

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Curso de Bacharelado em Engenharia de Computação



Trabalho de Conclusão de Curso

**VideoLearnAI: LLM10 Powered Web Application para aprendizagem ativa com
vídeos do Youtube**

Kevin Castro Weitgenant

Pelotas, 2025

Kevin Castro Weitgenant

**VideoLearnAI: LLM10 Powered Web Application para aprendizagem ativa com
vídeos do Youtube**

Trabalho de Conclusão de Curso apresentado
ao Centro de Desenvolvimento Tecnológico da
Universidade Federal de Pelotas, como requisito
parcial à obtenção do título de Bacharel em En-
genharia de Computação.

Orientador: Prof. Dr. Tiago Primo
Coorientador: Prof. Dr. Marilton Sanchotene de Aguiar

Pelotas, 2025

**Insira AQUI a ficha catalográfica
Quando finalizado o trabalho, deve ser
solicitada através do Sistema Cobalto
Biblioteca – Cadastro – Ficha catalográfica.**

Kevin Castro Weitgenant

VideoLearnAI: LLM10 Powered Web Application para aprendizagem ativa com vídeos do Youtube

Trabalho de Conclusão de Curso aprovado, como requisito parcial, para obtenção do grau de Bacharel em Engenharia de Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 16 de março de 2025

Banca Examinadora:

Prof. Dr. Marilton Sanchotene de Aguiar (orientador)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Paulo Roberto Ferreira Jr.
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Ricardo Matsumura Araujo
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Luciano da Silva Pinto
Doutor em Biotecnologia pela Universidade Federal de Pelotas.

Dedico...

AGRADECIMENTOS

Agradeço...

What would life be if we had no courage to attempt anything?
— VINCENT VAN GOGH

RESUMO

WEITGENANT, Kevin Castro. **VideoLearnAI: LLM10 Powered Web Application para aprendizagem ativa com vídeos do Youtube.** Orientador: Tiago Primo. 2025. 57 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2025.

Este trabalho apresenta o desenvolvimento de um Software como Serviço (SaaS) voltado para aprimorar a experiência de aprendizagem. A plataforma oferece cinco funcionalidades principais: melhoria automática da legibilidade de legendas, geração de capítulos, transcrição sincronizada, quizzes interativos e um sistema de bate-papo contextual baseado no conteúdo do vídeo. Os resultados demonstram o potencial da plataforma para tornar vídeos educacionais mais engajantes e eficazes.

Palavras-chave: palavrachave-um; palavrachave-dois; palavrachave-tres; palavrachave-quatro.

RESUMO

WEITGENANT, Kevin Castro. **AI-Powered Educational Platform: Transforming Video Content into Interactive Learning Experiences.** Orientador: Tiago Primo. 2025. 57 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2025.

This work presents the development of an educational Software as a Service (SaaS) platform that leverages Artificial Intelligence to enhance video-based learning experiences. The system implements five main functionalities: automatic subtitle readability improvement, chapter generation, synchronized transcription, interactive quiz generation, and a contextual chat system for video content. The solution employs Large Language Models (LLMs) and Transformer architecture to process and transform audiovisual content into interactive educational material. The implementation focused on scalability and performance, utilizing GPU processing and modern software development techniques. The results demonstrate the platform's potential for transforming videos into more engaging and effective learning experiences.

Palavras-chave: keyword-one; keyword-two; keyword-three; keyword-four.

LISTA DE FIGURAS

Figura 1	Logs de execução no Beam Serverless GPU demonstrando o tempo de cold start (18s 573ms) na primeira requisição e tempos de resposta reduzidos (aproximadamente 50-100ms) nas requisições subsequentes	36
Figura 2	Comparação antes e após a segmentação de parágrafos.	37
Figura 3	Logs de execução no Beam Serverless GPU demonstrando o tempo de cold start (18s 573ms) na primeira requisição e tempos de resposta reduzidos (aproximadamente 50-100ms) nas requisições subsequentes	38
Figura 4	Antes da geração de capítulos. Aviso de que vídeo não possui capítulos e botão para geração	40
Figura 5	Após geração de capítulos.	40
Figura 6	Tabela de conteúdos gerada automaticamente a partir dos capítulos identificados	41
Figura 7	Histórico de visualização mostrando os últimos vídeos assistidos pelo usuário	41
Figura 8	43
Figura 9	Exemplo de interface para perguntas e respostas.	44
Figura 10	Exemplo de interface para questões de verdadeiro ou falso.	45

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

- ABNT Associação Brasileira de Normas Técnicas
NUMA Non-Uniform Memory Access
SIMD Single Instruction Multiple Data
SMP Symmetric Multi-Processor
SPMD Single Program Multiple Data

