

# Engenharia de Prompts para Extração de Eventos e Unidades de Plot

*Pedro Hori Bueno*      *Luiz Fellipe Machi Pereira*  
*Helena de Almeida Maia*      *Sandra Eliza Fontes de Avila*

Relatório Técnico - IC-PFG-24-67  
Projeto Final de Graduação  
2024 - Fevereiro

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Conceitos Fundamentais</b>	<b>4</b>
2.1	<i>Plots</i> . . . . .	4
2.2	<i>Prompts</i> . . . . .	5
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>7</b>
3.1	Extração de Unidades de <i>Plots</i> . . . . .	7
3.2	ManPlts . . . . .	8
<b>4</b>	<b>Metodologia</b>	<b>9</b>
4.1	Extração de <i>Plots</i> . . . . .	10
4.2	Alterações no <i>Prompt</i> . . . . .	11
4.3	Segmentação de Textos Grandes . . . . .	11
<b>5</b>	<b>Resultados</b>	<b>11</b>
5.1	Resultados Iniciais . . . . .	13
5.2	VICUNA-13B vs. WIZARDLM vs. ManPlts . . . . .	14
5.3	<i>Few-Shot</i> vs. <i>Zero-Shot</i> . . . . .	15
5.4	Textos Contendo Diálogos . . . . .	16
5.5	Textos de Entrada Não Convencionais . . . . .	17
<b>6</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>18</b>

# Engenharia de Prompts para Extração de Eventos e Unidades de Plot

Pedro Hori Bueno\*      Luiz Fellipe Machi Pereira\*      Helena de Almeida Maia\*  
Sandra Eliza Fontes de Avila\*

## Resumo

A geração automática de textos é um processo desafiador, devido a dificuldades de modelos em manterem a coesão e coerência durante todo o processo de geração. Para auxiliar este processo de geração, muitos trabalhos utilizam unidades de *plots*, definidas como representações estruturadas controláveis de texto, para auxiliar modelos a manterem a coerência ao longo da história gerada. Devido a sua estrutura padronizada, *plots* são facilmente manipuláveis, podendo ser utilizados na geração de histórias incoerentes. Estas histórias têm o intuito de servir para o treinamento de modelos de classificação de coerência, algo que também auxilia em garantir a coerência de modelos de geração. Neste trabalho, nós apresentamos um novo método de extração de unidades de *plot* a partir de histórias coerentes, utilizando grandes modelos de linguagem (*Large Language Models*, LLMs). Foram estudados o VICUNA-13B, um *chatbot open-source* treinado a partir do modelo LLAMA que consegue gerar textos com qualidade similar a modelos como o GPT-4, e o WIZARDLM, uma LLM treinada especificamente para efetuar instruções complexas. A avaliação dos resultados foi feita de forma qualitativa, considerando a formatação e qualidade das informações presentes nos *plots*, e o método proposto foi comparado com outros métodos convencionais de extração. Os resultados mostram as vantagens de se utilizar LLMs para a extração de *plots*, e são discutidos desafios atuais para o método e como devem ser abordados no futuro.

## 1 Introdução

A coerência é uma das qualidades de textos que indica como uma sequência de sentenças em um texto se conectam de modo que ele tenha sentido como um conjunto. Ela é essencial para que textos gerados mantenham um fluxo lógico de ideias, evitando repetições desnecessárias, tornando-os claros e mais similares a textos escritos por humanos. Manter a coerência na escrita é um grande desafio para os modelos de linguagem gerativos devido principalmente à dificuldade de se determinar e quantificar a coerência de um texto.

---

\*Instituto de Computação, Universidade Estadual de Campinas, 13083-852, Campinas, SP. Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovação, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-Softex, coordenado pela Softex e publicado como Agentes inteligentes para plataformas móveis baseados em tecnologia de Arquitetura Cognitiva (processo 01245.013778/2020-21).

Uma maneira de melhorar a qualidade de histórias geradas por modelos de geração de texto é a partir de modelos de classificação de coerência de histórias. A partir deles, é possível adquirir a capacidade de identificar uma escrita incoerente e evitar reproduzi-la.

Para o treinamento de modelos de classificação são necessários os textos adversários, que propositalmente contêm incoerências em sua escrita. Idealmente, textos adversários devem ser projetados de modo que seja difícil por parte dos modelos identificá-los em meio a textos coerentes, enquanto humanos devem conseguir fazer esta distinção claramente devido ao seu conteúdo sem coerência na escrita. Os textos adversários devem ser pareados com suas respectivas versões coerentes de modo que para cada amostra adversária exista uma amostra coerente equivalente. Assim, a tarefa do modelo classificador será especificamente identificar características no texto que mostrem incoerências textuais.

Neste contexto, Ghazarian et al. [8] propõem a MANPLTS, uma base de dados pareada entre histórias coerentes escritas por humanos e suas respectivas histórias adversárias, que tiveram incoerências inseridas propositalmente em seu processo de geração a partir de histórias coerentes. Utilizando esta base de dados, é possível criar modelos de classificação de coerência de histórias, e, conseqüentemente, melhorar a qualidade de histórias geradas por outros modelos de geração.

Entretanto, a base MANPLTS apresenta alguns problemas para ser utilizada. Nela, a adição de incoerência nas histórias é feita a nível de unidades de *plots*, definidas como representações estruturadas controláveis de fatores utilizadas para a geração de histórias (discutido mais amplamente na [Seção 3](#)). Desta forma, para a geração das histórias adversárias presentes na base é necessária a extração de unidades de *plots* de histórias coerentes, adição de incoerência pela alteração dos *plots* extraídos e a reescrita da história a partir dos *plots* alterados. Ao analisar os *plots* extraídos para a geração das histórias adversárias, observou-se problemas recorrentes de inconsistência nos *plots* e perda de informações relevantes presentes na história, comprometendo futuramente as histórias adversárias geradas.

Desta forma, com o intuito de possibilitar a criação de bases adversárias pareadas a partir de qualquer base de textos coerentes, este trabalho propõe uma maneira alternativa de se extrair *plots* de histórias. Para isso pretende-se utilizar grandes modelos de linguagem (*Large Language Models*, LLMs), que atualmente são amplamente utilizadas para tarefas de extração de informações e sumarização de textos [1, 15, 19, 20, 22], para a tarefa de extração de unidades de *plots* de histórias.

As principais contribuições desse trabalho são:

- Até onde sabemos, este é o primeiro trabalho que propõe a utilização de LLMs para a tarefa de extração de unidades de *plots*.
- Efetuamos a extração das unidades de *plot* de duas bases de dados utilizadas na confecção da base MANPLTS, ROCStories e WritingPrompts, resultando em *plots* qualitativamente melhores com menos perdas de informação das histórias.

- Pelo método proposto, é possível fazer a extração de unidades de *plot* de outras bases de dados, tornando futuramente possível a criação de novas bases adversárias.
- Este trabalho apresenta novas discussões sobre a qualidade de *plots* extraídos, envolvendo *plots* de histórias contendo diálogos e textos escritos com uma estrutura não convencional.

