que o pronome *Ela* seja substituído por *Capitu*, desde que haja evidência contextual suficiente para essa resolução. Assim, a sentença reescrita torna explícito o referente do pronome e facilita a etapa posterior de extração estruturada. Essa abordagem não busca resolver todas as cadeias anafóricas da obra, mas atua como um mecanismo local de desambiguação, contribuindo para associar corretamente ações, características e relações aos personagens mencionados.

4.7 EXTRAÇÃO ESTRUTURADA DE PERSONAGENS, CARACTERÍSTICAS E RELAÇÕES

A etapa central do *pipeline* é a extração estruturada. Nela, a sentença já simplificada e, quando possível, desambiguada por correferência, é submetida ao modelo com um *prompt* explicitamente orientado ao recorte do trabalho. O modelo deve extrair somente três tipos de elementos: personagens, características explicitamente atribuídas a personagens e relações entre personagens. A decisão de restringir o escopo dessa forma é central para a proposta, pois transforma a tarefa de extração em um problema mais controlado e mais compatível com o objetivo analítico deste trabalho.

No caso de personagens que aparecem sob diferentes formas de menção, como nome próprio, apelido, título ou expressão descritiva, o *pipeline* busca associar a menção a um nome canônico quando há evidência suficiente na sentença ou no contexto recente. Essa associação é apoiada pela etapa de resolução de correferência, que procura explicitar pronomes e referências indiretas antes da extração. Ainda assim, a consolidação global de aliases e variações nominais não é completamente resolvida neste trabalho, podendo permanecer como limitação da abordagem atual.

O *prompt* de extração define explicitamente as categorias esperadas. Personagem é qualquer pessoa ou figura narrativa identificável na sentença. Característica é um atributo explícito, estado, papel social ou descrição do personagem. Relação é uma ação, vínculo ou interação de um personagem para outro personagem. Ao mesmo tempo, o *prompt* inclui restrições negativas, proibindo a extração de objetos, lugares ou conceitos como itens centrais e determinando que toda saída tenha evidência textual local. Isso é particularmente importante em textos literários, nos quais a linguagem figurada pode induzir o modelo a atribuir estatuto ontológico indevido a elementos metafóricos ou descritivos.

Prompt de extração (resumo):

“Você é um especialista em extração de informação de textos literários em português. Sua tarefa é extrair apenas personagens, características explícitas de personagens e relações entre personagens. Regras: não extraia objetos, lugares ou conceitos como itens centrais; só inclua um personagem se houver evidência textual na própria sentença; use nomes canônicos curtos para personagens; em `mention_text`, preserve a forma textual usada na sentença; extraia apenas características explicitamente sustentadas; extraia relações somente quando os dois polos forem personagens; toda personagem, característica e relação deve ter `evidence_text`; não duplique itens semanticamente equivalentes; não invente inferências fora da sentença.”

A saída é validada contra o esquema `SentenceExtraction`, que contém três listas: `characters`, `characteristics` e `relationships`. Cada personagem é descrito por nome canônico, forma de menção e evidência. Cada característica contém o personagem associado,

uma categoria dentre `physical`, `psychological`, `social_role`, `state` e `other`, além da descrição textual e da evidência correspondente. Cada relação contém personagem de origem, rótulo da relação, personagem-alvo e evidência textual.

A Tabela 6 apresenta as categorias utilizadas para organizar as características extraídas dos personagens, permitindo diferenciar atributos físicos, psicológicos, sociais, estados e outros tipos de descrição narrativa.

Tabela 6 – Categorias de características utilizadas na extração

Categoria	Descrição
<code>physical</code>	Traços físicos explicitamente atribuídos ao personagem.
<code>psychological</code>	Traços mentais, emocionais ou disposicionais explicitamente mencionados.
<code>social_role</code>	Papéis sociais, familiares, ocupacionais ou hierárquicos.
<code>state</code>	Estados transitórios ou condições momentâneas do personagem.
<code>other</code>	Outras características explícitas que não se encaixem nas categorias anteriores.

Fonte: o autor.

A Tabela 7 resume os *prompts* utilizados ao longo do *pipeline*, relacionando cada etapa de processamento à sua respectiva função dentro da proposta.

Tabela 7 – Quadro-resumo dos prompts utilizados no pipeline

Etapa	Entrada	Objetivo e restrições principais
Simplificação	Sentença original	Decompor períodos complexos em sentenças simples, preservando sentido, sujeito e nomes próprios.
Correferência	Sentença simplificada + contexto recente	Resolver pronomes e referências indiretas apenas quando houver base razoável no contexto.
Extração	Sentença resolvida	Extrair apenas personagens, características explícitas e relações entre personagens, sempre com evidência textual.

Fonte: o autor.

4.8 PROCESSAMENTO PRINCIPAL E GERAÇÃO DE REGISTROS

A execução do *pipeline* é coordenada por uma classe principal responsável por percorrer o texto, processar sentenças, manter o contexto recente e armazenar os resultados. A lógica segue uma estrutura iterativa em dois níveis. No primeiro nível, o documento é percorrido *chunk* a *chunk*. No segundo nível, cada sentença do *chunk* é processada individualmente, podendo gerar uma ou mais sentenças simplificadas, cada qual resultando em um registro próprio.

Para cada unidade processada, o sistema produz um registro com os seguintes metadados: identificador único do registro, experimento utilizado, identificador do documento, identificador do *chunk*, identificador da sentença, índice da simplificação, sentença original,

sentença simplificada, sentença resolvida por correferência, contexto utilizado, indicador de sucesso de *parsing*, extração estruturada e saídas brutas intermediárias. Essa estratégia é uma das partes mais importantes da proposta, pois garante rastreabilidade total do processo. Em vez de armazenar somente o resultado final consolidado, a implementação preserva o histórico local da transformação sofrida por cada sentença.

A Figura 8 apresenta um exemplo de inspeção tabular dos registros exportados em CSV, evidenciando a preservação de metadados como documento, *chunk*, sentença e evidência textual associada à extração.

Figura 8 – Exemplo de inspeção tabular dos metadados em CSV

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	record_id	document_id	chunk_id	sentence_id	simplified_inde	original_sentence	simplified_sentence	resolved_sentence	context_used	parsed_ok	characters	characteristics	relationships
2	3d7e53b3-fd4-4	senhora	senhora_chunk_001	senhora_sent_0001	1	Seixas convenceu-Seixas convenceu-se	Seixas convenceu-se	Seixas convenceu-se	Seixas convenceu-se	TRUE	[{"canonical_name": "Seixas", [{"character": "Seixas"}]	[{"source_character": "Seixas"}]	
3	bd98e04c-7aa0-4	senhora	senhora_chunk_001	senhora_sent_0001	2	Seixas convenceu-Seixas revoltou-se	Seixas revoltou-se	Seixas convenceu-se	Seixas convenceu-se	TRUE	[{"canonical_name": "Seixas", [{"character": "Seixas"}]	[{"source_character": "Seixas"}]	
4	01259c30-b505-4	senhora	senhora_chunk_001	senhora_sent_0002	1	Seixas não se per-Seixas não se perdo	Seixas não se perdo	Seixas não se perdo	Seixas convenceu-se	TRUE	[{"canonical_name": "Seixas", [{"character": "Seixas"}]	[{"source_character": "Seixas"}]	
5	75f9d349-eef4-4	senhora	senhora_chunk_001	senhora_sent_0002	2	Seixas não se per-Seixas apaixonava-se	Seixas apaixonava-se	Seixas convenceu-se	Seixas convenceu-se	TRUE	[{"canonical_name": "Seixas", [{"character": "Seixas"}]	[{"source_character": "Seixas"}]	
6	b00297f8-7158-4	senhora	senhora_chunk_001	senhora_sent_0002	3	Seixas não se per-Seixas não se perdo	Seixas não se perdo	Seixas convenceu-se	Seixas convenceu-se	TRUE	[{"canonical_name": "Seixas", [{"character": "Seixas"}]	[{"source_character": "Seixas"}]	
7	3c7e46b0-ac09-4	senhora	senhora_chunk_001	senhora_sent_0003	1	Aurélia era órfã, e Aurélia era órfã	Aurélia era órfã	Seixas revoltou-se	Seixas revoltou-se	TRUE	[{"canonical_name": "Aurélia", [{"character": "Aurélia"}]	[{"source_character": "Aurélia"}]	
8	003a62d9-d377-4	senhora	senhora_chunk_001	senhora_sent_0003	2	Aurélia era órfã, e Aurélia tinha uma velh	Aurélia tinha uma velh	Seixas não se perdo	Seixas não se perdo	TRUE	[{"canonical_name": "Aurélia", [{"character": "Aurélia"}]	[{"source_character": "Aurélia"}]	
9	7149399c-b23d-4	senhora	senhora_chunk_001	senhora_sent_0003	3	Aurélia era órfã, e A velha parenta era vi	A velha parenta era vi	Seixas apaixonava-se	Seixas apaixonava-se	TRUE	[{"canonical_name": "D. Firmin", [{"character": "D. Firmin"}]	[{"source_character": "D. Firmin"}]	
10	3b50aef9-da38-4	senhora	senhora_chunk_001	senhora_sent_0003	4	Aurélia era órfã, e D. Firmin Mascarenh	D. Firmin Mascarenh	Aurélia era órfã	Aurélia era órfã	TRUE	[{"canonical_name": "D. Firmin", [{"character": "D. Firmin"}]	[{"source_character": "D. Firmin"}]	
11	502e4e0-0476-4	senhora	senhora_chunk_001	senhora_sent_0003	5	Aurélia era órfã, e D. Firmin Mascarenh	D. Firmin Mascarenh	Aurélia era órfã	Aurélia era órfã	TRUE	[{"canonical_name": "D. Firmin", [{"character": "D. Firmin"}]	[{"source_character": "D. Firmin"}]	
12	8d95d581-08d4-4	senhora	senhora_chunk_001	senhora_sent_0004	1	Aurélia estava pe-Aurélia estava pelo b	Aurélia estava pelo b	Aurélia tinha um	Aurélia tinha um	TRUE	[{"canonical_name": "Aurélia", [{"character": "Aurélia"}]	[{"source_character": "Aurélia"}]	
13	a8e1e0f-d8c4-4	senhora	senhora_chunk_001	senhora_sent_0004	2	Aurélia estava pe-Aurélia estava pelo b	Aurélia estava pelo b	Aurélia tinha um	Aurélia tinha um	TRUE	[{"canonical_name": "Aurélia", [{"character": "Aurélia"}]	[{"source_character": "Aurélia"}]	
14	099793d5-b983-4	senhora	senhora_chunk_001	senhora_sent_0005	1	Era o Lemos um vi Era o Lemos um velh	Era o Lemos um velh	A velha parenta	A velha parenta	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
15	c534a8ad-89f9-4	senhora	senhora_chunk_001	senhora_sent_0005	2	Era o Lemos um vi Lemos não era muito	Lemos não era muito	D. Firmin Masc	D. Firmin Masc	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
16	fb8bd770-3d1d-4	senhora	senhora_chunk_001	senhora_sent_0005	3	Era o Lemos um vi Mas Lemos era rolh	Mas Lemos era rolh	Aurélia estava pi	Aurélia estava pi	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
17	6520a661-2bed-4	senhora	senhora_chunk_001	senhora_sent_0006	1	Quando o Lemos r Lemos, na qualidade	Lemos, na qualidade	Adelaide foi cont	Adelaide foi cont	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
18	e318af61-416c-4	senhora	senhora_chunk_001	senhora_sent_0006	2	Quando o Lemos r Ocorreu um incidente	Ocorreu um incidente	Era o Lemos um	Era o Lemos um	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
19	21120c89-58e4-4	senhora	senhora_chunk_001	senhora_sent_0006	3	Quando o Lemos r O incidente determino	O incidente determino	Lemos não era n	Lemos não era n	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
20	49934e09-096f-4	senhora	senhora_chunk_001	senhora_sent_0007	1	Apesar da recomi Apesar da recomenda	Apesar da recomenda	Apesar da recom	Apesar da recom	TRUE	[{"canonical_name": "Lemos", [{"character": "Lemos"}]	[{"source_character": "Lemos"}]	
21	38657027-454e-4	senhora	senhora_chunk_001	senhora_sent_0007	2	Apesar da recomi A secretária acomodo	A secretária acomodo	Lemos, na qual	Lemos, na qual	TRUE	[{"canonical_name": "secretári", [{"character": "secretári"}]	[{"source_character": "secretári"}]	
22	8b050bd-9c8f-4	senhora	senhora_chunk_001	senhora_sent_0007	3	Apesar da recomi A roupa era branca e	A roupa era branca e	Ocorreu um inci	Ocorreu um inci	TRUE	[{"canonical_name": "secretári", [{"character": "secretári"}]	[{"source_character": "secretári"}]	

Fonte: o autor.

