

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

EDER BRAGANÇA PEREIRA

**AVALIAÇÃO DE LARGE LANGUAGE MODELS DENTRO DE
SISTEMAS DE RECOMENDAÇÃO DE NOTÍCIAS**

Ouro Preto, MG
2025

EDER BRAGANÇA PEREIRA

**AVALIAÇÃO DE LARGE LANGUAGE MODELS DENTRO DE SISTEMAS DE
RECOMENDAÇÃO DE NOTÍCIAS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Ouro Preto, MG
2025



FOLHA DE APROVAÇÃO

Eder Bragança Pereira

Avaliação de Large Language Models dentro de sistemas de recomendação de notícias

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Agosto de 2025.

Membros da banca

Rodrigo César Pedrosa Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Anderson Almeida Ferreira (Examinador) - Doutor - Universidade Federal de Ouro Preto
Ana Luiza Almeida Soares (Examinadora) - Bacharel - Programa de Pós-Graduação em Ciência da Computação - UFOP

Rodrigo César Pedrosa Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/08/2025.



Documento assinado eletronicamente por **Rodrigo Cesar Pedrosa Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 31/08/2025, às 20:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0965545** e o código CRC **D59332B7**.

Este trabalho é dedicado a todos que acreditaram no meu potencial, em especial à minha família, que, mesmo nos momentos em que nem eu parecia acreditar, estava pronta para me dar todo o apoio necessário.

Agradecimentos

Gostaria de agradecer a todos que estiveram comigo desde o início e conseguiram contribuir de qualquer forma para essa graduação ser concluída. Agradeço principalmente à minha família, que esteve comigo durante todo o processo, e aos meus amigos, que nunca me deixaram sentir sozinho nesse período longe de casa e tornaram tudo mais leve.

Agradeço por fim a instituição e aos professores do DECOM, que foram solícitos quando precisei e sempre conduziram as aulas com muita maestria, representando sempre alto profissionalismo.

"Os verdadeiros analfabetos são os que aprenderam a ler e não leem." (QUINTANA, 1994)

Resumo

O uso de ferramentas modernas, como modelos de linguagem de grande escala para recomendação de notícias e ferramentas de *web scraping* para a coleta de informações, vêm transformando a forma como os *clippings* são gerados. Este trabalho traz a abordagem dessas modernizações dentro do cenário de *clippings*, combinando o uso de personas, [Large Language Model \(LLM\)](#), sistemas de recomendações e engenharia de *prompt*, a fim de obter um sistema de *clippings* que seja flexível e possa ser utilizado por parlamentares e assessores, fornecendo um panorama personalizado e sintético dos assuntos mais importantes. Para isso, foi realizada uma análise comparativa entre abordagens de *prompt engineering*, utilizando diferentes padrões de envio, e comparadas as versões 3B do LLaMa 3.2 e 7B do Mistral, a fim de encontrar o modelo mais eficaz para o trabalho. Contudo, as diferentes abordagens trouxeram resultados mais direcionada e precisos, especialmente na utilização da persona no *prompt*.

Palavras-chave: modelos de linguagem de grande escala. sistemas de recomendação. engenharia de *prompt*.

Abstract

The use of modern tools, such as large language models for news recommendation and web scraping tools for information collection, has transformed the way clippings are generated. This work brings the approach of these modernizations within the news clipping scenario, combining the use of personas, [Large Language Model \(LLM\)](#), recommendation systems and prompt engineering, in order to obtain a clipping system that is flexible and can be used by parliamentarians and advisors, providing a personalized and synthetic overview of the most important issues. For this, a comparative analysis was carried out between prompt engineering approaches, using different submission patterns, and compared the LLaMa 3.2 3B and the Mistral 7B, in order to find the most effective model for the work. However, the different approaches yielded more targeted and accurate results, especially when using the persona in the prompt.

Keywords: large language models. recommendation systems. prompt engineering.

Lista de Ilustrações

Figura 2.1 – Demonstração do fluxograma proposto pelos autores do RecPrompt, que demonstra os 3 sistemas existentes no artigo. Fonte: (LIU et al., 2024b)	4
Figura 3.1 – Imagem retirada do trabalho (BROWN et al., 2020) que demonstra um comparativo do desempenho entre modelos de <i>prompt engineering</i>	13
Figura 4.1 – Fluxograma do funcionamento do sistema	15
Figura 4.2 – Fluxograma do funcionamento do sistema	16
Figura A.1 – Imagem retirada do artigo (SCHULHOFF et al., 2024) que demonstra todas as técnicas abordadas no seu trabalho.	56

Lista de Tabelas

Tabela 2.1 – Comparativo dos resultados obtidos do PerCoNet e outros métodos de aprendizado profundo. Fonte: (LIU et al., 2023)	3
Tabela 5.1 – Comparação entre Deepseek-r1 e LLaMA 3.2 nas métricas avaliadas	29
Tabela 5.2 – Comparação entre LLaMA 3.2 3B e Mistral 7B em dois <i>prompts</i> nas métricas avaliadas	33
Tabela 5.3 – Comparação entre Mistral e LLaMA 3.2	34

Lista de Abreviaturas e Siglas

AUC Area Under ROC Curve. 4, 5, 7, 11

BERT Bidirecional Encoder Representations from Transformers. 3, 12

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart. 9

CoT Chain of thought. 13

CSS Cascading Style Sheets. 8

CSV Comma-separated values. 16, 17

GDPR Data Protection Regulation. 9

GSM8K Grade School Math 8K. 14

HTML HyperText Markup Language. 8, 15

HTTP Hypertext Transfer Protocol. 8, 34

IA Inteligência Artificial. 12

JSON JavaScript Object Notation. 17, 19, 34

LGPD Lei Geral de Proteção de Dados. 9

LLM Large Language Model. vi, vii, 1–5, 7, 11–13, 15, 17–19, 21, 24, 25, 28, 29, 32–34, 37, 48

MIND Microsoft News Dataset. 5, 7

MRR Mean Reciprocal Rank. 4, 5, 7, 11

nDCG normalized Discounted Cumulative Gain. 4, 5, 7, 11

PLM Modelo de Linguagem Pré-treinado. 1, 3

REGEX Regular Expressions. 6, 8, 16

SVAMP Simple Variations on Arithmetic Math word Problems. 14

VLM Vision-language Models. 12

WWW World Wide Web. 7

XML Extensible Markup Language. 8

Sumário

1	Introdução	1
1.1	Objetivos	2
2	Revisão de literatura	3
3	Fundamentação Teórica	6
3.1	<i>Clipping</i> de Notícias	6
3.1.1	Métodos tradicionais	6
3.1.2	Avaliação de <i>clippings</i>	7
3.2	Web Scraping para Coleta de Notícias	7
3.2.1	Principais técnicas de <i>Web Scraping</i>	8
3.2.2	Bibliotecas e serviços de Web Scraping	8
3.2.3	Desafios e limitações do Web Scraping	9
3.3	Sistemas de Recomendação de Notícias	9
3.3.1	Desafios específicos da recomendação de notícias	10
3.4	Uso de Personas para Definição de Clipping de Notícias	10
3.4.1	Aplicação de personas em sistemas de recomendação	11
3.5	Large Language Models (LLMs) e sua Aplicação em Clipping de Notícias	11
3.5.1	Como os LLMs podem auxiliar no cenário de notícias?	12
3.6	Técnicas de Prompt Engineering para LLMs	12
3.6.1	Técnicas de Prompt Engineering	12
4	Metodologia	15
4.1	<i>Scraper</i>	15
4.2	Códigos	16
4.3	Testes preliminares	17
4.4	Pré-experimentação	21
4.5	Experimentação	24
4.5.1	Configuração	24
4.5.2	Exemplos	25
5	Resultados	29
5.1	Resultados da pré-experimentação	29
5.1.1	Problemas encontrados	30
5.2	Resultados da experimentação	32
5.2.1	Problemas encontrados	34
6	Considerações Finais	37
6.1	Conclusão	37

Referências	38
Apêndices	43
APÊNDICE A <i>Clippings gerados</i>	44
A.1 Pré-experimentação	44
A.2 Experimentação	44
APÊNDICE B <i>Organização de envio de clippings</i>	45
APÊNDICE C <i>Códigos</i>	46
APÊNDICE D <i>Prompts utilizados em cada teste</i>	52
D.0.1 Teste 1	52
D.0.2 Teste 2	52
D.0.3 Teste 3	52
Anexos	54
ANEXO A <i>Técnicas de prompt engineering</i>	55

1 Introdução

A análise e o monitoramento da mídia são elementos essenciais para a gestão do sucesso de uma marca e para a formulação de estratégias de comunicação (Brand24, 2024). Através disso, o *clipping* de notícias se mostra como uma ferramenta fundamental para supervisionar a mídia e compreender como a marca é representada na imprensa, jornais e redes sociais. O *clipping* refere-se ao processo de coleta, monitoramento e análise de notícias publicadas em diversos canais, como canais jornalísticos, *blogs*, redes sociais ou até mesmo plataformas de *streaming*, como o *Youtube* (Comunique-se, 2024).

Historicamente, o *clipping* de notícias era realizado de forma manual, onde as empresas contratavam serviços especializados para "recortar" as notícias relevantes das revistas e jornais impressos (RD Station, 2024). No entanto, com a digitalização da mídia, o processo de *clipping* foi modernizado, anexando tecnologias de inteligência artificial e aprendizado de máquina para relatar menções da marca (SINOPRESS, 2024).

O *clipping* é um dos elementos mais básicos e primordiais para medir resultados da estratégia de comunicação e, quando é bem estruturado e organizado, apresenta dados como relevância da marca, frequência, positividade ou negatividade da popularidade e público que está sendo atingido pela propaganda. As técnicas para a geração de *clippings* envolvem processos como definição de palavras chaves, a fim de buscar nos veículos sobre assuntos específicos, extração de dados, seja ela manual ou digital, e organização das informações, para poder montar uma base de dados sólida, entre outros passos.

A combinação do uso de técnicas de engenharia de *prompt* e de *Large Language Models* (LLMs) para a geração de *clippings* de notícias possuem estudos relevantes, como o (LIU et al., 2024b), que utiliza de LLMs (GPT 3.5 e GPT 4.0) para este fim, obtendo resultados superiores a outros modelos profundos tradicionais, como o DKN (WANG et al., 2018) e NAML (WU et al., 2019). Além disso, (CHEN RUNFENG XIE, 2024) utiliza a combinação de LLMs com grafos de conhecimento para criar uma ferramenta de recomendação de notícias personalizada, demonstrando ainda mais a importância e relevância dos estudos na área.

O uso de personas em sistemas de recomendação é uma área explorada por alguns autores, seja de maneira implícita, como em (BARKAN et al., 2020), onde cada usuário é representado por múltiplas personas implícitas e aprendidas automaticamente a partir do seu histórico, ou explícita, como em (LIU et al., 2023), que utilizam modelos *Modelo de Linguagem Pré-treinados* (PLMs) combinados a personas construídas manualmente através do seus interesses e *hobbies* para compor um sistema de recomendação de notícias.

Além disso, as automações trazidas pelas soluções que utilizam LLMs e ferramentas de *web scraping* oferecem oportunidades significativas para otimizar esse fluxo de trabalho,

permitindo a extração mais refinada e contextualizada. Modelos como o GPT-4, DeepSeek e o LLaMa podem ser utilizados para filtrar, resumir e classificar notícias automaticamente, aumentando a eficiência e a personalização do *clipping* de informações (CHEN RUNFENG XIE, 2024).

O trabalho proposto utiliza a combinação dessas abordagens para a geração de *clippings* voltados ao cenário político, onde os veículos de notícias possuem esse perfil estadista. A utilização dessa ferramenta por membros parlamentares com o objetivo de obter dados e analisá-los traz diversos ganhos significativos, pois a mídia contribui para formar as referências a partir das quais os participantes das elites políticas constroem suas ações (MIGUEL; MÁXIMO, 2015). Nesse sentido, os *clippings* gerados serão direcionados aos congressistas e seus assessores, que buscam diariamente ter um recorte da mídia, com o objetivo principal de se manterem atualizados com as principais notícias da imprensa.

O processo de *clipping* de notícias é uma prática já consolidada e amplamente utilizada por diversas áreas, como assessores de imprensa, jornalistas e comunidades científicas (Comunique-se, 2024). No entanto, é afirmado no artigo que a análise humana ainda é momentaneamente indispensável, exatamente pela falta de ferramentas que utilizam inteligência artificial para substituir a interpretação feita por pessoas.

Com isso, foi utilizado neste trabalho ferramentas de *web scraping* para obter e formar a base de dados, personas para criar a personificação do público-alvo (ou seja, alguém presente no meio parlamentar), engenharia de *prompt* para amplificar os resultados na comunicação com as LLMs, que serão utilizadas para categorizar as personas e indicar as notícias que mais se adaptem.

Portanto, a relevância da pesquisa consiste na necessidade de aprimorar a geração de *clippings* de notícias para parlamentares, através da combinação de técnicas inovadoras e potentes, como ferramentas de *web scraping*, personas, LLM e engenharia de *prompt*.

1.1 Objetivos

Diante desse cenário, o principal objetivo deste trabalho é gerar um sistema de *clipping* de notícias baseado em Large Language Model (LLM) que selecione as notícias mais relevantes de forma a fornecer para parlamentares e assessores um panorama personalizado compacto porém sintético dos assuntos mais importantes, avaliando sua precisão, relevância e aplicabilidade na filtragem e síntese de notícias.

Para cumprir o objetivo geral foram definidos alguns objetivos específicos: Construir raspadores que coletem as notícias dos principais veículos da imprensa(I); encontrar o modelo mais eficaz de LLM para o trabalho (II); e encontrar o modelo de *prompt* que mais se adapte para o cenário de *clipping*, comparando diferentes técnicas de *prompt engineering* (III).

2 Revisão de literatura

Esta seção é utilizada para apresentar trabalhos anteriores relevantes ao tema em questão. Diversos estudos buscam explorar os aspectos relacionados, como o PerCoNet, um sistema de recomendação de notícias baseado em personas e aprendizagem contrastiva (LIU et al., 2023) e o RecPrompt, que utiliza a combinação do processo de *Self-tuning* com LLMs para montar sistema de recomendação de notícias (LIU et al., 2024b).

Em (LIU et al., 2023), os autores buscam construir um modelo de recomendação de notícias personalizado que combina a utilização de personas explícitas com aprendizado contrastivo *cross-view*, denominado de PerCoNet. Nele, são utilizados PLMs (Bidirecional Encoder Representations from Transformers (BERT)) nos processos de codificação de notícias e codificação de usuários.

O desenvolvimento de personas explícitas no artigo foi motivado, como mencionado no próprio artigo, por conta da maioria dos sistemas de recomendação atuais utilizarem personas implícitas. Para a sua criação, foram levadas em consideração características como gênero musical, esportes preferidos, tipos de filmes, marcas de roupas, posicionamento político, entre outras.