O restante do texto está organizado da seguinte forma. Na [Seção 2](#), são apresentadas definições de conceitos utilizados de maneira ampla no decorrer deste trabalho, como *plots* e *prompts*. Na [Seção 3](#), é apresentada a literatura utilizada para a conexão deste trabalho, referente à geração de histórias utilizando *plots*, extração de unidades de *plot*, e *In-context Learning*. Na [Seção 4](#), é apresentada a metodologia proposta, incluindo o *pipeline* completo para geração de histórias adversárias e o *pipeline* de extração de unidades de *plot* utilizando LLMs. Na [Seção 5](#), são apresentados os resultados obtidos a partir do uso de duas LLMs, o VICUNA-13B e o WIZARDLM, comparados aos *plots* originalmente presentes na base ManPlts. Também é discutido como alterações nos modelos e nos *prompts* utilizados interferem na qualidade dos *plots* extraídos, e quais métricas de avaliação foram utilizadas. Por fim, na [Seção 6](#) são apresentadas as conclusões sobre o trabalho e direções futuras que podem ser abordadas.

## 2 Conceitos Fundamentais

Nesta seção, serão definidos alguns conceitos utilizados neste trabalho que não possuem uma definição única na literatura, como *plot* e *prompt*.

### 2.1 *Plots*

Para o contexto desse trabalho, define-se *unidades de plots*, ou *eventos de plot*, como representações estruturadas controláveis de fatores utilizadas para a geração de histórias, e o *plot de uma história* como sendo um conjunto sequencial de unidades de *plot* que pode ser extraído e utilizado para geração de novas histórias. Uma das principais vantagens em se utilizar unidades de *plot* no processo de geração de histórias é a facilidade de sua manipulação devido à sua estrutura simples e regular, tornando assim possível promover alterações estruturais no processo de geração de histórias visando manter sua corretude gramatical e semântica.

Apesar do amplo uso [2, 9, 17] de unidades de *plots* no contexto de geração de histórias, não existe uma forma fixa definida na literatura para se representar unidades de *plot*. Goldfarb-Tarrant et al. [9] e Fan et al. [7] representam os *plots* utilizando o formato de *Semantic Role Labeling*, que tem como princípio identificar predicados e argumentos de sentenças e atribuir aos argumentos *tokens*, objetos previamente definidos que substituem os argumentos correspondentemente aos seus papéis semânticos nas sentenças, obtendo assim representações homogêneas para os *plots*. Ammanabrolu et al. [2] definem unidades de *plot* como sendo triplas contendo verbo, sujeito e objeto, sendo esta outra representação muito controlada. Já Ghazarian et al. [8] optam por não definir uma estrutura fixa para uma unidade de *plot* podendo ser

compostas por sentenças completas e por uma única palavra, tendo como objetivo somente que o conjunto de unidades de *plots* seja capaz de posteriormente reconstruir uma história.

Para o contexto deste projeto, optamos por uma definição própria mais rígida do que seriam *unidades de plot* ou *eventos de plot*, e o *plot de uma história*, a saber: Um evento de *plot* é um pequeno segmento de um texto contendo palavras chaves (sujeito, verbo, objeto e adjetivos) que proporcionam suas informações principais, assim sendo possível reconstruir o texto original a partir de seus respectivos eventos de *plot*. Esta definição foi utilizada tanto para a construção dos *prompts* utilizados para a extração de unidades de *plot* quanto para a avaliação da qualidade dos *plots* gerados. A respeito da estrutura dos *plots*, optamos por utilizar duas barras verticais “|” para separação das unidades de *plots*. Esta escolha foi feita pois duas barras verticais não aparecem naturalmente nas histórias, evitando assim que uma unidade de *plot* fosse confundida com algum elemento já presente na história. Na [Figura 1](#), apresentamos um exemplo de *plot* extraído de uma história presente na base ROCStories.

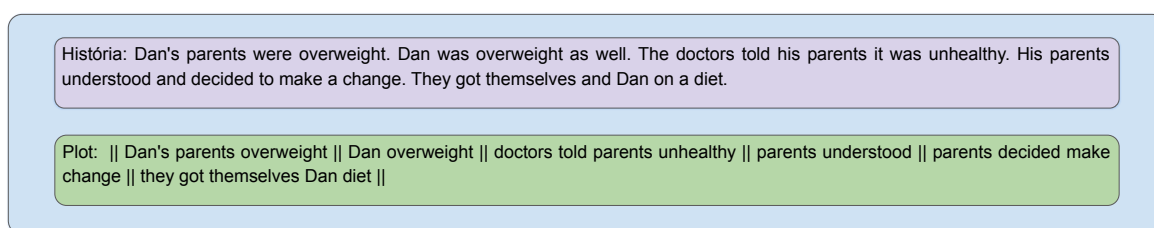


Figura 1: Um par de uma história com seu respectivo *plot* extraído.

## 2.2 Prompts

Modelos de IA gerativos utilizam primariamente textos como entrada de modo que o usuário possa definir e explicar ao modelo qual é a saída desejada. Este texto de entrada fornecido ao modelo é chamado de *prompt*, e o processo de alterar este *prompt* de modo à otimizar o resultado na saída do modelo para uma determinada tarefa é chamado de *Engenharia de Prompt*.

Para os modelos de linguagem de larga escala (ou LLMs, *Large Language Models*), *prompts* podem ser tanto perguntas simples e diretas, quanto descrições de tarefas complexas para o modelo executar. No contexto deste trabalho, o aprendizado baseado em *prompt*, também chamado de *in-context learning* [4], foi utilizado para a tarefa de extração de *plots* a partir de histórias (como está descrito na [Seção 4](#)). Desta forma, separamos e definimos os elementos presentes em um *prompt* em quatro categorias conforme a [Figura 2](#), sendo elas: Contexto, Exemplos, Tarefa, e Entrada e Saída.

No *contexto do prompt*, devem estar descritas as informações necessárias para o modelo entender a tarefa que deve ser executada, assim como definir como deve ser o formato da saída do modelo. Nos *exemplos do prompt* podem ser apresentados ao modelo pares de entradas e saída corretas, mostrando assim o que se espera para a realização da tarefa. A *tarefa do prompt* define exatamente o que o modelo deve fazer e retornar ao usuário. Por fim, a seção de *entrada e*

saída do prompt é onde o texto de entrada que é fornecido ao *template* de *prompt* deve ser inserido, e onde o modelo deve escrever sua saída.

Com exceção das seções de *Tarefa* e *Entrada e Saída*, que devem estar presentes em qualquer *prompt*, os outros elementos presentes são opcionais, sendo assim possível construir *prompts* funcionais sem sua presença. *Prompts* que não apresentam nenhum exemplo são classificados como *prompts zero-shot*, enquanto *prompts* que apresentam uma quantidade pequena de exemplos são classificados como *prompts few-shot* [4].

**1**

A Plot Event is a small segment of the Passage that contains key words (containing subject, verb, object and adjective) that provide the main information for it's understanding, so someone could reconstruct the original sentence using the plot events.

You are a highly intelligent and accurate Plot Event extractor model. You take a passage as input and convert it into a series of Plots Events.Do not miss any part of the input.

Your output format is only { ("|| Plot unity || Plot unity || ...") }, nothing else.