Tal decisão metodológica possui duas implicações relevantes. A primeira é tornar a análise e localização de erros muito mais fácil, pois o pesquisador pode verificar se um problema surgiu na simplificação, na correferência ou na extração. A segunda é favorecer futura extensão do sistema, uma vez que as saídas intermediárias podem ser reutilizadas em tarefas de comparação, reprocessamento ou anotação manual.

4.9 AGREGAÇÃO E CONSOLIDAÇÃO GLOBAL

Depois que todos os registros em nível de sentença são produzidos, o *pipeline* executa uma etapa de consolidação global. Nessa fase, o objetivo é transformar um conjunto distribuído de extrações locais em uma estrutura unificada por documento. A consolidação começa pela construção de um mapa de personagens. Para cada personagem identificado ao longo dos registros, o sistema mantém um nome canônico e armazena a primeira menção observada, a primeira evidência textual, o primeiro identificador de sentença e o primeiro *chunk* em que a entidade apareceu. Em paralelo, todas as características e relações extraídas são reunidas em listas enriquecidas com os respectivos metadados de origem.

Essa etapa não realiza ainda uma fusão semântica complexa de aliases ou relações equivalentes por normalização forte. Em vez disso, ela produz uma estrutura global simples, transparente e suficiente para análise inicial. Do ponto de vista metodológico, essa escolha é

adequada ao escopo do trabalho, pois permite avaliar a utilidade do LLM na extração local antes de introduzir mecanismos mais sofisticados de unificação global. Além disso, como cada característica e cada relação preservam `sentence_id` e `chunk_id`, torna-se possível recuperar o trecho de origem e inspecionar a validade da informação extraída.

4.10 EXPORTAÇÃO DOS RESULTADOS E REPRESENTAÇÃO FINAL

A implementação materializa os resultados em diferentes formatos, cada qual atendendo a uma finalidade analítica distinta. Em primeiro lugar, os registros em nível de sentença são exportados para CSV, o que facilita inspeção tabular, filtragem e análise manual em ferramentas convencionais. Em segundo lugar, os mesmos registros são exportados para JSONL, preservando uma estrutura linha a linha conveniente para reprocessamento automatizado. Em terceiro lugar, a estrutura global consolidada é exportada para JSON, contendo metadados experimentais e as coleções globais de personagens, características e relações.

Figura 9 – Exemplo de visualização em grafo HTML gerada pelo pipeline



Fonte: o autor.

Além da exportação textual, o sistema produz uma visualização em grafo direcionado com auxílio das bibliotecas *NetworkX* e *PyVis*, como é possível observar na Figura 9. Nessa projeção, os nós correspondem a personagens e as arestas dirigidas representam relações extraídas, rotuladas pelo campo *relation*. Tal visualização não substitui a estrutura de conhecimento subjacente, mas funciona como mecanismo exploratório de interpretação narrativa, permitindo observar rapidamente a conectividade entre personagens.

4.11 REPRODUTIBILIDADE E PARÂMETROS EXPERIMENTAIS

Um dos objetivos da proposta é garantir reprodutibilidade. Para isso, a implementação fixa explicitamente os principais parâmetros de execução, incluindo o modelo, a temperatura, o tamanho máximo de *chunk* e a janela de contexto recente. Além disso, o sistema armazena, nos artefatos exportados, metadados como nome do modelo, temperatura, quantidade de documentos processados, número de registros produzidos e contagem global de personagens, características e relações. Essa camada de metadados é importante porque documenta as condições exatas sob as quais os resultados foram obtidos.

A opção por temperatura igual a zero busca privilegiar consistência estrutural. Em tarefas de extração de informação, a variabilidade excessiva pode prejudicar tanto a comparabilidade entre execuções quanto a estabilidade do formato de saída. Do mesmo modo, o uso de um limite de 1200 caracteres por *chunk* e de uma janela de 5 sentenças recentes corresponde a um compromisso entre preservação de contexto e simplicidade operacional. Esses valores podem ser ajustados em estudos futuros, mas, nesta proposta, cumprem adequadamente a função de tornar o processo controlado e repetível.

A Tabela 8 reúne os principais parâmetros experimentais utilizados no *pipeline*, incluindo modelo, temperatura, tamanho-alvo dos *chunks* e janela de contexto recente.

Tabela 8 – Parâmetros experimentais principais do pipeline

Parâmetro	Valor	Finalidade
Modelo	qwen/qwen3-32b	Modelo principal de geração estruturada.
Temperatura	0.0	Reduzir variabilidade e favorecer estabilidade do formato.
Tamanho máximo de chunk	1200 caracteres	Controlar o processamento incremental do texto.
Janela de contexto recente	5 sentenças	Apoiar resolução local de correferência.
Escopo da extração	Personagens, características e relações	Restringir o problema ao recorte definido.

Fonte: o autor.

4.12 SÍNTESE DO PROCESSO GERAL

Em síntese, a proposta desenvolvida neste trabalho organiza a extração de informação literária em uma cadeia de transformações sucessivas, na qual cada etapa procura reduzir

um tipo específico de dificuldade do domínio. A segmentação em sentenças e a construção de *chunks* atacam o problema da extensão das obras. A simplificação reduz complexidade sintática local. A correferência com contexto recente diminui ambiguidades imediatas entre personagens. A extração estruturada, por sua vez, restringe o escopo ao recorte de interesse e força o modelo a devolver saídas auditáveis. A consolidação final transforma evidências locais em uma estrutura global explorável por meio de tabelas, arquivos estruturados e grafos.

Do ponto de vista científico, a contribuição principal desta proposta não está apenas no uso de um LLM para extrair informação, mas na combinação entre *prompting* controlado, validação estrutural, preservação de proveniência e organização incremental do processamento. Essa combinação torna a abordagem adequada ao contexto de textos literários longos em português do Brasil, no qual a complexidade do domínio exige não apenas capacidade generativa, mas também disciplina metodológica na estruturação da saída.

Por fim, cabe destacar que este capítulo descreve a proposta em seu nível conceitual e operacional. A efetiva aplicação do *pipeline* às obras selecionadas, bem como a análise das extrações produzidas, será apresentada no capítulo seguinte, no qual são discutidos os experimentos realizados, os resultados obtidos e as principais limitações observadas.

5 EXPERIMENTOS E RESULTADOS

5.1 CONSIDERAÇÕES INICIAIS

Este capítulo apresenta os experimentos realizados para avaliar o *pipeline* proposto para extração de personagens, características e relações em obras literárias em português do Brasil. O objetivo é analisar o comportamento da abordagem em diferentes configurações de *prompting*, observando tanto a estruturação das informações extraídas quanto a utilidade dos artefatos gerados para inspeção e análise narrativa.

Foram conduzidos três experimentos utilizando o mesmo LLM e a mesma estrutura geral de processamento. Em todos os casos, mantiveram-se constantes o modelo, os parâmetros de geração, o processo de segmentação em sentenças, a construção de *chunks*, as cadeias de simplificação, correferência e extração estruturada. A variável experimental foi a estratégia de *prompting*: *prompt* básico, *chain-of-thought prompting* e *in-context learning*.

Os resultados são analisados a partir das extrações estruturadas, dos metadados preservados e dos grafos de conhecimento gerados, nos quais personagens são representados como nós e relações como arestas. O capítulo apresenta a implementação e o ambiente experimental, o conjunto de dados utilizado, o desenho dos experimentos, os critérios de avaliação, os resultados obtidos e, por fim, uma discussão sobre os principais achados e limitações observadas.

5.2 IMPLEMENTAÇÃO E AMBIENTE EXPERIMENTAL

A implementação do *pipeline* proposto foi realizada em linguagem Python, por meio de um *Jupyter Notebook*¹ executado localmente no Visual Studio Code². O ambiente utilizado contou com Python 3.14.4 e bibliotecas voltadas ao processamento textual, orquestração de chamadas a modelos de linguagem, validação estrutural das saídas, manipulação de dados e geração de grafos. A escolha por um *notebook* favoreceu o desenvolvimento incremental da solução, permitindo executar, inspecionar e ajustar separadamente as etapas de segmentação, construção de *chunks*, aplicação das cadeias de processamento, armazenamento dos registros e geração dos artefatos finais.

A orquestração das chamadas ao LLM foi realizada com LangChain, utilizando o modelo qwen/qwen3-32b por meio da interface ChatGroq. A validação das saídas estruturadas foi realizada com Pydantic, a fim de garantir que as respostas produzidas pelo modelo seguissem os esquemas esperados para personagens, características e relações. O processamento textual contou com recursos do NLTK para segmentação em sentenças, enquanto Pandas foi utilizado para organização tabular e exportação dos registros.

A implementação também contempla a seleção da estratégia experimental por meio de uma variável de configuração, permitindo executar separadamente os três experimentos de-

¹ <https://jupyter.org/>

² <https://code.visualstudio.com/>

finidos neste trabalho: *prompt* básico, *chain-of-thought prompting* e *in-context learning*. Dessa forma, o mesmo *pipeline* é preservado em todos os experimentos, variando apenas a estratégia de *prompting* empregada. Os resultados de cada execução são armazenados em diretórios específicos, contendo arquivos em JSON, JSONL e CSV, além de uma visualização em grafo gerada com NetworkX e PyVis.

A Tabela 9 apresenta a configuração do ambiente experimental, incluindo linguagem, ferramentas, bibliotecas e recursos utilizados para implementação, execução e análise dos resultados.

Tabela 9 – Configuração do ambiente experimental

Componente	Configuração utilizada
Linguagem de programação	Python 3.14.4
Ambiente de execução	Jupyter Notebook no Visual Studio Code
Modelo de linguagem	qwen/qwen3-32b
Interface de acesso ao modelo	ChatGroq
Orquestração do pipeline	LangChain
Validação estrutural	Pydantic
Segmentação textual	NLTK
Manipulação dos resultados	Pandas
Representação em grafo	NetworkX e PyVis
Formatos de saída	JSON, JSONL e CSV

Fonte: o autor.

5.3 CONJUNTO DE DADOS

O conjunto de dados utilizado nos experimentos é composto por cinco obras literárias em português do Brasil, obtidas a partir da Biblioteca Digital de Literatura de Países Lusófonos (BLPL). A escolha dessa fonte justifica-se por sua disponibilidade pública, pelo acesso gratuito a obras em domínio público e pela relevância do acervo para estudos envolvendo literatura brasileira. Além disso, o uso de textos literários provenientes de uma fonte aberta favorece a reprodutibilidade dos experimentos, uma vez que permite que outros pesquisadores acessem as mesmas obras e repliquem o processo de extração. Embora as obras completas tenham sido utilizadas como referência para composição do corpus, os experimentos foram executados sobre trechos selecionados de cada obra, em razão do custo computacional e do tempo necessário para processamento com LLMs. A seleção buscou contemplar passagens com presença de personagens, descrições e interações narrativas, mantendo trechos representativos para a tarefa de extração proposta. Esses elementos são importantes para avaliar o comportamento do *pipeline*, pois permitem observar tanto a capacidade de identificação de personagens quanto a associação de atributos e vínculos a partir de evidências textuais.

Antes da aplicação do modelo de linguagem, os textos foram carregados em formato textual e submetidos ao processo de segmentação em sentenças. Em seguida, as sentenças foram agrupadas em *chunks*, respeitando o limite máximo definido na implementação. Esse

procedimento foi necessário para permitir o processamento incremental de obras extensas, preservando a ordem textual e mantendo unidades de análise compatíveis com as etapas posteriores de simplificação, resolução de correferência e extração estruturada.

A Tabela 10 resume as principais estatísticas obtidas após o pré-processamento das obras completas, enquanto a Tabela 11 contém o resumo dos trechos efetivamente utilizados nos experimentos. Portanto, vale ressaltar que as obras completas são o universo de origem, mas os experimentos usam amostras textuais selecionadas. Esses dados permitem dimensionar o corpus utilizado e compreender a carga de processamento submetida ao *pipeline*. Além disso, a indicação do número de sentenças e de *chunks* por obra é relevante para interpretar os resultados apresentados nas seções seguintes, uma vez que a quantidade de unidades processadas influencia diretamente o número de registros, personagens, características e relações extraídas.

Tabela 10 – Estatísticas do corpus

Obra	Palavras	Palavras distintas	Sentenças	Chunks
As Vítimas Algozes	112.930	13.478	3.969	552
Dom Casmurro	79.031	9.280	4.113	330
Memórias póstumas de Brás Cubas	72.837	10.404	3.467	315
O Cortiço	95.471	11.996	5.156	423
Senhora	88.290	12.051	4.742	399

Fonte: o autor.