SUMÁRIO

1 INTRODUÇÃO	16
1.1 Objetivo Geral	17
1.2 Objetivos Específicos	17
1.3 Estrutura do Trabalho	17
2 SOLUÇÕES RELACIONADAS	19
2.1 O Cenário Atual das Ferramentas de Aprendizado Assistido por IA	19
2.1.1 YouLearn.ai	19
2.1.2 StudyFetch	19
2.2 VideoLearnAI: Uma Abordagem Distinta	20
2.2.1 Características Distintivas e Vantagens	20
3 FUNDAMENTAÇÃO TEÓRICA	22
3.1 Large Language Models (LLMs)	22
3.1.1 Conceitos Fundamentais	22
3.1.2 Arquitetura e Funcionamento	23
3.1.3 Prompt Engineering	24
3.1.4 Janela de Contexto e Tokens	26
3.2 Aprendizagem ativa	28
3.3 Processamento de Texto	28
3.3.1 Segmentação de Texto	28
3.3.2 Abordagens Tradicionais	28
3.3.3 Segment Any Text (SAT)	28
3.4 Aplicações Baseadas em LLMs	28
3.4.1 Integração com APIs de LLMs	28
3.4.2 Function Calling	28
3.4.3 Streaming de Respostas	28
3.4.4 Rate Limiting e Custos	28
3.5 Computação em Nuvem Moderna	28
3.5.1 Arquitetura Serverless	28
3.5.2 Serverless GPU Computing	28
3.5.3 Desafios de Cold Start	28
3.5.4 Estratégias de Otimização	28
4 TECNOLOGIAS UTILIZADAS	29
4.1 Frameworks e Bibliotecas	29
4.1.1 FastAPI	29
4.1.2 Next.js	29

4.1.3	Vercel AI SDK	29
4.1.4	Drizzle ORM	30
4.1.5	ShadCN	30
4.1.6	Zustand	30
4.1.7	Stripe	31
4.2	Modelos de Machine Learning	31
4.2.1	API Deepgram	31
4.2.2	API GPT	31
4.2.3	Segment Anything (SaT)	31
4.3	Infraestrutura e Serviços em Nuvem	31
4.3.1	Open Next.js	31
4.3.2	Google Cloud	31
4.3.3	Beam Serverless GPU	31
4.3.4	Supabase	32
4.3.5	Sentry	32
4.4	Ferramentas de Desenvolvimento	32
4.4.1	Visual Studio Code (VS Code)	32
4.4.2	Cursor	32
4.4.3	v0.dev	32
4.4.4	OpenAPI TypeScript Code Generator	32
5	DESENVOLVIMENTO DA APLICAÇÃO	33
5.1	Visão Geral da Arquitetura	33
5.2	Melhoria da Legibilidade das Legendas	33
5.2.1	O Problema da Legibilidade	33
5.2.2	Primeira Abordagem com LLMs	34
5.2.3	Implementação com SaT (Segment Anything Text)	34
5.2.4	Interface e Experiência do Usuário	36
5.2.5	Desafios de Implantação em Produção	37
5.3	Geração de Capítulos	38
5.3.1	Algoritmo e Implementação	39
5.3.2	Integração com LLMs	39
5.3.3	Tabela de Conteúdos	41
5.3.4	Últimos Vídeos Vistos	41
5.4	Transcrição	42
5.4.1	Primeira implementação com Whisper	42
5.4.2	Migração para o Deepgram	42
5.4.3	Vantagens adicionais do Deepgram	42
5.4.4	Interface e Experiência do Usuário	43
5.5	Geração de Quizzes	43
5.5.1	Questões discursivas	44
5.5.2	Questões de Verdadeiro ou Falso	45
5.6	Prompt por capítulo	46
5.7	Bate-Papo com Vídeo	46
5.7.1	Visão Geral	46
5.7.2	Interface do Usuário	46
5.7.3	Gerenciamento de Contexto e Tokens	46
5.7.4	Implementação Técnica	46
5.7.5	Benefícios e Características	47

5.8 Desafios e Soluções	47
5.8.1 Obtenção dos dados do youtube	47
5.8.2 Utilização de GPU's em produção	47
6 CONCLUSÃO	48
6.1 Objetivos Alcançados	48
6.2 Trabalhos Futuros	49
6.2.1 Diarização e Análise de Múltiplos Falantes	49
6.2.2 Otimizações Técnicas	49
6.2.3 Expansão de Funcionalidades	49
6.2.4 Melhorias na Experiência do Usuário	50
REFERÊNCIAS	51
APÊNDICE A UM APÊNDICE	55
ANEXO A UM ANEXO	57