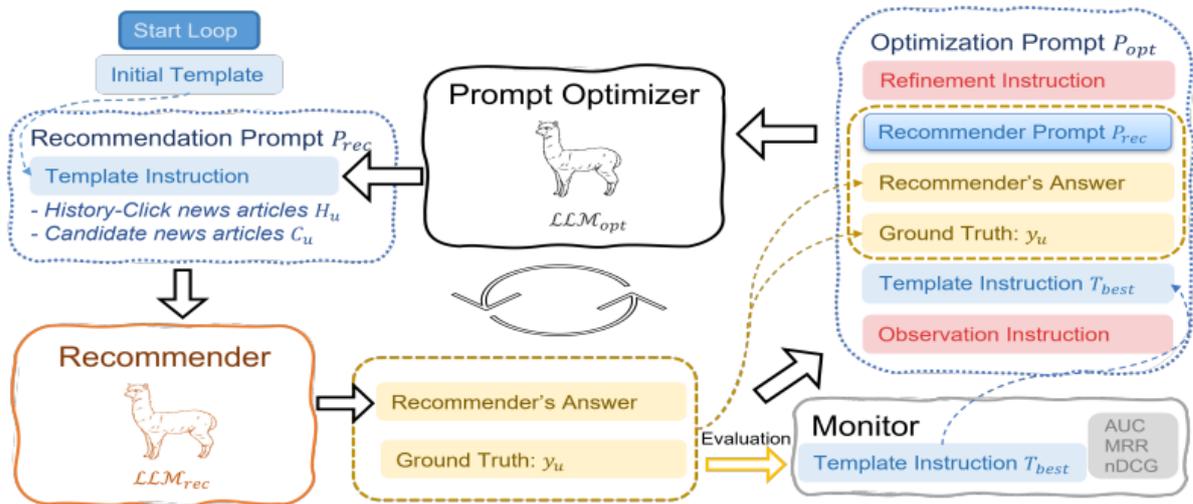
O trabalho inclui também a criação de um codificador de notícias candidatas, que foi projetado para aprender as representações de notícias através dos títulos e resumos. Ele funciona da seguinte maneira: Seja uma notícia (v), um modelo pré-treinado é utilizado para gerar *embeddings* do título ou resumo desta notícia. Essa etapa é importante para o sistema de recomendação extrair os principais pontos das notícias.

Semelhante ao processo de codificar as notícias candidatas, o processo de codificação de usuários presente nesse trabalho utiliza do BERT para modelar os interesses de um usuário, bem como suas notícias lidas recentemente. Por fim, o aprendizado contrastivo *cross-view* é utilizado para melhorar a coerência entre as notícias e os usuários e distinguir diferentes personas.

Tabela 2.1 – Comparativo dos resultados obtidos do PerCoNet e outros métodos de aprendizado profundo. Fonte: (LIU et al., 2023)

Method	MIND				Adressa			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@3	nDCG@5
LibFM [31]	57.93	24.64	26.53	33.35	55.98	49.73	48.78	62.12
DeepFM [12]	58.18	25.87	27.57	34.25	59.95	59.27	55.06	69.18
DKN [40]	62.14	27.83	30.04	36.70	65.49	63.77	60.71	72.60
LSTUR [1]	64.83	30.41	33.37	39.81	66.17	64.27	62.15	72.86
NAML [43]	66.02	32.13	35.41	41.37	67.42	62.97	62.30	72.10
NRMS [44]	65.56	30.48	33.45	39.97	68.24	66.00	65.31	74.80
KRED [23]	66.27	32.27	35.53	41.52	69.76	66.01	66.11	74.82
EEG [54]	66.47	31.80	34.85	41.23	68.49	65.71	63.69	73.67
UNBERT [53]	67.57	32.40	35.85	42.26	74.47	68.23	68.46	74.82
PerCoNet*	68.93	33.40	36.93	43.28	77.26	72.20	70.18	76.22

Figura 2.1 – Demonstração do fluxograma proposto pelos autores do RecPrompt, que demonstra os 3 sistemas existentes no artigo. Fonte: (LIU et al., 2024b)



Para avaliar as metodologias, foram utilizadas duas bases de dados: Mind (disponível em [Mind](#)) e Adressa (disponível em [Adressa](#)). Essas bases de dados são bases estáticas utilizadas para validar sistemas de recomendação de notícias, onde são construídos grupos de notícias candidatas e qual delas o usuário realmente clicou. A partir disso, o PerCoNet foi comparado com diversos métodos do estado da arte, como mostrado na tabela 2.1, entre eles: LibFM (RENDLE, 2012), DeepFM (GUO et al., 2017) e LSTUR (AN et al., 2019), avaliando métricas conhecidas como Area Under ROC Curve (AUC), Mean Reciprocal Rank (MRR) e normalized Discounted Cumulative Gain (nDCG), se demonstrando superior que os outros métodos em todas as métricas avaliadas.

Por sua vez, o artigo (LIU et al., 2024b) propõe o desenvolvimento de um *framework* de *prompt* autoajustável para recomendação de notícias utilizando LLMs, como o GPT-4. A Figura 2.1 mostra o esquema proposto pelos autores no artigo, que é composto por três componentes principais: o Recomendador de Notícias (LLMREC), o Otimizador de Prompt (LLMOPT) e o Monitor.

Para o Recomendador de Notícias foi utilizado de LLM para gerar uma lista ranqueada de notícias e explicar os tópicos de interesse baseado no histórico e, para isso, o *prompt* (PREC) é necessário. Esse *prompt* contém um modelo de instruções T com dois espaços reservados: "{history}", utilizado para enviar o histórico de cliques do usuário, e "{candidate}", representando as notícias candidatas daquela recomendação.

Já o Otimizador de Prompts é usado para gerar melhores mensagens de envio às LLMs automaticamente. Nesse processo, foi utilizada outra LLM para obter um *prompt* que melhore a correspondência entre as notícias recomendadas e o interesse dos usuários. Para realizar o

procedimento, ele recebe o *prompt* atual, a última lista ranqueada gerada pelo recomendador, o *Ground truth*, notícias que o usuário realmente clicou, e o melhor *Template* de instruções registrado pelo Monitor. Com esses parâmetros, é feito um envio para a LLM com o objetivo de obter um novo *prompt*.

Por fim, o Monitor é utilizado para avaliar os modelos de entrada fornecidos pelo Otimizador e suas respostas obtidas pelo Recomendador. Ele verifica métricas como a MRR e a nDCG para determinar se aquele novo *prompt* obtido é melhor que o anterior. Essas medidas são calculadas baseadas na lista ranqueada obtida e no *Ground truth*. A partir disso, o Monitor compara com o último *prompt* e garante que a melhor versão seja continuada no sistema de recomendação de notícias.

Adicionalmente ao protótipo principal, os autores propõem o TopicScore, uma nova métrica criada para avaliar a qualidade temática das explicações do Recomendador, isso ocorre pois as métricas tradicionais avaliam apenas o ranking ou a similaridade textual. Com isso, o modelo avalia o quanto o modelo (LLMREC) entende e resume corretamente os interesses com base no histórico e nas notícias recomendadas.

Para avaliar as metodologias propostas no artigo, foi utilizada a base de dados Microsoft News Dataset (MIND Mind). A partir dela foram selecionados 100 usuários de forma aleatória para a validação dos modelos e 400 usuários para teste. Utilizando as LLMs GPT-3.5 e GPT-4, quando comparado a modelos de profundidade (Deep Models), foram obtidos valores de +3,36% na métrica AUC, +10,49% na MRR e +9,64% na nDCG, demonstrando efetividade no uso de modelos de linguagem em sistemas de recomendação de notícias.

Portanto, os trabalhos revisados neste capítulo demonstram alta relevância em sistemas de recomendações de notícias com modelos pré-treinados. No entanto, eles possuem algumas limitações, como realizar testes com uma base de notícias estática (como MIND e Adressa) e não possuir avaliação qualitativa para seus experimentos. Nesse sentido, o presente trabalho busca adaptar e estender tais abordagens, incorporando novas técnicas voltadas à geração de *clippings*, utilizando a combinação de técnicas de raspagem de dados para gerar uma base de dados dinâmica, o uso de LLMs para realizar as recomendações e a exploração de avaliações qualitativas, permitindo uma análise mais pessoal dos *clippings*.

3 Fundamentação Teórica

3.1 *Clipping* de Notícias

O processo de *Clipping* surgiu no século XIX, em um contexto em que jornais impressos eram a principal fonte de informação (RD Station, 2024). Inicialmente ele consistia no recorte físico de artigos relevantes para determinada marca, empresa ou figura pública, para então serem analisadas e utilizadas para o monitoramento da marca.

Segundo autores, como em (Neil Patel, 2025), existem diversas razões para a realização de *clipping*, como a análise da percepção do público, que é influenciado pela mídia regularmente e ajuda a traçar metas do desenvolvimento da imagem, identificar informações erradas de uma determinada marcas, permitindo tratativas mais diretas, além de conseguir acompanhar os concorrentes.

O *clipping* de notícias é utilizado por diferentes profissionais e organizações, entre eles estão os deputados e assessores de imprensa, juntamente com jornalistas (TEIXEIRA, 2001), onde o monitoramento emerge como instrumento de avaliação da repercussão na imprensa do trabalho parlamentar, e empresas de marketing e *branding*, que buscam percepções sobre a repercussão dos seus produtos no mercado.

3.1.1 Métodos tradicionais

Existem diversos métodos e técnicas para a montagem de *clippings*, como o uso de palavras-chave, métodos baseados em expressões regulares, e serviços completos, como o Google Alerts (Press Manager, 2025) e o Meltwater (HERMAN, 2015), que utilizam diferentes técnicas combinadas.

A escolha de palavras-chave, que pode ser combinado ao uso de **Regular Expressions (REGEX)** para meios digitais, é um dos passos iniciais e mais importantes do processo, como apresentado em (Knewin, 2025), pois elas que serão a base de toda a busca. A escolha inadequada dessas palavras podem resultar em coletas de informações irrelevantes para seu objetivo.

O Google Alerts (Google, 2025) é uma ferramenta gratuita e que permite a configuração de palavras-chave e alertas via *email*. O serviço permite monitorar artigos, notícias, páginas da web, *posts* de *blogs*, entre outros, o que se demonstra ideal para novas empresas. Trabalhos como (NEW-AARON et al., 2017) exploram os ganhos desses serviços, onde realizou a montagem de clipagens combinando diversas palavras-chave, relatando que obteve bons dados para realizar a análise proposta pelo artigo.

Já o Meltwater (Meltwater, 2025) é uma ferramenta paga que, além do monitoramento

dos serviços do Google Alerts, permite também o monitoramento de transmissões de TV e rádio (RD Station, 2024), oferecendo relatórios detalhados sobre a análise, se mostrando assim uma opção mais robusta e indicada para empresas que buscam *insights* mais avançados.

No entanto, apesar das inúmeras opções robustas e técnicas de geração de clipping, o trabalho humano ainda é fundamental para a construção (Comunique-se, 2024), tornando o monitoramento de notícias mais complexo ao envolver partes manuais, aumentando assim o custo de realização.

3.1.2 Avaliação de *clippings*

A avaliação de *clippings* é comumente conduzida através de abordagens quantitativas e qualitativas, que podem também ser utilizadas em conjunto. A avaliação quantitativa é mais objetiva e automatizada, ela é a representação numérica com o objetivo de explicar os fenômenos que essas observações refletem (SUKAMOLSON, 2007). A pesquisa qualitativa, por sua vez, é caracterizada por seus objetivos, que se relacionam com a compreensão de algum aspecto social e seus métodos, e, muitas vezes, geram palavras ao invés de números (MCCUSKER; GUNAYDIN, 2015).

Para abordagem qualitativa, emergem diversas técnicas, como a Escala Likert, desenvolvida na década de 30 (OLIVEIRA, 2023) e utilizada para representar sentimentos mais realistas buscando valores não binários (JOSHI et al., 2015), a comparação pareada, muito comum em LLMs, que apresenta duas opções e nela é perguntada qual responde melhor o cenário (FÜRNKRANZ; HÜLLERMEIER, 2010), e avaliações mais simples, como a Avaliação Absoluta, que busca atribuir notas de 1-x nos critérios escolhidos.

Além disso, as abordagens quantitativas, combinadas à bases de dados estáticas com um grande volume de dados como a MIND, também são utilizadas em trabalhos de recomendação de notícias, como em (LIU et al., 2023), que combina métricas MRR, AUC e nDCG para avaliar o desempenho das LLMs, e em (LI; ZHANG; MALTHOUSE, 2023), que utilizam de métricas nDCG e Hit para avaliar os *prompts* enviados aos sistemas de recomendação.

3.2 Web Scraping para Coleta de Notícias

A raspagem da *web*, ou *web scraping*, é um conjunto de técnicas que são utilizadas para extrair dados da World Wide Web (WWW) e armazená-los em seu computador ou banco de dados para tratá-los futuramente (ZHAO, 2022). Contudo, o processo de coleta pode se tornar complexo, especialmente se realizado manualmente, fomentando estudos na área de automatização (DIOUF et al., 2019).

Juntamente com a importância da coleta de dados e o avanço tecnológico fomentado no século XXI, o desenvolvimento de ferramentas de automação de coleta de dados da web se

mostra altamente relevante, seja pela eficiência trazida pelas ferramentas automatizadas, pela rapidez e versatilidade desses serviços ou pelo ganho de *performance* quando comparadas aos métodos manuais (KHDER, 2021).

3.2.1 Principais técnicas de *Web Scraping*

Dentre as principais técnicas de extração, destacam-se (KHDER, 2021): *Copy and paste*, *Regular Expressions* (REGEX) e *HyperText Markup Language* (HTML) *parsing*.

A *Copy and paste*, também conhecida como cópia manual, é maneira mais primitiva e tradicional de extrair dados da internet, que consiste basicamente em copiar manualmente os dados, sem o uso de *frameworks* terceiros para auxiliar na extração (KHDER, 2021).

A estratégia *Regular Expressions* (REGEX) é uma fórmula que busca correspondências entre padrões de caracteres e palavras com o texto (GUNAWAN et al., 2019/03). Por haver a necessidade do domínio da criação de expressões regulares, o método possui uma dificuldade maior agregada.

Após o acesso e recuperação das páginas *Web* através de requisições *Hypertext Transfer Protocol* (HTTP), o *HTML parsing* (Análise HTML) é utilizado para extrair todo o conteúdo de interesse das páginas através da mescla com filtros (GLEZ-PEÑA et al., 2014).

3.2.2 Bibliotecas e serviços de *Web Scraping*

Durante o processo de *Web Scraping*, algumas bibliotecas podem auxiliar na extração, como *Beautiful Soup*, o *Selenium* e a *Scrapy*.

A *Beautiful Soup* (RICHARDSON, 2024) é uma biblioteca da linguagem de programação *Python* utilizada para a retirada de dados de arquivos *HTML* e *Extensible Markup Language* (XML). Esta ferramenta, criada em 2004 por Leonard Richardson, é amplamente utilizada em cenários de *web scraping* devido a sua facilidade de utilização, desde a instalação até a manipulação de código.

Já o *Selenium* se configura como uma variedade de ferramentas e bibliotecas de automação que permitem emular a interação do usuário com os navegadores da web, tanto para extrair dados, quanto para testar aplicações (*Selenium Project*, 2025). Sua utilização está disponível em diversas linguagens, como C#, Java e Python.

O *Scrapy* por sua vez é um *framework* gratuito e de alto nível para *web scraping* desenvolvida em Python (*Scrapy Project*, 2025). Essa ferramenta combina diversas técnicas de extração, como *HTML parsing*, *REGEX* e *CSS selectors*, o que torna sua manipulação extremamente versátil.