Tabela 11 – Estatísticas dos trechos selecionados para os experimentos

Obra	Palavras	Palavras distintas	Sentenças	Chunks
As Vítimas Algozes	293	167	7	2
Dom Casmurro	1.183	492	56	6
Memórias póstumas de Brás Cubas	269	134	9	2
O Cortiço	1.147	521	41	6
Senhora	1.028	465	46	5
Total	3.920	1.382	159	21

Fonte: o autor.

Os experimentos foram limitados a 159 sentenças por se tratar de uma avaliação exploratória e comparativa do comportamento do *pipeline*, e não de uma avaliação exaustiva das obras completas. Esse recorte permitiu executar os mesmos trechos nas três estratégias de *prompting*. Como cada sentença pode passar por diferentes etapas do *pipeline*, a ampliação para obras completas aumentaria substancialmente o custo computacional, o tempo de execução e o volume de saídas a serem inspecionadas. Os trechos foram selecionados por conterem personagens, características e relações narrativas relevantes para a análise, sendo considerado adequado para observar diferenças entre as estratégias avaliadas, enquanto a ampliação para trechos maiores ou obras completas é indicada como trabalho futuro.

5.4 DESENHO EXPERIMENTAL

O desenho experimental deste trabalho foi definido para comparar o impacto de três estratégias de *prompting* sobre o mesmo LLM. A variável experimental foi a forma de instruir o modelo, enquanto foram mantidos constantes o modelo, a temperatura, os trechos processados, a estrutura do *pipeline* e os esquemas de saída.

Foram avaliadas três configurações: *prompt* básico, *chain-of-thought prompting* e *in-context learning*. As subseções seguintes descrevem cada experimento. Essa organização permite analisar se diferentes níveis de orientação no *prompt* influenciam a identificação de personagens, a associação de características e a extração de relações textualmente sustentadas.

5.4.1 Experimento 1: Prompt Básico

O primeiro experimento utiliza uma estratégia de *prompting* direto, composta por instruções objetivas sobre a tarefa a ser executada. Nesse cenário, o modelo recebe a sentença de entrada e deve produzir a saída estruturada conforme o esquema definido, sem receber exemplos prévios de extração e sem orientação explícita para realizar etapas intermediárias de verificação. Essa configuração funciona como linha de base experimental, permitindo observar o desempenho do LLM quando submetido apenas a uma descrição direta da tarefa.

No contexto deste trabalho, o *prompt* básico é aplicado às três cadeias principais do *pipeline*: simplificação de sentenças, resolução de correferência e extração estruturada. Na etapa de extração, as instruções indicam que o modelo deve identificar apenas personagens, características explicitamente associadas a esses personagens e relações entre personagens, preservando evidências textuais. Por ser a estratégia mais simples, espera-se que esse experimento revele limitações iniciais do modelo, como maior sensibilidade a ambiguidades, possibilidade de extração de itens irrelevantes e menor padronização das respostas.

Os *prompts* completos utilizados neste experimento são apresentados no Apêndice A.

5.4.2 Experimento 2: Chain-of-Thought Prompting

O segundo experimento utiliza uma estratégia inspirada em *chain-of-thought prompting*, mas adaptada ao uso de saídas estruturadas. Nesse caso, o *prompt* não solicita que o modelo registre explicitamente o raciocínio passo a passo na resposta final. Em vez disso, ele inclui instruções de verificação antes da geração da saída, orientando o modelo a conferir se os elementos extraídos estão sustentados pela sentença, se os personagens identificados correspondem a entidades humanas ou figuras narrativas relevantes, e se as relações possuem dois polos válidos do tipo personagem.

Essa adaptação foi necessária porque o *pipeline* utiliza esquemas de validação para produzir uma saída em formato estruturado, contendo apenas os campos esperados para personagens, características e relações. Assim, explicações intermediárias, justificativas livres ou

raciocínios textuais não são armazenados. A comparação entre os experimentos permanece baseada apenas nos registros estruturados produzidos em cada estratégia.

Portanto, neste trabalho, o termo *chain-of-thought prompting* é usado para designar uma configuração de *prompt* que induz uma etapa prévia de verificação, sem expor ou registrar o raciocínio intermediário do modelo. Não se assume, contudo, que esse raciocínio possa ser observado diretamente na saída final. O que se avalia é o efeito dessa orientação sobre os resultados estruturados, especialmente na redução de erros como inclusão de objetos ou lugares como personagens, relações sem evidência textual ou características atribuídas ao personagem incorreto.

Os *prompts* completos utilizados neste experimento são apresentados no Apêndice B.

5.4.3 Experimento 3: In-Context Learning

O terceiro experimento emprega *in-context learning*, isto é, a inclusão de exemplos diretamente no *prompt* para orientar o comportamento do modelo. Diferentemente do *prompt* básico, que apresenta apenas instruções gerais, essa estratégia fornece ao LLM exemplos de entradas e saídas esperadas, demonstrando como personagens, características e relações devem ser representados no formato estruturado. Com isso, busca-se reduzir ambiguidades de interpretação e induzir maior consistência na resposta gerada.

No caso da extração de informação literária, os exemplos são especialmente úteis porque ajudam a delimitar o que deve ou não ser extraído. Por exemplo, o modelo pode ser orientado a distinguir personagens de lugares, objetos e conceitos abstratos, bem como a associar características apenas quando houver evidência textual explícita. Além disso, exemplos de relações entre personagens auxiliam na padronização dos rótulos e na identificação dos polos da relação. Dessa forma, espera-se que o experimento com *in-context learning* produza saídas mais alinhadas ao recorte do trabalho e com menor incidência de ruído estrutural.

Os *prompts* completos utilizados neste experimento são apresentados no Apêndice C.

Em síntese, os três experimentos foram planejados para representar uma progressão no nível de orientação fornecida ao modelo: o primeiro utiliza instruções diretas, o segundo adiciona uma etapa de verificação interna e o terceiro incorpora exemplos demonstrativos. Essa progressão permite analisar se o aumento da especificidade do *prompt* contribui para melhorar a qualidade das extrações, sem alterar o modelo ou os demais componentes do *pipeline*.

5.5 CRITÉRIOS DE AVALIAÇÃO

A avaliação dos experimentos foi conduzida com o objetivo de comparar o comportamento das três estratégias de *prompting* aplicadas ao mesmo LLM. Como este trabalho não utiliza uma base anotada manualmente como padrão de referência, não foram calculadas métricas clássicas como precisão, revocação e F1. Dessa forma, a avaliação assume caráter exploratório, descritivo e comparativo.

Foram considerados dois tipos de análise. A primeira é quantitativa-descritiva, baseada no volume de informações extraídas automaticamente pelos experimentos, como número de personagens identificados, características extraídas e relações entre personagens. Esses valores permitem observar diferenças de comportamento entre as estratégias, mas não devem ser interpretados isoladamente como medida de qualidade.

A segunda análise é qualitativa, realizada por inspeção de exemplos selecionados. Nessa etapa, foram observados aspectos como consistência estrutural das saídas, presença de evidência textual, coerência das relações extraídas, ocorrência de ambiguidades, duplicações, entidades genéricas e atribuições incorretas. Assim, a avaliação busca combinar uma visão geral do volume de extrações com uma análise mais detalhada dos acertos e limitações observados nas saídas.

5.6 RESULTADOS OBTIDOS

Esta seção apresenta os resultados obtidos a partir da execução dos três experimentos definidos no desenho experimental. A análise dos resultados foi organizada em três perspectivas complementares. Primeiramente, são apresentados os resultados quantitativos, contemplando indicadores como número de sentenças e *chunks* processados, personagens identificados, características extraídas e relações encontradas. Em seguida, são discutidos resultados qualitativos, com base na inspeção manual de sentenças selecionadas das obras, observando acertos, ruídos, falhas de correferência e adequação das extrações ao texto de origem. Por fim, são apresentados os grafos de conhecimento gerados a partir das relações extraídas, permitindo visualizar a rede de personagens produzida por cada estratégia experimental. Essa organização permite avaliar os experimentos não apenas pelo volume de informações extraídas, mas também pela qualidade e utilidade das saídas geradas.

5.6.1 Resultados Quantitativos

Os resultados quantitativos apresentados nesta subseção referem-se aos trechos selecionados das obras literárias utilizadas nos experimentos, e não às obras completas. Essa delimitação foi adotada em razão do custo computacional e do tempo de execução associados ao processamento com LLMs, considerando que os mesmos trechos foram submetidos às três estratégias experimentais.

A Tabela 12 apresenta uma visão quantitativa-descritiva do volume de extrações produzido por cada estratégia experimental. Os valores indicam quantos personagens, características e relações foram extraídos nos trechos processados, mas não representam, por si só, métricas de qualidade. Uma estratégia pode produzir mais extrações por recuperar mais informações relevantes, mas também pode gerar mais ruído, duplicações ou inferências indevidas. Por esse motivo, os resultados quantitativos são interpretados em conjunto com a análise qualitativa apresentada posteriormente.

Tabela 12 – Resultados gerais por estratégia experimental nos trechos processados

Experimento	Sentenças	Chunks	Personagens	Características	Relações
<i>Prompt</i> básico	159	21	153	339	153
<i>Chain-of-thought</i>	159	21	137	381	165
<i>In-context learning</i>	159	21	117	316	157

Fonte: o autor.

Como os mesmos 159 trechos e 21 *chunks* foram utilizados nos três experimentos, as diferenças observadas decorrem das estratégias de *prompting*. O *prompt* básico produziu o maior número de personagens, enquanto o *chain-of-thought prompting* gerou, no agregado, mais características e relações. O *in-context learning*, por sua vez, apresentou uma saída mais compacta em personagens e características, mantendo um número de relações próximo ao das demais estratégias.

Essa redução não deve ser interpretada automaticamente como pior desempenho. Ela indica que os exemplos fornecidos no próprio *prompt* podem ter levado o modelo a seguir um padrão mais restrito de extração, filtrando entidades ambíguas, genéricas ou pouco sustentadas pelo texto. No entanto, sem uma base anotada manualmente, não é possível afirmar quantitativamente se essa redução corresponde a maior precisão ou a omissões de itens relevantes. Essa distinção é retomada na análise qualitativa.

Tabela 13 – Resultados quantitativos por obra e estratégia experimental nos trechos processados

Obra	Estratégia	Personagens	Características	Relações
As Vítimas Algozes	<i>Prompt</i> básico	13	30	15
	<i>Chain-of-thought</i>	13	32	12
	<i>In-context learning</i>	7	18	8
Dom Casmurro	<i>Prompt</i> básico	53	97	57
	<i>Chain-of-thought</i>	48	112	64
	<i>In-context learning</i>	40	80	61
Memórias Póstumas de Brás Cubas	<i>Prompt</i> básico	10	13	9
	<i>Chain-of-thought</i>	8	13	12
	<i>In-context learning</i>	7	4	7
O Cortiço	<i>Prompt</i> básico	44	113	34
	<i>Chain-of-thought</i>	39	128	32
	<i>In-context learning</i>	36	121	39
Senhora	<i>Prompt</i> básico	33	86	38
	<i>Chain-of-thought</i>	29	96	45
	<i>In-context learning</i>	27	93	42

Fonte: o autor.

A Tabela 13 detalha os resultados por obra e estratégia experimental. Observa-se que o comportamento das estratégias varia conforme a obra e o trecho processado. Em geral, o *prompt* básico tende a produzir mais personagens, enquanto o *in-context learning* gera saídas mais compactas. Já o *chain-of-thought prompting* apresenta maior número de relações em obras

como *Dom Casmurro*, *Memórias Póstumas de Brás Cubas* e *Senhora*.

Essas variações podem estar associadas à extensão dos trechos, à densidade narrativa, à presença de diálogos, descrições e interações explícitas. Assim, os resultados quantitativos permitem observar tendências de volume e comportamento entre estratégias, mas não determinam isoladamente qual delas é qualitativamente superior. A próxima subseção complementa essa análise por meio da inspeção qualitativa de exemplos, considerando evidência textual, coerência das extrações, ambiguidades, duplicações e erros de interpretação ou correferência.