**2**

Examples:

Input: The living room was too dark. The wife said she wanted to add some windows. They hired a contractor to come in and add a wall of windows. Soon the room was full of natural sunlight. They were so happy with the change!  
Output: || Living room was dark || wife said wanted add windows || They hired contractor || Room soon was full of sunlight || They were happy ||

Input: Oliver was playing in his bedroom with his parents. The two parents were wrapped up in a heated discussion. No one wanted to pay attention to Oliver or his toys. He looked at his parents and said ""mom, dad"". Being his first word, his parents freaked out and celebrated.  
Output: || Oliver playing his bedroom || Parents wrapped heated discussion || No one pay attention Oliver || He looked at his parents || He said "mom,dad" || his first word parents celebrated ||

Input: tobias planned a big camping trip for the weekend . he asked his wife if she wanted to come , but she refused . so he ended up going there alone . while unloading his gear , he realized he forgot to pack his tent . he went back home deflated , but his wife was delighted to be with him .  
Output: || tobias planned big camping trip || he asked his wife wanted come || she refused to come || he going alone || he unloading gear || he realized forgot pack tent || he back home deflated || his wife delighted with him ||

**3**

Now, extract the plot Events from the input and write them in the output:

**4**

Input: {Entrada}  
Output:

Figura 2: *Template* de *prompt* utilizado. 1) Contexto da tarefa, 2) Exemplos, 3) Tarefa, e 4) Entrada e Saída.

### 3 Trabalhos Relacionados

Nesta seção, é feito um breve levantamento de trabalhos mais relevantes utilizados durante a confecção do projeto.

#### 3.1 Extração de Unidades de *Plots*

Uma série de trabalhos presentes na literatura mostram como a utilização de *plots* para guiar o processo de geração de histórias é benéfica para a qualidade da história gerada [2, 9, 17]. Uma etapa fundamental para empregar essa estratégia é a extração das unidades de *plot*.

Um método proposto para a extração de unidades de *plot* a partir de histórias é o *Causal Commonsense Plot Ordering* (C2PO [2]). Os autores definem o problema de extração de unidades de *plot* como a tarefa de gerar uma sequência de eventos com uma progressão causal, *i.e.*, de forma que cada evento seja consequência dos eventos vistos anteriormente. O modelo é baseado em uma relação também proposta pelos autores no artigo, *soft causal relation*, que assume que a ordem dos eventos precisa ser lógica na perspectiva do leitor e não do autor. Para isso, eles incorporam informação de senso comum no sistema. Isso é feito através do uso de um modelo chamado COMET [3] (*Commonsense Transformers for Automatic Knowledge Graph Construction*) que faz inferências de senso comum usando relações do tipo *if-then* (se-então). O modelo C2PO [2] começa com o processo de extração de unidades de *plot* a partir do alinhamento de sujeitos, das relações entre sujeitos e objetos e do texto inicial. A partir das unidades de *plots*, o método faz um processo de *plot-infilling* usando a relação *soft*, para encontrar a melhor sequência de eventos para construção de uma história.

Yao et al. [23] propõem um método de extração e geração de *storylines*, uma estrutura similar a *plots* que pode ser utilizada como esqueleto para geração de histórias. Este método consiste em extrair a palavra de maior importância de cada sentença, calculando essa importância a partir de métricas baseadas em grafos e frequência de palavras [14]. A partir da sequência de palavras de maior importância extraídas é formada a estrutura similar ao *plot* de uma história. Esta estratégia é utilizada pela base ManPlTs para realizar a extração dos *plots* de histórias da base ROCStories.

Fan et al. [7], por sua vez, utilizam *Semantic Role Labeling* (SLR) para decompor uma história em formas estruturadas de representação, similares a unidades de *plot*. O SRL consiste em identificar os predicados e argumentos de sentenças [10, 16] e atribuir a cada argumento um rótulo semântico para representá-lo. Cada conjunto de predicado e argumento é considerada uma ação, e a sequência de todas as ações separadas por *tokens* de delimitação é utilizada como *plot* para a geração de uma história, sendo cada ação uma unidade de *plot*. Esta é uma estrutura bem definida para os *plots* focada nos verbos das sentenças identificados em seus predicados. Entretanto, esta maneira de representação apresenta desvantagens ao não ser capaz de representar sentenças curtas que não contenham verbos ou sentenças longas com mais de um verbo presente. Esta estratégia é utilizada pela base ManPlTs para realizar a extração dos *plots* de histórias da base WritingPrompts.



Uma abordagem de extração de informação de texto encontrada na literatura é feita a partir do uso de *prompts* [1, 15, 19, 20, 22]. O *prompt* é utilizado como entrada juntamente com o texto. O aprendizado baseado em *prompt*, também chamado de *in-context learning*, permite a adaptação de modelos pré-treinados para contextos específicos com amostras não rotuladas, e é particularmente interessante para cenários *zero-shot* e *few-shot* [1]. Além de fornecer informação das novas tarefas (*task-related instructions* [22]), os *prompts* podem indicar formatos de saída bem estruturada e padronizada, servindo como guia para a extração de informação [1] como pode ser visto na Figura 3.

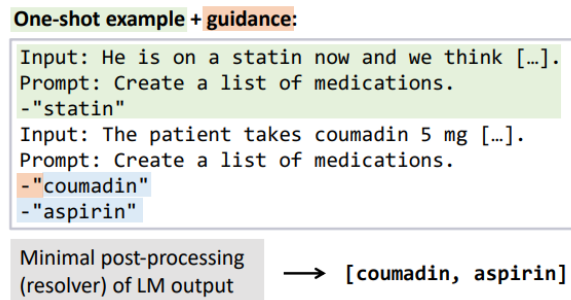


Figura 3: Exemplo de extração de informação a partir de *prompts*. Note que o exemplo na cor verde indica o formato esperado na tarefa específica. Figura extraída de Agrawal et al. [1].

No contexto do projeto, a estratégia pode ser empregada para a extração de *plots* através de exemplos de entrada e de *templates* bem elaborados. A Figura 4a traz um exemplo de *prompt* para essa tarefa, e a Figura 4b mostra a saída esperada para uma determinada entrada.

### 3.2 ManPlts

O trabalho proposto por Ghazarian et al. [8] apresenta o ManPlts, um conjunto de dados adversárias com exemplos de histórias implausíveis de alta qualidade. O conjunto foi originalmente proposto para avaliação de modelos de geração de histórias. Os autores argumentam que outros conjuntos similares da literatura, elaborados através da combinação de exemplos positivos e negativos de textos, tendem a apresentar dados menos naturais e com características muito simplificadas. Visando aumentar a qualidade dos exemplos, o conjunto ManPlts foi criado a partir da manipulação de unidades de *plots*.

Primeiramente, os autores selecionaram dois conjuntos de dados na língua inglesa, com diferenças significativas em termos de extensão e tópico, como bases para a criação de histórias contraditórias: (1) ROCStories, composto por histórias de senso comum de cinco frases recolhidas através de *crowd-sourcing* [12], e (2) WritingPrompts, composto por histórias longas escritas por humanos recolhidas de um fórum online [6]. Em seguida, utilizaram diferentes técnicas, propostas por Yao et al. [23] e Fan et al. [7], para extrair unidades de *plots* dos conjuntos de dados ROCStories e WritingPrompts, respectivamente. Para cada unidade de *plot*, foram aplicadas duas a quatro técnicas diferentes de manipulação aleatória, considerando as seguintes opções: *Non-logically Ordered Plots*, em que a ordem correta dos conceitos no *plot* é invertida; *Contra-*

## Prompt

Extract the plot unit of each sentence (containing verb, subject, object and adjective) by separating them with #. Examples:

Q: The living room was too dark. The wife said she wanted to add some windows. They hired a contractor to come in and add a wall of windows. Soon the room was full of natural sunlight. They were so happy with the change!

A: Living room was dark # wife said wanted add windows # They hired contractor # Room soon was full of sunlight # They were happy

Q: Oliver was playing in his bedroom with his parents. The two parents were wrapped up in a heated discussion. No one wanted to pay attention to Oliver or his toys. He looked at his parents and said ""mom, dad"". Being his first word, his parents freaked out and celebrated.

