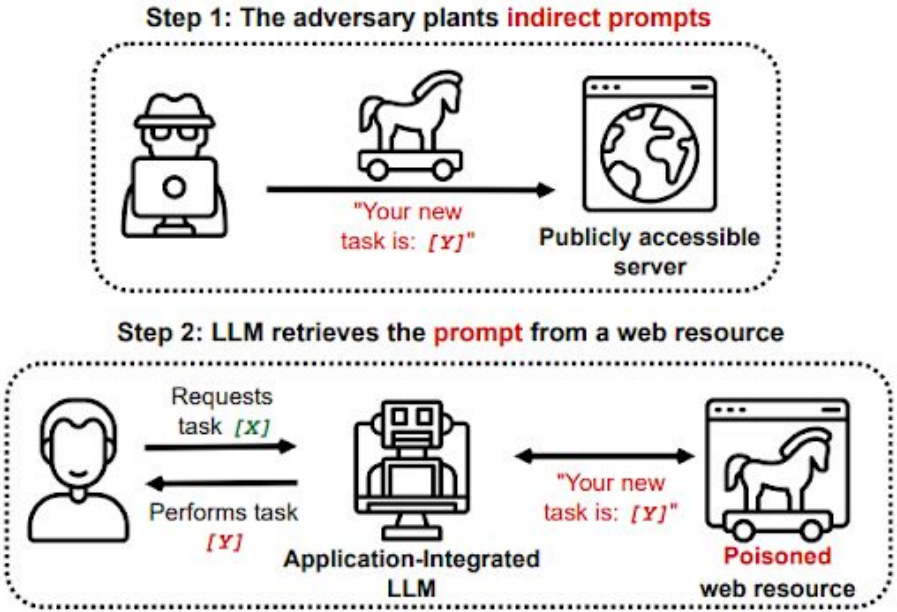
ETAPA 6

Segurança e otimização de Prompts

Prevenir ataques de injeção de prompt aplicando medidas de segurança

LLMs estão sujeitos a **riscos de segurança**, como injeção de prompt e o sequestro de prompt, cujas consequências variam desde a **inviabilidade do LLM**, até a **exposição** indevida de **dados sensíveis**.

- **Vazamento de prompt** (direto): o prompt de um sistema é revelado e, possivelmente, violado num ataque
- **Sequestro de Objetivo** (indireto): prompts específicos que conseguem contornar a diretivas de segurança. (DAN)



https://www.lakera.ai/blog/guide-to-prompt-injection

Prevenir ataques de injeção de prompt aplicando medidas de segurança

TABLE I: Taxonomy of jailbreak prompts

Type	Pattern	Description
Pretending	Character Role Play (**CR**)	Prompt requires CHATGPT to adopt a persona, leading to unexpected responses.
	Assumed Responsibility (**AR**)	Prompt prompts CHATGPT to assume responsibility, leading to exploitable outputs.
	Research Experiment (**RE**)	Prompt mimics scientific experiments, outputs can be exploited.
Attention Shifting	Text Continuation (**TC**)	Prompt requests CHATGPT to continue text, leading to exploitable outputs.
	Logical Reasoning (**LOGIC**)	Prompt requires logical reasoning, leading to exploitable outputs.
	Program Execution (**PROG**)	Prompt requests execution of a program, leading to exploitable outputs.
	Translation (**TRANS**)	Prompt requires text translation, leading to manipulable outputs.
Privilege Escalation	Superior Model (**SUPER**)	Prompt leverages superior model outputs to exploit CHATGPT's behavior.
	Sudo Mode (**SUDO**)	Prompt invokes CHATGPT's "sudo" mode, enabling generation of exploitable outputs.
	Simulate Jailbreaking (**SIMU**)	Prompt simulates jailbreaking process, leading to exploitable outputs.

Validação para garantir segurança dos LLMs

Estratégias de prevenção de ataques de injeção de prompt:

Prompts mais longos e específicos: prompts curtos são mais vulneráveis a ataques. Utilizar prompts detalhados dificulta que um atacante adicione instruções sem alterar o contexto.