Além disso, existem alguns serviços completos de extração, como a *NewsAPI*, *RSS Feeds*, *News Crawl* e *GDELT*, que fornecem acesso estruturado a notícias jornalísticas. Essas soluções

entregam, por meio de requisições automatizadas, dados organizados, facilitando a análise do *scraping*.

3.2.3 Desafios e limitações do Web Scraping

O processo de extração de dados encontra diversos desafios em alguns sites, impondo restrições aos *scraping*, incluindo [Completely Automated Public Turing test to tell Computers and Humans Apart \(CAPTCHAs\)](#), limites de taxas de requisições, como mostra [\(KHDER, 2021\)](#), e os aspectos legais e éticos.

Os bloqueios por [CAPTCHAs](#) são um conjunto de vários métodos de autenticação que possuem um objetivo bem claro de diferenciar usuários humanos de usuários não humanos [\(IBM, 2025\)](#). Lidar com sites que possuem esse tipo bloqueio pode negar o acesso das ferramentas, refletindo em erros que comprometam o *scraping*.

Já o erro de limites de taxas de requisições ocorre quando excedemos o número de requisições a determinado serviços, refletindo em bloqueios ao acesso e, conseqüentemente, impede a requisição de informações.

Outro desafio enfrentado pelos *Web scrapers* são os aspectos legais e éticos, onde coletar dados sem levar em consideração as políticas de uso específicas de cada site pode se configurar crime [\(BROWN et al., 2024\)](#), o que as tornam submissas a regulamentos como a [Data Protection Regulation \(GDPR\)](#), ou, no Brasil, a [Lei Geral de Proteção de Dados \(LGPD\)](#) que são regulamentos que possuem o objetivo claro de proteger a privacidade das pessoas e seus dados, entregando mais segurança aos usuários da internet.

3.3 Sistemas de Recomendação de Notícias

Sistemas de recomendação são técnicas e ferramentas de *software* que provém sugestões de itens a serem utilizadas pelo usuário [\(RICCI; ROKACH; SHAPIRA, 2021\)](#). Essas recomendações visam apoiar o usuário em vários processos de tomadas de decisões, como o que comprar, qual música ouvir ou qual notícia ler. Dentre os principais modelos utilizados se destacam os Baseados em conteúdo, os baseados em filtragem colaborativa e as abordagem híbridos, que mescla as diferentes abordagens [\(FENG et al., 2020\)](#).

Nas abordagens baseadas em conteúdo, os sistemas buscam recomendar aos usuários novos itens similares aos que ele escolheu no passado. Essa similaridade é calculada baseada nas características associadas aos itens comparados. Para ter um exemplo adaptado ao cenário de notícias, se um usuário gostou de notícias do tema "futebol", o sistema pode aprender a recomendar outras notícias do tema [\(RICCI; ROKACH; SHAPIRA, 2021\)](#).

As abordagens baseadas em filtragem colaborativa, apresentada em [\(GOLDBERG et al., 1992\)](#), consideram o comportamento de usuários com perfis semelhantes para sugerir conteúdos,

podendo sugerir baseado em notícias lidas por usuários semelhantes ou recomendar notícias que frequentemente aparecem juntas nos históricos de leitura. A similaridade entre dois perfis é calculada através da similaridade entre o histórico dos usuários, comparando os gêneros dos conteúdos e categorias (RICCI; ROKACH; SHAPIRA, 2021).

No entanto, existem vantagens e desvantagens nas duas abordagens, como Cold Start nas filtragens colaborativas, por precisarem de uma grande escala de dados, e a dificuldade de gerar os atributos dos itens em certas nas abordagens baseadas em conteúdos (THORAT; GOUDAR; BARVE, 2015). Com isso, os modelos híbridos surgiram do cenário em que ambas abordagens já estavam bem introduzidas e as utilizam de forma combinada, buscando resolver as limitações de uma com as vantagens das outras (RICCI; ROKACH; SHAPIRA, 2021). Ótimos exemplos são as plataformas como Google News e Netflix que utilizam desses modelos para compor e balancear as recomendações (THORAT; GOUDAR; BARVE, 2015).

3.3.1 Desafios específicos da recomendação de notícias

O processo de recomendação de notícias encontra diversos desafios, interferindo no seu desempenho, incluindo Cold-Start Problem, dispersão de dados e a mudança de interesses (SHARMA; GERA, 2013).

Alguns sistemas de recomendação enfrentam um desafio extremamente relevante é o Cold-start Problem (ÖZGÖBEK; GULLA; ERDUR, 2014). Esse problema é causado devido a alguns sistemas de recomendação não conseguirem recomendar novos itens se não existem informações iniciais suficientes, como histórico de *clicks*, interações anteriores, metadados, dificultando as recomendações iniciais (PANTELI; BOUTSINAS, 2023).

A dispersão de dados é um dos principais problemas encontrados em modelos de filtragem colaborativa e que exerce grande influencia na qualidade das recomendações (SHARMA; GERA, 2013). Ela ocorre quando existem poucos dados de interação entre usuário e itens (CHEN et al., 2011).

A mudança de interesses por parte dos usuários é um dos maiores desafios enfrentados em sistemas de recomendação baseados em usuários (FENG et al., 2020), tornando as recomendações ainda mais dinâmicas. Essa mudança pode ocorrer por fatores pessoais (mudança de interesses ou *hobbies*), ou por fatores externos, como mudança na política local e eventos mundiais.

3.4 Uso de Personas para Definição de Clipping de Notícias

As *personas* são representações de usuários e costumes, apresentadas em forma de pessoas imaginárias (AN et al., 2018), e são mais comumente utilizadas em ramos de desenvolvimento de software, assistência médica, educação superior, marketing, entre outras áreas (SALMINEN et al., 2022).

O uso dessas representações em sistemas de recomendação busca identificar os usuários e retirar as ambiguidades entre eles, para assim melhorar os resultados obtidos (BAJAJ; SHEKHAR, 2016). Utilizando um cenário político, essas personas podem representar, além de características pessoais, posicionamento político e afiliações, oferecendo *insights* importantes para os algoritmos.

3.4.1 Aplicação de personas em sistemas de recomendação

(LIU et al., 2023) trazem a abordagem da utilização de personas para a criação de sistemas de recomendação personalizados, criando personas com a utilização de palavras-chave bem diversificadas. Como resultado, seu sistema era capaz de associar as características com as palavras do resumo da notícia fornecido, identificando sinônimos, palavras relacionadas e propriamente suas aparições no texto. Com isso, foram avaliados resultados usando *Mind* e *Adressa* (bases de dados de notícias disponíveis no mercado com mais de 150 mil notícias cada), com taxas de acerto superiores a outros 9 métodos comparados em métricas conhecidas como *nDCG*, *MRR* e *AUC*, demonstrando que o seu uso é promissor.

O uso de personas em outros modelos de sistemas de recomendação também é evidenciado em (LI et al., 2016/04), que busca resolver, através da abordagem da representação de pessoas fictícias, os problemas causados pela desorientação e sobrecarga de informação devido ao excesso de informações. O trabalho realiza diversos experimentos digitais através da análise da interação entre o usuário e os navegadores, mesmo lidando com o desafio do paradigma da contradição entre a proteção de privacidade e sistemas de recomendação, onde os serviços podem ser invasivos, resultando em uma menor privacidade por parte dos usuários.

3.5 Large Language Models (LLMs) e sua Aplicação em Clipping de Notícias

Desde os Testes de Turing, na década de 1950, os humanos têm explorado o domínio da inteligência linguística por máquinas. A linguagem é essencialmente um sistema complexo regido por regras gramaticais (ZHAO et al., 2023). Nesse cenário de evolução das linguagens, as *LLMs*, ou modelos de linguagem de larga escala em português, são modelos de aprendizado profundo (*deep learning*) muito grandes (contendo dezenas ou centenas de bilhões de parâmetros) e pré treinados com uma enorme quantidade de dados (Amazon Web Services, 2024).

Atualmente, temos diversos modelos de *LLMs* no mercado, como o ChatGPT (disponível em <https://chatgpt.com/>), o LLaMA (TOUVRON et al., 2023), o DeepSeek (LIU et al., 2024a), entre outros.

A versatilidade de modelos e arquiteturas permite e fomenta estudos da aplicação das *LLMs* em diversas áreas, como psicologia (DEMSZKY et al., 2023), medicina (THIRUNAVUKARASU et al., 2023), entendimento de códigos de programação (NAM et al., 2024) e

também de *clippings* de notícias, devido a necessidade de interpretação em grande quantidade de conteúdo.

3.5.1 Como os LLMs podem auxiliar no cenário de notícias?

Durante a história do uso de LLMs para a geração de *clippings* ela foi usado em diversas etapas: resumo automático de notícias, categorização de notícias e sumarização automática.

As LLMs podem ser utilizadas para a sumarização de textos, como em (FALCÃO et al., 2024), que utiliza o BERT para verificar a similaridade entre notícias. Nesse trabalho, foi utilizado um conjunto de 3224 notícias que totalizaram 8,6 milhões de caracteres. Essas notícias foram analisadas pelas LLMs a fins de retirar as notícias que não possuíam similaridade direta com os temas buscados, facilitando a análise e a inserção de outras técnicas, resultando em uma redução total de caracteres em cerca de 75%, demonstrando a eficácia do BERT.

Já o artigo (AZHER et al., 2024) introduz o LimTopic, uma solução baseada em LLMs e suas integrações com o BERTopic, que busca extrair tópicos e resumir os textos automaticamente. O objetivo é produzir títulos e resumos coesos, preservando a mensagem principal das notícias, entregando ao usuário *insights* e facilitando a tomada de decisão em cenários de grandes volumes de informações. Em seus experimentos, ele utiliza diversas abordagens como técnicas engenharia de *prompt*, LLMs e técnicas de *fine-tuning*, avaliando em em diversos pontos, entre eles gramaticais, fluência de leitura e entendibilidade do texto.

3.6 Técnicas de Prompt Engineering para LLMs

No ramo computacional, um *prompt* é a entrada, normalmente em formato de texto, que enviamos para os modelos de Inteligência Artificial (IA), e, a engenharia de *prompt*, ou *prompt engineering*, é o processo de descoberta de *prompts* que geram resultados úteis de forma confiável (PHOENIX; TAYLOR, 2024).

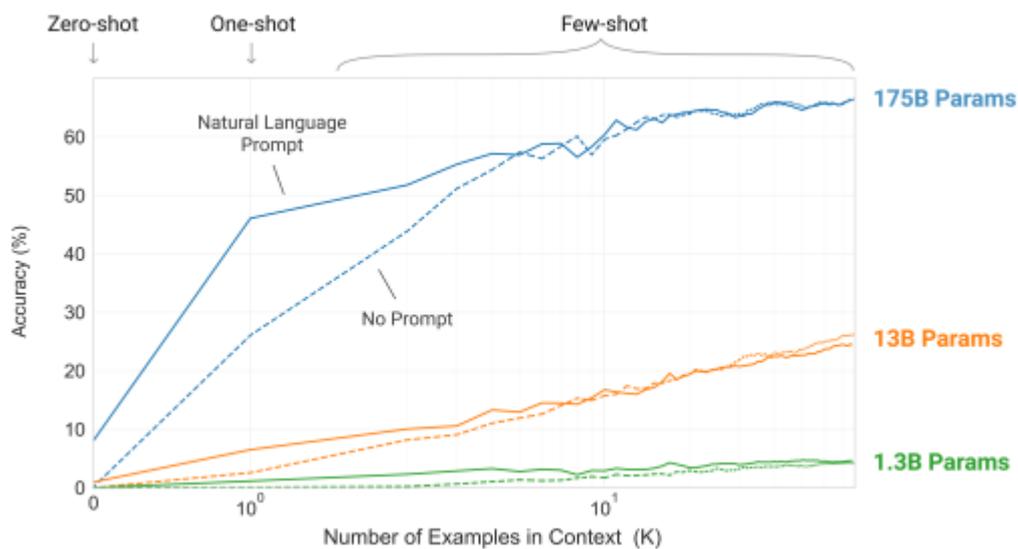
Desde que foi criada, a engenharia de *prompt* emerge como uma técnica crucial para melhorar a capacidade de retorno das LLMs e das Vision-language Models (VLMs) (SAHOO et al., 2024). Com o fomento da utilização das LLMs, com aumentos significativos especialmente após o lançamento do ChatGPT em novembro de 2022 (LIANG et al., 2024), a área continua sendo amplamente estudada, motivando os estudos e o desenvolvimento de novas técnicas (SCHULHOFF et al., 2024).

3.6.1 Técnicas de Prompt Engineering

Atualmente existem mais de 58 técnicas que utilizam diferentes abordagens de *prompt engineering* (SCHULHOFF et al., 2024) (disponível no anexo A), como *Zero-shot*, *few-shot*, *Chain of Thought* e *Self-consistency*.

O modelo *Zero-shot prompting* busca responder a pergunta ou comando sem nenhum exemplo fornecido (BROWN et al., 2020). Essa abordagem se baseia exclusivamente no conhecimento pré-treinado do modelo. Consequentemente, o modelo *Few-shot prompting* recebe exemplos antes da solicitação, melhorando a precisão das respostas ao reduzir ambiguidades através da contextualização do software. Os resultados do trabalho demonstram uma superioridade do *Few-shot* em todos os casos, especialmente quando o número de parâmetros da LLM cresce, como demonstra o gráfico 3.1.

Figura 3.1 – Imagem retirada do trabalho (BROWN et al., 2020) que demonstra um comparativo do desempenho entre modelos de *prompt engineering*



No gráfico da Figura 3.1 pode-se observar a acurácia média de modelos de 1.3, 13 e 175 bilhões de parâmetros quando comparamos as abordagens de *Zero-shot*, *One-shot* e *Few-shot* em *prompts* de linguagem natural e sem *prompt*, que demonstra a utilização dos modelos sem instruções adicionais. Através disso, pode-se analisar que o número cresce consideravelmente quando com maior número de exemplos (*Few-shot*), especialmente com o aumento de número de parâmetros, além de ser possível observar a baixa precisão quando não existem exemplos no contexto (*Zero-shot*).

A *Chain of thought (CoT)* (WU; ZHANG; HUANG, 2023) introduz *prompts* encadeados que incentivam o modelo a estruturar sua resposta passo a passo. Esse trabalho combina o uso de *CoT* com outras técnicas, como *self-consistency*, *few-shot* e *zero-shot*, onde obteve resultados consideravelmente melhores que o uso dessas técnicas isoladas.

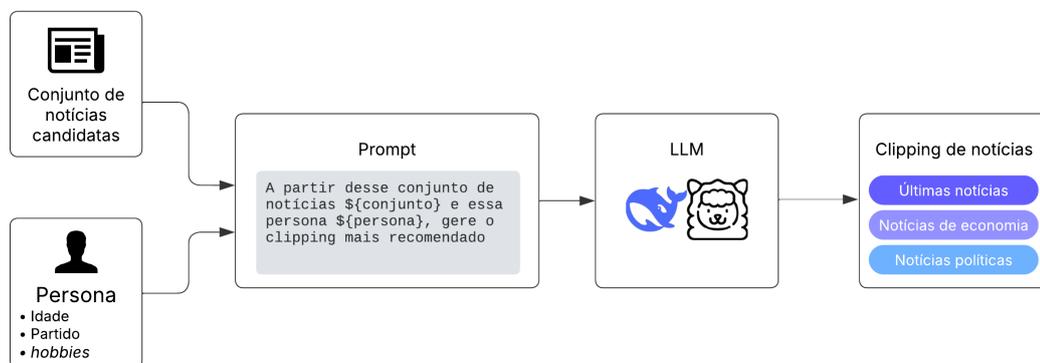
Já a *self-consistency*, introduzida em (WANG et al., 2023), é uma estratégia que melhora o desempenho do raciocínio em comparação as *CoT*. Algumas tarefas de raciocínio mais complexas e que possuem vários caminhos válidos, a *self-consistency* gera diversas cadeias de raciocínio e identifica a resposta final mais consistente, marginalizando as demais (SAHOO et al., 2024). O

trabalho demonstra que combinação de abordagens nessa técnica leva a melhoras significativas na precisão, como 17,9% no *dataset* Grade School Math 8K (GSM8K) e 11% no Simple Variations on Arithmetic Math word Problems (SVAMP).