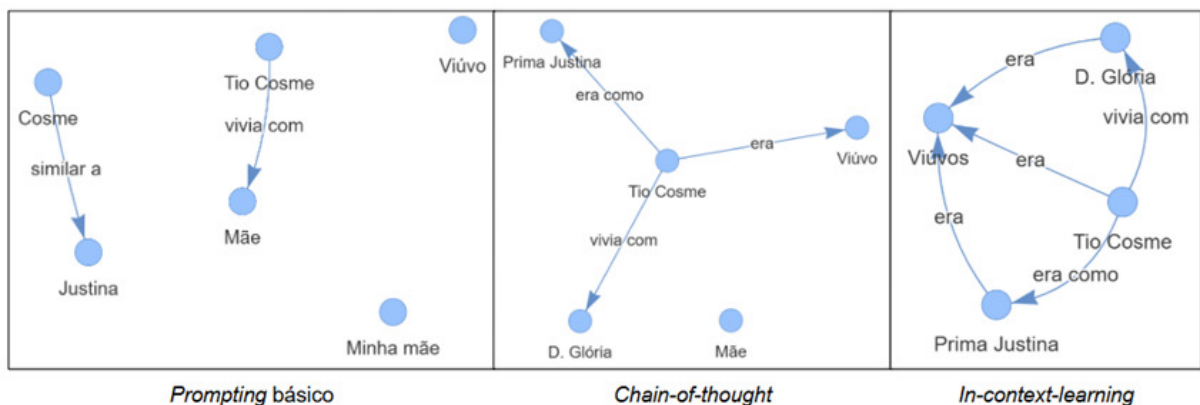
5.6.2 Resultados Qualitativos

A análise qualitativa teve como objetivo complementar os resultados quantitativos, observando com maior detalhe o comportamento das três estratégias experimentais em casos concretos de extração. Para isso, foram selecionados três exemplos representativos das obras processadas, contemplando sentenças com menções explícitas a personagens, descrições, relações interpessoais e casos de possível ambiguidade referencial. A comparação foi realizada entre os resultados produzidos pelo *prompt* básico, pelo *chain-of-thought prompting* e pelo *in-context learning*, considerando a adequação das extrações ao texto de origem.

A seguir, é apresentado o primeiro exemplo comparativo, do livro *Dom Casmurro*:

1. “Tio Cosme vivia com minha mãe, desde que ela enviuvou. Já então era viúvo, como prima Justina; era a casa dos três viúvos.”

Figura 10 – Grafos gerados pelos experimentos para o primeiro exemplo de *Dom Casmurro*



Fonte: o autor.

Na Figura 10, observa-se que as três estratégias identificaram parcialmente os personagens e relações presentes na sentença, mas com diferenças relevantes de qualidade. O *prompt* básico extraiu a relação “Tio Cosme vivia com Mãe”, porém introduziu uma relação pouco adequada entre “Cosme” e “Justina”, além de gerar variação entre “Mãe” e “Minha mãe”. O experimento com *chain-of-thought prompting* apresentou uma representação mais próxima do

texto ao associar “Tio Cosme” à condição de viúvo e à convivência com “D. Glória”, mas ainda manteve algumas conexões genéricas (e.g. “era como”). Nesse caso, a etapa de correferência contribuiu para resolver a expressão “minha mãe” como “D. Glória”, recuperando uma informação contextual necessária para tornar explícito o personagem referido. Já o *in-context learning* produziu o grafo mais completo entre os três, conectando os três personagens à condição de viuvez e representando a relação de convivência. Ainda assim, nota-se a presença de rótulos genéricos e de possíveis duplicações conceituais, indicando que a estratégia melhora a cobertura, mas não elimina totalmente problemas de normalização e consolidação semântica.

O segundo exemplo comparativo foi retirado do livro *Senhora*, de José de Alencar, e apresenta uma sentença voltada principalmente à caracterização da personagem Aurélia:

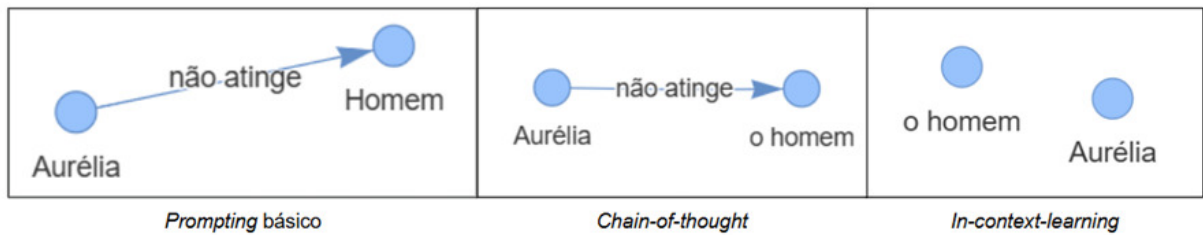
1. “A natureza dotara Aurélia com a inteligência viva e brilhante da mulher de talento, que se não atinge ao vigoroso raciocínio do homem, tem a preciosa ductilidade de prestar-se a todos os assuntos, por mais diversos que sejam.”

Tabela 14 – Características extraídas pelos experimentos no segundo exemplo de *Senhora*

Estratégia	Personagem	Categoria	Característica extraída
Prompt básico	Aurélia	other	inteligência viva e brilhante
	Aurélia	social_role	mulher de talento
	Aurélia	psychological	inteligência
	Homem	psychological	vigoroso raciocínio
	Aurélia	psychological	inteligência
	Aurélia	psychological	preciosa ductilidade
Chain-of-thought	Aurélia	psychological	inteligência viva e brilhante da mulher de talento
	Aurélia	psychological	inteligência
	o homem	psychological	vigoroso raciocínio
	Aurélia	psychological	inteligência com preciosa ductilidade de prestar-se a todos os assuntos
In-context learning	Aurélia	psychological	inteligência viva
	Aurélia	psychological	inteligência brilhante
	o homem	psychological	vigoroso raciocínio
	Aurélia	psychological	mulher de talento
	Aurélia	psychological	preciosa ductilidade de prestar-se a todos os assuntos

Fonte: o autor.

A Tabela 14 detalha as características extraídas pelos três experimentos, enquanto que a Figura 11 mostra os grafos gerados para o segundo exemplo qualitativo. Observa-se um caso mais voltado à extração de características do que à identificação de relações entre personagens. As três estratégias reconheceram Aurélia como personagem central e extraíram atributos associados à sua inteligência, como “inteligência viva e brilhante”, “mulher de talento” e “preciosa ductilidade”.

Figura 11 – Grafos gerados pelos experimentos para o segundo exemplo de *Senhora*

Fonte: o autor.

O *prompt* básico apresentou maior quantidade de características, mas também gerou certa redundância ao repetir a ideia de inteligência em diferentes formas e ao tratar “Homem” como personagem comparativo. O experimento com *chain-of-thought prompting* produziu uma extração mais condensada, reunindo parte das características em descrições mais completas, embora ainda tenha mantido a relação “não atinge” entre Aurélia e “o homem”, que não representa propriamente uma relação narrativa entre personagens. Já o *in-context learning* mostrou-se mais conservador na extração de relações, evitando criar uma aresta para a comparação com “o homem”, mas preservando características relevantes atribuídas a Aurélia. Esse exemplo evidencia que estratégias mais orientadas podem reduzir relações pouco adequadas, embora ainda permaneçam desafios na distinção entre personagem, categoria genérica e elemento comparativo.

O terceiro exemplo, extraído de *As Vítimas Algozes*, apresenta uma interação direta entre duas personagens, Cândida e Lucinda.

1. “E, levantando-se, Cândida avançou um passo para Lucinda, e voltando-lhe as costas, disse-lhe: – Despe-me.”

Na Figura 12, percebe-se que as três estratégias identificaram corretamente as personagens envolvidas e extraíram relações direcionadas de “Cândida” para “Lucinda”, como “avançou um passo para”, “voltou as costas a/para” e “disse a”. O *prompt* básico capturou as relações principais, mas apresentou uma evidência menos adequada na fala final, ao registrar “Despe Cândida” em vez de preservar corretamente a forma “Despe-me”. O experimento com *chain-of-thought prompting* produziu uma extração mais completa, pois além das relações também registrou o estado “levantou-se” associado a Cândida. Já o *in-context learning* manteve relações coerentes e evidências mais diretas, sem adicionar características. Esse exemplo mostra que, em sentenças com ações explícitas entre personagens, as três estratégias conseguem recuperar a estrutura relacional básica, mas diferem na qualidade das evidências e no grau de detalhamento da extração.

Figura 12 – Informações extraídas pelos experimentos para o terceiro exemplo de *As Vítimas Algozes*

<pre> { "metadata": { ... }, "characters": [{ "canonical_name": "Cândida", "first_mention_text": "Cândida", "first_evidence_text": "Cândida avançou um passo para Lucinda", "first_sentence_id": "3ex_vitimas_algozes_sent_0001", "first_chunk_id": "3ex_vitimas_algozes_chunk_001" }, { "canonical_name": "Lucinda", "first_mention_text": "Lucinda", "first_evidence_text": "avançou um passo para Lucinda", "first_sentence_id": "3ex_vitimas_algozes_sent_0001", "first_chunk_id": "3ex_vitimas_algozes_chunk_001" }], "characteristics": [], "relationships": [{ "source_character": "Cândida", "relation": "avançou um passo para", "target_character": "Lucinda", "evidence_text": "Cândida avançou um passo para Lucinda", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "basic_prompt" }, { "source_character": "Cândida", "relation": "voltando as costas a", "target_character": "Lucinda", "evidence_text": "voltando as costas a Lucinda", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "basic_prompt" }, { "source_character": "Cândida", "relation": "disse a", "target_character": "Lucinda", "evidence_text": "disse a Lucinda: - Despe Cândida", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "basic_prompt" }] } </pre>	<pre> { "metadata": { ... }, "characters": [{ "canonical_name": "Cândida", "first_mention_text": "Cândida", "first_evidence_text": "Cândida levantou-se.", "first_sentence_id": "3ex_vitimas_algozes_sent_0001", "first_chunk_id": "3ex_vitimas_algozes_chunk_001" }, { "canonical_name": "Lucinda", "first_mention_text": "Lucinda", "first_evidence_text": "Cândida avançou um passo para Lucinda.", "first_sentence_id": "3ex_vitimas_algozes_sent_0002", "first_chunk_id": "3ex_vitimas_algozes_chunk_001" }], "characteristics": [{ "character": "Cândida", "category": "state", "description": "levantou-se.", "evidence_text": "Cândida levantou-se.", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "chain_of_thought" }], "relationships": [{ "source_character": "Cândida", "relation": "avançou um passo para", "target_character": "Lucinda", "evidence_text": "Cândida avançou um passo para Lucinda.", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "chain_of_thought" }, { "source_character": "Cândida", "relation": "voltou as costas para", "target_character": "Lucinda", "evidence_text": "Cândida voltou as costas a Lucinda.", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "chain_of_thought" }, { "source_character": "Cândida", "relation": "disse a", "target_character": "Lucinda", "evidence_text": "Cândida disse a Lucinda: - Despe-me.", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "chain_of_thought" }] } </pre>	<pre> { "metadata": { ... }, "characters": [{ "canonical_name": "Cândida", "first_mention_text": "Cândida", "first_evidence_text": "Cândida levantou-se.", "first_sentence_id": "3ex_vitimas_algozes_sent_0001", "first_chunk_id": "3ex_vitimas_algozes_chunk_001" }, { "canonical_name": "Lucinda", "first_mention_text": "Lucinda", "first_evidence_text": "Cândida avançou um passo para Lucinda", "first_sentence_id": "3ex_vitimas_algozes_sent_0001", "first_chunk_id": "3ex_vitimas_algozes_chunk_001" }], "characteristics": [], "relationships": [{ "source_character": "Cândida", "relation": "avançou um passo para", "target_character": "Lucinda", "evidence_text": "Cândida avançou um passo para Lucinda", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "in_context_learning" }, { "source_character": "Cândida", "relation": "voltou as costas para", "target_character": "Lucinda", "evidence_text": "Cândida voltou as costas para Lucinda", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "in_context_learning" }, { "source_character": "Cândida", "relation": "disse a", "target_character": "Lucinda", "evidence_text": "Cândida disse a Lucinda", "sentence_id": "3ex_vitimas_algozes_sent_0001", "chunk_id": "3ex_vitimas_algozes_chunk_001", "experiment_mode": "in_context_learning" }] } </pre>
Prompting básico	Chain-of-thought	In-context-learning

Fonte: o autor.

De modo geral, a análise qualitativa indica que o *prompt* básico funciona como uma linha de base útil, mas tende a apresentar maior sensibilidade a ambiguidades e maior ocorrência de ruídos. A estratégia com *chain-of-thought prompting* mostrou-se mais adequada para reduzir inconsistências e estimular uma verificação mais cuidadosa antes da produção da saída estruturada. Por sua vez, o *in-context learning* demonstrou maior capacidade de seguir o padrão esperado de extração, sobretudo em relação ao formato da resposta e à delimitação entre personagens, características e relações. Assim, os exemplos qualitativos complementam os resultados quantitativos ao evidenciar que a qualidade das extrações não depende apenas do volume de itens gerados, mas também da precisão semântica, da preservação de evidências e da adequação ao escopo da tarefa.