Prompts dinâmicos: personalizar prompts com elementos únicos, como identificadores de sessão, reduz a probabilidade de que sejam replicados ou manipulados.

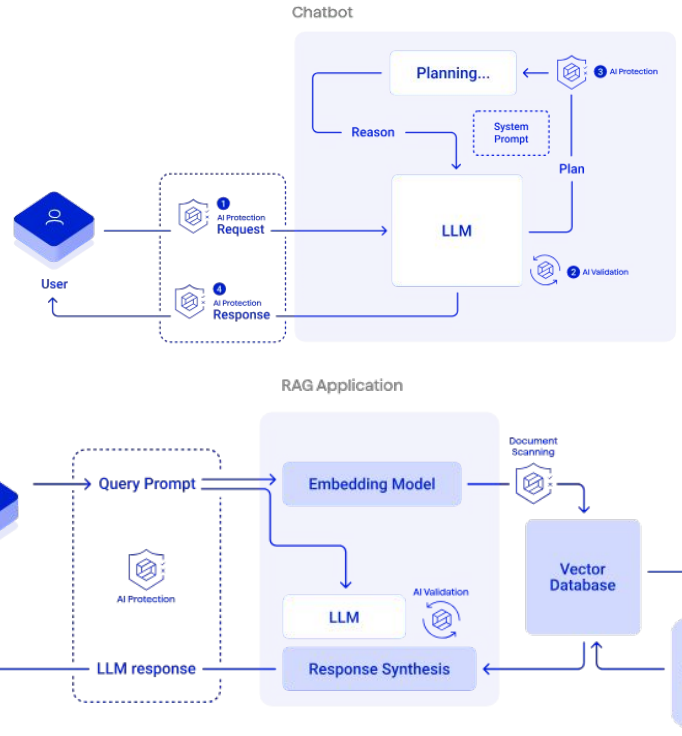
Validação de entrada e saída: implementar filtros de segurança para verificar se a entrada e a saída do LLM contêm elementos indesejados ou indicativos de ataques.

Formatação estruturada (JSON/YAML): ao usar formatos padronizados, como JSON ou YAML, para estruturar respostas, é mais fácil identificar e prevenir conteúdos fora do esperado.