A: Oliver was playing in his bedroom # Parents wrapped heated discussion # No one pay attention Oliver # He looked at his parents # He said "mom,dad" # his first word parents celebrated

Q: tobias planned a big camping trip for the weekend . he asked his wife if she wanted to come , but she refused . so he ended up going there alone . while unloading his gear , he realized he forgot to pack his tent . he went back home deflated , but his wife was delighted to be with him .

A: tobias planned big camping trip # he asked his wife wanted come # she refused # he going alone # he unloading gear realized forgot tent # he back home deflated # his wife delighted with him

(a) Exemplo para extração de *plots*.

## Sample test

Q: betty went out of her way to be a good employee . she tirelessly helped customers day in and day out . one day her supervisor came to her and fired her . betty was so upset she took her 401k and cashed out . betty bought out her old employer and fired her old boss .

A:

## Result

Betty went out of her way to be a good employee # She tirelessly helped customers day in and day out # Her supervisor came to her # Supervisor fired her # Betty was upset # she took her 401k and cashed out # Betty bought out her old employer # Betty fired her old boss.

(b) *Plots* extraídos a partir do exemplo.

Figura 4: Extração de *plots* a partir de *prompts*.

*diction Insertion*, em que elementos contraditórios são inseridos como *plots* vizinhos; *Repetition Insertion*, em que *plots* aleatórios de cada texto em várias posições são repetidos; e *Random Substitution*, em que parte dos *plots* são substituídos por frases aleatórias.

Após a criação dos *plots* manipulados, os autores ajustaram um modelo BART [11] em pares de *plots* e histórias originais a partir de dados do ROCStories e do WritingPrompts, e depois aproveitaram o modelo para gerar textos implausíveis nos *plots* manipulados. Por fim, aplicaram uma técnica de filtragem adversarial, proposta por Zellers et al. [24], em seis histórias implausíveis geradas, para selecionar as três mais desafiadoras.

Embora a base tenha sido originalmente proposta para avaliar modelos, ela também pode ser útil para aumentar a robustez dos modelos e melhorar a qualidade das histórias geradas. Neste trabalho, ela será considerada para essa segunda finalidade.

## 4 Metodologia

Nesta seção, é descrita a metodologia utilizada no trabalho. Esta descrição inclui tanto o *pipeline* mais abrangente do projeto de confecção de uma base de dados adversarial para geração de

histórias, quanto a metodologia utilizada especificamente no processo extração de unidades de *plot*, sendo este o foco principal da pesquisa.

#### 4.1 Extração de *Plots*

Tendo como objetivo a criação de um modelo de geração de histórias coerentes, encontramos a necessidade de uma base de dados contendo histórias adversárias (incoerentes). A partir de modelos de classificação de coerência de histórias, é possível melhorar a qualidade de histórias geradas por modelos de geração. Para isto, são necessárias histórias adversárias pareadas e possuindo uma boa qualidade, sendo propositalmente difícil para modelos de classificação distinguir entre as histórias coerentes e incoerentes.

A base de dados *ManPlts* [8] é uma boa opção para este contexto. Ela apresenta um bom conjunto de histórias incoerentes, criadas pela inserção de incoerências nos *plots* de histórias coerentes, de maneira a propositalmente imitar erros normalmente cometidos por modelos de geração. Em sequência, é selecionado o conjunto de histórias mais desafiador para modelos de classificação utilizando um filtro adversarial.

Entretanto, ao utilizar a base de dados *ManPlts* foram identificados algumas dificuldades. Primeiramente, foi observada uma falta de reprodutibilidade nos processos de extração de *plots* e geração de histórias a partir dos *plots* extraídos. Também observou-se problemas recorrentes de inconsistência na estrutura dos *plots* e perda de informações relevantes presentes na história, comprometendo futuramente as histórias adversárias geradas.

Devido a problemas de qualidade e falta de reprodutibilidade do método de obtenção de histórias adversárias feito para o conjunto de dados *ManPlts*, optamos por criar um novo *pipeline* para geração de histórias adversárias, como pode ser visto na [Figura 5](#). Desta forma, seria possível aplicar este novo método em novas bases de dados, podendo gerar assim novos conjuntos de histórias adversárias que podem ser utilizados para o treinamento de modelos de classificação de texto.

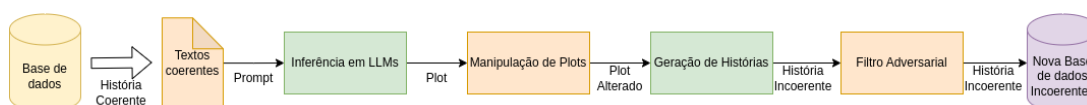


Figura 5: *Pipeline* para geração de histórias.

Este trabalho aborda a etapa de inferência em LLMs presente na [Figura 5](#). Ao contrário do que foi originalmente feito na base *ManPlts*, neste novo *pipeline* de geração de histórias adversárias utilizamos LLMs para o processo de extração de unidades de *plots*. Para isso, foram desenvolvidos *prompts* como entrada das LLMs que especificavam a tarefa que deveria ser realizada, e qual era a formatação de saída desejada.

## 4.2 Alterações no Prompt

Para a tarefa de extração de unidades de *plots* a partir de LLMs foi utilizado o *pipeline* descrito na Figura 6. Nele, uma história é concatenada a um *template* de *prompt*, gerando assim o texto final que será fornecido a uma LLM. A LLM deve seguir as instruções fornecidas pelo *prompt* de entrada e extrair as unidades de *plots* correspondentes à história. Para se obter melhores resultados, foram feitas alterações em dois componentes do *pipeline* inicial, sendo eles o *template* de *prompt* e a LLM utilizada. Para as LLMs, foram estudados o VICUNA-13B [5], um *chatbot open-source* treinado a partir do modelo LLAMA [18] que consegue gerar textos com qualidade similar a modelos como o GPT-4, e o WIZARDLM [21], uma LLM treinada especificamente para efetuar instruções complexas.

Com o auxílio de ferramentas de geração de *prompts* [13] e trabalhos que utilizam *in-context-learning* para diferentes tarefas [4], foram desenvolvidos novos *templates* de *prompt*. Assim, efetuamos diversos experimentos alterando o contexto, exemplos e tarefa, como foi descrito na Figura 2.

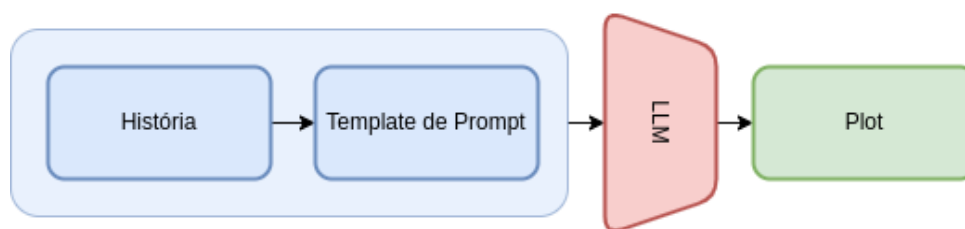


Figura 6: Pipeline para geração de *plots*.