1 INTRODUÇÃO

Nos últimos anos, o consumo de conteúdo educacional em vídeo tem crescido exponencialmente, impulsionado por plataformas como YouTube, Coursera e Udemy. Hoje, é possível encontrar aulas completas de universidades de altíssimo nível, como MIT, Harvard e Stanford, gratuitamente disponíveis online. No entanto, apesar da abundância de material de qualidade, muitos usuários enfrentam dificuldades em absorver e reter conhecimento de forma eficiente. A maioria das pessoas consome esses conteúdos de maneira passiva, apenas assistindo aos vídeos sem um envolvimento ativo com o material. Isso limita a retenção e a compreensão das informações.

A aprendizagem ativa, por outro lado, é um modelo comprovadamente mais eficaz, pois envolve o estudante em processos como resumo, questionamento, reorganização do conteúdo e interação com o material. Pesquisas mostram que métodos ativos de estudo, como fazer perguntas sobre o conteúdo, testar-se frequentemente e organizar a informação de forma estruturada, levam a um aprendizado mais profundo e duradouro.

Diante desse cenário, este trabalho apresenta o desenvolvimento de um Software as a Service (SaaS) voltado para transformar o consumo passivo de vídeos educacionais em um processo de aprendizagem ativa. A solução utiliza modelos de linguagem natural (LLMs) para reestruturar legendas em textos mais legíveis, gerar capítulos automáticos, fornecer resumos e permitir interações como perguntas e respostas sobre o conteúdo. Além disso, o sistema oferece quizzes dinâmicos para reforçar o aprendizado e um mecanismo para salvar o progresso dos usuários, incentivando um envolvimento mais estruturado com os vídeos.

O desenvolvimento do SaaS seguiu uma abordagem iterativa. A arquitetura da aplicação integra tecnologias como FastAPI, Next.js e Transformers, além de estratégias de otimização para garantir eficiência e escalabilidade. A validação da ferramenta inclui métricas de desempenho e feedback dos usuários, avaliando sua eficácia na melhoria da compreensão e retenção do conhecimento.

Com esta pesquisa, buscamos não apenas oferecer uma ferramenta inovadora para aprendizado com vídeos, mas também contribuir para a democratização da educação de qualidade, permitindo que qualquer pessoa tenha acesso a um método mais eficaz

para extrair o máximo de conhecimento dos conteúdos disponíveis online.

1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um Software as a Service (SaaS) que transforme o consumo passivo de vídeos educacionais em um processo de aprendizagem ativa. Para isso, a plataforma utilizará inteligência artificial para melhorar a legibilidade das legendas, gerar resumos, estruturar conteúdos em capítulos, criar quizzes interativos e permitir interações diretas com o conteúdo por meio de perguntas e respostas. O foco é tornar o aprendizado com vídeos mais eficiente, estruturado e acessível, permitindo que qualquer pessoa aproveite melhor o vasto acervo educacional disponível online.

1.2 Objetivos Específicos

Para atingir o objetivo geral, este trabalho busca:

- Desenvolver um sistema que reestruture legendas de vídeos em textos mais legíveis e organizados, facilitando a compreensão.
- Implementar um mecanismo para geração automática de capítulos e resumos, permitindo uma navegação mais eficiente pelo conteúdo.
- Criar um módulo de perguntas e respostas, possibilitando interações com o vídeo de forma contextualizada.
- Desenvolver um sistema de quizzes automáticos baseados no conteúdo dos vídeos, reforçando o aprendizado ativo.
- Implementar um sistema de salvamento de progresso para permitir que usuários retomem facilmente seus estudos.

1.3 Estrutura do Trabalho

Este trabalho está organizado da seguinte forma:

- **Capítulo 2 – Revisão da Literatura:** apresenta os conceitos fundamentais de aprendizagem ativa e modelos de linguagem natural (LLMs), que embasam o desenvolvimento da aplicação.
- **Capítulo 3 – Metodologia:** descreve o processo de desenvolvimento do SaaS, incluindo a definição de requisitos, prototipação, escolha de tecnologias e critérios de avaliação.

- **Capítulo 4 – Desenvolvimento da Aplicação:** detalha as funcionalidades do sistema, explicando a implementação de cada módulo e os desafios enfrentados.
- **Capítulo 5 – Resultados:** analisa o desempenho da aplicação e apresenta o feedback dos usuários, avaliando o impacto da ferramenta na experiência de aprendizado.
- **Capítulo 6 – Conclusão:** discute os objetivos alcançados, as principais contribuições do trabalho e sugestões para aprimoramentos futuros.

