

“Do conhecimento acadêmico à transformação sustentável: inovação com validação científica”**Um estudo da transferência de aprendizado empregando modelos de aprendizado profundo: aplicação de LLMs (*Large Language Models*) no contexto educacional**Ulisses D. de P. A. Junior¹ (IC), Isabela Neves Drummond (PQ)¹¹Universidade Federal de Itajubá.**Palavras-chave:** Inteligência Artificial. LLM. Otimização. Planejamento Educacional. *Prompt Engineering*.**Introdução**

A integração de Modelos de Linguagem de Grande Escala (*Large Language Models* - LLMs) no campo da educação revolucionou a forma como o aprendizado é abordado, permitindo experiências personalizadas e adaptadas às necessidades individuais dos estudantes. Embora o avanço em modelos maiores muitas vezes leve a custos computacionais elevados, o Mistral 7B demonstra que um modelo cuidadosamente projetado pode alcançar alto desempenho com eficiência (MISTRAL AI, 2023), com isso o modelo possui forte capacidade de aprendizado *Few-Shot*, o que permite a personalização das experiências de aprendizado sem a necessidade de grandes conjuntos de dados de treinamento específicos para a tarefa.

A abordagem inovadora de Sistemas de Aprendizagem Adaptativa, que emprega análises de dados com LLMs para projetar currículos personalizados, demonstrou melhorias notáveis tanto no engajamento quanto na retenção de conhecimento dos alunos, conforme evidenciado por Li et al. (2025). As avaliações realizadas em diversos ambientes educacionais validaram a flexibilidade e o impacto positivo da estrutura nos resultados de aprendizado, com o objetivo de revolucionar as práticas educacionais tradicionais.

Neste contexto, o presente trabalho aborda a aplicação de Inteligência Artificial, especificamente Modelos de Linguagem de Grande Escala (LLMs), para aprimorar a elaboração de grades curriculares de alunos. O objetivo principal é desenvolver e avaliar um sistema que, com base nas disciplinas já concluídas por um estudante e em sua disponibilidade de horários, sugira uma grade otimizada para o próximo semestre. Uma abordagem desse tipo pode ser mais eficiente que o processo manual de montagem de grade, que frequentemente resulta em atrasos na formação, conflitos de horários e dificuldades para priorizar disciplinas essenciais. O método empregado envolve a combinação de um algoritmo tradicional de otimização de grades, que segue

uma sequência de prioridades, com técnicas de *prompt engineering* para permitir que um LLM interprete a intenção do usuário a partir de linguagem natural, superando a necessidade de grandes volumes de dados para treinamento. O procedimento inclui a definição de uma estrutura de dados para as disciplinas, o desenvolvimento de funções de lógica para a otimização e a implementação de um pipeline de processamento que integra a interpretação do LLM com a lógica de negócio para gerar as recomendações finais.

1. Modelos de Linguagem de Grande Escala (LLMs)

Modelos de Linguagem de Grande Escala, ou LLMs, são uma classe de modelos de aprendizado de máquina treinados em vastos volumes de dados textuais para entender, gerar e interagir com a linguagem humana. Sua arquitetura, frequentemente baseada em transformadores, permite que processem sequências de texto de maneira a capturar o contexto e a semântica de forma extremamente eficaz, assim como o modelo Mistral 7B. Essa capacidade os torna ferramentas poderosas para tarefas como sumarização, tradução, análise de sentimentos e, no contexto deste trabalho, a interpretação da intenção do usuário a partir de consultas em linguagem natural (Xiao & Zhu, 2025).

A aplicação de LLMs na automação do planejamento acadêmico, tema deste estudo, representa uma nova fronteira, buscando tornar a jornada do estudante mais fluida e eficiente, assim como demonstrado por Li et al. (2025).

Para a execução deste trabalho, foi empregado um modelo principal chamado Mistral-7B, um modelo eficiente e de alta performance, otimizado para tarefas de instrução, sendo particularmente adequado para extração de informações.