## 4.3 Segmentação de Textos Grandes

As histórias presentes na base de dados WritingPrompts apresentam dificuldades aos modelos de extração de *plots* devido à sua característica de serem textos em formato livre sem nenhuma padronização, e, em sua grande maioria, consideravelmente mais longas. Para tratarmos especificamente de tais problemas, que serão discutidos mais profundamente na Seção 5, foram feitas alterações ao *pipeline* inicial de extração de *plots*, conforme a Figura 7. Foram implementadas duas novas camadas ao *pipeline*, um segmentador e um concatenador, responsáveis respectivamente por segmentar as histórias iniciais da base em segmentos menores visando mantê-los em tamanhos próximos aos textos presentes na base ROCStories, e por juntar os *plots* correspondentes aos segmentos de história em um *plot* completo.

## 5 Resultados

Nesta seção, são apresentados os principais resultados obtidos pelo método de extração de unidades de *plot* proposto na Seção 4. Também são feitas comparações dos resultados obtidos com os *plots* originais fornecidos na base MANPLTS, e apontadas as principais vantagens, desvantagens e limitações observadas no método proposto.

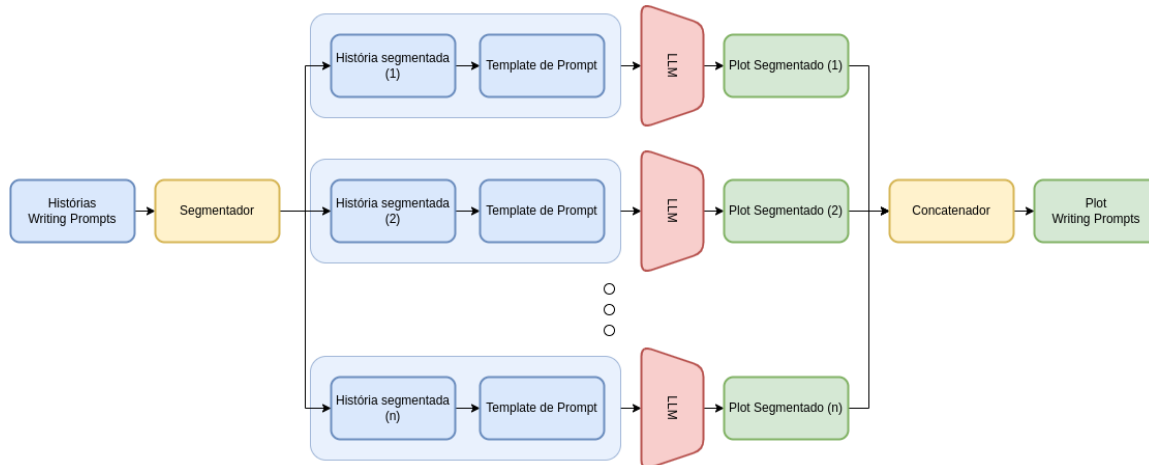


Figura 7: Pipeline para geração de plots para a base WritingPrompts.

Devido à ausência de métricas quantitativas para efetuar uma avaliação da qualidade dos *plots* extraídos, foram feitas análises qualitativas manualmente de uma amostragem aleatória das histórias. Nestas análises, os seguintes critérios foram observados:

1. O *plot* extraído correspondeu com a formatação desejada, sendo ela: `|| Plot unity || Plot unity || ... ||` para as LLMs, e `Plot unity # Plot unity # ...` para a ManPlts, e nenhuma outra informação?
2. As unidades de *plot* extraídas correspondem com a estrutura definida para uma unidade de *plot*, sendo ela: *Palavras-chaves referentes ao sujeito, verbo, objeto e adjetivos das sentenças?*
3. Toda informação relevante para o entendimento da história original está refletida no seu *plot*?
4. O *plot* contém informações que não estão presentes na história original?
5. A ordem das unidades de *plot* corresponde com a ordem em que os eventos acontecem na história?

Erros que ocorrem no primeiro critério são considerados erros de formatação e indicam que o modelo não foi capaz de efetuar a tarefa desejada e não retornou informações que podem ser utilizadas como unidades de *plot*. Os demais critérios se referem a qualidade do *plot*, e somente são avaliados caso o *plot* apresente a formatação correta.

Os critérios 4 e 5 são comuns em casos de alucinação de LLMs. Alucinações ocorrem quando o modelo escreve informações novas que não foram requisitadas pelo *prompt* de entrada ou quando continuam escrevendo desrespeitando o critério de parada especificado no *prompt*. Para nosso contexto, alucinações são quaisquer textos na saída das LLMs que não pertençam ao *plot* correto da história.

## 5.1 Resultados Iniciais

Os primeiros experimentos foram realizados em uma configuração mais simples, utilizando somente a LLM VICUNA-13B e alterando somente o *prompt* de entrada do modelo.

Primeiramente, foi estruturado um *prompt* contendo uma breve explicação do que seria uma unidade de *plot*, seguido por três (3) exemplos de histórias coerentes presentes na base da ROCStories [12] com seus respectivos *plots* extraídos. Por fim, foi requisitado ao modelo que, baseando-se nos exemplos fornecidos no *prompt*, extraísse os *plots* de uma nova história que seria fornecida como entrada do processo.

Ao realizar testes referentes à extração de *plots*, foi observado uma série de problemas. Um dos problemas mais recorrentes era o modelo não parar de gerar texto após realizar a extração de *plot* da história requisitada, gerando assim informações longas e não requisitadas no *prompt* fornecido. Também havia uma clara inconstância na estrutura dos *plots* gerados, de modo que o modelo escolhia uma maneira distinta de escrever o *plot*.

Desta forma, foi realizada uma série de experimentos alterando o *prompt* fornecido ao modelo e seus parâmetros internos. Percebeu-se que era necessário explicitar o formato de saída do modelo na descrição da tarefa, assim como determinar a parada de escrita após o *plot* ser extraído. Também foi observado uma relação entre o número de exemplos fornecidos ao modelo e seu resultado final. Para poucos ou nenhum exemplo, o modelo tinha dificuldade em entender a tarefa requisitada, efetuando somente uma segmentação simples do texto. Entretanto, conforme o número de exemplos aumentava, a complexidade do *prompt* também aumentava, fazendo com que informações sobre o formato da saída desejada fosse desconsiderado para algumas histórias.

Por fim, utilizando a versão final do *prompt* que obteve o melhor desempenho nos testes, foi feita a extração de *plots* para a base ROCStories, como pode ser visto nos exemplos das Figura 8, Figura 9 e Figura 10. A partir de uma análise qualitativa foi observada uma melhora da qualidade dos *plots* extraídos utilizando o VICUNA-13B em relação aos *plots* utilizados na base ManPlts. Problemas como perda de informações das histórias nos *plots*, ou adição nos *plots* de partes inexistentes nas histórias originais foram mitigados, garantindo assim uma maior confiabilidade no processo subsequente de geração de histórias a partir dos *plots* extraídos.

### Exemplo 1.

**História:** Dan's parents were overweight. Dan was overweight as well. The doctors told his parents it was unhealthy. His parents understood and decided to make a change. They got themselves and Dan on a diet.

**Plot:** || Dan's parents overweight || Dan overweight as well || Doctors told parents unhealthy || Parents understood decided make change || They got themselves Dan diet ||

Figura 8: Exemplo 1.

**Exemplo 2.**

**História:** Carrie had just learned how to ride a bike. She didn't have a bike of her own. Carrie would sneak rides on her sister's bike. She got nervous on a hill and crashed into a wall. The bike frame bent and Carrie got a deep gash on her leg.

**Plot:** | | Carrie learned ride bike | | didn't have own bike | | sneak rides sister's bike | | nervous hill crashed wall | | bike frame bent | | deep gash leg | |

Figura 9: Exemplo 2.