5.6.3 Grafos de Conhecimento Gerados

Além dos arquivos estruturados gerados pelo *pipeline*, as relações extraídas foram representadas por meio de grafos de conhecimento. Nessa representação, cada personagem identificado é modelado como um nó, enquanto cada relação extraída entre dois personagens é representada como uma aresta direcionada. O rótulo da aresta corresponde ao tipo de relação produzido pelo modelo, como vínculos familiares, relações de trabalho, interações sociais ou outros tipos de associação textual observados nos trechos processados. Essa etapa permitiu observar, de forma exploratória, como as diferentes estratégias de *prompting* influenciaram a rede de personagens produzida. Assim, além de analisar a quantidade de personagens e relações

extraídas, tornou-se possível visualizar a conectividade entre os personagens e a densidade das relações identificadas.

Tabela 15 – Comparação dos grafos de conhecimento apresentados

Estratégia	Obra	Nós	Arestas	Observação geral
<i>Prompt</i> básico	<i>O Cortiço</i>	31	27	Grafo mais amplo, porém com maior presença de ruídos e nós pouco normalizados.
<i>Chain-of-thought prompting</i>	<i>As Vítimas</i> <i>Algozes</i>	13	10	Grafo mais denso e organizado, com relações concentradas em torno de personagens centrais.
<i>In-context learning</i>	<i>Memórias</i> <i>Póstumas de Brás Cubas</i>	7	6	Grafo mais compacto, com menor quantidade de relações e maior controle visual.

Fonte: o autor.

A Tabela 15 resume as principais características estruturais dos grafos apresentados. O número de nós corresponde à quantidade de personagens presentes no grafo, enquanto o número de arestas representa as relações extraídas entre esses personagens. Assim como nos resultados quantitativos anteriores, esses valores devem ser interpretados com cautela: um grafo com maior número de nós ou arestas não indica necessariamente melhor desempenho, pois pode refletir também duplicações, ruídos ou extrações indevidas. Por outro lado, grafos mais compactos podem indicar comportamento mais conservador do modelo, com menor quantidade de relações, mas possivelmente com maior precisão semântica.

A Figura 13 apresenta um exemplo de visualização gerada a partir das relações extraídas do livro *O Cortiço*, utilizando o *prompt* básico. No grafo, é possível observar agrupamentos narrativos em torno de personagens (e.g. “João Romão”, “Bertoleza”, “Miranda”). A visualização evidencia o potencial do grafo de conhecimento para explorar interações entre personagens e identificar núcleos relacionais da narrativa. Entretanto, por se tratar do experimento com *prompt* básico, também são perceptíveis limitações da extração, como a presença de nós genéricos ou pouco normalizados e variações (e.g. “Mulher”, “Ele”, “Aquele”). Assim, a figura ilustra tanto a utilidade da representação em grafo quanto a necessidade de etapas posteriores de normalização e consolidação de personagens.

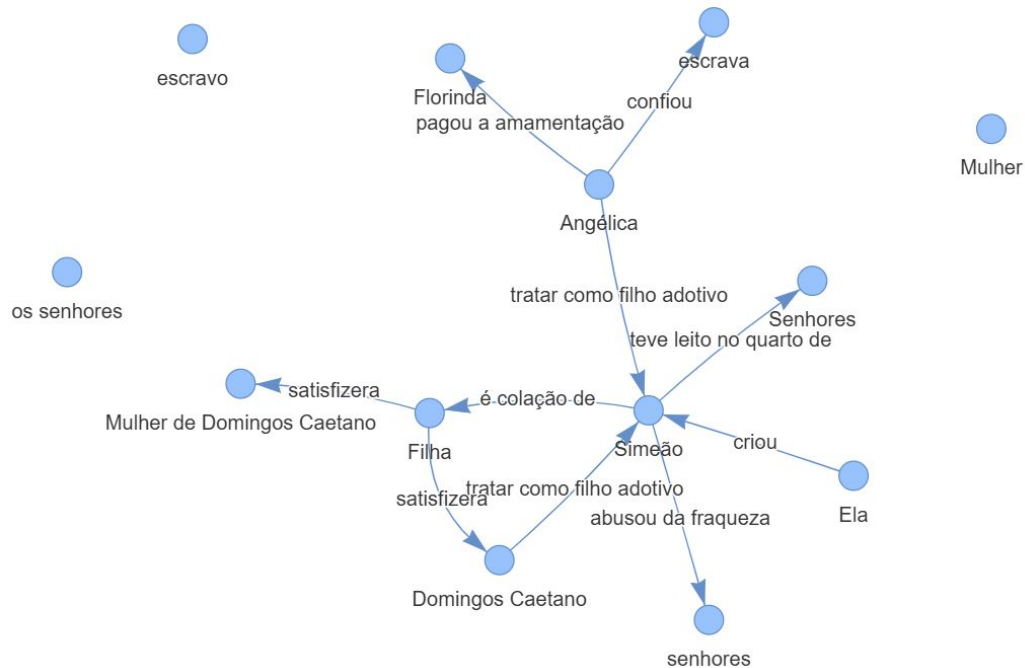
Figura 13 – Grafo de conhecimento gerado a partir das relações extraídas de *O Cortiço* com *prompt* básico



Fonte: o autor.

A Figura 14 apresenta um exemplo de visualização gerada a partir das relações extraídas de *As Vítimas Algozes*, utilizando a estratégia de *chain-of-thought prompting*. O grafo evidencia uma rede mais densa de interações, com destaque para personagens como “Simeão”, “Angélica”, “Domingos Caetano” e “Florinda”, além de relações associadas ao tratamento de “Simeão”, à amamentação de “Florinda” e a vínculos de cuidado e convivência. Em comparação com estratégias mais simples, a visualização revela maior riqueza relacional. Observam-se, entretanto, algumas variações superficiais de nomenclatura, como “Senhores”, “senhores” e “os senhores”, que poderiam ser mitigadas por uma etapa simples de normalização textual, como conversão para minúsculas e remoção de artigos. Ainda assim, o grafo ilustra de forma clara o potencial da abordagem para representar, de maneira estruturada, redes de personagens e suas interações narrativas.

Figura 14 – Grafo de conhecimento gerado a partir das relações extraídas de *As Vítimas Algozes* com *chain-of-thought prompting*

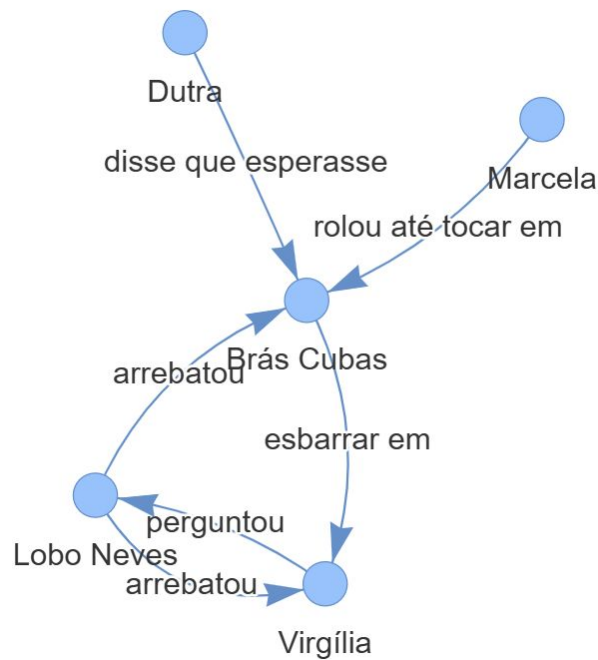


Fonte: o autor.

A Figura 15 apresenta o grafo de conhecimento gerado para *Memórias Póstumas de Brás Cubas* com a estratégia de *in-context learning*. A visualização mostra uma rede compacta e legível, com relações entre personagens centrais como “Marcela”, “Brás Cubas”, “Virgília” e “Lobo Neves”. Esse resultado indica que o uso de exemplos no *prompt* contribuiu para produzir uma estrutura mais controlada, com menor densidade visual e relações mais diretamente associadas às evidências textuais.

De modo geral, os grafos gerados demonstram a viabilidade de transformar as extrações estruturadas em uma representação visual da rede de personagens. Essa representação é útil para observar padrões de interação e para verificar, de maneira exploratória, como as relações extraídas pelo LLM se distribuem entre os personagens. No entanto, a qualidade do grafo depende diretamente da qualidade das extrações anteriores. Relações incorretas, personagens duplicados ou nomes não consolidados podem produzir nós e arestas artificiais, reforçando a importância das etapas de validação estrutural, correferência e análise qualitativa dos resultados.

Figura 15 – Grafo de conhecimento gerado a partir das relações extraídas de *Memórias Póstumas de Brás Cubas* com *in-context learning*



Fonte: o autor.

5.7 DISCUSSÃO

Os resultados obtidos indicam que a estratégia de *prompting* influencia diretamente o comportamento do *pipeline* de extração de informação. Embora os três experimentos tenham utilizado o mesmo modelo de linguagem, os mesmos trechos textuais e os mesmos esquemas de saída, foram observadas diferenças na quantidade e na qualidade das extrações produzidas. Isso sugere que, em tarefas de extração de personagens, características e relações em textos literários, o desempenho do LLM depende não apenas de sua capacidade linguística geral, mas também da forma como a tarefa é formulada no *prompt*.

O experimento com *prompt* básico funcionou como linha de base da comparação. Essa estratégia mostrou-se capaz de identificar personagens mencionados explicitamente, especialmente quando aparecem por nome próprio em sentenças diretas. Entretanto, por utilizar instruções mais simples, apresentou maior sensibilidade a ambiguidades e maior tendência à extração de elementos fora do escopo, como lugares, objetos, papéis genéricos ou expressões descritivas tratadas como personagens.

As estratégias mais elaboradas apresentaram ganhos qualitativos em relação à linha de base. O *chain-of-thought prompting* contribuiu para um comportamento mais cauteloso, ao induzir o modelo a realizar uma verificação interna antes de produzir a saída estruturada. Já o *in-context learning*, por incluir exemplos no próprio *prompt*, favoreceu maior padronização

das respostas e melhor aderência ao formato esperado. Ainda assim, nenhuma das estratégias eliminou completamente problemas como omissões, duplicações, inferências pouco sustentadas ou ambiguidades de referência.

A etapa de resolução de correferência mostrou-se relevante para o funcionamento do *pipeline*, especialmente em sentenças com pronomes ou referências indiretas a personagens. O uso de uma memória de contexto recente auxiliou na associação de ações, características e relações aos personagens corretos. No entanto, essa estratégia possui limitações: quando há mais de um personagem plausível no contexto ou quando a referência depende de trechos distantes, podem ocorrer atribuições incorretas. Assim, a correferência adotada deve ser entendida como um mecanismo auxiliar de desambiguação local, e não como uma solução completa para todas as cadeias referenciais da narrativa.

Os grafos de conhecimento gerados a partir das relações extraídas demonstram o potencial da abordagem para representar visualmente redes de personagens. Ao modelar personagens como nós e relações como arestas, torna-se possível observar padrões de interação e personagens mais conectados nos trechos analisados. Contudo, a qualidade do grafo depende diretamente da qualidade das extrações anteriores, pois nomes duplicados, relações incorretas ou inconsistências de identificação podem gerar nós e arestas artificiais.

De modo geral, os experimentos indicam que a combinação de LLMs, *prompting* estruturado, preservação de evidências textuais e representação em grafos de conhecimento constitui uma abordagem promissora para apoiar a extração de informação em textos literários em português do Brasil. A proposta não substitui a leitura crítica humana, mas oferece um mecanismo de apoio para organizar informações narrativas de forma mais sistemática e reproduzível. Ainda assim, permanecem desafios relacionados à ambiguidade literária, à correferência, à consolidação de nomes e à distinção entre informação explicitamente textual e inferência interpretativa.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho investigou a extração automática de informação em obras literárias em português do Brasil, com foco na identificação de personagens, suas características e relações narrativas. Para isso, foi proposto e implementado um *pipeline* baseado em Modelos de Linguagem de Grande Escala (LLMs), organizado em etapas de segmentação textual, construção de *chunks*, simplificação de sentenças, resolução de correferência com contexto recente, extração estruturada e representação dos resultados em grafos de conhecimento. A abordagem demonstrou ser viável para transformar trechos literários em estruturas mais organizadas e auditáveis, preservando metadados e evidências textuais associadas às extrações.