2. Prompt Engineering

Prompt Engineering é a disciplina de projetar e refinar as entradas (*prompts*) para modelos de linguagem a fim de obter saídas mais precisas e desejadas. Em vez de

“Do conhecimento acadêmico à transformação sustentável: inovação com validação científica”

modificar a arquitetura interna do modelo (como no *fine-tuning*), esta técnica foca em como a instrução é formulada para extrair o máximo do potencial do modelo pré-treinado. Essa abordagem é fundamental para moldar a interação com modelos generativos e depende do entendimento das capacidades e limitações dos LLMs, bem como da estruturação de instruções, perguntas, dados de entrada e exemplos (Amatriain, 2024). Neste contexto, duas abordagens de *prompt engineering* foram testadas para guiar o LLM na interpretação das solicitações dos alunos: *Few-Shot* e *Chain-of-Thought* (CoT). A técnica CoT, conforme abordado por Ma et al. (2024), utiliza o encadeamento de raciocínio passo a passo para melhorar o desempenho do modelo, o que é relevante para a interpretação de intenções complexas dos usuários.

Metodologia

1. Abordagem Baseada em Algoritmo de Otimização

Para simular e validar a lógica de recomendação, foi desenvolvido um algoritmo de otimização de grades curriculares que opera com base em um conjunto de regras e prioridades bem definidas. Este algoritmo é alimentado com duas fontes de dados principais: a estrutura curricular completa do curso, contendo informações como códigos de disciplina, pré-requisitos e período recomendado, e a lista de disciplinas ofertadas para o semestre em questão, com seus respectivos horários.

A lógica de negócio do algoritmo é construída em torno de uma sequência de prioridades para a seleção das disciplinas, garantindo que o aluno curse primeiro as matérias mais críticas para sua progressão acadêmica. A hierarquia de prioridades é a seguinte:

- **Prioridade 1:** Disciplinas obrigatórias que, segundo o plano de curso, deveriam ter sido cursadas em semestres anteriores ao atual do aluno. A inclusão de matérias atrasadas neste nível de prioridade visa acelerar a conclusão do curso e evitar o acúmulo de dependências.
- **Prioridade 2:** Disciplinas obrigatórias recomendadas para o semestre atual. O foco aqui é manter o aluno alinhado com a progressão padrão do curso.
- **Prioridade 3:** Disciplinas obrigatórias de semestres futuros. Esta prioridade permite que alunos com capacidade de adiantamento possam avançar em seus estudos, desde que os pré-requisitos tenham sido cumpridos.

- **Prioridade 4:** Disciplinas optativas. Estas matérias, por serem flexíveis, são consideradas apenas depois que todas as necessidades de disciplinas obrigatórias são atendidas.

O algoritmo funciona de forma iterativa, percorrendo cada nível de prioridade e, para cada disciplina, verifica se todos os seus pré-requisitos foram concluídos e se não há conflitos de horário com as disciplinas já selecionadas. A primeira disciplina que atende a todos os critérios é adicionada à grade e seus horários são marcados como indisponíveis, garantindo a otimização da grade e a eliminação de sobreposições.

2. Abordagem Baseada em LLM com Prompt Engineering

Com o objetivo de testar uma LLM em uma tarefa e pesquisar sobre os limites da mesma, foi desenvolvida uma segunda abordagem para interpretar as solicitações dos alunos feitas em linguagem natural. Em vez de exigir que o usuário insira manualmente seus dados em um formulário, um LLM é utilizado para extrair as informações relevantes a partir de um texto livre.

O método central desta abordagem é o *prompt engineering* e, para isso, foram testadas duas técnicas:

- **Few-Shot:** Esta técnica fornece ao modelo exemplos de textos de alunos e os respectivos resultados esperados em formato JSON. Ao ver esses exemplos, o modelo aprende a extrair e formatar as informações por similaridade, como o semestre atual, a lista de matérias cumpridas e a disponibilidade de horários.
- **Chain-of-Thought (CoT):** Com o intuito de melhorar a capacidade de raciocínio do modelo, esta técnica instrui o LLM a "pensar em voz alta" e detalhar como chegou à resposta. A expectativa era que este processo passo a passo resultasse em uma extração de dados mais precisa e confiável.

A arquitetura do sistema que integra o LLM com o algoritmo de otimização funciona em um pipeline de três etapas:

Interpretação do LLM: O texto de entrada do aluno é processado pelo LLM, que, com base nas instruções e em um mapeamento de sinônimos fornecido, gera uma saída estruturada em formato JSON.

Tradução de Disponibilidade: O JSON gerado pelo LLM, que descreve a disponibilidade de horários em formato semântico (ex: {"dia": "segunda", "turno": "manhã"}), é traduzido por um parser para o formato de slots de horário compatível com o algoritmo de

“Do conhecimento acadêmico à transformação sustentável: inovação com validação científica”

otimização (ex: ['2M1', '2M2', '2M3', '2M4']).