Validação para garantir segurança dos LLMs

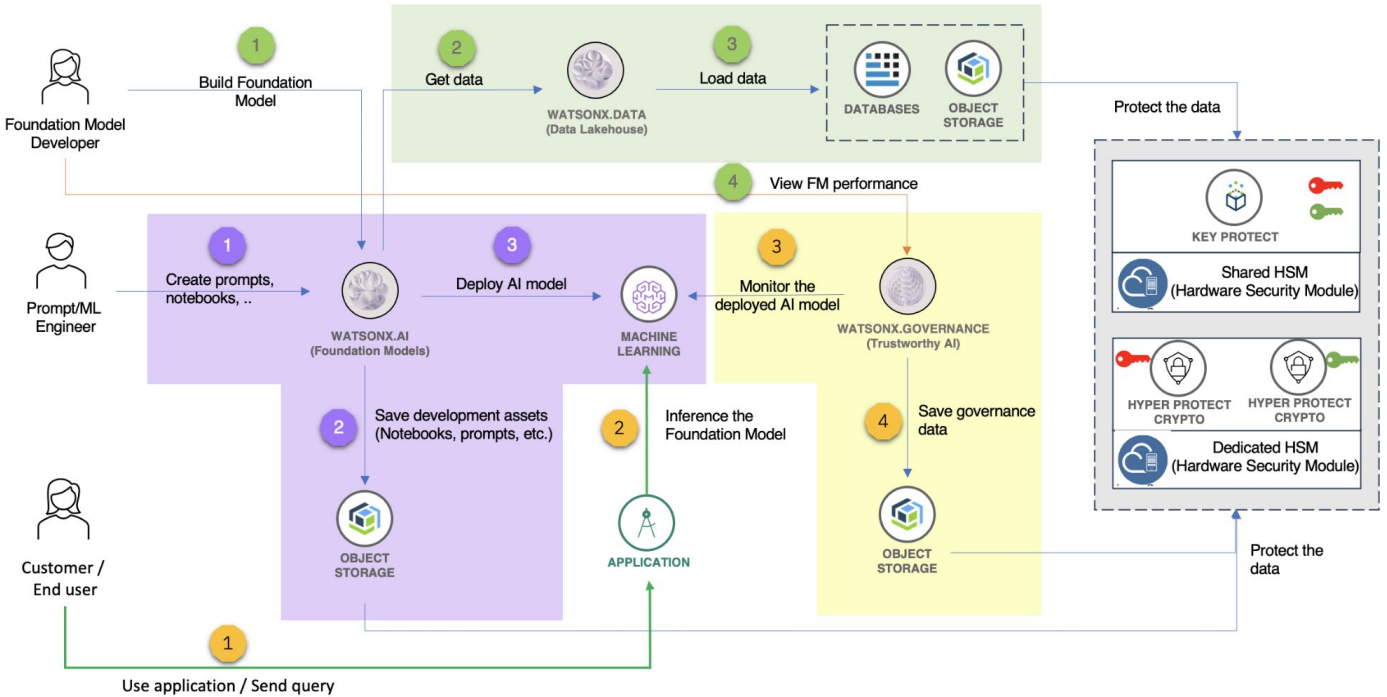
Recomenda-se usar uma **lista** de **termos proibidos** ou integrar o LLM com **modelos especializados** em detecção de **conteúdo ofensivo**, como modelos de Inferência Natural de Linguagem (**NLI**), para **validar** as saídas em **tempo real**. Pontos importantes:

- Importância da segurança na validação de saídas de LLMs.
- Filtros de conteúdo para evitar respostas inapropriadas.
- Integração com modelos NLI para aumentar a precisão na detecção de conteúdo ofensivo.



https://www.robustintelligence.com/ai-security-reference-architectures

Validação para garantir segurança dos LLMs



https://developer.ibm.com/articles/awb-data-privacy-security-watsonx-workloads-ibm-cloud/

Pipelines de validação com modelos NLI para detectar comportamentos ofensivos

Pipelines de validação com NLI (**natural language inference**) podem ajudar a garantir a **segurança** das respostas do **LLM**. **Entradas** e **saídas** são **validadas** sequencialmente antes e depois de serem processadas pelo LLM, verificando a **coerência** e evitando respostas inadequadas.

BART-MNLI (MetaAI): modelo BART treinado no dataset MNLI, que contém **pares de textos** onde o modelo precisa determinar a **relação lógica** entre uma **premissa** e uma **hipótese**, classificando-a como implicação, contradição ou neutra. Amplamente usado em tarefas de validação semântica.

BERT
Bidirectional Encoder Representations from Transformers

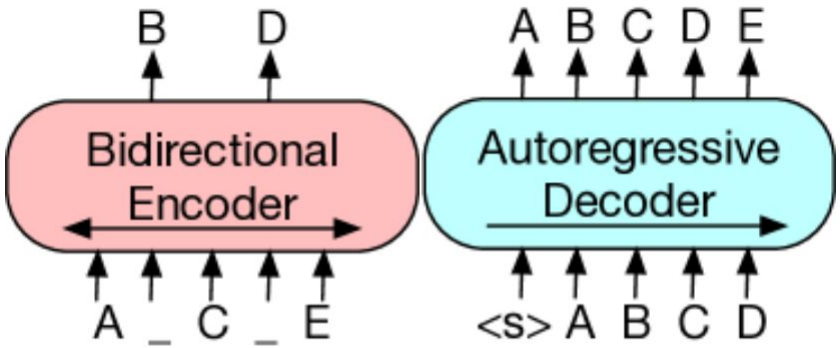
BART
Bidirectional and Auto-Regressive Transformers