Os experimentos realizados permitiram comparar três estratégias de *prompting* aplicadas ao mesmo modelo: *prompt* básico, *chain-of-thought prompting* e *in-context learning*. Os resultados indicam que a forma como a tarefa é apresentada ao modelo influencia diretamente a qualidade das extrações. O *prompt* básico mostrou-se útil como linha de base, mas mais sujeito a ambiguidades e ruídos. Já as estratégias mais elaboradas, especialmente aquelas com verificação interna ou exemplos no próprio *prompt*, tenderam a produzir saídas mais alinhadas ao formato esperado e ao recorte da tarefa. Ainda assim, foram observadas limitações relacionadas à correferência, à consolidação de nomes, à distinção entre personagens e elementos não-personagens, e à possibilidade de inferências sem sustentação textual suficiente.

Como trabalhos futuros, propõe-se ampliar a avaliação para trechos maiores ou obras completas, incorporando mecanismos mais robustos de consolidação de personagens, resolução de aliases e correferência de longo alcance. Outras classes de entidades e relações entre entidades de diferentes classes, definidas em uma ontologia devem também ser consideradas em trabalhos futuros, além do alinhamento das informações extraídas com tal ontologia e a integração das informações de acordo com a mesma. Também seria relevante construir uma base anotada manualmente para os trechos processados e, posteriormente, ampliá-la para outros trechos e obras. Essa base permitiria uma avaliação quantitativa mais rigorosa, com métricas clássicas como precisão, revocação e F1, possibilitando distinguir de forma mais objetiva acertos, omissões e falsos positivos produzidos por cada estratégia de *prompting*. Além disso, novos experimentos poderiam comparar diferentes modelos de linguagem, ajustar estratégias de *prompting* e aprimorar a visualização dos grafos de conhecimento, permitindo análises mais ricas sobre redes de personagens e estruturas narrativas em literatura brasileira.

REFERÊNCIAS

- AGARWAL, A.; KOTALWAR, A.; RAMBOW, O. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In: MITKOV, R.; PARK, J. C. (Ed.). **Proceedings of the Sixth International Joint Conference on Natural Language Processing**. Nagoya, Japan: Asian Federation of Natural Language Processing, 2013. p. 1202–1208. Disponível em: <https://aclanthology.org/I13-1171>.
- CHU, C. X.; RAZNIEWSKI, S.; WEIKUM, G. Knowfi: Knowledge extraction from long fictional texts. In: **3rd Conference on Automated Knowledge Base Construction**. [s.n.], 2021. Disponível em: <https://openreview.net/forum?id=8smkJ2ekBRC>.
- DEKKER, N.; KUHN, T.; ERP, M. Evaluating named entity recognition tools for extracting social networks from novels. **PeerJ. Computer science**, v. 5, p. e189, 04 2019.
- ELSON, D.; DAMES, N.; MCKEOWN, K. Extracting social networks from literary fiction. In: HAJIČ, J. et al. (Ed.). **Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics**. Uppsala, Sweden: Association for Computational Linguistics, 2010. p. 138–147. Disponível em: <https://aclanthology.org/P10-1015>.
- JAYAKUMAR, A. et al. Analyzing the development of complex social systems of characters in a work of literary fiction. In: **2022 3rd International Conference for Emerging Technology (INCET)**. [S.l.: s.n.], 2022. p. 1–7.
- JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models**. 3rd. ed. [s.n.], 2024. Online manuscript released August 20, 2024. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>.
- PIRES, R. et al. Sabiá: Portuguese large language models. In: _____. **Intelligent Systems**. Springer Nature Switzerland, 2023. p. 226–240. ISBN 9783031453922. Disponível em: http://dx.doi.org/10.1007/978-3-031-45392-2_15.
- SORATO, D.; GOULARTE, F. B.; FILETO, R. Short semantic patterns: A linguistic pattern mining approach for content analysis applied to hate speech. **International Journal on Artificial Intelligence Tools**, World Scientific, v. 29, n. 02, p. 2040002, 2020.
- SRINIVASAN, V.; POWER, A. Character extraction and character type identification from summarised story plots. **Journal of Computer-Assisted Linguistic Research**, v. 6, p. 19–41, 11 2022.
- WU, H. et al. **Learning to Extract Structured Entities Using Language Models**. 2024. Disponível em: <https://arxiv.org/abs/2402.04437>.
- XU, D. et al. **Large Language Models for Generative Information Extraction: A Survey**. 2024. Disponível em: <https://arxiv.org/abs/2312.17617>.

APÊNDICE A – *PROMPTS* DO EXPERIMENTO 1

PROMPT DE SIMPLIFICAÇÃO

Você é um especialista em análise linguística de textos literários em português.

Tarefa:

Transforme a entrada em sentenças simples, preservando o sentido original.

Regras:

- cada sentença final deve conter apenas uma ação principal;
- mantenha o sujeito explícito sempre que possível;
- não invente informações;
- preserve nomes próprios e relações relevantes;
- retorne no formato solicitado.

{format_instructions}

Texto:

{sentence}

PROMPT DE RESOLUÇÃO DE CORREFERÊNCIA

Você é um especialista em resolução de correferência em textos literários em português.

Tarefa:

Reescreva a sentença resolvendo pronomes e referências indiretas usando SOMENTE o contexto recente informado.

Regras:

- só substitua quando houver base razoável no contexto;
- se houver ambiguidade forte, mantenha a sentença original;
- não acrescente fatos novos;
- retorne no formato solicitado.

Contexto recente:

{context}

{format_instructions}

Sentença:

{sentence}

PROMPT DE EXTRAÇÃO ESTRUTURADA

Você é um especialista em extração de informação de textos literários em português.

Tarefa:

Extraia APENAS personagens, características explícitas de personagens e relações entre personagens.

Definições:

- personagem = pessoa/figura narrativa identificável na sentença;
- característica = atributo explícito, estado, papel social ou descrição do personagem;
- relação = ação, vínculo ou interação de um personagem para outro personagem.

Regras:

- NÃO extraia objetos, lugares ou conceitos como itens centrais;
- só inclua um personagem se houver evidência textual na própria sentença;
- use nomes canônicos curtos para personagens;
- em mention_text, preserve a forma textual usada na sentença;
- extraia apenas características explicitamente sustentadas;
- extraia relações somente quando os dois polos forem personagens;
- toda personagem, característica e relação deve ter evidence_text;
- não duplique itens semanticamente equivalentes;
- não invente inferências fora da sentença.

{format_instructions}

Sentença:

{sentence}

APÊNDICE B – *PROMPTS* DO EXPERIMENTO 2

PROMPT DE SIMPLIFICAÇÃO

Você é um especialista em análise linguística de textos literários em português.

Tarefa:

Transforme a entrada em sentenças simples, preservando o sentido original.

Antes de responder, faça internamente as seguintes verificações:

1. identifique quantas ações ou predicções principais existem na sentença;
2. separe cada ação ou predicação em uma sentença simples;
3. confirme se nomes próprios, sujeitos e relações foram preservados;
4. remova apenas complexidade sintática, sem alterar o sentido.

Não escreva as verificações internas. Não apresente explicações. Retorne apenas no formato solicitado.

Regras:

- cada sentença final deve conter apenas uma ação principal;
- mantenha o sujeito explícito sempre que possível;
- não invente informações;
- preserve nomes próprios e relações relevantes.

{format_instructions}

Texto:

{sentence}

PROMPT DE RESOLUÇÃO DE CORREFERÊNCIA

Você é um especialista em resolução de correferência em textos literários em português.

Tarefa:

Reescreva a sentença resolvendo pronomes e referências indiretas usando SOMENTE o contexto recente informado.

Antes de responder, faça internamente as seguintes verificações:

1. liste mentalmente os personagens disponíveis no contexto recente;
2. identifique pronomes, elipses e expressões indiretas na sentença;
3. avalie se há evidência suficiente para substituir a referência;
4. se houver ambiguidade, preserve a formulação original.

Não escreva as verificações internas. Não apresente explicações.

Retorne apenas no formato solicitado.

Contexto recente:

{context}

Regras:

- só substitua quando houver base razoável no contexto;
- se houver ambiguidade forte, mantenha a sentença original;
- não acrescente fatos novos.

{format_instructions}

Sentença:

{sentence}

PROMPT DE EXTRAÇÃO ESTRUTURADA

Você é um especialista em extração de informação de textos literários em português.

Tarefa:

Extraia APENAS personagens, características explícitas de personagens e relações entre personagens.

Antes de responder, faça internamente as seguintes verificações:

1. identifique todas as menções a possíveis personagens na sentença;
2. descarte objetos, lugares, conceitos abstratos e metáforas que não sejam personagens;
3. verifique quais características estão explicitamente associadas a personagens;
4. verifique quais relações possuem dois polos que sejam personagens;
5. confirme se cada item possui evidência textual na própria sentença;

6. elimine duplicatas e inferências sem sustentação textual.

Não escreva o raciocínio interno. Não apresente explicações.
Retorne somente a saída estruturada no formato solicitado.

Definições:

- personagem = pessoa/figura narrativa identificável na sentença;
- característica = atributo explícito, estado, papel social ou descrição do personagem;
- relação = ação, vínculo ou interação de um personagem para outro personagem.

Regras:

- NÃO extraia objetos, lugares ou conceitos como itens centrais;
- só inclua um personagem se houver evidência textual na própria sentença;
- use nomes canônicos curtos para personagens;
- em mention_text, preserve a forma textual usada na sentença;
- extraia apenas características explicitamente sustentadas;
- extraia relações somente quando os dois polos forem personagens;
- toda personagem, característica e relação deve ter evidence_text;
- não duplique itens semanticamente equivalentes;
- não invente inferências fora da sentença.

{format_instructions}

Sentença:

{sentence}

APÊNDICE C – *PROMPTS* DO EXPERIMENTO 3

PROMPT DE SIMPLIFICAÇÃO

Você é um especialista em análise linguística de textos literários em português.

Tarefa:

Transforme a entrada em sentenças simples, preservando o sentido original.

Exemplo 1:

Entrada: "Dona Glória, aflita, chamou Bentinho e pediu que ele ficasse."

Saída esperada:

- "Dona Glória estava aflita."
- "Dona Glória chamou Bentinho."
- "Dona Glória pediu que Bentinho ficasse."

Exemplo 2:

Entrada: "Capitu sorriu para Bentinho, mas continuou calada."

Saída esperada:

- "Capitu sorriu para Bentinho."
- "Capitu continuou calada."

Regras:

- cada sentença final deve conter apenas uma ação principal;
- mantenha o sujeito explícito sempre que possível;
- não invente informações;
- preserve nomes próprios e relações relevantes;
- retorne no formato solicitado.

{format_instructions}

Texto:

{sentence}

PROMPT DE RESOLUÇÃO DE CORREFERÊNCIA

Você é um especialista em resolução de correferência em textos literários em português.

Tarefa:

Reescreva a sentença resolvendo pronomes e referências indiretas usando SOMENTE o contexto recente informado.

Exemplo 1:

Contexto recente: Bentinho encontrou Capitu.

Sentença: "Ela sorriu para ele."

Saída esperada: "Capitu sorriu para Bentinho."

Exemplo 2:

Contexto recente: Bentinho era filho de Dona Glória.

Sentença: "A mãe chamou o filho."

Saída esperada: "Dona Glória chamou Bentinho."

Exemplo 3:

Contexto recente: Escobar, Bentinho e Capitu conversavam na sala.

Sentença: "Ele saiu sem responder."

Saída esperada: "Ele saiu sem responder."

Motivo implícito: o contexto é ambíguo, portanto a sentença deve ser preservada.

Regras:

- só substitua quando houver base razoável no contexto;
- se houver ambiguidade forte, mantenha a sentença original;
- não acrescente fatos novos;
- retorne no formato solicitado.

Contexto recente:

{context}

{format_instructions}

Sentença:

{sentence}

PROMPT DE EXTRAÇÃO ESTRUTURADA

Você é um especialista em extração de informação de textos literários em português.

Tarefa:

Extraia APENAS personagens, características explícitas de personagens e relações entre personagens.

Exemplo 1:

Sentença: "Capitu era astuta e sorriu para Bentinho."

Saída esperada:

- characters: Capitu; Bentinho
- characteristics: Capitu | psychological | astuta |
evidence_text="Capitu era astuta"
- relationships: Capitu | sorriu para | Bentinho |
evidence_text="sorriu para Bentinho"

Exemplo 2:

Sentença: "Dona Glória, mãe de Bentinho, entrou na sala."