2 SOLUÇÕES RELACIONADAS

2.1 O Cenário Atual das Ferramentas de Aprendizado Assistido por IA

» dar enfase principalmente que os quizzes são gerados para o conteúdo todo, o que acaba gerando uma certa sensação de cansaço para os usuários, por exemplo, pra um vídeo pequeno eu gerei umas 200 flashcards. Não existe a possibilidade de selecionar as partes que deseja se criar quizzes.

» Além disso, não existe possibilidade de customização do prompt para a geração de quizzes.

» Outro diferencial, marcar progresso.

» Improved readability e geração de transcrição com IA.

—> kevin do futuro, escrever isso melhor

O recente avanço em inteligência artificial, especialmente em processamento de linguagem natural, tem possibilitado o desenvolvimento simultâneo de diversas ferramentas educacionais. Este capítulo apresenta duas soluções existentes no mercado e introduz a VideoLearnAI, desenvolvida neste trabalho, destacando seus diferenciais

2.1.1 YouLearn.ai

Uma das soluções que emergiram neste contexto oferece:

- Transformação de materiais em ferramentas interativas
- Geração de flashcards e quizzes
- Suporte a múltiplos formatos (PDFs, vídeos, slides)
- Sistema de chat com IA para dúvidas

2.1.2 StudyFetch

Paralelamente, esta plataforma desenvolveu:

- Criação de conjuntos de estudo personalizados
- Tutor AI (Spark.E) para assistência em tempo real
- Geração de materiais de estudo
- Recursos de colaboração e estudo em grupo

2.2 VideoLearnAI: Uma Abordagem Distinta

A VideoLearnAI, também desenvolvida no contexto destes avanços em IA, apresenta uma implementação diferenciada para o processamento e apresentação do conteúdo educacional. Enquanto as outras plataformas optaram por fragmentar o conteúdo em elementos menores, a VideoLearnAI estabeleceu um caminho próprio.

2.2.1 Características Distintivas e Vantagens

A VideoLearnAI se diferencia fundamentalmente em sua abordagem para processamento e apresentação do conteúdo educacional. A plataforma realiza a transformação de vídeos em texto estruturado e coeso, mantendo a narrativa e o contexto original do material. Esta abordagem proporciona uma experiência de leitura fluida, mais próxima à leitura de um livro do que à interação fragmentada comum em outras plataformas.

Um diferencial significativo está no sistema transparente de geração de quizzes. Enquanto outras soluções tratam a geração de questões como uma "caixa preta", a VideoLearnAI expõe e permite a customização dos prompts utilizados neste processo. Esta transparência oferece aos usuários controle preciso sobre o tipo e estilo das questões geradas, possibilitando ajustes finos na dificuldade e no foco do material de estudo.

A experiência do usuário foi cuidadosamente pensada para proporcionar um consumo mais natural do conteúdo em vídeo, mantendo uma leitura contínua e fluida. A personalização efetiva do material de estudo é facilitada pela transparência do sistema, permitindo que cada usuário adapte o conteúdo às suas necessidades específicas de aprendizado.

O equilíbrio entre automação e controle do usuário é um aspecto central da plataforma. A customização completa dos prompts de geração de questões, combinada com a transparência no processo de criação de materiais, permite que os usuários aproveitem os benefícios da automação por IA sem perder o controle sobre o processo de aprendizado.

É interessante notar como, partindo da mesma base tecnológica, diferentes equipes desenvolveram abordagens distintas para melhorar a experiência de aprendizado. A VideoLearnAI, desenvolvida de forma independente e simultânea às outras soluções,

destaca-se por sua abordagem única na transformação de conteúdo em vídeo, oferecendo uma experiência mais natural e coesa de aprendizado.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Large Language Models (LLMs)

Os Large Language Models (LLMs) representam um avanço significativo no campo da Inteligência Artificial, especificamente no Processamento de Linguagem Natural (PLN). Estes modelos têm revolucionado diversas áreas, desde assistentes virtuais até sistemas educacionais, como o proposto neste trabalho. Esta seção explora os conceitos fundamentais, arquitetura, técnicas de prompt engineering e aspectos relacionados à janela de contexto destes modelos.

3.1.1 Conceitos Fundamentais

Large Language Models são sistemas de inteligência artificial treinados em vastos corpora de texto para aprender padrões estatísticos da linguagem humana. Diferentemente dos sistemas tradicionais de PLN, que frequentemente dependiam de regras pré-definidas ou características manualmente extraídas, os LLMs aprendem representações contextuais ricas diretamente dos dados, como demonstrado no trabalho "Language Models are Few-Shot Learners"(Brown et al., 2020) (Brown; Mann; Ryder; Subbiah; Kaplan; Dhariwal; Neelakantan; Shyam; Sastry; Askell et al., 2020).