Geração da Grade: As informações extraídas e traduzidas (matérias cursadas e slots de horário) são enviadas ao algoritmo de otimização, que as utiliza para gerar a grade final recomendada, seguindo a hierarquia de prioridades.

Resultados e discussão

Os testes foram baseados em um conjunto de cinco casos de teste cuidadosamente selecionados, cada um projetado para desafiar diferentes aspectos da capacidade dos modelos de linguagem. O objetivo foi simular uma variedade de interações reais, desde um *prompt* mais direto e básico até cenários com muita informalidade e pouco contexto. Os casos incluíram:

- **Prompt Básico e direto:** Um teste de referência para avaliar a capacidade fundamental do modelo de processar solicitações claras.
- **Prompt com disponibilidade de horários complexa e sinônimos:** Um desafio para a compreensão de horários detalhados e a interpretação de termos variados que significam a mesma coisa.
- **Prompt visando o adiamento de disciplinas:** Um teste da habilidade do modelo em entender a intenção do usuário de avançar na grade curricular.
- **Prompt com linguagem informal e sem contexto de semestre:** Um teste para observar como o modelo lida com a falta de formalidade e informações cruciais.
- **Prompt com os termos "todos" e "exceto":** O teste final, focado na capacidade do modelo de processar negações e exceções, que é um ponto frequentemente problemático.

A análise do desempenho da LLM revelou contrastes notáveis entre as abordagens. A abordagem *Few-Shot Prompt*, utilizando o modelo Mistral-7B, mostrou-se promissora, obtendo sucesso em 3 dos 5 casos de teste, ou seja, os modelos de saída esperados foram análogos às respostas de saída do LLM. Como o caso de teste #3, com a descrição de “Adiantamento de matérias” referenciado na Tabela 1 para o seguinte exemplo:

- **Texto de Entrada:** "quero adiantar matérias, estou no 3º semestre. Posso estudar nas tardes de terça e nas primeiras duas aulas da noite de quarta."
- **Resultado Esperado (Formato JSON):**

```
{'semestre_atual': 3, 'materias_cumpridas': [], 'disponibilidade': [{'dia': 'terca', 'turno': 'tarde', 'detalhe': 'completo'}, {'dia': 'quarta', 'turno': 'noite', 'detalhe': 'primeiras duas aulas'}]}
```

- **Resultado Obtido (Formato JSON):**
{'semestre_atual': 3, 'materias_cumpridas': [], 'disponibilidade': [{'dia': 'terca', 'turno': 'tarde', 'detalhe': 'completo'}, {'dia': 'quarta', 'turno': 'noite', 'detalhe': 'primeiras duas aulas'}]}

Neste caso, o modelo falhou ao interpretar a disponibilidade do usuário, indicando a noite de quarta-feira como "não disponível" e adicionando a noite de quinta-feira, que não foi mencionada no texto de entrada. Isso demonstra que, embora a abordagem *Few-Shot* seja eficaz em muitos cenários, ainda pode apresentar imprecisões na extração e interpretação de informações específicas. Essa abordagem demonstrou capacidade de interpretar a intenção do usuário e extrair as informações necessárias, mesmo com um número limitado de exemplos. No entanto, suas falhas indicam a necessidade de melhorias para lidar com a complexidade da linguagem humana. A falha no caso #4 descrito na Tabela 1, por exemplo, demonstrou uma interpretação incorreta de "noites de seg e qua", enquanto a falha no caso #5, também referenciado na Tabela 1, evidenciou a dificuldade em processar a negação em "todas as noites, exceto quarta".

Tabela 1 – Resumo dos resultados dos testes de LLM

Caso de Teste	Descrição	Few-Shot	CoT
#1	Prompt Básico	Sucesso	Sucesso
#2	Disponibilidade complexa	Sucesso	Falha
#3	Adiantamento de matérias	Sucesso	Falha
#4	Linguagem informal	Falha	Falha
#5	Termos "todos" e "exceto"	Falha	Falha

Em contrapartida, a abordagem *Chain-of-Thought*, que busca aprimorar o raciocínio do modelo, teve um desempenho menos favorável, falhando em 4 dos 5 casos. As falhas foram principalmente causadas por pequenas imprecisões na formatação da saída e pela incapacidade de lidar com a ausência de contexto. O

“Do conhecimento acadêmico à transformação sustentável: inovação com validação científica”

baixo desempenho da abordagem *Chain-of-Thought*, que falhou em 80% dos casos, demonstra que o modelo não soube lidar com as pequenas imprecisões na formatação e com a ausência de contexto, características comuns em prompts humanos. A Tabela 1 resume o desempenho das duas abordagens, destacando os pontos de sucesso e falha em cada caso de teste.