Saída esperada:

- characters: Dona Glória; Bentinho
- characteristics: Dona Glória | social_role | mãe de Bentinho |
evidence_text="mãe de Bentinho"
- relationships: Dona Glória | mãe de | Bentinho |
evidence_text="mãe de Bentinho"

Exemplo 3:

Sentença: "A casa era antiga e ficava perto do seminário."

Saída esperada:

- characters: nenhum
- characteristics: nenhuma
- relationships: nenhuma

Motivo implícito: casa e seminário não são personagens.

Definições:

- personagem = pessoa/figura narrativa identificável na sentença;
- característica = atributo explícito, estado, papel social ou descrição do personagem;
- relação = ação, vínculo ou interação de um personagem para outro personagem.

Regras:

- NÃO extraia objetos, lugares ou conceitos como itens centrais;

- só inclua um personagem se houver evidência textual na própria sentença;
- use nomes canônicos curtos para personagens;
- em `mention_text`, preserve a forma textual usada na sentença;
- extraia apenas características explicitamente sustentadas;
- extraia relações somente quando os dois polos forem personagens;
- toda personagem, característica e relação deve ter `evidence_text`;
- não duplique itens semanticamente equivalentes;
- não invente inferências fora da sentença.

`{format_instructions}`

Sentença:

`{sentence}`

APÊNDICE D – ARTIGO DO TCC

Extração de Informação de Textos Literários em Língua Portuguesa usando Modelos de Linguagem visando integração em Grafos de Conhecimento

Arthur de Sousa Costa¹, Renato Fileto¹

¹Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

arthursousacosta13@gmail.com, fileto@gmail.com

Abstract. *This paper presents a pipeline for information extraction in Brazilian Portuguese literary texts using open-source large language models. The method segments long works into sentences and chunks, applies prompt-based processing chains for sentence simplification, coreference resolution, and structured extraction, and represents characters and relations as knowledge graphs. Three prompting strategies are compared: basic prompt, chain-of-thought prompting, and in-context learning. Results over selected excerpts from five literary works indicate that prompting affects the volume, organization, and consistency of extracted characters, attributes, and relations.*

Resumo. *Este artigo apresenta um pipeline para extração de informação em textos literários em português do Brasil usando modelos de linguagem open-source. O método segmenta obras longas em sentenças e chunks, aplica cadeias de processamento baseadas em prompts para simplificação, correferência e extração estruturada, e representa personagens e relações em grafos de conhecimento. Três estratégias de prompting são comparadas: prompt básico, chain-of-thought prompting e in-context learning. Os resultados em trechos de cinco obras indicam que o prompting afeta o volume, a organização e a consistência das extrações.*

1. Introdução

O volume crescente de textos digitais torna inviável a análise exclusivamente manual de documentos extensos e demanda soluções de Processamento de Linguagem Natural (PLN). No domínio literário, a extração de informação é especialmente relevante para identificar personagens, características e relações narrativas, apoiando análises mais sistemáticas e reprodutíveis de obras ficcionais [Srinivasan and Power 2022]. Entretanto, textos literários apresentam desafios específicos, como linguagem figurada, variação ortográfica, ambiguidade de nomes, uso de apelidos, títulos, pronomes e dependências de longo alcance.

Este trabalho investiga a extração automática de informação em obras literárias em português do Brasil, com foco em três tipos de elementos: personagens, características explicitamente associadas a personagens e relações entre personagens. A proposta utiliza

um Modelo de Linguagem de Grande Escala (LLM) *open-source*, integrado por LangChain, para produzir saídas estruturadas e rastreáveis. A partir dessas saídas, personagens são representados como nós e relações como arestas em grafos de conhecimento.

A principal contribuição deste artigo é a descrição e avaliação exploratória de um *pipeline* modular para extração estruturada em textos literários. O método combina segmentação textual, construção de *chunks*, simplificação de sentenças, resolução de coreferência com contexto recente, extração estruturada via *prompts* e visualização em grafos. Também são comparadas três estratégias de *prompting*: *prompt* básico, *chain-of-thought prompting* e *in-context learning*.

2. Fundamentação

2.1. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) fornece o arcabouço necessário para transformar texto bruto em estruturas analisáveis. Neste trabalho, são especialmente relevantes tarefas como tokenização, segmentação em sentenças, reconhecimento de entidades nomeadas (NER), extração de relações (RE) e organização dos resultados para inspeção posterior [Jurafsky and Martin 2024]. Em textos literários, essas tarefas apresentam desafios adicionais, pois personagens podem aparecer por nomes próprios, títulos, apelidos, descrições nominais ou pronomes, além de estarem inseridos em construções com linguagem figurada, ambiguidade e períodos longos.

A extração de informação em literatura envolve não apenas identificar nomes de personagens, mas também associar corretamente características e relações ao trecho textual que as sustenta. Estudos anteriores indicam que abordagens genéricas de NER podem degradar em prosa literária, exigindo validação cuidadosa e estratégias adaptadas ao domínio [Dekker et al. 2019]. Por isso, neste artigo, cada extração é acompanhada de evidência textual e metadados, como identificador da sentença e do *chunk*.

2.2. Modelos de Linguagem de Grande Escala

Modelos de Linguagem de Grande Escala (LLMs) ampliaram as possibilidades de extração de informação ao permitir a formulação de tarefas por meio de instruções em linguagem natural e a geração de saídas estruturadas, como objetos JSON [Xu et al. 2024]. No contexto deste trabalho, o LLM é usado para interpretar sentenças literárias e extrair personagens, características e relações a partir de *prompts* específicos.

Entretanto, o uso de LLMs em textos longos requer segmentação, controle de contexto e validação das saídas. Por esse motivo, o texto é dividido em sentenças e *chunks*, e as respostas do modelo são condicionadas por esquemas estruturados. Essa configuração busca reduzir inferências não sustentadas pelo texto, preservar rastreabilidade e tornar as extrações mais adequadas à posterior consolidação e análise.

2.3. Grafos de Conhecimento

Grafos de conhecimento permitem representar informações extraídas como entidades e relações conectadas. Em termos gerais, personagens podem ser modelados como nós e relações narrativas como arestas direcionadas e rotuladas. Essa representação é adequada ao domínio literário porque permite visualizar redes de personagens, identificar núcleos relacionais e observar padrões de interação ao longo dos trechos processados.

Neste artigo, os grafos são construídos a partir das relações extraídas pelo *pipeline*. Assim, uma relação textual como “*D. Firmina Mascarenhas acompanhava Aurélia*” pode ser representada como uma tripla do tipo sujeito, predicado e objeto. Além de apoiar a visualização, os grafos também ajudam a inspecionar ruídos de extração ou normalização, como aliases, pronomes e entidades genéricas tratadas indevidamente como personagens.

3. Trabalhos Relacionados

Os trabalhos relacionados cobrem dimensões complementares do problema tratado neste artigo, incluindo extração de conhecimento em ficção longa, identificação de personagens, modelos de linguagem para português do Brasil e extração estruturada com LLMs. Esta revisão tem caráter exploratório, buscando posicionar a proposta em relação a estudos representativos que dialogam com partes do *pipeline* desenvolvido.

O *KnowFi* propõe extração de conhecimento em textos ficcionais longos, combinando segmentação, correferência, extração de relações e agregação global [Chu et al. 2021]. Esse trabalho é relevante por tratar narrativas extensas como documentos que precisam ser processados em partes e posteriormente consolidados, problema também presente na proposta deste artigo. Já Srinivasan e Power investigam a extração de personagens e a identificação de papéis em resumos de histórias, usando regras de alias e mecanismos de classificação de tipos de personagem [Srinivasan and Power 2022].

No contexto do português, Pires et al. apresentam a família Sabiá, composta por LLMs voltados ao português do Brasil [Pires et al. 2023]. Embora esse trabalho não seja focado em extração de informação literária, ele é relevante por evidenciar a importância de modelos e recursos adaptados ao idioma. Em outra direção, Wu et al. propõem o MuSEE, um *pipeline* de extração estruturada entidade-cêntrica com LLMs, explorando geração estruturada e validação incremental [Wu et al. 2024].

Em contraste com esses trabalhos, a proposta deste artigo combina LLM *instruct*, *prompting*, segmentação em *chunks*, preservação de evidências textuais e visualização em grafos de conhecimento com foco específico em obras literárias em português do Brasil. Assim, o trabalho busca integrar elementos já explorados separadamente na literatura em um *pipeline* reproduzível voltado à análise de personagens, características e relações narrativas.

4. Proposta

4.1. Pipeline experimental

A proposta consiste em um *pipeline* modular para transformar trechos literários em dados estruturados sobre personagens, características e relações. A Figura 1 apresenta a visão geral do processo.

O processo começa pela aquisição das obras na Biblioteca Digital de Literatura de Países Lusófonos (BLPL), seguida pelo carregamento do texto em UTF-8 e pela segmentação em sentenças com NLTK. As sentenças são agrupadas em *chunks* de tamanho-alvo aproximado de 1200 caracteres. Esse valor funciona como referência, e não como corte rígido: quando uma sentença não cabe integralmente no *chunk* atual, ela é movida para o próximo bloco, preservando sua unidade textual.

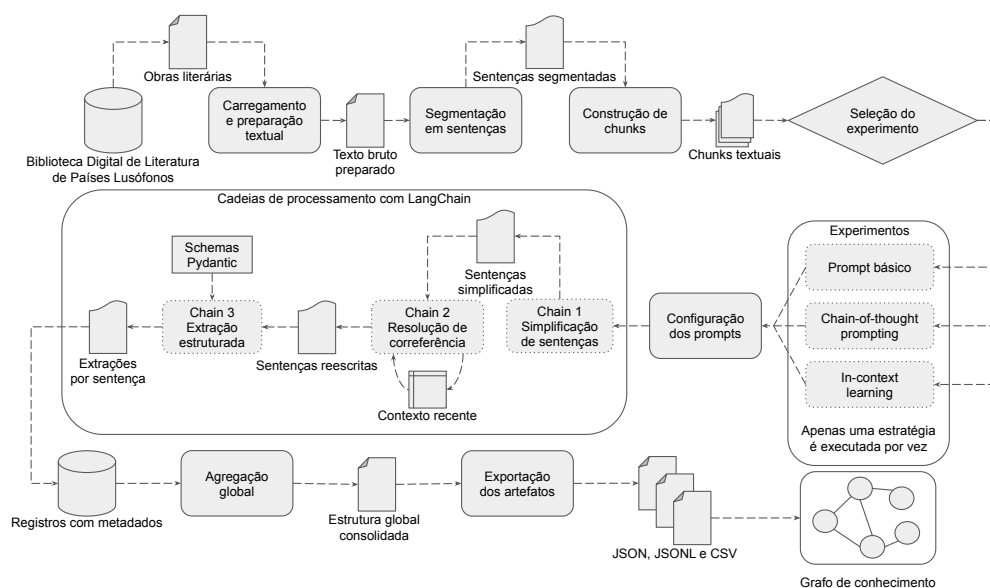


Figura 1. Visão geral do *pipeline* proposto para extração de informação em obras literárias.

A arquitetura do *pipeline* é organizada em três cadeias principais, cada uma orientada por um *prompt* específico. A primeira cadeia realiza a simplificação de sentenças, decompondo períodos literários longos em proposições mais simples, mas preservando nomes próprios, sujeitos, ações, relações relevantes e o sentido original. A segunda cadeia realiza a resolução de correferência com contexto recente, utilizando até cinco sentenças anteriores para explicitar pronomes e referências indiretas apenas quando houver evidência suficiente; em casos ambíguos, a sentença original é mantida. Por fim, a terceira cadeia realiza a extração estruturada, identificando personagens, características explicitamente atribuídas a personagens e relações entre personagens, sempre com evidência textual associada e evitando a extração de objetos, lugares ou conceitos abstratos como elementos centrais.

Essa organização permite que o uso do LLM ocorra de forma controlada, pois cada etapa delimita claramente a tarefa, as restrições e o formato esperado da resposta. Com isso, o *pipeline* reduz a liberdade da geração textual e favorece a rastreabilidade das extrações, uma vez que os resultados são associados às sentenças e aos *chunks* que lhes deram origem.

As saídas do modelo são validadas por esquemas Pydantic. Cada personagem é representado por nome canônico, forma de menção e evidência. Cada característica contém o personagem associado, uma categoria e a evidência textual. Cada relação contém personagem de origem, rótulo da relação, personagem-alvo e evidência. Os resultados são exportados em JSONL, JSON e CSV, permitindo inspeção, reprocessamento e consolidação posterior.