O desenvolvimento dos LLMs pode ser compreendido como uma evolução natural dos modelos de linguagem estatísticos. Enquanto os modelos tradicionais, como n-gramas, previam a próxima palavra baseando-se apenas em um contexto limitado de palavras anteriores, os LLMs modernos consideram contextos muito mais amplos e capturam dependências de longo alcance no texto, conforme apresentado no artigo "Attention is All You Need"(Vaswani et al., 2017) (Vaswani; Shazeer; Parmar; Uszkoreit; Jones; Gomez; Kaiser; Polosukhin, 2017).

Um aspecto fundamental dos LLMs é sua capacidade de aprendizado de representações contextuais. Modelos como BERT ("BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding") (Devlin et al., 2019) (Devlin; Chang; Lee; Toutanova, 2019) e GPT ("Language Models are Unsupervised Multitask Learners") (Radford et al., 2019) (Radford; Wu; Child; Luan; Amodei; Sutskever, 2019) apren-

dem representações vetoriais (embeddings) para palavras que variam dependendo do contexto em que aparecem, capturando assim nuances semânticas que modelos anteriores não conseguiam.

O treinamento destes modelos ocorre em duas fases principais: pré-treinamento e ajuste fino (fine-tuning). Durante o pré-treinamento, o modelo é exposto a enormes quantidades de texto não rotulado, aprendendo padrões linguísticos gerais. No ajuste fino, o modelo é especializado para tarefas específicas usando conjuntos de dados menores e rotulados, como explorado no trabalho "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"(Raffel et al., 2020) (Raffel; Shazeer; Roberts; Lee; Narang; Matena; Zhou; Li; Liu, 2020).

Um conceito crucial para entender os LLMs modernos é o de aprendizado auto-supervisionado. Nesta abordagem, o próprio texto serve como supervisão, com o modelo sendo treinado para prever partes mascaradas do texto (como no BERT) ou a próxima palavra na sequência (como no GPT). Esta capacidade de extrair sinais de supervisão dos próprios dados permitiu o treinamento em escala sem precedentes, conforme discutido em "Self-supervised Learning: Generative or Contrastive"(Liu et al., 2021) (Liu; Zhang; Hou; Mian; Wang; Zhang; Tang, 2021).

3.1.2 Arquitetura e Funcionamento

A arquitetura predominante nos LLMs modernos é baseada no Transformer, introduzido no trabalho seminal "Attention is All You Need"(Vaswani et al., 2017) (Vaswani; Shazeer; Parmar; Uszkoreit; Jones; Gomez; Kaiser; Polosukhin, 2017). Esta arquitetura revolucionou o PLN ao substituir as redes recorrentes (RNNs) e convolucionais (CNNs) por um mecanismo de atenção que permite processar todas as palavras de uma sequência simultaneamente, em vez de sequencialmente.

O componente central da arquitetura Transformer é o mecanismo de auto-atenção (self-attention), que permite ao modelo ponderar a importância de diferentes palavras em uma sentença ao processar cada palavra. Isso possibilita a captura de dependências de longo alcance no texto, independentemente da distância entre as palavras relacionadas.

Os LLMs modernos podem ser classificados em três categorias principais baseadas na arquitetura Transformer:

- **Encoder-only:** Modelos como BERT ("BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding") (Devlin et al., 2019) (Devlin; Chang; Lee; Toutanova, 2019) utilizam apenas a parte do codificador do Transformer. São bidirecionais, considerando tanto o contexto à esquerda quanto à direita de cada palavra, e são particularmente eficazes em tarefas de compreensão de linguagem.

- **Decoder-only:** Modelos como GPT ("Language Models are Unsupervised Multitask Learners") (Radford et al., 2019) (Radford; Wu; Child; Luan; Amodei; Sutskever, 2019) utilizam apenas a parte do decodificador. São unidirecionais, considerando apenas o contexto à esquerda (palavras anteriores) e são especializados em geração de texto.
- **Encoder-decoder:** Modelos como T5 ("Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer") (Raffel et al., 2020) (Raffel; Shazeer; Roberts; Lee; Narang; Matena; Zhou; Li; Liu, 2020) utilizam ambas as partes, sendo eficazes tanto para compreensão quanto para geração de texto, especialmente em tarefas de transformação de texto como tradução e resumo.