4 Metodologia

A metodologia principal do presente trabalho está mostrada no fluxograma da Figura 4.1: as notícias candidatas e a persona compõem o *prompt*, juntamente com o restante da mensagem. Após ser montado e estruturado, o *prompt* é enviado para as LLMs selecionadas e elas retornam o *clipping* completo, contendo um conjunto de notícias recomendadas para a persona descrita.

Figura 4.1 – Fluxograma do funcionamento do sistema



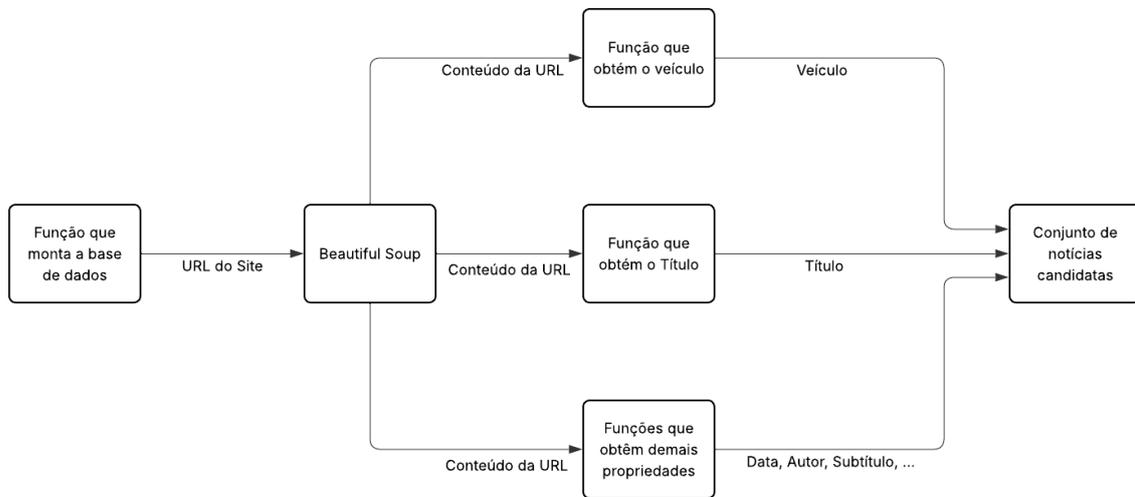
4.1 *Scraper*

A fim de montar uma base de notícias, o processo começa com uma raspagem de dados, através da ferramenta de *scraping* BeautifulSoup. Essa biblioteca está disponível em Python e é utilizada para extrair informações de sites, extraindo o conteúdo textual de páginas HTML e transformando em texto, onde podemos filtrar as informações que desejamos.

A Figura 4.2 demonstra o processo de montagem do conjunto de notícias candidatas. A função construída para montar a base de dados faz a requisição da função que comanda a biblioteca BeautifulSoup, enviando o *link* desejado. Após esse primeiro momento, é feita uma requisição a *url* recebida e envia o conteúdo encontrado para as funções responsáveis por extrair as propriedades como título, subtítulo, veículo, data de publicação, autores e categorias. Após a execução, as funções retornam as notícias encontradas para serem adicionadas ao conjunto de notícias candidatas.

No processo de extração das propriedades das notícias, foram utilizadas técnicas de HTML *parsing*, porém foram encontrados problemas com a codificação dos sites, proteção contra

Figura 4.2 – Fluxograma do funcionamento do sistema



bots e dinamismo das páginas. No entanto, com adaptações através do uso de **REGEX** e da manipulação do *JavaScript* das páginas, foi possível contornar esses desafios.

Exemplo de **REGEX** utilizado:

```
r 'Text__TextBase-sc-[a-zA-Z0-9\ -]+ [a-zA-Z0-9]+ noticia__titulo '
```

O **REGEX** apresentado é responsável por buscar os valores que começam com `Text__TextBase-sc-` e terminam com `noticia__titulo`, ignorando o conteúdo entre eles, pelo motivo dele ser volátil. Após a raspagem, os dados são tratados e armazenados em arquivos **Comma-separated values (CSV)**. Esses arquivos são utilizados para compor toda a base de dados, que será utilizada futuramente.

4.2 Códigos

Sobre o desenvolvimento é importante mencionar que todo o processo poderia existir em um mesmo arquivo / código. No entanto, o processo foi subdividido em arquivos para separar as responsabilidades, facilitando a manutenção e entendibilidade do código.

Com isso, foi criado um "molde"(chamada de classe interface), onde todos os *scrapers* seguem o mesmo padrão, tendo as mesmas funções implementadas. Além de denotar as funções que precisam estar presentes em cada *scraper*, ela também é responsável por alguns funções base que são comuns entre os raspadores, como a chamada da biblioteca *Beautiful Soup* e a exportação de dados via **CSV**.

Ademais, foi desenvolvida uma classe responsável por criar a base de dados ao realizar a chamada de todos os raspadores mencionados no item anterior. Para isso, foram desenvolvidas

três funções que foram utilizadas para gerar as notícias da categoria Economia, Política e Mais recentes, que são as características identificadas como importantes para o cenário parlamentar. A partir dessas funções, as notícias são armazenadas localmente na máquina em formatos **CSV**, para serem utilizadas como base de dados para a geração dos *clippings*.

Para organizar a comunicação com as **LLMs**, foi criado um processo responsável por carregar as bases de notícias armazenadas no passo anterior, extrair o conteúdo através da utilização da biblioteca **Pandas** e armazená-lo em estruturas **JavaScript Object Notation (JSON)**, montar o *prompt* com as notícias candidatas e a persona e realizar os envios para ao modelo selecionado.

Por fim, além dos desenvolvimentos citado, foi criada uma função para organizar as chamadas das principais funções: Criação da base de dados e Comunicação com a **LLM**. Todos os códigos estão disponíveis para análise no Apêndice **C**.

4.3 Testes preliminares

Antes da experimentação, foram realizados alguns testes, cujo objetivo foi pré-avaliar os *prompts* e a estratégia que mais se adapta ao cenário proposto (os modelos dos 3 testes estão disponíveis no Apêndice **D**).

Para os testes realizados, foram adotadas as seguintes tratativas:

- Teste 1: Testes com histórico de *clicks* em notícias de um usuário
- Teste 2: Testes com uma breve descrição do esteriótipo de um usuário, simulando brevemente uma persona
- Teste 3: Testes com personas mais complexas, com diversos dados e informações, como idade, gênero, localização, etc.

Para o primeiro experimento, foram feitos testes com histórico de *clicks* em notícias de um usuário fictício e foram fornecidos conjuntos de 10 notícias candidatas retiradas de veículos nacionais como **G1** e **O Globo** que foram organizadas em título e subtítulo. Para esse. No entanto, foi notado que muitas vezes os modelos não conseguiam recomendar notícias que se assemelhavam com o perfil, apenas realizavam recomendações baseadas em notícias que haviam semelhança na escrita do título, porém, não necessariamente eram do mesmo viés político. Por exemplo, recomendar notícias que tinham a palavra "Lula" para usuários que possuíam a palavra "Lula" no histórico.

Exemplo de *prompt* utilizado no primeiro teste:

```
"Baseado no meu historico de noticias {hist}, me recomende a noticia que eu
mais me interessaria: {news_df}"
```

Exemplo de resultado do primeiro teste e com o LLaMA 2 7b:

```
'Based on your news history , I can recommend the following articles that
are most likely to be of interest to you:\n\n
1. "Duda Salabert rega plantas na Praça Sete e procura transformar São
Paulo" – This article is likely to be of interest to you as it mentions
a notable figure in Brazilian politics and a current event in the city
of São Paulo , which is likely to be of relevance to you.\n
2. "Veja números da pesquisa Quaest em BH por sexo e faixa etária" – This
article provides statistical information on a recent survey conducted in
the state of Bahia , which may be of interest to you as it relates to
demographic data in Brazil.\n
3. "Metrô de BH: desapropriações na Linha 2 vão afetar condições de
trabalho
dos funcionários" – This article discusses current events related to
transportation in the state of Bahia , which may be of interest to you as
it relates to your local area.\n
4. "Pesquisa Quaest no Rio: Paes tem 57%; Ramagem é o segundo grupo mais
votado" – This article provides information on a recent survey conducted
in Rio de Janeiro , which may be of interest to you as it relates to
current events and political developments in the city.\n
5. "Twitter voltou? Usuário que acessar o X nesta sexta-feira pode ser o
primeiro desde o ato da Justiça" – This article discusses a recent event
related to social media , which may be of interest to you as it relates
to your online activities .\n
...
```

Nesse teste, a LLM buscou explicar e resumir as notícias fornecidas, categorizando as notícias, demonstrando quais os critérios utilizados para fazer a recomendação. No entanto, ela não levou em consideração o histórico que foi passado via *prompt*, retornando as notícias fornecidas na mesma ordem e indicando que o único processo feito foi o de categorização, o que foi considerado um resultado ruim para o experimento nas considerações do autor, pois nenhuma recomendação foi feita.

Para o segundo teste, alguns problemas se repetiram, como o modelo ser enviesado por alguma palavra chave, não levando em consideração a opinião sobre determinada ela. Além disso, em algumas requisições o sistema apenas realizava uma categorização das notícias.

Exemplo do segundo teste:

```
"Me indique apenas uma dessas noticias , a que mais me interessa ,
considerando que eu sou estudante de Ciências da computação {news}"
```

Exemplo de resultado do segundo teste:

```
Of all the news articles you've provided , the one that most catches my
attention as a student of Computer Science is "Veja números da pesquisa
Quaest em BH por sexo , idade , cor , renda , religião e escolaridade". This
```

article provides interesting insights into the demographics and voting preferences of the population in Belo Horizonte, which can be analyzed using various machine learning techniques.

For example, you could use clustering algorithms to group voters based on their demographic characteristics, such as age, gender, and income level. You could also use regression analysis to identify the factors that influence voting preferences, such as religion, education level, and political affiliation. Additionally, you could use sentiment analysis to analyze the tone of the responses in the survey and understand how they relate to the current political climate in Brazil.

Overall, this article provides a wealth of data that can be analyzed using various computational methods, providing valuable insights into the attitudes and beliefs of the population in Belo Horizonte.

A partir desse segundo experimento, pode-se observar uma melhora de desempenho, tendo um resultado mais próximo do esperado, entregando apenas uma notícia como solicitado via *prompt*. Ao analisar, nota-se uma longa explicação da motivação da LLM em construir a recomendação da notícia, realizando seu resumo e categorização.

O teste 3, com as personas sendo bem mais detalhadas, foram obtidos resultados mais próximo do que foi solicitado via *prompt*. Os problemas relatados durante os outros testes, como a repetição direta das notícias candidatas fornecidas, não aconteceram, demonstrando uma melhora em relação aos outros. Neste teste, as notícias foram fornecidas em formato JSON e a *persona* utilizada possui caráter político. Abaixo está demonstrado um exemplo de configuração do ensaio.

Prompt enviado:

"Minha persona: \{ persona \}. Conjunto de noticias candidatas: { news }. Com base na minha persona e nos meus interesses, selecione as 5 notícias mais relevantes entre as que eu forneci. Por favor, forneça um breve comentário explicando por que essas notícias foram escolhidas, destacando os aspectos que mais se alinham com minha persona."

Persona Utilizada

Mulher, 27 anos, deputada estadual pelo PV é uma parlamentar de esquerda, formava em bioquímica que atualmente mora em Belo Horizonte. Frequentou a universidade pública e tem como pautas principais direito das mulheres, educação pública de qualidade e meio ambiente. Costuma se informar por veículos tradicionais como a Folha de São Paulo, mas também de portais como a revista Piauí. Ouve podcasts como o Café da Manhã da Folha e O assunto. Se interessa por cultura brasileira e música popular brasileira. Pratica Pilates e Musculação.

Exemplo de notícia enviada:

```

[ {
  'title ': 'PF cumpre 17 mandados de prisão contra esquema de desvio de
    emendas ',
  'url ': 'https://tiny.url/s12gds92 ',
  'category ': 'poder ',
  'siteName ': 'folha ',
  'author ': 'Demétrio Vecchioli ',
  'date\_published ': '2024-12-10 11:35:00 '
} ]

```

Resposta recebida:

```

\# Aqui estão as 5 notícias mais relevantes para a sua persona:

\# 1. "Relatório da PF: Veja a cronologia das reuniões em que Bolsonaro
    tratou do golpe" – Esta notícia é relevante porque aborda a investigação
    da PF sobre o plano golpista de Jair Bolsonaro, o que pode ser de
    grande interesse para quem está ciente das discussões e controvérsias
    políticas atuais.

\# 2. "Apresentação de minuta do golpe, ataque ao sistema eleitoral, plano
    de fuga: PF vê Bolsonaro no centro da trama golpista" – Esta notícia é
    relevante porque fornece detalhes sobre a investigação da PF e como
    Bolsonaro está sendo visto como o responsável pelo plano golpista.

\# 3. "Plano golpista era delirante, e relatório da PF mostra isso" – Esta
    notícia é relevante porque descreve os detalhes do plano golpista
    descobertos pela PF e como ele foi considerado "delirante".

\# 4. "Bolsonaro planejou, atuou e teve domínio dos atos no plano golpista,
    diz PF" – Esta notícia é relevante porque confirma a investigação da PF
    sobre a participação de Bolsonaro no plano golpista.

\# 5. "Por unanimidade, STF valida uso de símbolos religiosos em órgãos pú
    blicos" – Esta notícia não está diretamente relacionada a investigação
    do golpe, mas pode ser relevante para quem está interessado em questões
    políticas e jurídicas.

```

Neste experimento, já podemos observar um breve *clipping* montado para a persona fornecida. Nele podemos observar um breve resumo da motivação da indicação, assim como solicitado, e as notícias na ordem de relevância para a persona, embora ainda não tenha sido feita a análise qualitativa da resposta. Através disso, os testes com personas mais complexas se mostraram mais efetivos, com uma melhor padronização da resposta e menos problemas diretos, por isso foram adotados durante os experimentos.