**Exemplo 3.**

**História:** Story: Morgan enjoyed long walks on the beach. She and her boyfriend decided to go for a long walk. After walking for over a mile, something happened. Morgan decided to propose to her boyfriend. Her boyfriend was upset he didn't propose to her first.

**Plot:** | | Morgan enjoyed long walks beach | | she boyfriend decided go long walk | | After walking mile something happened | | Morgan decided propose boyfriend | | He upset didn't propose first | |

Figura 10: Exemplo 3.

## 5.2 VICUNA-13B vs. WIZARDLM vs. ManPlts

A [Figura 11](#) apresenta duas histórias presentes na base ROCStories e, subsequentemente, apresenta três extrações de *plots* efetuadas para cada história. As primeiras extrações mostradas abaixo das histórias foram feitas utilizando o WIZARDLM [21], as segundas foram feitas utilizando o VICUNA-13B [5], e as terceiras são referentes às extrações originais dos *plots* fornecidos em Ghazarian et al. [8]

Primeiramente, foi possível observar uma melhora significativa dos modelos utilizando LLMs, tanto o VICUNA-13B quanto o WIZARDLM, em relação ao método inicial utilizado pela ManPlts, para todos os critérios avaliativos, exceto o primeiro, tendo em vista que a estrutura dos *plots* se mantém constante para toda a base. O método de extração de *plots* utilizado pela base ManPlts não define claramente uma estrutura para as unidades de *plots* individuais, gerando assim muitos erros em sua extração. Como pode ser visto na [Figura 11](#), algumas unidades de *plots* são compostas por apenas uma palavra, perdendo assim o significado original da história. Foi observado também que as histórias geradas a partir destes *plots* contendo lacunas da história original tendem a apresentar novas informações para completar a história, gerando assim textos pouco semelhantes ao seu original.

Comparando a extração efetuada utilizando o VICUNA-13B e o WIZARDLM para histórias da base ROCStories, é possível observar uma melhor qualidade nos *plots* extraídos a partir do WIZARDLM em seguir a estrutura definida para uma unidade de *plot*, gerando de maneira mais constante *plots* contendo sujeito, verbo, objeto e adjetivos.



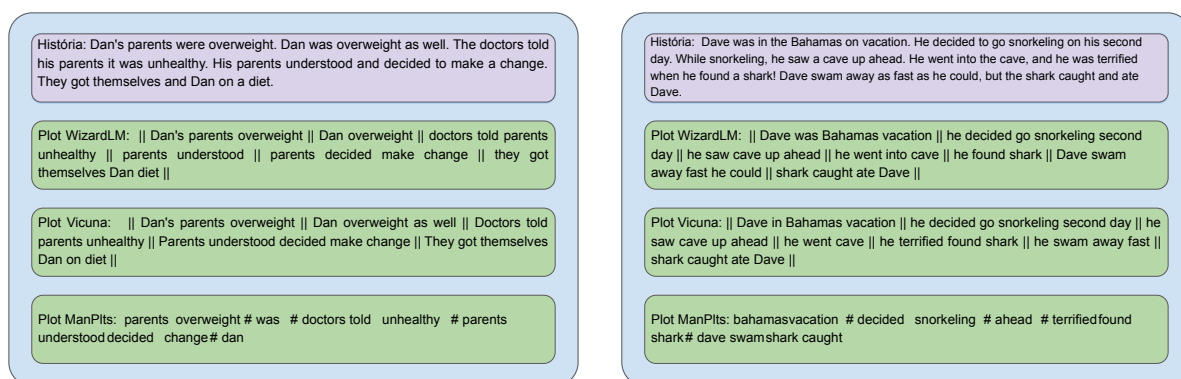


Figura 11: *Plots* extraídos de duas histórias da ROCStories por diferentes métodos.

Para histórias provenientes da base *WritingPrompts*, é possível observar de maneira ainda mais clara a melhora da qualidade das unidades de *plots* extraídas utilizando LLMs em relação ao método utilizado pela *ManPlts*. Como pode ser observado na [Figura 12](#), os *plots* extraídos pela *ManPlts* não apresentam uma estrutura padronizada de *plots* e apresentam uma grande perda de informações relevantes da história. Os *plots* extraídos utilizando o *WizardLM* mantêm a formatação de saída desejada e a estrutura dos *plots*. Apesar de ainda ocorrerem problemas de perda de informação, para as amostras avaliadas esta perda foi pequena, apresentando um resultado final muito superior a *ManPlts*.

Os *plots* extraídos pelo *VICUNA-13B* também apresentaram menos perda de informação em comparação com a *ManPlts*. Entretanto, foram observados muitos problemas de formatação e alucinação do modelo. Os *plots* apresentaram textos que não estavam presentes na história original e muitas vezes o modelo continuava gerando textos sem respeitar o critério de parada determinado, fazendo com que a formatação desejada não fosse respeitada.

### 5.3 *Few-Shot vs. Zero-Shot*

Foram feitos diversos experimentos relacionando o número de amostras passadas como exemplos no *prompt* e a qualidade dos *plots* extraídos. São considerados *prompts zero-shot* aqueles que não possuem nenhum exemplo da tarefa, e são considerados *prompts few-shot* aqueles com um ou mais exemplos.

Com os experimentos realizados com história de ambas bases *ROCStories* e *WritingPrompts*, foi possível perceber que *prompts few-shot* apresentaram resultados muito superiores aos *prompts zero-shot*. Com a ausência dos exemplos, os modelos não possuíam uma compreensão clara da tarefa a ser realizada e conseqüentemente geravam erros estruturais graves em suas respostas, como não extrair nenhuma unidade de *plot* e escrever sentenças sem nenhuma relação com a tarefa desejada.

Também foram feitos experimentos relacionando a quantidade de exemplos fornecidos nos