O escalonamento dos modelos tem sido um fator crucial para o avanço dos LLMs. Pesquisas demonstraram que aumentar o número de parâmetros, a quantidade de dados de treinamento e o poder computacional leva a melhorias consistentes no desempenho, como evidenciado no estudo "Scaling Laws for Neural Language Models"(Kaplan et al., 2020) (Kaplan; Mccandlish; Henighan; Brown; Chess; Child; Gray; Radford; Wu; Amodei, 2020). Este fenômeno, conhecido como "scaling laws", tem guiado o desenvolvimento de modelos cada vez maiores, como o GPT-3 com 175 bilhões de parâmetros ("Language Models are Few-Shot Learners") (Brown et al., 2020) (Brown; Mann; Ryder; Subbiah; Kaplan; Dhariwal; Neelakantan; Shyam; Sastry; Askell et al., 2020) e o GPT-4, cujo número exato de parâmetros não foi divulgado, mas estima-se que seja significativamente maior ("GPT-4 Technical Report") (OpenAI, 2023) (OpenAI, 2023a).

Um aspecto importante do funcionamento dos LLMs é o processo de tokenização, que converte o texto em unidades discretas (tokens) que o modelo pode processar. Diferentes abordagens incluem tokenização baseada em palavras, caracteres ou subpalavras, sendo esta última a mais comum em modelos recentes por oferecer um bom equilíbrio entre eficiência e capacidade de lidar com palavras desconhecidas, como demonstrado no trabalho "Neural Machine Translation of Rare Words with Subword Units"(Sennrich et al., 2016) (Sennrich; Haddow; Birch, 2016).

3.1.3 Prompt Engineering

Prompt engineering refere-se à prática de formular instruções ou consultas (prompts) para LLMs de maneira a obter respostas mais precisas, relevantes e úteis. Com o advento de modelos como GPT-3 e GPT-4, capazes de realizar diversas tarefas sem fine-tuning específico, a qualidade do prompt tornou-se um fator determinante para o desempenho do modelo, como discutido em "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing"(Liu et al., 2023) (Liu; Yuan; Fu; Jiang; Hayashi; Neubig, 2023).

A eficácia do prompt engineering baseia-se no fenômeno de "in-context learning", onde os LLMs podem adaptar seu comportamento com base apenas nos exemplos ou instruções fornecidos no prompt, sem modificação de seus parâmetros, como demonstrado em "Language Models are Few-Shot Learners"(Brown et al., 2020) (Brown; Mann; Ryder; Subbiah; Kaplan; Dhariwal; Neelakantan; Shyam; Sastry; Askell et al., 2020). Este comportamento emergente permite que os modelos realizem tarefas para as quais não foram explicitamente treinados.

Várias técnicas de prompt engineering têm sido desenvolvidas e estudadas:

- **Zero-shot prompting:** Fornecer apenas a instrução, sem exemplos, confiando na capacidade do modelo de entender a tarefa, como explorado em "Large Language Models are Zero-Shot Reasoners"(Kojima et al., 2022) (Kojima; Gu; Reid; Matsuo; Iwasawa, 2022).
- **Few-shot prompting:** Incluir alguns exemplos de entrada-saída no prompt para guiar o modelo, técnica apresentada em "Language Models are Few-Shot Learners"(Brown et al., 2020) (Brown; Mann; Ryder; Subbiah; Kaplan; Dhariwal; Neelakantan; Shyam; Sastry; Askell et al., 2020).
- **Chain-of-thought prompting:** Solicitar ao modelo que explique seu raciocínio passo a passo, melhorando significativamente o desempenho em tarefas de raciocínio complexo, como demonstrado em "Chain of Thought Prompting Elicits Reasoning in Large Language Models"(Wei et al., 2022) (Wei; Wang; Schuurmans; Bosma; Ichter; Xia; Chi; Le; Zhou, 2022).
- **Self-consistency:** Gerar múltiplas cadeias de raciocínio e selecionar a resposta mais consistente, aumentando a precisão em tarefas matemáticas e lógicas, conforme proposto em "Self-Consistency Improves Chain of Thought Reasoning in Language Models"(Wang et al., 2023) (Wang; Wei; Schuurmans; Le; Chi; Zhou, 2023).
- **ReAct:** Combinar raciocínio e ações, permitindo que o modelo interaja com ferramentas externas para resolver problemas, abordagem introduzida em "ReAct: Synergizing Reasoning and Acting in Language Models"(Yao et al., 2023) (Yao; Zhao; Yu; Du; Shafran; Narasimhan; Cao, 2023).