Otimização de Grade

Em contraste com os desafios enfrentados pelos LLMs, a implementação do algoritmo de otimização de grade em Python juntamente com a técnica de *Few-Shot* se mostraram eficazes. Ele foi capaz de processar os dados de entrada de forma consistente, gerando uma grade otimizada com base nas prioridades definidas. O sistema demonstrou sua robustez ao identificar disciplinas elegíveis em diferentes níveis de prioridade e selecionar aquelas que se encaixavam nos horários disponíveis do aluno.

A Tabela 2 ilustra essa capacidade, apresentando uma grade recomendada para um aluno no 4º semestre que já concluiu um conjunto de disciplinas e tem todas as noites livres.

Tabela 2 – Grade recomendada para um aluno de Sistemas de Informação no 4º semestre

Código	Nome da Disciplina	Horário
SMAC03	Grafos	3N1, 3N2, 5N1, 5N2
SRSC02	Sistemas Operacionais	2N3, 2N4, 3N3, 3N4
XPAD01	Banco de Dados I	4N1, 4N2, 6N1, 6N2
ADM54H	Gestão de Carreira	4N3, 4N4

A abordagem algorítmica para a otimização foi validada, resultando em uma grade de horários recomendada e adequada para o aluno.

Limitações e Desafios Futuros

A principal limitação observada foi a dificuldade dos LLMs em lidar com as nuances da linguagem humana, como as negações e a necessidade de conversão exata para um formato pré-determinado, como observado nos casos #4 e #5. Essas falhas sugerem que, embora os modelos de linguagem sejam ferramentas poderosas, eles ainda requerem ajustes significativos para tarefas que exigem alta precisão.

Conclusões

Este estudo demonstrou a viabilidade de utilizar LLMs em conjunto com técnicas de *prompt engineering* para interpretar a intenção de alunos e automatizar a montagem de grades curriculares. O método se mostrou eficaz em cenários mais diretos, com a abordagem *Few-Shot* alcançando maior sucesso na extração de dados do que a *Chain-of-Thought*. No entanto, as dificuldades em interpretar contextos mais complexos, como negações e detalhes de horários, indicam que a precisão da ferramenta ainda pode ser melhorada. A integração do LLM com a técnica *Few-Shot* demonstrou uma boa taxa de sucesso quando combinada a flexibilidade da linguagem natural com a precisão de um sistema baseado em regras. Pesquisas futuras poderiam focar no desenvolvimento de prompts mais robustos, além de uma reavaliação da viabilidade de *fine-tuning* com um conjunto de dados mais diversificado.

Agradecimentos

Agradeço à Universidade Federal de Itajubá (UNIFEI) pelo apoio e pela infraestrutura que possibilitaram a realização deste trabalho. Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo fomento à pesquisa e à inovação tecnológica.

Referências

- LI, Yongjie et al. Adaptive Learning Systems: Personalized Curriculum Design Using LLM-Powered Analytics. 2025. Disponível em: <https://arxiv.org/html/2507.18949v1>. Acesso em: 7 ago. 2025.
- MA, Xilai; LI, Jing; ZHANG, Min. Chain of Thought with Explicit Evidence Reasoning for Few-shot Relation Extraction. 2024. Disponível em: <https://arxiv.org/html/2311.05922v3>. Acesso em: 24 jun. 2025.
- MISTRAL AI. Mistral-7B-Instruct-v0.2. 2023. Disponível em: <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2>. Acesso em: 07 jul. 2025.
- JIANG, Albert Q. et al. Mistral 7B. arXiv, 2023. Disponível em: <https://arxiv.org/pdf/2310.06825>. Acesso em: 07 jul. 2025.
- AMATRIAIN, Xavier. Prompt Design and Engineering: Introduction and Advanced Methods. 2024. Disponível em: <https://arxiv.org/abs/2401.14423>. Acesso em: 06 jun. 2025.
- XIAO, Tong; ZHU, Jingbo. Foundations of large language models. [S. l.]: NLP Lab, Northeastern University; NiuTrans Research, 2025. Disponível em: <https://arxiv.org/abs/2501.09223v2>. Acesso em: 02 ago. 2025.