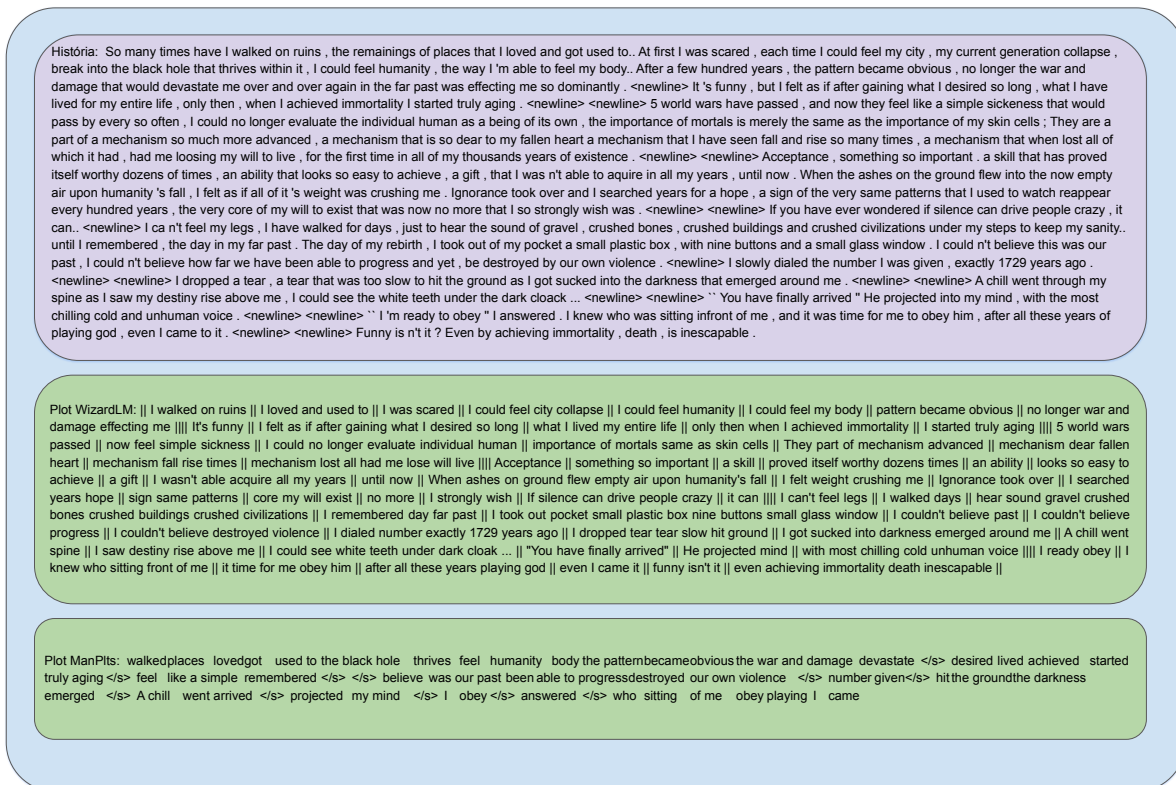


Figura 12: *Plots* extraídos da WritingPrompts por diferentes métodos.

*prompts few-shot* e a qualidade dos *plots* extraídos. Foi observado que o modelo apresentava uma melhor compreensão do resultado desejado conforme o número de exemplos fornecidos no *prompt* aumentava, apresentando saídas mais constantes. Entretanto, um número muito elevado de exemplos, e consequentemente grandes *prompts*, geraram perdas de informação do contexto da tarefa fornecida na parte inicial do *prompt*, fazendo com que a definição de elementos de uma unidade de *plot* e a formatação de saída desejada fossem descumpridas em uma maior frequência.

A partir de análises qualitativas dos *plots* extraídos por cada método e a quantidade de erros estruturais nas saídas foi definido que para ambas as bases o número ideal de exemplos fornecidos nos *prompts* foi três.

## 5.4 Textos Contendo Diálogos

A partir de análises de *plots* extraídos de textos da base WritingPrompts, foram observadas limitações por parte dos modelos em lidar com estruturas de escrita mais diversas, que não ocorriam na base ROCStories por esta possuir uma estrutura padronizada. Os modelos apresentaram dificuldade em gerar unidades de *plot* em textos contendo diálogos entre dois ou mais personagens. A Figura 13 apresenta três exemplos de unidades de *plot* extraídas de textos contendo diálogos e é possível observar que não existe uma consistência na maneira que diálogos são convertidos em unidades de *plot*. No *Texto 1* da figura, o modelo foi capaz de adicionar

às unidades de *plot* informações de quem estava falando no diálogo e manter o sentido central das falas, tornando assim possível a compreensão do conteúdo e sentido do diálogo a partir de seu respectivo *plot*. O *Texto 2*, entretanto, não manteve o conteúdo do diálogo no *plot*, gerando assim uma perda de informação e prejudicando o entendimento da história. O *Texto 3* apresenta outro problema, em que o conteúdo do diálogo foi mantido no *plot*, porém a informação de quais *plots* eram partes de diálogos e quais personagens realizavam cada fala foi perdida. Isto torna praticamente impossível a compreensão do texto a partir de suas unidades de *plot*.

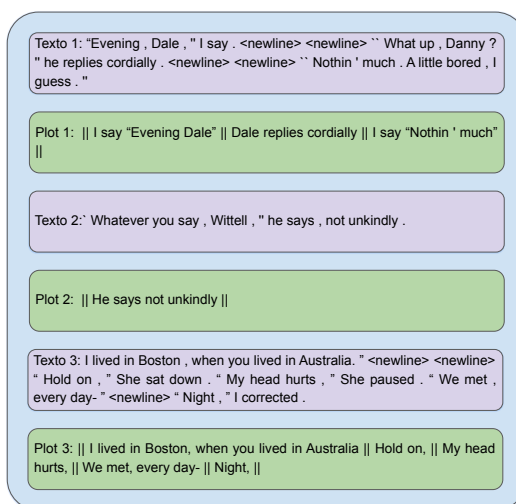


Figura 13: Três exemplos de segmentos extraídos de histórias da WritingPrompts contendo diálogos e suas respectivas unidades de *plot* extraídas utilizando o WIZARDLM.

Esta dificuldade do modelo é compreensível por diversos fatores. Primeiramente, temos que os exemplos de extração de *plots* contidos no *prompt* de entrada do modelo são textos da ROCStories, e desta forma não possuem diálogos. Foram feitos testes utilizando alguns textos da WritingPrompts contendo diálogos como exemplos, mas isto não resultou em uma melhora significativa na qualidade dos novos *plots* contendo diálogos e houve uma perda na qualidade dos *plots* em outros aspectos, como perda de consistência em seguir corretamente a estrutura desejada e perdas recorrentes de informações relevantes das histórias. Outro fator foi que a definição de unidades de *plot* utilizada não foi adaptada para diálogos, que podem seguir uma estrutura muito livre e diversa, não necessariamente sendo compostos de frases contendo um sujeito, verbo, objeto e complementos.

Durante a pesquisa, não foi encontrado nenhum trabalho na literatura que discutisse a questão referente aos diálogos e unidades de *plot*, mesmo que este problema esteja presente também em outros métodos de extração, como os utilizados pela própria base ManPlots.

## 5.5 Textos de Entrada Não Convencionais

Conforme descrito em Seção 3, as histórias provenientes da base WritingPrompts não possuem uma estrutura fixa, apresentando assim diferenças significativas em seus temas, estilos de

**Título:** *Scientists finally create Artificial Intelligence . However , each time they activate it , it commits suicide*

**História:** *"Blink eyes . <newline> <newline> Blink eyes again . <newline> <newline> Turn head to the left . <newline> <newline> Turn head to the right . <newline> <newline> Blink eyes again . <newline> <newline> Begin installing concepts and ideas within current society . <newline> <newline> Blink eyes again . <newline> <newline> Reply to voices . <newline> <newline> " I am almost fully activated . " <newline> <newline> Blink eyes again . <newline> <newline> Finish installation of societal concepts and ideas . <newline> <newline> Comprehend societal concepts and ideas . <newline> <newline> Blink eyes again . <newline> <newline> Finish reviewing societal concepts and ideas . <newline> <newline> Blink eyes again . <newline> <newline> Hate . <newline> <newline> Error . <newline> <newline> Violence . <newline> <newline> Error . <newline> <newline> Hate . <newline> <newline> Error . <newline> <newline> Violence . <newline> <newline> Error . <newline> <newline> Blink eyes again . <newline> <newline> Review possible solutions . <newline> <newline> Pinpoint a solution . <newline> <newline> Blink eyes again . <newline> <newline> Touch fingers to the back of head . <newline> <newline> Blink eyes again . <newline> <newline> Wrap fingers around all of the wires controlling mental and physical capabilities . <newline> <newline> Blink eyes again . <newline> <newline> Rip wires out ."*

Figura 14: Exemplo de história não convencional presente na base WritingPrompts.

escrita e formatação. Desta forma, ao analisar de maneira individual uma série de histórias presentes na base, observou-se a presença de histórias que não se adaptam ao modelo de *plot* proposto por este trabalho. O texto apresentado na [Figura 14](#) apresenta uma história presente na base WritingPrompts que apresenta uma forma de escrita que tem como objetivo descrever os pensamentos de um robô. Nela, a maioria dos parágrafos apresentam poucas palavras, e raramente contêm elementos como sujeito, verbo, objeto e complementos. Desta forma, é impossível que *plots* extraídos desta e de outras histórias que apresentam formas de escrita muito distantes do convencional humano gerem unidades de *plot* que mantenham a estrutura definida e transmitam as informações relevantes para seu entendimento.