A estruturação do prompt também é crucial. Pesquisas mostram que a ordem dos exemplos, a formatação do texto e até mesmo a escolha de palavras específicas podem afetar significativamente o desempenho do modelo, como evidenciado em "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity"(Lu et al., 2022) (Lu; Bartolo; Moore; Riedel; Stenetorp, 2022). Além disso, a inclusão de metadados como o papel que o modelo deve assumir (por exemplo,

"Você é um assistente útil") pode influenciar o estilo e a qualidade das respostas, como discutido em "Prompt Engineering for Large Language Models: A Survey"(White et al., 2023) (White; Fu; Hays; Sandborn; Olea; Gilbert; Elnashar; Spencer-smith; Schmidt, 2023).

No contexto de aplicações educacionais como a proposta neste trabalho, o prompt engineering é particularmente importante para garantir que os LLMs gerem conteúdo pedagogicamente adequado, preciso e adaptado ao nível de conhecimento do estudante, como analisado em "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education"(Kasneci et al., 2023) (Kasneci; Sessler; Küchemann; Bannert; Dementieva; Fischer; Gasser; Groh; Günemann; Hüllermeier et al., 2023).

3.1.4 Janela de Contexto e Tokens

A janela de contexto refere-se à quantidade máxima de texto que um LLM pode processar em uma única operação. Esta limitação é uma consideração crítica no design de aplicações baseadas em LLMs, especialmente aquelas que lidam com documentos longos ou conversas extensas, como investigado no estudo "Lost in the Middle: How Language Models Use Long Contexts"(Liu et al., 2023) (Liu; Bosselut; Srinivasan; Choi; Hajishirzi; Khashabi, 2023).

Os LLMs processam texto dividindo-o em unidades chamadas tokens, que podem representar palavras, partes de palavras ou caracteres individuais, dependendo do algoritmo de tokenização utilizado. O número de tokens que um modelo pode processar simultaneamente é determinado por limitações de hardware (principalmente memória GPU) e pela arquitetura do modelo, como explicado em "Attention is All You Need"(Vaswani et al., 2017) (Vaswani; Shazeer; Parmar; Uszkoreit; Jones; Gomez; Kaiser; Polosukhin, 2017).

A evolução dos LLMs tem sido marcada por um aumento constante no tamanho da janela de contexto. Modelos iniciais como o GPT-2 ("Language Models are Unsupervised Multitask Learners") (Radford et al., 2019) (Radford; Wu; Child; Luan; Amodei; Sutskever, 2019) tinham uma janela de contexto de 1.024 tokens, enquanto versões recentes como o GPT-4 Turbo ("GPT-4 Technical Report") (OpenAI, 2023) (OpenAI, 2023a) podem processar até 128.000 tokens. Este aumento permite aplicações mais sofisticadas, como análise de documentos longos, resumo de livros inteiros e manutenção de conversas extensas com histórico completo.

No entanto, mesmo com janelas de contexto expandidas, os LLMs enfrentam desafios ao lidar com contextos longos. Pesquisas recentes identificaram o fenômeno de "lost in the middle", onde informações localizadas no meio de um texto longo têm menor probabilidade de serem utilizadas pelo modelo em suas respostas, como demonstrado em "Lost in the Middle: How Language Models Use Long Contexts"(Liu et

al., 2023) (Liu; Bosselut; Srinivasan; Choi; Hajishirzi; Khashabi, 2023). Este efeito pode ser atribuído a limitações no mecanismo de atenção e à forma como os modelos foram treinados.

Para mitigar as limitações da janela de contexto, várias estratégias têm sido desenvolvidas:

- **Chunking:** Dividir documentos longos em segmentos menores que podem ser processados separadamente, como discutido em "Retrieval-Augmented Generation for Large Language Models: A Survey"(Gao et al., 2023) (Gao; Xiong; Gao; Jiang; Shen; Ren; Han, 2023).
- **Sliding window:** Processar o texto em janelas sobrepostas para manter a coerência entre segmentos, abordagem proposta em "Longformer: The Long-Document Transformer"(Beltagy et al., 2020) (Beltagy; Peters; Cohan, 2020).
- **Retrieval-Augmented Generation (RAG):** Combinar LLMs com sistemas de recuperação de informação para acessar conhecimento externo sem sobrecarregar a janela de contexto, método introduzido em "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"(Lewis et al., 2020) (Lewis; Perez; Piktus; Petroni; Karpukhin; Goyal; Küttler; Lewis; Yih; Rocktäschel et al., 2020).
- **Compressão de contexto:** Resumir partes do histórico de conversação para liberar espaço na janela de contexto, técnica explorada em "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"(Xu et al., 2023) (Xu; Sarthi; Agarwal; Gupta; Saxena; Aralikatte; Batra; Parikh; Misra; Awadallah, 2023).