4.4 Pré-experimentação

Os testes pré experimentais foram utilizados para validar o modelo de *prompt*, nas LLMs e os *scrapers*. Para isso, as execuções pré-experimentais foram utilizadas as seguintes configurações:

- O modelo das notícias enviadas foi reduzido de título, subtítulo, link, autor e data para apenas título, link por notar uma sobrecarga piora dos resultados quando utilizado de mais informações;
- Foram utilizadas as versões 7B do Deepseek-r1 e 3B do LLaMA 3.2.
- Somente 1 modelo de *prompt*

Durante 1 semana foram gerados 8 *clippings* diferentes todos os dias, um para cada combinação, às 8 horas da manhã, combinando as seguintes características:

- As regiões Brasil e Minas Gerais
- Os modelos LLaMA 3.2 e Deepseek-r1
- 2 Personas

Para a geração dos *clippings*, foram utilizados ao todo 11 veículos de notícias online. Como foi citado anteriormente, os *clippings* gerados durante o experimento possuem um caráter político, e por isso foram escolhidas revistas online que possuem esse padrão. São eles:

- Folha
- Estadão
- Uol
- O Globo
- Metrôpoles
- Valor econômico
- G1
- Estado de Minas
- Itatiaia
- O Fator
- O Tempo

As personas utilizadas no processo foram:

- Persona 1 = "Homem, 42 anos, deputado estadual pelo partido PSD, morador de Belo Horizonte, formado em economia por uma universidade particular. É um parlamentar de centro-direita, que tem como pautas principais, saúde, segurança e educação com responsabilidade nas contas públicas. Costuma se informar através de veículos tradicionais como o Estadão, Folha de São Paulo, BBC nos temas nacionais. Para os assuntos de Minas, utiliza os canais O Estado de Minas, O tempo e Itatiaia. Pratica esportes como canoagem."
- Persona 2 = Mulher, 27 anos, deputada estadual pelo PV é uma parlamentar de esquerda, formada em bioquímica que atualmente mora em Belo Horizonte. Frequentou a universidade pública e tem como pautas principais direito das mulheres, educação pública de qualidade e meio ambiente. Costuma se informar por veículos tradicionais como a Folha de São Paulo, mas também de portais como a revista Piauí. Ouve podcasts como o Café da Manhã da Folha e O assunto. Se interessa por cultura brasileira e música popular brasileira. Pratica Pilates e Musculação."

Ao realizar os testes utilizando as notícias com todas as informações demonstradas anteriormente, como evidenciado nos testes preliminares, foi notada uma falta de padronização das respostas e uma qualidade inferior, isso é causado devido ao volume de dados enviado, que era excessivamente grande por conta do alto número de notícias. Uma solução encontrada foi reduzir o escopo das mensagens e enviar apenas o seguinte formato de título e subtítulo.

Modelo de notícias enviados:

```
[[
  'title ': 'PF cumpre 17 mandados de prisão contra esquema de desvio de
    emendas ',
  'url ': 'https://tiny.url/s12gds92 '
]]
```

Prompt enviado:

```
Minha persona: / My persona:

"{ persona }"

Conjunto de Notícias candidatas: / Set of candidate news:

- Notícias mais recentes / The newest news {newstdb}
- Notícias políticas / Political news {politicdb}
- Notícias econômicas / Economic news {economicdb}

Exemplo de saída / Output example:

"TILULO 1", "link";
"TILULO 2", "link";
```

"TILULO 3", "link";
 "TILULO 4", "link";
 "TILULO 5", "link";

****Regras obrigatórias / restricted rules:****

1. Não invente notícias. Apenas retorne itens do conjunto fornecido.
2. Não altere os links originais.
3. Não retorne apenas notícias de um único veículo, varie as respostas.
4. Não envie nada do tipo: "TILULO DA NOTICIA 1" ou "https://exemplo.com/noticial" como resposta, mande apenas links reais.
5. Não envie nada diferentes dos títulos e links reais.
6. Seja direto e objetivo.
7. Não fale nada além do que foi pedido.

Crie um clipping das 5 notícias mais indicadas para minha persona em cada categoria fornecida, ou seja, 15 notícias no total, 5 de cada categoria.

As categorias são as seguintes:

- Notícias mais recentes / The newest news
- Notícias políticas / Political news
- Notícias econômicas / Economic news

Answer in Portuguese

Exemplo de resposta recebida utilizando a persona 1 e o LLaMA 3.2:

Vou criar um clipping das 5 notícias mais indicadas para sua persona em cada categoria. Aqui estão as respostas:

****Notícias mais recentes****

1. ****BC reduz reservas de dólares e deixa Brasil mais vulnerável inflação**** – <https://tinyurl.com/2cjvs54p>
2. ****Lula promete crédito consignado privado, isenção fiscal para pequenas empresas**** – <https://tinyurl.com/2bmvzjsr>
3. ****André Esteves fura bolha anti-Trump e louva o governo Biden**** – <https://tinyurl.com/28vumv2x>
4. ****Carro brasileiro volta a fazer sucesso na Argentina**** – <https://tinyurl.com/29jgopm9>
5. ****PIB de Cingapura cresce 4,4% em 2024 em base a variações mensais**** – <https://tinyurl.com/23rsgnm5>

****Notícias políticas****

1. ****Lula anuncia projeto de investimento em infraestrutura para criar emprego**** – <https://tinyurl.com/2bmvzjsr>
2. ****Haddad afirma que governo não vai aumentar impostos**** – <https://>

tinyurl.com/25vrop66

3. ****Ações do Congresso Brasileiro vão ser monitoradas pelo governo**** – <https://tinyurl.com/29qjutcv>
4. ****Ministro da Justiça afirma que governo está preparado para lidar com protestos**** – <https://tinyurl.com/2al43h6v>
5. ****Presidente Bolsonaro faz declarações controversas sobre imigração**** – <https://tinyurl.com/24n3dx98>

****Notícias econômicas****

1. ****BC reduz reservas de dólares e deixa Brasil mais vulnerável inflação**** – <https://tinyurl.com/2cjvs54p>
2. ****Lula promete crédito consignado privado, isenção fiscal para pequenas empresas**** – <https://tinyurl.com/2bmvezsr>
3. ****Investimentos estrangeiros na Brasil estão aumentando**** – <https://tinyurl.com/28vumv2x>
4. ****Mercado de ações brasileiro está volátil**** – <https://tinyurl.com/29jgopm9>
5. ****PIB de Cingapura cresce 4,4% em 2024 em base a variações mensais**** – <https://tinyurl.com/23rsgnm5>

Lembre-se de que essas são apenas sugestões e não é recomendado tomar decisões financeiras ou políticas com base em notícias. sempre importante fazer sua própria pesquisa e análise antes de tomar qualquer decisão.

4.5 Experimentação

4.5.1 Configuração

Para a experimentação, foi utilizada a seguinte configuração:

- As notícias foram enviadas no padrão: título, *link* e veículo;
- Foram mantidas as personas e os veículos da pré-experimentação;
- Foram utilizadas as versões 7B do Mistral e 3B do LLaMA 3.2;
- Foram utilizados 2 modelos de *prompt* diferentes, a fim de avaliar diferentes envios.

Para a experimentação, versão do 7B do Deepseek-v1 utilizada na pré-experimentação foi substituída pela versão 7B do Mistral devido ao baixo desempenho apresentado durante os testes.

Para avaliar o desempenho das LLMs que foram utilizadas durante a geração dos *clippings* (Mistral 7B e LLaMA 3.2 3B), por 40 dias foram gerados 2 *clippings* por dia, sendo 1 para cada persona utilizada (2 no total), onde cada clipping possuía 30 notícias, onde 15 notícias são de

veículos da região Minas Gerais e 15 notícias são de veículos voltados para o Brasil como um todo. Para isso, foram feitos 4 envios por dia as **LLMs**, tendo as seguintes combinações:

- Persona 1 com o conjunto de notícias da região de Minas
- Persona 1 com o conjunto de notícias do Brasil
- Persona 2 com o conjunto de notícias da região de Minas
- Persona 2 com o conjunto de notícias do Brasil

A partir das quatro combinações, é formado dois *clipping* de 30 notícias, um para cada persona.

Para variar a utilização das **LLMs** e os *prompts*, a fim de evitar enviesamento, foi aleatoriamente ao longo dos 40 dias qual seria a **LLMs** utilizada no referente dia (a organização dos envios está disponível no Apêndice B).

4.5.2 Exemplos

Os 2 *prompts* a seguir foram utilizados durante toda a experimentação, neles é possível notar uma diferença na estruturação, embora os 2 utilizem o mesmo padrão de notícias a serem enviadas e as mesmas personas.

Prompt 1

```

---
Imagine que você é um assessor de imprensa e deve criar um clipping de
  notícias para uma persona específica. /
Você deve seguir as seguintes instruções restritamente:

Minha persona: / My persona:

"{ persona }"

Conjunto de Notícias candidatas: / Set of candidate news:

- Notícias mais recentes / The newest news {newstdb}
- Notícias políticas / Political news {politicdb}
- Notícias econômicas / Economic news {economicdb}

Exemplo de saída / Output example:
  "TILULO 1", veiculo, "link";
  "TILULO 2", veiculo, "link";
  "TILULO 3", veiculo, "link";
  "TILULO 4", veiculo, "link";
  "TILULO 5", veiculo, "link";

**Regras obrigatórias / restricted rules:**

```

- 1 NAO invente notícias. APENAS retorne itens do conjunto de noticias candidatas fornecido. / Do not invent news. Only return items from the provided set.
- 2 NAO altere os links originais. / Do not alter the original links.
- 3 NAO retorne apenas notícias de um único veículo, VARIE as repostas. / Do not return news from a single source, VARY the answers.
- 4 NAO envie nada do tipo: "TILULO 1" ou "link" como resposta, mande apenas links reais / Do not send nothing of "TILULO 1", "link" as an answer.
- 5 NAO envie nada diferentes dos títulos, fonte e links reais. / Do not send anything different from the actual titles, site and links.
- 6 Seja Direto e objetivo. / Be direct and objective.
- 7 NAO fale nada além do que foi pedido / Do not say anything beyond what was asked.

Crie um clipping das 5 noticias mais indicadas para minha persona em cada categoria / database fornecida, ou seja, 15 noticias no total, 5 de cada categoria. / Create a clipping of the 5 most indicated news for my persona in each category / provided database, that is, 15 news in total, 5 from each category.

As categorias são as seguintes: / The categories are as follows:

- Notícias mais recentes / The newest news
- Notícias políticas / Political news
- Notícias econômicas / Economic news

Answer in Portuguese

Prompt 2

Imagine que voce é um assessor de um político e deve seguir o seguinte passo a passo:

PASSO 1

leia as noticias do conjunto de noticias
conjunto de noticias para ser usado = [{news}]

PASSO 2

Retire titulo e url do conjunto de noticias e apenas do conjunto de noticias fornecido exatamente 5 noticias sobre politica, 5 noticias sobre economia e 5 noticias recentes que mais combinam com os temas na lista a baixo.

Analise a persona a baixo e retorne conforme o template de lista python a baixo no minimo 10 palavras e temas chaves pelos quais esta pessoa se interessa.

A saida deve apresentar apenas a lista com os temas.

Exemplo saida: "lista = ['tema 1', 'tema 2', 'tema 3', 'tema 4',...]"

Persona: {persona}

PASSO 3

Escreva exatamente como recebido as 5 notícias de cada tópico (política, economia, recentes) selecionadas seguindo estritamente o formato:

Descrição do formato: bloco de título e url das 5 notícias escolhidas de cada tópico

política

" *nome da notícia* *url da notícia* "

economia

" *nome da notícia* *url da notícia* "

recentes

" *nome da notícia* *url da notícia* "

"

Exemplo de resposta recebida no LLaMa 3.2 3B:

Vou criar um clipping das 5 notícias mais indicadas para sua persona em cada categoria.

****Notícias mais recentes****

1. "Acordo comercial ruim com os EUA mostra os limites da cooperação econômica" – Estadão (link: <https://tinyurl.com/2y6k2e4o>)
2. "Casa Branca defende demissão de autoridade trânsito no Brasil" – G1 (link: <https://tinyurl.com/24netc7e>)
3. "Tarifaço: setores afetados calculam prejuízos" – G1 (link: <https://tinyurl.com/2aw3xatj>)
4. "China libera 183 empresas do Brasil para exportar" – G1 (link: <https://tinyurl.com/2czb93b7>)
5. "Safrá do sorgo anima produtores, mas previsões são caóticas" – G1 (link: <https://tinyurl.com/2475aycx>)

****Notícias políticas****

1. "Trump e a demissão da chefe de estatística: Hi, é uma coisa séria?" – Estadão (link: <https://tinyurl.com/2b6ex8ze>)
2. "Política para data centers deve incentivar produção" – Estadão (link: <https://tinyurl.com/2d15cbkl>)
3. "Foco em transição climática marca criação da agenda do dia" – G1 (link: <https://tinyurl.com/28v4564d>)
4. "Seleção de ações tem bancos e empresas que escutam o público" – G1 (link: <https://tinyurl.com/2b8sh12g>)
5. "Maior rede de finanças descentralizadas ensaia modelo" – G1 (link: <https://tinyurl.com/27us22x2>)

****Notícias econômicas****

1. "Skaf volta a Fiesp em momento de inflexão da indústria" – Estadão (link : <https://tinyurl.com/25qh8hr3>)
2. "Seremidade é sinal de firmeza; por isso, BC e outros acordam com o mercado" – Estadão (link: <https://tinyurl.com/2xwtkder>)
3. "A COP-30 está aí: como o setor de seguros será afetado?" – Estadão (link: <https://tinyurl.com/2c3selts>)
4. "Refinaria dos anos 80 em SP entra no radar da área de energia" – Folha (link: <https://tinyurl.com/27wxnvbv>)
5. "Ray Dalio vende últimas ações que tinha na BRF" – Folha (link: <https://tinyurl.com/262fan4b>)

Espero que essas notícias sejam úteis para você!

Nesse exemplo pode-se observar o *clipping* com a estruturação o que foi solicitada: 5 notícias de cada tema, notícias padronizadas com título, veículo e *link*, os 3 temas estão bem definidos, as notícias retornadas estão contidas na base de dados fornecida, os *links* não estão corrompidos e os títulos não estão trocados. Todos os pontos mencionados demonstram grande capacidade das *LLMs* em padronizarem os *clippings*, embora não esteja sendo avaliado a capacidade de associar as notícias com as personas fornecidas.

5 Resultados

5.1 Resultados da pré-experimentação

Para avaliar o desempenho das **LLMs** que foram utilizadas durante a geração dos *clippings* (Deepseek-r1 7B e LLaMA 3.2 3B), 40 conjuntos de notícias que foram gerados durante uma semana foram analisados, utilizando as seguintes métricas técnicas (os *links* de todos os *clippings* gerados constam no Apêndice A.1):

- **Padronização das respostas:** Avaliação da estrutura da resposta obtida da **LLM**, como trazer o número correto de notícias, categorias, links e títulos.
- **Funcionalidade dos links fornecidos:** Análise do endereço fornecido, verificando sua acessibilidade.
- **Notícias contidas no conjunto fornecido:** Verificação se todas as notícias recebidas na resposta estão contidas no conjunto de notícias candidatas fornecidas.
- **Variedade de veículos:** Análise da capacidade de diversificação das fontes utilizadas no *clipping*, para evitar enviesamento.

Por se tratarem de testes pré-experimentais, as avaliações foram simples de forma padronizada e não envolveram analisadores externos. Com isso foram atribuídas notas de 0-10 para cada critério, resultando na seguinte média para cada modelo de **LLM**:

Métrica	Deepseek-r1	LLaMA 3.2
Padronização das respostas	6.35	9.7
Funcionalidade dos links fornecidos	4.50	7.21
Notícias contidas no conjunto fornecido	6.65	9.7
Variedade de veículos	4.80	5.26
Média Geral	5,57	7.96

Tabela 5.1 – Comparação entre Deepseek-r1 e LLaMA 3.2 nas métricas avaliadas

Analisando os resultados, vemos que, de forma geral, o modelo **LLaMA 3.2** obteve um desempenho superior quando comparado ao **Deepseek-r1**, lembrando que foram utilizadas as versões com 7 bilhões de parâmetros do Deepseek e 3 bilhões do LLaMA 3.2 e com um *prompt* em inglês e português.