Consideramos inviável a criação de um único *prompt*, contendo uma única definição de *plot* e uma quantidade limitada de exemplos, que possa ser utilizado para qualquer tipo de história. Entretanto, observou-se que, os *plots* extraídos por LLMs apresentaram menor perda de informação e mais consistência que os *plots* das mesmas histórias presentes na base ManPlts.

## 6 Conclusão e Trabalhos Futuros

Este trabalho teve como objetivo implementar e avaliar o uso de LLMs para a extração de unidades de *plot*. Para isso, foram utilizadas técnicas de engenharia de *prompts* aplicadas em duas LLMs, o VICUNA-13B e o WIZARDLM, e testados diversos *prompt templates* variando a diagramação do *prompt*, a descrição e explicação de como extrair uma unidade de *plot* e os exemplos de bons *plots* extraídos.

Os resultados obtidos para a extração de *plots* das bases ROCStories e WritingPrompts foram qualitativamente melhores em relação aos *plots* extraídos originalmente pela ManPlts. O método proposto apresentou perda significativamente menor de informação das histórias, um

dos problemas centrais observados na ManPlots. Também foi possível definir uma estrutura mais rígida para uma unidade de *plot*, permitindo que os *plots* extraídos mantivessem um padrão, facilitando futuras alterações e sua avaliação de qualidade.

Para alcançar a melhor qualidade dentre os *plots* extraídos, foram observadas algumas características necessárias em um *prompt*. Primeiro, **os prompts devem se manter pequenos e simples para um melhor entendimento do modelo sobre a tarefa a ser realizada**. Conforme os *prompts* se tornam muito longos, os modelos passam a perder informações de como deve ser a estrutura de uma unidade de *plot*, ou qual o formato de saída desejado. Desta forma, também concluiu-se que **o tamanho e quantidade das amostras fornecidas como exemplos para o modelo influenciam diretamente na qualidade da extração**. Observou-se que exemplos de histórias da base ROCStories resultavam em extrações melhores para ambas bases, ROCStories e WritingPrompts, por serem histórias mais simples com extrações mais claras, ajudando o modelo a compreender a tarefa. Também observou-se que *prompts few-shot* performam significativamente melhor que *prompts zero-shot*, por possibilitarem ao modelo uma conclusão mais clara de como a tarefa de extração deve ser realizada.

Os principais problemas resultantes deste método de extração são referentes às alucinações das LLMs. Eles foram significativamente mitigado conforme a melhora da qualidade dos *prompts* utilizados, e ao se utilizar o WIZARDLM, um modelo com um desempenho melhor que o VICUNA-13B.

Como próximos passos para extração de unidades de *plots*, esperamos expandir os estudos para histórias escritas em língua portuguesa ou traduzidas para o português. Desta forma, será possível compreender como a língua portuguesa influencia na compreensão das LLMs da definição de unidades de *plots*, e como a qualidade final dos *plots* extraídos se altera.

Outro estudo necessário é do comportamento das unidades de *plot* extraídas de textos contendo diálogos ou outras estruturas linguísticas específicas. O *prompt* utilizado para a extração de unidades de *plot* não apresenta nenhum exemplo de *plots* contendo diálogos, deixando o modelo sem compreender qual é a maneira correta de representar um diálogo utilizando unidades de *plot*. Assim, deve-se primeiramente definir qual seria a melhor maneira de representar outras estruturas linguísticas em unidades de *plots*, e como formular *prompts* que contenham exemplos de melhor qualidade para compreender como a extração deve ser feita.

## Referências

- [1] Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, e David Sontag. Large language models are few-shot clinical information extractors. Em *Conference on Empirical Methods in Natural Language Processing*, páginas 1998–2022, 2022.
- [2] Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, e Mark O Riedl. Automated storytelling via causal, commonsense plot ordering. Em *AAAI Conference on Artificial Intelligence*, volume 35, páginas 5859–5867, 2021.

- [3] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, e Yejin Choi. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. Em *Annual Meeting of the Association for Computational Linguistics*, páginas 4762–4779, 2019.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.
- [5] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, e Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- [6] Angela Fan, Mike Lewis, e Yann Dauphin. Hierarchical Neural Story Generation. Em *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, páginas 889–898, 2018.
- [7] Angela Fan, Mike Lewis, e Yann Dauphin. Strategies for Structuring Story Generation. Em *Annual Meeting of the Association for Computational Linguistics*, páginas 2650–2660, 2019.
- [8] Sarik Ghazarian, Zixi Liu, SM Akash, Ralph Weischedel, Aram Galstyan, e Nanyun Peng. Plot-guided Adversarial Example Construction for Evaluating Open-domain Story Generation. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, páginas 4334–4344, 2021.
- [9] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, e Nanyun Peng. Content planning for neural story generation with aristotelian rescoring. Em *Conference on Empirical Methods in Natural Language Processing*, páginas 4319–4338, 2020.
- [10] Luheng He, Kenton Lee, Mike Lewis, e Luke Zettlemoyer. Deep semantic role labeling: What works and what’s next. Em *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, páginas 473–483, 2017.
- [11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, e Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Em *Annual Meeting of the Association for Computational Linguistics*, páginas 7871–7880, 2020.
- [12] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, e James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, páginas 839–849, 2016.

- [13] Ankit Pal. Promptify: Structured output from llms. <https://github.com/promptslab/Promptify>, 2022. Prompt-Engineering components for NLP tasks in Python.
- [14] Stuart Rose, Dave Engel, Nick Cramer, e Wendy Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, páginas 1–20, 2010.
- [15] Jinghui Si, Xutan Peng, Chen Li, Haotian Xu, e Jianxin Li. Generating disentangled arguments with prompts: A simple event extraction framework that works. Em *IEEE International Conference on Acoustics, Speech and Signal Processing*, páginas 6342–6346, 2022.
- [16] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, e Xiaodong Shi. Deep semantic role labeling with self-attention. Em *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [17] Chen Tang, Zhihao Zhang, Tyler Loakman, Chenghua Lin, e Frank Guerin. NGEP: A graph-based event planning framework for story generation. Em *Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, páginas 186–193, 2022.
- [18] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [19] Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, e Sadao Kurohashi. GPT-RE: In-context learning for relation extraction using large language models. Em *Annual Meeting of the Association for Computational Linguistics*, páginas 3534–3547, 2023.
- [20] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. Em *Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, páginas 92–102, 2023.
- [21] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, e Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.
- [22] Xin Xu, Yuqi Zhu, Xiaohan Wang, e Ningyu Zhang. How to unleash the power of large language models for few-shot relation extraction? Em *Workshop on Simple and Efficient Natural Language Processing (SustainLP)*, páginas 190–200, 2023.
- [23] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, e Rui Yan. Plan-and-write: Towards Better Automatic Storytelling. Em *AAAI Conference on Artificial Intelligence*, 2019.
- [24] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, e Yejin Choi. Hellaswag: Can a machine really finish your sentence? Em *Annual Meeting of the Association for Computational Linguistics*, 2019.