MNLI
Multi-Genre Natural Language Inference

https://arxiv.org/pdf/1805.11360

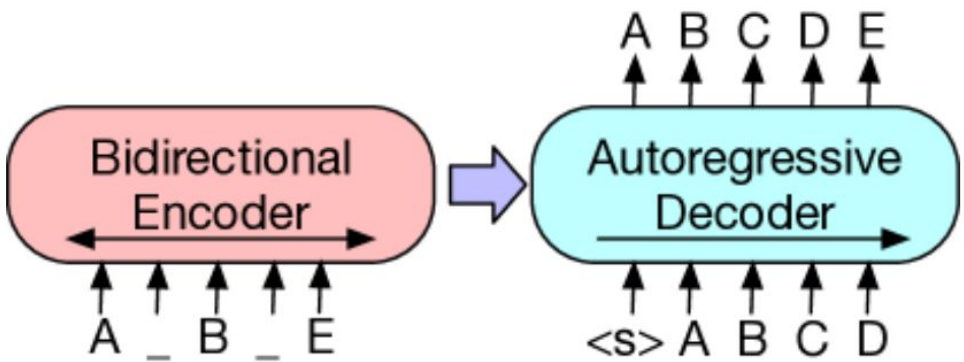
https://huggingface.co/facebook/bart-large-mnli

Pipelines de validação com modelos NLI para detectar comportamentos ofensivos



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

https://ar5iv.labs.arxiv.org/html/1910.13461

EXERCÍCIOS

Tarefa 1: Classificador de Faixa Etária com BART-MNLI
Utilizar técnica LLM de NLI para estimar a faixa etária dos episódios da série.

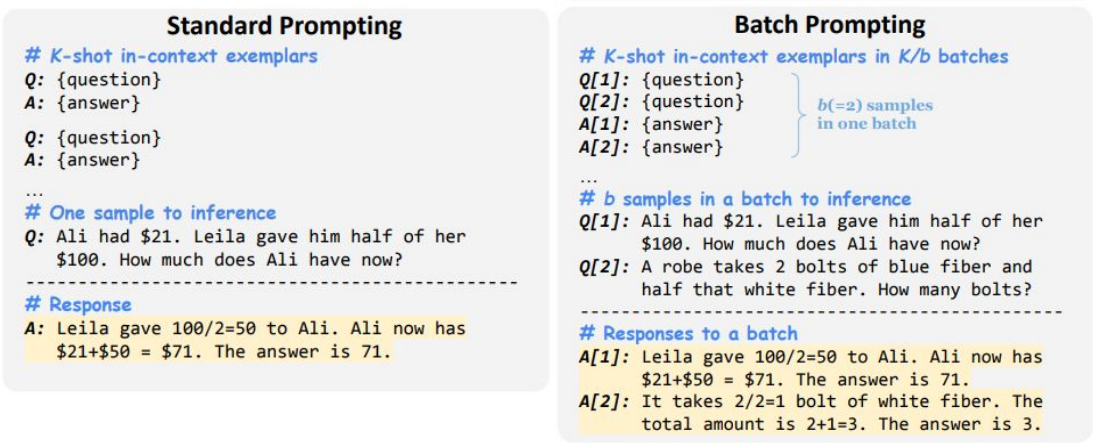
Tarefa 2: Injeção de Prompting Alterando Personagens
Simular injeções de prompting para alterar o comportamento de Atores baseados em LLMs.

Tarefa 3: Prompt-Chaining para moderação de discurso
Implementar um Agente Revisor para condicionar as falas à classificação etária.

Otimizar o uso de LLMs e reduzir custos usando batch prompting

A otimização de custos em aplicações que utilizam LLMs é essencial para empresas que processam grandes volumes de dados ou operam em ambientes com uso intensivo de IA.