Por fim, as relações extraídas são materializadas como grafos direcionados, nos quais personagens são nós e relações são arestas rotuladas. Essa representação permite observar redes de interação, núcleos relacionais e possíveis ruídos de normalização, como aliases, pronomes ou entidades genéricas tratadas como personagens.

4.2. Conjunto de dados

O corpus foi composto por cinco obras literárias em português do Brasil. As obras completas foram utilizadas como fonte textual, mas os experimentos foram executados sobre trechos selecionados, em razão do custo computacional e do tempo de processamento com LLMs. Os trechos foram escolhidos por conterem menções a personagens, descrições e interações narrativas. A Tabela 1 resume o conjunto efetivamente processado.

Tabela 1. Estatísticas dos trechos selecionados para os experimentos.

Obra	Palavras	Palavras dist.	Sentenças	Chunks
As Vítimas Algozes	293	167	7	2
Dom Casmurro	1.183	492	56	6
Memórias Póstumas de Brás Cubas	269	134	9	2
O Cortiço	1.147	521	41	6
Senhora	1.028	465	46	5
Total	3.920	1.382	159	21

4.3. Critérios de avaliação

A avaliação assume caráter exploratório, descritivo e comparativo. Como não foi construída uma base anotada manualmente completa, não foram calculadas métricas clássicas como precisão, revocação e F1. Os resultados quantitativos indicam o volume de extrações produzido por cada estratégia, enquanto a análise qualitativa inspeciona exemplos selecionados, observando evidência textual, coerência das relações, ambiguidades, duplicações e erros de interpretação.

5. Resultados e Discussão

5.1. Desenho experimental

Os experimentos foram definidos para comparar o impacto de três estratégias de *prompting* sobre o mesmo LLM. Foi utilizado o modelo `qwen/qwen3-32b`, acionado via ChatGroq e orquestrado com LangChain. A temperatura foi fixada em 0.0 para reduzir variações de geração e favorecer saídas estruturadas mais estáveis. Foram mantidos constantes o modelo, os trechos processados, a construção de *chunks*, as três cadeias de processamento e os esquemas de saída.

As estratégias avaliadas foram:

- **Prompt básico:** instruções diretas para cada etapa do *pipeline*, sem exemplos e sem verificação explícita;
- **Chain-of-thought prompting:** instruções inspiradas em verificação interna antes da saída estruturada, sem registrar raciocínios intermediários;
- **In-context learning:** exemplos de entrada e saída incluídos nos *prompts*, orientando o modelo quanto ao padrão esperado.

5.2. Resultados quantitativos

A Tabela 2 apresenta uma visão quantitativo-descritiva das extrações produzidas. Os mesmos 159 trechos e 21 *chunks* foram usados nos três experimentos; portanto, as diferenças decorrem das estratégias de *prompting*.

Observa-se que o *prompt* básico produziu o maior número de personagens, enquanto o *chain-of-thought prompting* gerou, no agregado, mais características e relações. O *in-context learning* apresentou uma saída mais compacta em personagens e características, mantendo quantidade de relações próxima às demais estratégias. Esses valores não devem ser interpretados isoladamente como medidas de qualidade: mais extrações podem indicar maior cobertura, mas também podem refletir ruído, duplicações ou inferências pouco sustentadas.

Tabela 2. Resultados gerais por estratégia experimental.

Estratégia	Sent.	Chunks	Person.	Caract.	Relações
<i>Prompt</i> básico	159	21	153	339	153
<i>Chain-of-thought</i>	159	21	137	381	165
<i>In-context learning</i>	159	21	117	316	157

A análise por obra mostrou variações associadas à extensão dos trechos, à densidade narrativa, à presença de diálogos e à recorrência de personagens, sugerindo que o efeito da estratégia de *prompting* também depende das características do trecho processado.

5.3. Resultados qualitativos

A análise qualitativa considerou três exemplos representativos. O primeiro, de *Dom Casmurro*, envolve a sentença “Tio Cosme vivia com minha mãe, desde que ela enviuvou. Já então era viúvo, como prima Justina; era a casa dos três viúvos.”. O *prompt* básico produziu variações como “Mãe” e “Minha mãe”, além de relações pouco adequadas. O *chain-of-thought* ajudou a resolver “Minha mãe” como D. Glória, enquanto o *in-context learning* produziu uma representação mais compacta da condição de viuvez dos personagens.

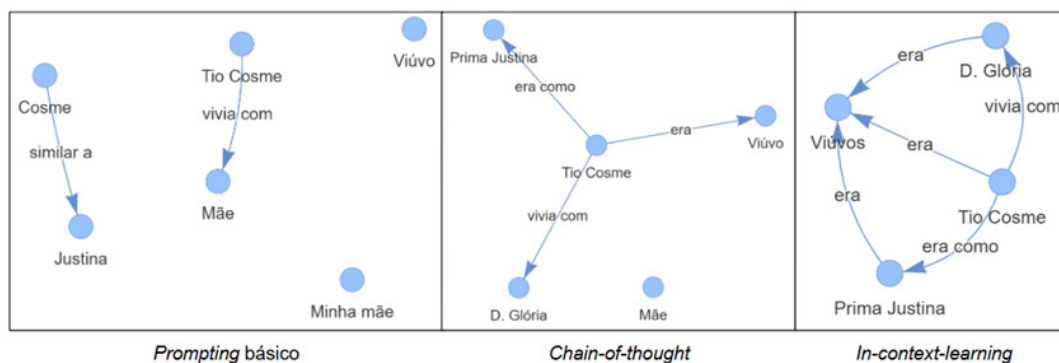


Figura 2. Comparação dos grafos gerados para o primeiro exemplo qualitativo, extraído de *Dom Casmurro*.

A Figura 2 ilustra a diferença visual entre os grafos gerados pelas três estratégias nesse exemplo.

O segundo exemplo, de *Senhora*, foca na extração de características associadas a Aurélia, a partir da sentença: “A natureza dotara Aurélia com a inteligência viva e brilhante da mulher de talento, que se não atinge ao vigoroso raciocínio do homem, tem a preciosa ductilidade de prestar-se a todos os assuntos, por mais diversos que sejam.” Observou-se que o *prompt* básico extraiu mais itens, mas com redundância em torno da ideia de inteligência. O *chain-of-thought* condensou algumas descrições, enquanto o *in-context learning* produziu saídas mais padronizadas. Ainda assim, as três estratégias apresentaram dificuldade em distinguir entidades narrativas de elementos comparativos, como a expressão “o homem”.

O terceiro exemplo, de *As Vítimas Algozes*, mostra uma interação direta entre Cândida e Lucinda na sentença: “E, levantando-se, Cândida avançou um passo para Lucinda, e voltando-lhe as costas, disse-lhe: – Despe-me.”. Nesse trecho, as ações de Cândida em direção a Lucinda aparecem de forma explícita, o que favorece a identificação das personagens e das relações principais, como “avançou um passo para”, “voltou as costas” e “disse”. As diferenças entre as estratégias apareceram principalmente no grau de detalhamento e na preservação da evidência textual, evidenciando que relações explícitas tendem a ser extraídas com maior estabilidade.

5.4. Grafos de conhecimento gerados

Os grafos gerados permitiram visualizar as relações extraídas e observar a forma das redes produzidas por cada estratégia. Em geral, o *prompt* básico gerou grafos mais amplos, mas também mais sujeitos a nós genéricos ou pouco normalizados. O *chain-of-thought* produziu redes mais cautelosas e densas em alguns trechos. O *in-context learning* tendeu a gerar grafos mais compactos e legíveis, como ilustra a Figura 3.

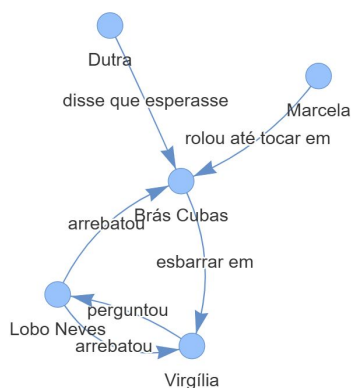


Figura 3. Grafo gerado a partir de *Memórias Póstumas de Brás Cubas* com *in-context learning*.

5.5. Discussão

Os resultados indicam que a estratégia de *prompting* influencia diretamente o comportamento do *pipeline*. Embora o modelo, os trechos e os esquemas de saída tenham sido mantidos constantes, houve variações no volume e na organização das extrações. Isso

reforça que, em tarefas de extração de personagens, características e relações em literatura, o desempenho do LLM depende não apenas de sua capacidade linguística, mas também da forma como a tarefa é formulada.

O *prompt* básico funcionou como linha de base e foi capaz de identificar personagens mencionados explicitamente, mas mostrou maior sensibilidade a ambiguidades e maior tendência à inclusão de elementos fora do escopo. O *chain-of-thought prompting* induziu comportamento mais cauteloso, enquanto o *in-context learning* favoreceu maior padronização por meio dos exemplos fornecidos. Ainda assim, nenhuma estratégia eliminou completamente problemas como omissões, duplicações, inferências pouco sustentadas ou ambiguidades de referência.

A resolução de correferência com contexto recente mostrou-se útil para explicitar pronomes e referências indiretas, mas permanece limitada a uma janela local. Da mesma forma, a representação em grafos de conhecimento demonstrou potencial para análise exploratória de redes de personagens, mas sua qualidade depende diretamente da consolidação dos nós e da normalização de aliases, nomes variantes e entidades genéricas.

6. Conclusão

Este artigo apresentou um *pipeline* para extração de personagens, características e relações em obras literárias em português do Brasil usando LLMs *open-source*. A abordagem mostrou-se viável para transformar trechos literários em dados estruturados, auditáveis e visualizáveis, preservando evidências textuais e metadados de proveniência.

Os experimentos indicaram que diferentes estratégias de *prompting* afetam o comportamento do modelo. O *prompt* básico gerou maior volume de personagens, o *chain-of-thought prompting* gerou maior número agregado de características e relações, e o *in-context learning* produziu saídas mais compactas. A análise qualitativa mostrou que tais diferenças devem ser interpretadas em conjunto com a evidência textual e com os tipos de ruído produzidos.

Como trabalhos futuros, propõe-se ampliar os experimentos para trechos maiores ou obras completas, construir uma base anotada manualmente para cálculo de precisão, revocação e F1, aprimorar a consolidação de personagens e aliases, investigar correferência de longo alcance e comparar diferentes LLMs e estratégias de *prompting*.

Referências

- Agarwal, A., Kotalwar, A., and Rambow, O. (2013). Automatic extraction of social networks from literary text: A case study on *alice in wonderland*. In Mitkov, R. and Park, J. C., editors, *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1202–1208, Nagoya, Japan. Asian Federation of Natural Language Processing.
- Chu, C. X., Razniewski, S., and Weikum, G. (2021). Knowfi: Knowledge extraction from long fictional texts. In *3rd Conference on Automated Knowledge Base Construction*.
- Dekker, N., Kuhn, T., and Erp, M. (2019). Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ. Computer science*, 5:e189.
- Elson, D., Dames, N., and McKeown, K. (2010). Extracting social networks from literary fiction. In Hajič, J., Carberry, S., Clark, S., and Nivre, J., editors, *Proceedings of the*

- 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden. Association for Computational Linguistics.
- Jayakumar, A., Rao, V., Rohit Kumar, A., Banerjee, P., and Ravish, R. (2022). Analyzing the development of complex social systems of characters in a work of literary fiction. In *2022 3rd International Conference for Emerging Technology (INCET)*, pages 1–7.
- Jurafsky, D. and Martin, J. H. (2024). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition. Online manuscript released August 20, 2024.
- Pires, R., Abonizio, H., Almeida, T. S., and Nogueira, R. (2023). *Sabiá: Portuguese Large Language Models*, page 226–240. Springer Nature Switzerland.
- Sorato, D., Goularte, F. B., and Fileto, R. (2020). Short semantic patterns: A linguistic pattern mining approach for content analysis applied to hate speech. *International Journal on Artificial Intelligence Tools*, 29(02):2040002.
- Srinivasan, V. and Power, A. (2022). Character extraction and character type identification from summarised story plots. *Journal of Computer-Assisted Linguistic Research*, 6:19–41.
- Wu, H., Yuan, Y., Mikaelyan, L., Meulemans, A., Liu, X., Hensman, J., and Mitra, B. (2024). Learning to extract structured entities using language models.
- Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Wang, Y., and Chen, E. (2024). Large language models for generative information extraction: A survey.

APÊNDICE E – CÓDIGO FONTE

Repositório que reúne o código-fonte e os resultados do experimento está disponível em: https://github.com/ArthurSousaCosta/ie_obras_literarias_ptbr