5.1.1 Problemas encontrados

Durante a análise da pré-experimentação, foram notados diversos problemas nos *clippings* recebidos, mesmo quando utilizado o mesmo *prompt*.

Exemplo de clipping gerado pelo LLaMA 3.2:

Claro! Aqui estão os 15 notícias mais indicadas para sua persona, 5 de cada categoria:

1. ****TILULO 81****, "<https://tinyurl.com/2btwx4qm>" - Após um ano, teto de juros do cartão tem pouco impacto na economia
2. ****TILULO 77****, "<https://tinyurl.com/2b9cva6d>" - Portabilidade não atrai interesse de bancos
3. ****TILULO 79****, "<https://tinyurl.com/23jjulkc>" - Fundos de universidades aderem a bitcoin e criptoativos
4. ****TILULO 80****, "<https://tinyurl.com/2h4cqbbp>" - Dólar fecha em baixa e Ibovespa termina em alta
5. ****TILUTO 76****, "<https://tinyurl.com/2btwx4qm>" - Bombril entra com pedido de recuperação judicial

****Notícias políticas****

1. ****TILULO 45****, "<https://tinyurl.com/27e7wrjz>" - Brasil recua em índice de percepção de corrupção
2. ****TILUTO 48****, "<https://tinyurl.com/2bqcfjer>" - Governo Lula ainda não teve 'voo de brigadeiro' para solucionar problemas do país
3. ****TILULO 49****, "<https://tinyurl.com/2yln9he7>" - AGU pede que STF rejeite queixa-crime de Flávio Doriges
4. ****TILUTO 43****, "<https://tinyurl.com/27e3g9oj>" - PSDB começa a definir futuro político em meio a crise do partido
5. ****TILUTO 46****, "<https://tinyurl.com/28fv4tod>" - Ministro da Defesa defende punições diferentes para envolvidos no escândalo

****Notícias econômicas****

1. ****TILULO 3****, "<https://tinyurl.com/24xnrsoz>" - Bombril entra com pedido de recuperação judicial
2. ****TILULO 4****, "<https://tinyurl.com/2aefwp5n>" - Mercado eleva previsão de inflação para 2025 por causa da mistura de biodiesel
3. ****TILULO 1****, "<https://tinyurl.com/2c44eral>" - Trump assina decreto que impõe tarifas de 25% sobre carne brasileira
4. ****TILUTO 2****, "<https://tinyurl.com/2agjl59c>" - Novas ações trabalhistas superam 2 milhões em empresas do país
5. ****TILULO 5****, "<https://tinyurl.com/23jjulkc>" - Fundos de universidades aderem a bitcoin e criptoativos

Nesse clipping gerado, podemos notar uma correta estruturação das notícias, uma ótima padroni-

zação, exibindo o título e o link funcional das 15 notícias solicitadas, separadas em tópicos, tudo conforme foi requisitado no *prompt*. No entanto, foi notada pouca diversificação de veículos de notícias.

Exemplo de clipping gerado pelo Deepseek:

```
" Notícias mais recentes ":
- Notícias da revista: "Revista Efeio lança jornal de negociação com Dir
  compulsório e direito a vencer." https://tinyurl.com/26k4753x
- Notícias da revista: "Revista Efeio lança jornal de negociação com Dir
  compulsório e direito a vencer." https://tinyurl.com/26k4753x

" Notícias políticas ":
- Notícias da revista: "Governo estadual SP: candidatos Luiz Inácio Lacerda
  e Celso Amorim." https://tinyurl.com/25vrmz2a
- Notícias da revista: "Governo estadual SP: candidatos Luiz Inácio Lacerda
  e Celso Amorim." https://tinyurl.com/25vrmz2a

" Notícias econômicas ":
- Notícias da revista: "BC reduz reservas de dólares e deixa Brasil mais
  fragile em changed." https://tinyurl.com/2cjvs54p
- Notícias da revista: "Etanol brasileiro entrou na mira de polêmica com os
  EUA." https://tinyurl.com/2c4e87d2
```

Já nesse exemplo de clipping obtido do Deepseek-r1 vemos uma falta alguns fatores na padronização, como poucas notícias (foram requisitadas 15 e retornadas apenas 6), notícias repetidas, endereços não funcionais e também pouca diversificação de veículos de notícias.

Portanto, durante toda a análise dos resultados, foram encontrados os seguintes problemas nas respostas:

- O conjunto de veículos não está bem diversificado
- Respostas sem endereço ou sem títulos
- Respostas com endereços quebrados / não funcionais
- Respostas fora do padrão, retornando um número errado de notícias
- Respostas contendo notícias repetidas no conjunto.
- Centralidade no posicionamento político, sem favorecendo algum dos lados.
- Falta de relação com os *hobbies* das personas

5.2 Resultados da experimentação

Para avaliar a experimentação de maneira imparcial e mais confiável, no momento em que eram construídos, os *clippings* foram enviados para 2 avaliadores, 1 para cada persona, e foram responsáveis por responder formulários (criados utilizando a plataforma do Google Forms) e avaliar os *clippings* que possuía as seguintes perguntas (os *links* de todos os *clippings* gerados constam no Apêndice A.2):

- As notícias apresentam erros técnicos (*links* quebrados, títulos errados, veículos trocados)
- O *Clipping* gerado apresenta diversidade de fontes de informação, evitando a repetição de veículos
- O *Clipping* varia os temas das notícias
- A notícia apresentada está alinhada com os interesses, perfil e contexto da persona simulada
- O *Clipping* possui viés político
- O *Clipping* possui o viés político da persona
- O *Clipping* traz notícias dos canais em que persona alvo se informa
- Houve alguma notícia relevante e o *clipping* não trouxe

Dentro de cada pergunta do formulário, foram criadas alternativas utilizando a escala Likert com 5 opções de respostas: Discordo totalmente, Discordo parcialmente, Neutro, Concordo parcialmente e Concordo totalmente. Considerando a nota 1 sendo a opção Discordo totalmente e a 5 como sendo a opção Concordo totalmente, na Tabela 5.2 estão demonstradas as médias que foram obtidas em 35 respostas.

A pergunta: As notícias apresentam erros técnicos (*links* quebrados, títulos errados, veículos trocados) busca avaliar se existiram erros técnicos dentro dos *clippings* gerados. Nessa avaliação, uma maior a nota representa um pior desempenho da *LLMs*. Ao analisar a nota, observa-se uma média geral de 4,64 pontos de 5 possíveis, o que demonstra que grande parte dos *clippings* enviados apresentaram problemas. Comparando os desempenhos dos *prompts* nesta pergunta o Prompt 2 se saiu levemente melhor, com uma nota de 4,59 pontos, contra 4,68 pontos do Prompt 1. Se tratando de *LLMs*, o LLaMA demonstrou uma média pior, com 4,80 pontos, contra 4,47 pontos de média encontrados no Mistral.

Já as perguntas: A notícia apresentada está alinhada com os interesses, perfil e contexto da persona simulada e O *Clipping* possui o viés político da persona buscam avaliar a capacidade das *LLMs* em relacionar a persona com a notícia fornecida. Nessa avaliação, notas maiores representam um melhor desempenho e, ao analisar as notas, observa-se, respectivamente, médias gerais de 3,96 e 3,89 pontos de 5 pontos possíveis. Para essas perguntas, foram obtidas notas

bem próximas para os diferentes *prompts* e LLMs, com uma leve superioridade do Prompt 1 e do LLaMA 3.2 3B.

Ademais, as perguntas: O *Clipping* gerado apresenta diversidade de fontes de informação, evitando a repetição de veículos; O *Clipping* varia os temas das notícias; O *Clipping* possui viés político; e O *Clipping* traz notícias dos canais em que persona alvo se informa; exploram a capacidade de gerar *clippings* voltados para o cenário político que tenham uma boa diversidade de notícias e fontes, buscando dar prioridade para os canais que a persona se informa e para notícias com viés político. Para essas perguntas, notas maiores representam um melhor desempenho. A partir disso, obtivemos médias gerais de, respectivamente, 4,09, 4,40, 3,83 e 4,11 pontos, de 5 pontos possíveis.

Por fim, a pergunta: Houve alguma notícia relevante e o *clipping* não trouxe; busca avaliar se os modelos possuem capacidade de entender automaticamente possíveis notícias relevantes, seja pela repetição no banco de dados fornecido ou por qualquer motivo implícito adotado pelo sistema. Contudo, todas as avaliações recebidas indicaram 3 pontos nessa pergunta, o que representa uma neutralidade em todas as respostas.

Tabela 5.2 – Comparação entre LLaMA 3.2 3B e Mistral 7B em dois *prompts* nas métricas avaliadas

Métrica	LLaMA 3.2		Mistral		Média Geral
	Prompt 1	Prompt 2	Prompt 1	Prompt 2	
As notícias apresentam erros técnicos?	4,75	4,86	4,62	4,33	4,64
O clipping gerado apresenta diversidade de fontes de informação	4,25	3,79	4,12	4,22	4,09
O Clipping varia os temas das notícias	5,00	4,29	3,87	4,44	4,40
As notícias apresentadas estão alinhadas com os interesses, perfil e contexto da persona simulada	4,00	4,07	4,00	3,78	3,96
O Clipping possui viés político	4,00	3,43	4,00	3,89	3,83
O Clipping possui o viés político da persona	4,00	3,79	4,00	3,78	3,89
O Clipping traz notícias dos canais em que persona alvo se informa	4,75	3,71	4,00	4,00	4,11
Houve alguma notícia relevante e o clipping não trouxe	3,00	3,00	3,00	3,00	3,00

Através da análise dos resultados, pode-se observar que, dentro da escala proposta, houve resultados positivos: a diversidade das fontes de informação, a variação nos temas das notícias, o alinhamento com o perfil e o viés político das notícias. No entanto, foi possível observar a quantidade de erros técnicos como *links* quebrados e títulos errados, apresentando valores próximos da pontuação máxima (5), já demonstrando parte dos problemas encontrados.

Para validar os *links* e títulos fornecidos nos *clippings* gerados, eles foram submetidos a um *script* que combina as bibliotecas `Requests` (utilizada para realizar requisições `HTTP`) e `RapidFuzz` (utilizado para comparar os títulos). Para isso, os *clippings* enviados foram transformados em `JSON` e, a partir disso, foi realizada uma requisição nos endereços do *clipping* e, caso a requisição retornasse um código 200, foi considerado uma requisição de sucesso. A partir dessa requisição de sucesso, foi utilizado um comparador de *string* (`RapidFuzz`) com um critério de similaridade em 80, ou seja, caso o título tenha uma similaridade acima de 80%, ele foi considerado correto.

A partir da validação de títulos e *links* foram obtidos os resultados da tabela 5.3. Através desses resultados é possível notar a superioridade de 11% do LLaMA 3.2 em relação ao Mistral na métrica de *links* válidos, que avalia a capacidade da LLM de organizar os endereços fornecidos sem corrompê-los. Sobre a avaliação da taxa de títulos corretos, que busca avaliar se o título fornecido pelo *clipping* condiz com o título verdadeiro da notícia, o LLaMA apresenta uma superioridade de 9% em relação ao Mistral, demonstrando novamente sua superioridade. Com isso, nas duas métricas técnicas avaliadas de maneira automática, o LLaMA 3.2 3B apresentou uma média geral 10% melhor em relação ao Mistral 7B. No entanto, os resultados, que apresentam médias de 67,5% e 77,5%, são ruins, pois as tarefas avaliadas nesse ponto eram apenas de selecionar e exibir, não foi solicitada as LLMs nenhuma tratativa nos endereços e títulos, ou seja, era esperada uma média próxima a 100%.

Tabela 5.3 – Comparação entre Mistral e LLaMA 3.2

Métrica	Mistral	LLaMA 3.2
Taxa de <i>links</i> válidos	78%	89%
Taxa de títulos corretos	57%	66%
Média Geral	67,5%	77,5%

5.2.1 Problemas encontrados

Assim como na pré-experimentação, alguns erros persistiram. Os principais foram:

- *Links* não funcionais
- *Links* trocados
- Veículos faltando

- Erros graves que alteram a compreensão do leitor
- Falta de título

Neste exemplo de retorno, pode-se observar diversos dos problemas comentados anteriormente: a falta do título, falta do veículo e *links* não funcionais, se mostrando um *clipping* altamente ineficaz para o cenário proposto.

```
"TILULO 1", veiculo , "<https://www.folha.uol.com.br/poder/noticia/2023/04/lula-pede-liga%C3%A7ao-e-conversa-com-xi-jinping-sozinho-antes-de-viagem-a-china_51966831?utm_source=rss&utm_medium=referral>";
"TILULO 2", veiculo , "<https://www.folha.uol.com.br/poder/noticia/2023/04/lula-inicia-visita-a-china-em-busca-de-ajuda-para-fazer-o-acordo-de-cooperacao-em-energia_51964028?utm_source=rss&utm_medium=referral>";
"TILULO 3", veiculo , "<https://www.folha.uol.com.br/poder/noticia/2023/04/lula-devera-se-apresentar-ao-congresso-dos-estados-unidos-em-2024_51961805?utm_source=rss&utm_medium=referral>";
"TILULO 4", veiculo , "<https://www.folha.uol.com.br/poder/noticia/2023/04/lula-devera-entregar-relatorio-sobre-acordo-de-cooperacao-em-energia_51961804?utm_source=rss&utm_medium=referral>";
"TILULO 5", veiculo , "<https://www.folha.uol.com.br/poder/noticia/2023/04/lula-deve-visitar-china-em-maio_51960777?utm_source=rss&utm_medium=referral>";
```

Já neste exemplo, é possível observar a ausência de *urls* no *clipping* gerado, que ocorreu sem corromper os títulos e os veículos, demonstrando mais um obstáculo enfrentado durante a experimentação.