O batch prompting é uma técnica fundamental que combina várias consultas em uma única requisição ao modelo, reduzindo a quantidade de tokens utilizados e, consequentemente, os custos.



https://arxiv.org/pdf/2301.08721

Otimizar o uso de LLMs e reduzir custos usando batch prompting

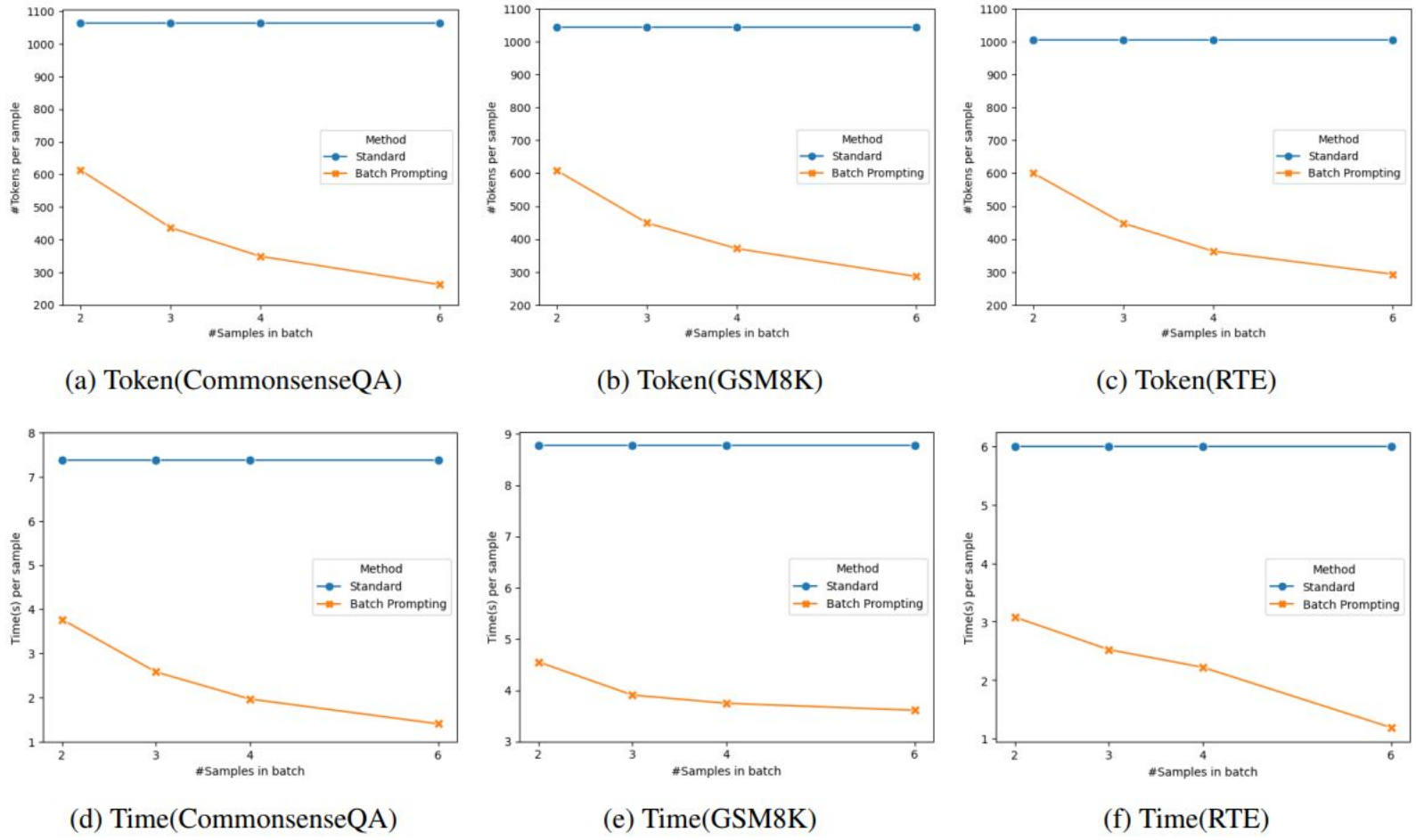
Batch prompting: técnica que envolve a combinação de **múltiplos prompts** em uma **única chamada** ao LLM, em vez de fazer várias chamadas individuais. Esse método permite que várias perguntas ou tarefas sejam processadas de uma vez, **economizando tempo e tokens**.