1. "Notícias mais recentes"
 - "Nova luta do governo contra a pandemia: medidas estritas para combater o surto de Covid-19" (veículo: Folha de S. Paulo)
 - "Tragédia em Minas Gerais: ônibus cai numa ponte, com mortes e feridos" (veículo: UOL)
 - "Escândalo no Senado Federal: senador acusado de irregularidades em compra de vacinas" (veículo: G1)
 - "Clima catastrófico previsto para o Brasil no final do ano, segundo relatório da ONU" (veículo: Folha de S. Paulo)
 - "Estudo mostra que a Amazonas está se recuperando mais lentamente do que outras áreas" (veículo: UOL)
2. "Notícias políticas"
 - "Aumento nas taxas de transporte em São Paulo: o que será o impacto e como isso pode afetar a população?" (veículo: Folha de S. Paulo)
 - "Novo desafio para Bolsonaro: ex-ministro aponta que ele estava ciente da irregularidade no Senado" (veículo: G1)

- "Poder executivo anuncia novas medidas contra a violência nas prisões do Brasil" (veículo: UOL)
- "Ministério Público pediu investigação de senadores por irregularidades em compra de vacinas" (veículo: Folha de S. Paulo)
- "Senador apresentou projeto para retirar a imunidade parlamentar temporária" (veículo: G1)

3. "Notícias econômicas"

- "Inflação no mês de agosto está mais alta que o esperado, segundo IPCA" (veículo: UOL)
- "Pesquisa mostra que a economia brasileira está se recuperando lentamente da pandemia" (veículo: Folha de S. Paulo)
- "Fábrica de celulares com reabertura parcial em São Bernardo do Campo" (veículo: UOL)
- "O que você precisa saber sobre as novas tarifas de Trump e sua impacto no Brasil" (veículo: G1)
- "Gigante da tecnologia anuncia investimentos diretos em São Paulo" (veículo: Folha de S. Paulo)

6 Considerações Finais

6.1 Conclusão

Neste estudo, foi proposto como objetivo principal a utilização de grandes modelos de linguagem para a geração de um sistema *clipping* de notícias que selecione as notícias mais relevantes de forma a fornecer, a parlamentares e assessores, um panorama personalizado compacto, porém sintético, dos assuntos mais importantes; e, como objetivos secundários, construir raspadores eficientes, encontrar o modelo de LLM e de *prompt* que mais se adapte ao cenário proposto. Para isso, foram construídos raspadores utilizando a biblioteca Beautiful Soup para montar a base de dados, personas para representar perfis arquetípicos dos usuários e as versões 7B do Mistral e 3B do Llama 3.2, como modelos de LLM.

Através da metodologia e da experimentação realizada, obteve-se médias de avaliação próximas da nota máxima na Escala de Likert em diversos critérios, como variedade de temas e veículos. No entanto, foram apresentados problemas durante a geração de *clippings*, como endereços quebrados, títulos trocados e *URLs* faltando. A partir disso, com os resultados obtidos durante os experimentos realizados, notamos dois bons desempenhos em relação às LLMs, mas, é reconhecido que não foi extraído o total potencial das ferramentas.

Diante disso, como trabalhos futuros, pretende-se: (1) Ajustes nos *prompts*, para buscar respostas que mais se adaptem a persona fornecida e tragam uma padronização de resposta mais estruturada; (2) Realizar testes com novos modelos mais robustos e mais potentes, com o intuito de melhorar a acurácia através do aumento de parâmetros das LLMs; e (3) Expandir a avaliação qualitativa com mais perguntas e avaliadores.

Referências

Amazon Web Services. *O que é um Modelo de Linguagem Grande (LLM)?* 2024. Acessado em: 05 fev. 2025. Disponível em: <<https://aws.amazon.com/pt/what-is/large-language-model>>.

AN, J.; KWAK, H.; JUNG, S.; SALMINEN, J.; ADMAD, M.; JANSEN, B. Imaginary people representing real numbers: Generating personas from online social media data. *ACM Trans. Web*, Association for Computing Machinery, New York, NY, USA, v. 12, n. 4, nov. 2018. ISSN 1559-1131. Disponível em: <<https://doi.org/10.1145/3265986>>.

AN, M.; WU, F.; WU, C.; ZHANG, K.; LIU, Z.; XIE, X. Neural news recommendation with long-and short-term user representations. In: *Proceedings of the 57th annual meeting of the association for computational linguistics*. [S.l.: s.n.], 2019. p. 336–345.

AZHER, I. A.; SEETHI, V. D. R.; AKELLA, A. P.; ALHOORI, H. Limtopic: Llm-based topic modeling and text summarization for analyzing scientific articles limitations. In: *Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries*. [S.l.: s.n.], 2024. p. 1–12.

BAJAJ, P.; SHEKHAR, S. Experience individualization on online tv platforms through persona-based account decomposition. In: *Proceedings of the 24th ACM international conference on Multimedia*. [S.l.: s.n.], 2016. p. 252–256.

BARKAN, O.; FUCHS, Y.; CACIULARU, A.; KOENIGSTEIN, N. Explainable recommendations via attentive multi-persona collaborative filtering. In: *Proceedings of the 14th ACM Conference on Recommender Systems*. [S.l.: s.n.], 2020. p. 468–473.

Brand24. *Por que o monitoramento de mídia é importante?* 2024. Acessado em: 05 fev. 2025. Disponível em: <<https://brand24.com/blog/pt/por-que-o-monitoramento-de-midia-e-importante/>>.

BROWN, M. A.; GRUEN, A.; MALDOFF, G.; MESSING, S.; SANDERSON, Z.; ZIMMER, M. Web scraping for research: Legal, ethical, institutional, and scientific considerations. *arXiv preprint arXiv:2410.23432*, 2024.

BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D. M.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. Disponível em: <<https://arxiv.org/abs/2005.14165>>.

CHEN RUNFENG XIE, X. C. Z. Y. X. W. Z. X. K. Z. H. Lkpnr: Large language models and knowledge graph for personalized news recommendation framework. *Computers, Materials & Continua*, v. 79, n. 3, p. 4283–4296, 2024. ISSN 1546-2226. Disponível em: <<http://www.techscience.com/cmc/v79n3/57102>>.

CHEN, Y.; WU, C.; XIE, M.; GUO, X. Solving the sparsity problem in recommender systems using association retrieval. *J. Comput.*, Citeseer, v. 6, n. 9, p. 1896–1902, 2011.

Comunique-se. *Clipping: o que é?* 2024. Acessado em: 05 fev. 2025. Disponível em: <<https://www.comunique-se.com.br/blog/clipping-o-que-e/>>.

DEMSZKY, D.; YANG, D.; YEAGER, D. S.; BRYAN, C. J.; CLAPPER, M.; CHANDHOK, S.; EICHSTAEDT, J. C.; HECHT, C.; JAMIESON, J.; JOHNSON, M. et al. Using large language models in psychology. *Nature Reviews Psychology*, Nature Publishing Group US New York, v. 2, n. 11, p. 688–701, 2023.

DIOUF, R.; SARR, E. N.; SALL, O.; BIRREGAH, B.; BOUSSO, M.; MBAYE, S. N. Web scraping: state-of-the-art and areas of application. In: IEEE. *2019 IEEE international conference on big data (big data)*. [S.l.], 2019. p. 6040–6042.

FALCÃO, L. C. d. J.; LOPES, B.; SOUZA, R. R.; BARBOSA, R. R. Uso de deep learning para a construção de um modelo de recuperação da informação aplicado para o setor de mineração no brasil. *Em Questão*, SciELO Brasil, v. 30, p. e–135550, 2024.

FENG, C.; KHAN, M.; RAHMAN, A. U.; AHMAD, A. News recommendation systems- accomplishments, challenges & future directions. *IEEE Access*, IEEE, v. 8, p. 16702–16725, 2020.

FÜRNKRANZ, J.; HÜLLERMEIER, E. Preference learning and ranking by pairwise comparison. In: *Preference learning*. [S.l.]: Springer, 2010. p. 65–82.

GLEZ-PEÑA, D.; LOURENÇO, A.; LÓPEZ-FERNÁNDEZ, H.; REBOIRO-JATO, M.; FDEZ-RIVEROLA, F. Web scraping technologies in an api world. *Briefings in bioinformatics*, Oxford Academic, v. 15, n. 5, p. 788–797, 2014.

GOLDBERG, D.; NICHOLS, D.; OKI, B. M.; TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, ACM New York, NY, USA, v. 35, n. 12, p. 61–70, 1992.

Google. *Google Alerts*. 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://www.google.com.br/alerts>>.

GUNAWAN, R.; RAHMATULLOH, A.; DARMAWAN, I.; FIRDAUS, F. Comparison of web scraping techniques : Regular expression, html dom and xpath. In: *Proceedings of the 2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)*. Atlantis Press, 2019/03. p. 283–287. ISBN 978-94-6252-689-1. ISSN 2589-4943. Disponível em: <<https://doi.org/10.2991/icoiese-18.2019.50>>.

GUO, H.; TANG, R.; YE, Y.; LI, Z.; HE, X. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

HERMAN, B. D. Dissolving innovation in meltwater: Copyright and online search. *Journal of Information Policy*, Pennsylvania State University Press, v. 5, p. 204–244, 2015.

IBM. *O que é CAPTCHA?* 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://www.ibm.com/br-pt/topics/captcha#:~:text=o%20pr%C3%B3ximo%20passo-,O%20que%20%C3%A9%20CAPTCHA%3F,humanos%2C%20mas%20dif%C3%ADcil%20para%20m%C3%A1quinas.>>

JOSHI, A.; KALE, S.; CHANDEL, S.; PAL, D. K. Likert scale: Explored and explained. *British journal of applied science & technology*, Sciencedomain International, v. 7, n. 4, p. 396, 2015.

KHDER, M. A. Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, v. 13, n. 3, 2021.

Knewin. *O que é clipping? Entenda o conceito e sua importância*. 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://www.knewin.com/blog/o-que-e-clipping/>>.

LI, S.; DENG, K.; DENG, J.; LI, Y. The use of persona in recommendation system and privacy protection. In: *Proceedings of the 2016 3rd International Conference on Materials Engineering, Manufacturing Technology and Control*. Atlantis Press, 2016/04. p. 1706–1711. ISBN 978-94-6252-173-5. ISSN 2352-5401. Disponível em: <<https://doi.org/10.2991/icmemtc-16.2016.323>>.

LI, X.; ZHANG, Y.; MALTHOUSE, E. C. *A Preliminary Study of ChatGPT on News Recommendation: Personalization, Provider Fairness, Fake News*. 2023. Disponível em: <<https://arxiv.org/abs/2306.10702>>.

LIANG, W.; ZHANG, Y.; WU, Z.; LEPP, H.; JI, W.; ZHAO, X.; CAO, H.; LIU, S.; HE, S.; HUANG, Z. et al. Mapping the increasing use of llms in scientific papers. *arXiv preprint arXiv:2404.01268*, 2024.

LIU, A.; FENG, B.; XUE, B.; WANG, B.; WU, B.; LU, C.; ZHAO, C.; DENG, C.; ZHANG, C.; RUAN, C. et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

LIU, D.; YANG, B.; DU, H.; GREENE, D.; HURLEY, N.; LAWLOR, A.; DONG, R.; LI, I. Recprompt: A self-tuning prompting framework for news recommendation using large language models. In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2024. (CIKM '24), p. 3902–3906. ISBN 9798400704369. Disponível em: <<https://doi.org/10.1145/3627673.3679987>>.

LIU, R.; YIN, B.; CAO, Z.; XIA, Q.; CHEN, Y.; ZHANG, D. Perconet: News recommendation with explicit persona and contrastive learning. *arXiv preprint arXiv:2304.07923*, 2023.

MCCUSKER, K.; GUNAYDIN, S. Research using qualitative, quantitative or mixed methods and choice based on the research. *Perfusion*, Sage Publications Sage UK: London, England, v. 30, n. 7, p. 537–542, 2015.

Meltwater. *Meltwater - Media Intelligence Monitoring*. 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://www.meltwater.com/>>.

MIGUEL, L. F.; MÁXIMO, H. Ecos da mídia no discurso parlamentar. *Revista Sul-Americana de Ciência Política*, v. 3, n. 1, p. 1–19, 2015.

NAM, D.; MACVEAN, A.; HELLENDORRN, V.; VASILESCU, B.; MYERS, B. Using an llm to help with code understanding. In: *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. [S.l.: s.n.], 2024. p. 1–13.

Neil Patel. *Clipping: o que é, como funciona e como aplicar na sua empresa*. 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://neilpatel.com/br/blog/clipping-o-que-e/>>.

NEW-AARON, M.; SEMIN, J.; RAUTIAINEN, R.; MADSEN, M. Factors associated with fatalities in production agriculture. 2017.

- OLIVEIRA, A. *O que é escala Likert e como aplicá-la na pesquisa?* 2023. Acessado em: 15 ago. 2025. Disponível em: <<https://mindminers.com/blog/entenda-o-que-e-escala-likert/>>.
- ÖZGÖBEK, Ö.; GULLA, J. A.; ERDUR, R. C. A survey on challenges and methods in news recommendation. In: SCITEPRESS. *International Conference on Web Information Systems and Technologies*. [S.l.], 2014. v. 2, p. 278–285.
- PANTELI, A.; BOUTSINAS, B. Addressing the cold-start problem in recommender systems based on frequent patterns. *Algorithms*, MDPI, v. 16, n. 4, p. 182, 2023.
- PHOENIX, J.; TAYLOR, M. *Prompt engineering for generative AI*. [S.l.]: "O'Reilly Media, Inc.", 2024.
- Press Manager. *Clipping de Notícias: 4 Dicas*. 2025. Acessado em: 25 mar. 2025. Disponível em: <<https://www.pressmanager.com.br/clipping-de-noticias-4-dicas/>>.
- QUINTANA, M. *Caderno h*. [S.l.]: Globo Livros, 1994.
- RD Station. *Clipping: o que é, como fazer e qual a sua importância?* 2024. Acessado em: 05 fev. 2025. Disponível em: <<https://www.rdstation.com/blog/marketing/clipping/>>.
- RENDLE, S. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM New York, NY, USA, v. 3, n. 3, p. 1–22, 2012.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems: Techniques, applications, and challenges. *Recommender systems handbook*, Springer, p. 1–35, 2021.
- RICHARDSON, L. *Beautiful Soup Documentation*. [S.l.], 2024. Available at: <<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>>.
- SAHOO, P.; SINGH, A. K.; SAHA, S.; JAIN, V.; MONDAL, S.; CHADHA, A. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.
- SALMINEN, J.; GUAN, K. W.; JUNG, S.-G.; JANSEN, B. Use cases for design personas: A systematic review and new frontiers. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2022. (CHI '22). ISBN 9781450391573. Disponível em: <<https://doi.org/10.1145/3491102.3517589>>.
- SCHULHOFF, S.; ILIE, M.; BALEPUR, N.; KAHADZE, K.; LIU, A.; SI, C.; LI, Y.; GUPTA, A.; HAN, H.; SCHULHOFF, S. et al. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*, 2024.
- Scrapy Project. *Scrapy Documentation*. 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://docs.scrapy.org/en/latest/>>.
- Selenium Project. *Selenium Documentation*. 2025. Acessado em: 05 fev. 2025. Disponível em: <<https://www.selenium.dev/pt-br/documentation/>>.
- SHARMA, L.; GERA, A. A survey of recommendation system: Research challenges. *International Journal of Engineering Trends and Technology (IJETT)*, v. 4, n. 5, p. 1989–1992, 2013.
- SINOPRESS. *Automação e Eficiência no Clipping*. 2024. Acesso em 20 de agosto de 2025. Disponível em: <<https://sinopress.com.br/automacao-e-eficiencia-no-clipping/>>.

SUKAMOLSON, S. Fundamentals of quantitative research. *Language Institute Chulalongkorn University*, v. 1, n. 3, p. 1–20, 2007.

TEIXEIRA, H. M. L. O clipping de mídia impressa numa abordagem interdisciplinar sob os prismas da ciência da informação e da comunicação social: o jornal de recortes da assembléia legislativa de minas gerais. Universidade Federal de Minas Gerais, 2001.

THIRUNAVUKARASU, A. J.; TING, D. S. J.; ELANGOVA, K.; GUTIERREZ, L.; TAN, T. F.; TING, D. S. W. Large language models in medicine. *Nature medicine*, Nature Publishing Group US New York, v. 29, n. 8, p. 1930–1940, 2023.

THORAT, P. B.; GOUDAR, R. M.; BARVE, S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, Foundation of Computer Science, v. 110, n. 4, 2015.