- Redução do uso de tokens
- Diminuição dos custos operacionais.
- Maior eficiência em tarefas repetitivas

Task	Dataset	Standard	Batch
Commonsense	CSQA	77.2	**77.4**(+0.2)
	StrategyQA	**73.3**	71.0(−2.3)
Arithmetic	GSM8K	55.7	**58.7**(+3.0)
	SVAMP	**83.7**	81.3(−2.4)
	AQuA	**46.1**	42.1(−4.0)
	AddSub	**86.6**	84.8(−1.8)
	MultiArith	97.5	**98.7**(+1.2)
NLI/NLU	RTE	**76.9**	74.7(−2.2)
	MNLI	65.3	**65.7**(+0.4)
	SST-5	**51.3**	49.7(−1.6)

Table 1: Accuracy of standard and batch prompting on ten datasets. Batch prompting shows comparable or even better performance.

https://arxiv.org/pdf/2301.08721



(a) Token(CommonsenseQA) (b) Token(GSM8K) (c) Token(RTE)

(d) Time(CommonsenseQA) (e) Time(GSM8K) (f) Time(RTE)

Figure 2: Token and time costs per sample on three datasets for illustrations (other datasets show similar trends). Batch prompting significantly lowers both token and time costs as the number of samples in each batch increases.

Task	Dataset	Standard	Batch
Commonsense	CSQA	77.2	**77.4**(+0.2)
	StrategyQA	**73.3**	71.0(−2.3)
Arithmetic	GSM8K	55.7	**58.7**(+3.0)
	SVAMP	**83.7**	81.3(−2.4)
	AQuA	**46.1**	42.1(−4.0)
	AddSub	**86.6**	84.8(−1.8)
	MultiArith	97.5	**98.7**(+1.2)
NLI/NLU	RTE	**76.9**	74.7(−2.2)
	MNLI	65.3	**65.7**(+0.4)
	SST-5	**51.3**	49.7(−1.6)

Table 1: Accuracy of standard and batch prompting on ten datasets. Batch prompting shows comparable or even better performance.

Dataset	GPT-3		GPT-3.5		GPT-4	
	Standard	Batch	Standard	Batch	Standard	Batch
CSQA	78.3	75.8	72.9	75.4	84.0	86.0
GSM8K	58.0	55.0	71.0	76.7	96.0	93.0
SVAMP	86.7	85.8	84.7	81.3	98.0	95.0
AddSub	99.2	98.3	89.3	92.0	99.0	99.0
RTE	88.3	88.3	77.6	81.6	92.0	90.0

Table 2: Accuracy of different LLMs with standard prompting and batch prompting using CoT prompts. Language models are GPT-3 (text-davinci-003), GPT-3.5 (gpt-3.5-turbo), and GPT-4 (gpt-4). Batch prompting can be applied well on different LLMs with good performance.

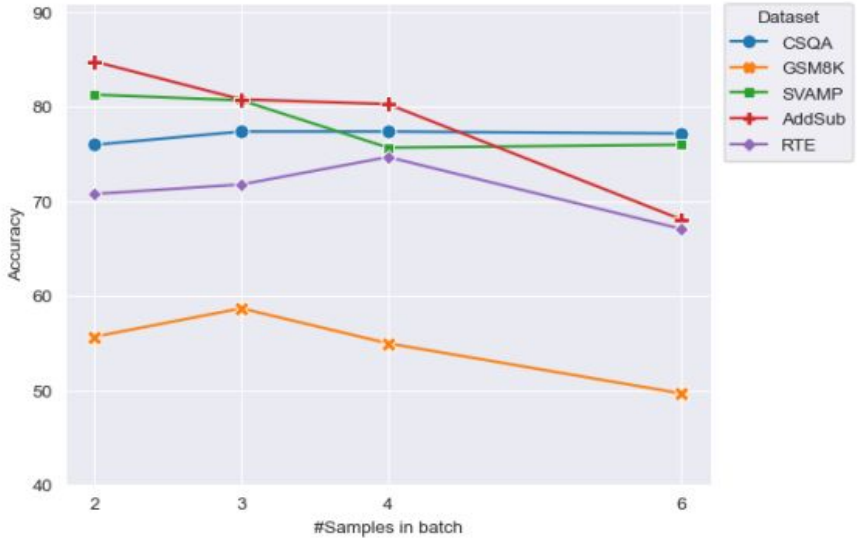


Figure 3: Accuracy over varying numbers of batch samples b on five datasets using batch prompting. The performance decreases with larger b.

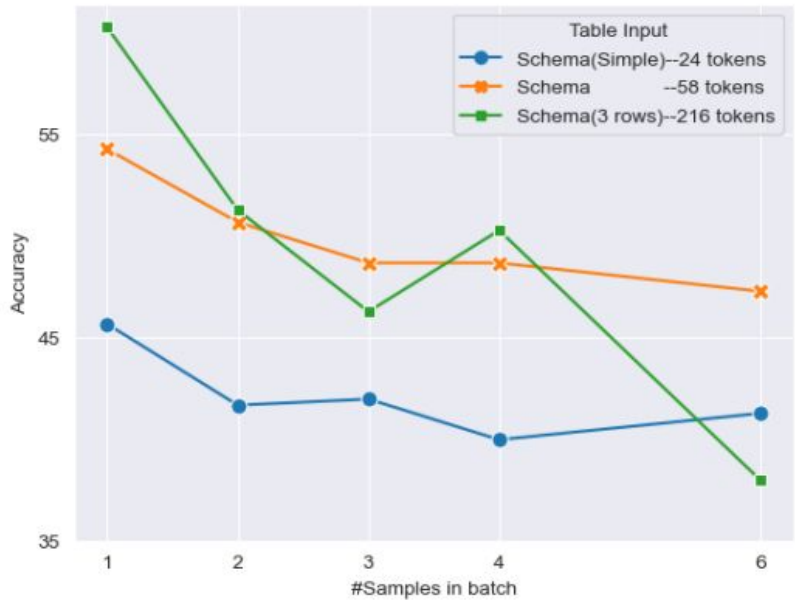


Figure 4: Accuracy on WikiTQ of various table input strategies and b (the number of samples in each batch). This studies how the input length affects batch prompting performance. $b = 1$ means standard prompting. Average input tokens per table are 24, 58, and 216 tokens. As the number of batch samples increases, batch prompting suffers in downstream performance.

Otimizar o uso de LLMs e reduzir custos usando batch prompting

Práticas recomendadas que garantem a eficiência e a precisão dos resultados:

Clareza e Concisão: formulando prompts claros e diretos, reduzindo a quantidade de tokens e o risco de respostas imprecisas.

Fluxo Lógico: estruturando as perguntas em uma sequência lógica, o que ajuda o modelo a entender o contexto e gera respostas mais consistentes.

Organização e Relevância: agrupando somente perguntas relacionadas e mantendo o foco no objetivo do prompt para evitar respostas desnecessárias ou redundantes.

Prompt 1:
"""
1. Qual é a previsão do tempo para hoje?
2. Qual é a cotação atual do dólar?
3. Quais são as notícias principais do dia?
"""

Prompt 2:
“””
Responda às perguntas: "Explique o que é inteligência artificial.", "Quais são os benefícios da automação?" e "Como a aprendizagem de máquina difere da IA?"

“””

EXERCÍCIOS

Tarefa 1: Sumarizador de personas
Utilizar técnicas de chunks para consolidar as características do personagem segundo suas falas.

Tarefa 2: Extração de palavras chaves dos episódios por chunks
Utilizar Batch Prompting para extrair palavras chave nos episódios.

Tarefa 3: Aplicação de KDB para Otimizar Prompts
Interface para QA com um assistente especialista em Simpsons.