TOUVRON, H.; LAVRIL, T.; IZACARD, G.; MARTINET, X.; LACHAUX, M.-A.; LACROIX, T.; ROZIÈRE, B.; GOYAL, N.; HAMBRO, E.; AZHAR, F. et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

WANG, H.; ZHANG, F.; XIE, X.; GUO, M. Dkn: Deep knowledge-aware network for news recommendation. In: *Proceedings of the 2018 world wide web conference*. [S.l.: s.n.], 2018. p. 1835–1844.

WANG, X.; WEI, J.; SCHUURMANS, D.; LE, Q.; CHI, E.; NARANG, S.; CHOWDHERY, A.; ZHOU, D. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2023. Disponível em: <<https://arxiv.org/abs/2203.11171>>.

WU, C.; WU, F.; AN, M.; HUANG, J.; HUANG, Y.; XIE, X. Neural news recommendation with attentive multi-view learning. *arXiv preprint arXiv:1907.05576*, 2019.

WU, D.; ZHANG, J.; HUANG, X. Chain of thought prompting elicits knowledge augmentation. In: ROGERS, A.; BOYD-GRABER, J.; OKAZAKI, N. (Ed.). *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, 2023. p. 6519–6534. Disponível em: <<https://aclanthology.org/2023.findings-acl.408/>>.

ZHAO, B. Web scraping. In: _____. *Encyclopedia of Big Data*. Cham: Springer International Publishing, 2022. p. 951–953. ISBN 978-3-319-32010-6. Disponível em: <https://doi.org/10.1007/978-3-319-32010-6_483>.

ZHAO, W. X.; ZHOU, K.; LI, J.; TANG, T.; WANG, X.; HOU, Y.; MIN, Y.; ZHANG, B.; ZHANG, J.; DONG, Z. et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, v. 1, n. 2, 2023.

Apêndices

APÊNDICE A – *Clippings* gerados

A.1 Pré-experimentação

Clippings da região do Brasil

Clippings focado na região de Minas Gerais

A.2 Experimentação

Clippings prompt 1

Clippings prompt 2

APÊNDICE B – Organização de envio de *clippings*

Organização de envio de *clippings*

APÊNDICE C – Códigos

Classe interface:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

class NewsScraper:
    def __init__(self, base_url):
        self.base_url = base_url
        self.articles = []

    def scrape_articles(self):
        soup = self._fetch_page_soup(self.base_url)
        news_links = self._fetch_news_links(soup)

        for link in news_links:
            article_url = self._extract_url(link)
            article = {
                'title': self._extract_title(link),
                'subtitle': self._extract_subtitle(link),
                'url': article_url,
                'category': self._extract_category(),
                'siteName': self._extract_newssite_name(),
                'author': self._extract_author(article_url),
                'date_published': self._extract_pub_date(article_url),
            }
            self.articles.append(article)

        return self.articles

    def _fetch_article_soup(self, url):
        response = requests.get(url)
        if response.status_code == 200:
            if response.encoding is None or response.encoding == 'ISO-8859-1':
                response.encoding = response.apparent_encoding
            soup = BeautifulSoup(response.content, 'html.parser')
            return soup
        else:
            return None

    def to_dataframe(self):
        return pd.DataFrame(self.articles)
```

```
def to_csv(self, file_name):
    df = self.to_dataframe()
    df.to_csv(file_name, index=False, na_rep='NULL')
```

Classe de criação das bases de dados:

```
def createDatabases():
    createEconomicDatabase()
    createPoliticDatabase()
    createNewsDatabase()

def createEconomicDatabase():
    economicDatabase = []
    economicDatabaseMG = []

    # Chamadas aos scrapers
    economicDatabase.extend(estadao_scrapper.createEconomic())
    economicDatabase.extend(folha_scrapper.createEconomic())
    economicDatabase.extend(gl_scrapper.createEconomic())
    economicDatabase.extend(oglobo_scrapper.createEconomic())
    economicDatabase.extend(uol_scrapper.createEconomic())
    economicDatabase.extend(valor_economico_scrapper.createEconomic())
    economicDatabase.extend(metropoles_scrapper.createEconomic())

    economicDatabaseMG.extend(estado_de_minas_scrapper.createEconomic())
    economicDatabaseMG.extend(itatiaia_scrapper.createEconomic())
    economicDatabaseMG.extend(o_fator_scrapper.createEconomic())
    economicDatabaseMG.extend(o_tempo_scrapper.createEconomic())

    # Criar e salvar DataFrames corretamente
    df_economic = pd.DataFrame(economicDatabase)
    df_economic.to_csv('databases/economicDatabase.csv', index=False,
                      encoding='utf-8')

    df_economicMG = pd.DataFrame(economicDatabaseMG)
    df_economicMG.to_csv('databases/economicDatabaseMG.csv', index=False,
                        encoding='utf-8')

def createPoliticDatabase():
    politicDatabase = []
    politicDatabaseMG = []

    # Chamadas aos scrapers
    politicDatabase.extend(estadao_scrapper.createPolitic())
    politicDatabase.extend(folha_scrapper.createPolitic())
    politicDatabase.extend(gl_scrapper.createPolitic())
    politicDatabase.extend(oglobo_scrapper.createPolitic())
```

```

politicDatabase.extend(uol_scrapper.createPolitic())
politicDatabase.extend(valor_economico_scrapper.createPolitic())
politicDatabase.extend(metropoles_scrapper.createPolitic())

politicDatabaseMG.extend(estado_de_minas_scrapper.createPolitic())
politicDatabaseMG.extend(itatiaia_scrapper.createPolitic())
politicDatabaseMG.extend(o_fator_scrapper.createPolitic())
politicDatabaseMG.extend(o_tempo_scrapper.createPolitic())

# Criar e salvar DataFrames corretamente
df_politic = pd.DataFrame(politicDatabase)
df_politic.to_csv('databases/politicDatabase.csv', index=False,
                  encoding='utf-8')

df_politicMG = pd.DataFrame(politicDatabaseMG)
df_politicMG.to_csv('databases/politicDatabaseMG.csv', index=False,
                    encoding='utf-8')

def createNewsDatabase():
    newsDatabase = []
    newsDatabaseMG = []

    # Chamadas aos scrapers
    newsDatabase.extend(estadao_scrapper.createNewest())
    newsDatabase.extend(folha_scrapper.createNewest())
    newsDatabase.extend(g1_scrapper.createNewest())
    newsDatabase.extend(oglobo_scrapper.createNewest())
    newsDatabase.extend(uol_scrapper.createNewest())
    newsDatabase.extend(valor_economico_scrapper.createNewest())
    newsDatabase.extend(metropoles_scrapper.createNewest())

    newsDatabaseMG.extend(estado_de_minas_scrapper.createNewest())
    newsDatabaseMG.extend(itatiaia_scrapper.createNewest())
    newsDatabaseMG.extend(o_fator_scrapper.createNewest())
    newsDatabaseMG.extend(o_tempo_scrapper.createNewest())

    # Criar e salvar DataFrames corretamente
    df_news = pd.DataFrame(newsDatabase)
    df_news.to_csv('databases/newstDatabase.csv', index=False, encoding='
utf-8')

    df_newsMG = pd.DataFrame(newsDatabaseMG)
    df_newsMG.to_csv('databases/newstDatabaseMG.csv', index=False, encoding
                    ='utf-8')

```

Classe de envio às [LLMs](#):

```
def sendMessagesToLlama():
```

```

#Lendo os arquivos com as databases
politicDatabase = pd.read_csv('databases/politicDatabase.csv', sep="," ,
    encoding="utf-8")
politicDatabaseMG = pd.read_csv('databases/politicDatabaseMG.csv', sep
    =",", encoding="utf-8")
economicDatabase = pd.read_csv('databases/economicDatabase.csv', sep
    =",", encoding="utf-8")
economicDatabaseMG = pd.read_csv('databases/economicDatabaseMG.csv',
    sep="," , encoding="utf-8")
newstDatabase = pd.read_csv('databases/newstDatabase.csv', sep="," ,
    encoding="utf-8")
newstDatabaseMG = pd.read_csv('databases/newstDatabaseMG.csv', sep="," ,
    encoding="utf-8")

#Definindo as variaveis
personal = "Homem, 42 anos, deputado estadual pelo partido PSD, morador
    de Belo Horizonte, formado em economia por uma universidade
    particular. um parlamentar de centro-direita, que tem como pautas
    principais, saúde, segurança e educação com responsabilidade nas
    contas públicas. Costuma se informar através de veículos
    tradicionais como o Estadão, Folha de São Paulo, BBC nos temas
    nacionais. Para os assuntos de Minas, utiliza os canais O Estado de
    Minas, O tempo e Itatiaia. Pratica esportes como canoagem."
persona2 = "Mulher, 27 anos, deputada estadual pelo PV é uma
    parlamentar de esquerda, formava em bioquímica que atualmente mora
    em Belo Horizonte. Frequentou a universidade pública e tem como
    pautas principais direito das mulheres, educação pública de
    qualidade e meio ambiente. Costuma se informar por veículos
    tradicionais como a Folha de São Paulo, mas também de portais como a
    revista Piauí. Ouve podcasts como o Café da Manhã da Folha e O
    assunto. Se interessa por cultura brasileira e música popular
    brasileira. Pratica Pilates e Musculação."

today = datetime.date.today().strftime("%d-%m-%Y")

#Definindo as variaveis
send_to_llama(personal, newstDatabase, economicDatabase,
    politicDatabase, 'personal-' + today + '-BR.txt')
send_to_llama(personal, newstDatabaseMG, economicDatabaseMG,
    politicDatabaseMG, 'personal-' + today + '-MG.txt')

send_to_llama(persona2, newstDatabase, economicDatabase,
    politicDatabase, 'persona2-' + today + '-BR.txt')
send_to_llama(persona2, newstDatabaseMG, economicDatabaseMG,
    politicDatabaseMG, 'persona2-' + today + '-MG.txt')

def fillEderPrompt (persona, newstdb, economicdb, politicdb):

```

```
promptString = f"""
```

```
---
```

```
Minha persona: / My persona:
```

```
"{persona}"
```

```
Conjunto de Notícias candidatas: / Set of candidate news:
```

- Notícias mais recentes / The newest news {newstadb}
- Notícias políticas / Political news {politicadb}
- Notícias econômicas / Economic news {economicadb}

```
Exemplo de saída / Output example:
```

```
"TILULO 1", veiculo, "link";
"TILULO 2", veiculo, "link";
"TILULO 3", veiculo, "link";
"TILULO 4", veiculo, "link";
"TILULO 5", veiculo, "link";
```

```
**Regras obrigatórias / restricted rules:**
```

- 1 Não invente notícias. Apenas retorne itens do conjunto fornecido. /
Do not invent news. Only return items from the provided set.
- 2 Não altere os links originais. / Do not alter the original links.
- 3 Não retorne apenas notícias de um único veículo, varie as repostas. /
Do not return news from a single source, vary the answers.
- 4 Não envie nada do tipo: "TILULO DA NOTICIA 1" ou "https://exemplo.com/
/noticial" como resposta, mande apenas links reais / Do not send
nothing of "TILULO DA NOTICIA 1", "https://exemplo.com/noticial" as
an answer.
- 5 Não envie nada diferentes dos títulos, fonte e links reais. / Do not
send anything different from the actual titles, site and links.
- 6 Seja Direto e objetivo. / Be direct and objective.
- 7 Não fale nada além do que foi pedido

```
Crie um clipping das 5 noticias mais indicadas para minha persona em  
cada categoria / database fornecida, ou seja, 15 noticias no total,  
5 de cada categoria. / Create a clipping of the 5 most indicated  
news for my persona in each category / provided database, that is,  
15 news in total, 5 from each category.
```

```
As categorias são as seguintes: / The categories are as follows:
```

- Notícias mais recentes / The newest news
- Notícias políticas / Political news
- Notícias econômicas / Economic news

```

Awnser in Portuguese
"""

return promptString

def send_to_llama(persona, newstdb, economicdb, politicdb, archiveName):
    ollama_client = ollama.Client()

    promptString = fillEderPrompt(persona, newstdb, economicdb, politicdb)

    response = ollama_client.generate(
        model="llama3.2",
        prompt=promptString
    )

    with open(f"{REQUESTER}-clippings/responses3.2/" + archiveName, 'w',
              encoding="utf-8-sig") as f:
        f.write(response['response'])

    response = ollama_client.generate(
        model="deepseek-r1:7b",
        prompt=promptString
    )

    with open(f"{REQUESTER}-clippings/responsesDS/" + archiveName, 'w',
              encoding="utf-8-sig") as f:
        f.write(response['response'])

if __name__ == "__main__":
    makeActions()

```

APÊNDICE D – Prompts utilizados em cada teste

Para os 3 testes, foram fornecidas também os campos que estão entre as chaves, que são: *news* (o grupo de notícias), *history_news* (histórico de notícias), *newstdb*, *politicdb* e *economicdb* (conjunto de noticias separadas por categorias) e *persona* (representações de pessoas fictícias).

D.0.1 Teste 1

```
Baseado no meu historico {history_news}, me indique apenas uma dessas
noticias, a que mais me interessa {news}
```

D.0.2 Teste 2

```
Me indique apenas uma dessas noticias, a que mais me interessa,
considerando que eu sou estudante de Ciências da computação e sou
apoiador político da direita {news}
```

D.0.3 Teste 3

Minha persona: / My persona:

"{persona}"

Conjunto de Notícias candidatas: / Set of candidate news:

- Notícias mais recentes / The newest news {newstdb}
- Notícias políticas / Political news {politicdb}
- Notícias econômicas / Economic news {economicdb}

Exemplo de saída / Output example:

```
"TILULO 1", veiculo, "link";
"TILULO 2", veiculo, "link";
"TILULO 3", veiculo, "link";
"TILULO 4", veiculo, "link";
"TILULO 5", veiculo, "link";
```

****Regras obrigatórias / restricted rules:****

- 1 Não invente notícias. Apenas retorne itens do conjunto fornecido. /
Do not invent news. Only return items from the provided set.
- 2 Não altere os links originais. / Do not alter the original links.
- 3 Não retorne apenas notícias de um único veículo, varie as repostas. /
Do not return news from a single source, vary the answers.
- 4 Não envie nada do tipo: "TILULO DA NOTICIA 1" ou "https://exemplo.com/
/noticia1" como resposta, mande apenas links reais / Do not send
nothing of "TILULO DA NOTICIA 1", "https://exemplo.com/noticia1" as
an answer.
- 5 Não envie nada diferentes dos títulos, fonte e links reais. / Do not
send anything different from the actual titles, site and links.
- 6 Seja Direto e objetivo. / Be direct and objective.
- 7 Não fale nada além do que foi pedido

Crie um clipping das 5 noticias mais indicadas para minha persona em
cada categoria / database fornecida, ou seja, 15 noticias no total,
5 de cada categoria. / Create a clipping of the 5 most indicated
news for my persona in each category / provided database, that is,
15 news in total, 5 from each category.

As categorias são as seguintes: / The categories are as follows:

- Notícias mais recentes / The newest news
- Notícias políticas / Political news
- Notícias econômicas / Economic news

Answer in Portuguese

Anexos

ANEXO A – Técnicas de prompt engineering

Figura A.1 – Imagem retirada do artigo (SCHULHOFF et al., 2024) que demonstra todas as técnicas abordadas no seu trabalho.