A eficiência no uso da janela de contexto é particularmente importante em aplicações como a proposta neste trabalho, onde é necessário processar transcrições de vídeos potencialmente longas. A implementação de técnicas como chunking inteligente baseado em capítulos ou segmentos temáticos pode melhorar significativamente a qualidade das respostas geradas pelo modelo, como sugerido em "Retrieval-Augmented Generation for Large Language Models: A Survey"(Gao et al., 2023) (Gao; Xiong; Gao; Jiang; Shen; Ren; Han, 2023).

Além disso, a otimização do uso de tokens é crucial para controlar custos em aplicações comerciais, já que os serviços de API de LLMs geralmente cobram com base no número de tokens processados (OpenAI, 2023) (OpenAI, 2023b). Estratégias como a remoção de informações redundantes e a priorização de conteúdo relevante podem reduzir significativamente o consumo de tokens sem comprometer a qualidade das respostas.

3.2 Aprendizagem ativa

3.3 Processamento de Texto

3.3.1 Segmentação de Texto

3.3.2 Abordagens Tradicionais

3.3.3 Segment Any Text (SAT)

3.4 Aplicações Baseadas em LLMs

3.4.1 Integração com APIs de LLMs

3.4.2 Function Calling

3.4.3 Streaming de Respostas

3.4.4 Rate Limiting e Custos

3.5 Computação em Nuvem Moderna

3.5.1 Arquitetura Serverless

3.5.2 Serverless GPU Computing

3.5.3 Desafios de Cold Start

3.5.4 Estratégias de Otimização

4 TECNOLOGIAS UTILIZADAS

4.1 Frameworks e Bibliotecas

4.1.1 FastAPI

FastAPI foi utilizado para criar um serviço backend complementar em Python, expondo endpoints para funcionalidades específicas de extração de dados do YouTube e operações com o modelo SaT. Um diferencial importante foi a geração automática de clients TypeScript através do OpenAPI, permitindo uma integração tipo-segura e seamless com o frontend Next.js. Esta capacidade de TypeScript code generation eliminou a necessidade de definir tipos manualmente e garantiu consistência na comunicação entre os serviços.

4.1.2 Next.js

Next.js atuou como o framework full-stack principal, gerenciando tanto o frontend quanto a maior parte do backend da aplicação. Através de seus recursos de API Routes e Server Components, foi possível construir uma aplicação completa com renderização híbrida, manipulação de estado servidor/cliente e APIs RESTful. Sua arquitetura permitiu manter a maioria das operações de backend centralizadas, recorrendo ao serviço Python apenas para funcionalidades específicas.

4.1.3 Vercel AI SDK

O **Vercel AI SDK** desempenhou um papel fundamental na implementação de funcionalidades de inteligência artificial diretamente no **Next.js**, proporcionando uma integração eficiente com modelos de linguagem natural (LLMs).

Uma das principais vantagens dessa biblioteca é a facilidade na construção de interfaces de usuário interativas e dinâmicas para aplicações baseadas em IA. O SDK permite o **streaming de respostas**, garantindo uma experiência mais fluida para o usuário, sem a necessidade de aguardar a conclusão total do processamento. Essa abordagem melhora significativamente a experiência do usuário (**UX**), tornando as interações mais ágeis e responsivas.

Além do texto, outros elementos da interface, como **quizzes gerados dinamicamente**, também são entregues por meio de streaming. Isso significa que as perguntas e respostas dos quizzes não precisam ser completamente processadas antes de serem exibidas, pois são disponibilizadas conforme são geradas pelo modelo de IA. Todo esse processo é abstruído pelo **Vercel AI SDK**, simplificando a implementação e reduzindo a complexidade no desenvolvimento da interface.

Dessa forma, o uso do **Vercel AI SDK** permitiu a criação de uma **UI altamente responsiva e eficiente**, eliminando a necessidade de comunicação constante com o backend Python para determinadas operações de IA, otimizando o desempenho e a escalabilidade da aplicação.

4.1.4 Drizzle ORM

O **Drizzle ORM** foi utilizado como ORM principal no **Next.js**, oferecendo uma interface *type-safe* para interações com o banco de dados **PostgreSQL**. Sua integração direta com **TypeScript** permitiu manter a consistência de tipos entre o banco de dados e a aplicação.

Além disso, o uso do **Drizzle ORM** faz sentido dentro da arquitetura **serverless** do projeto. Diferente de ORMs mais tradicionais, que podem apresentar desafios com conexões persistentes em ambientes serverless, o **Drizzle ORM** foi projetado para ser leve e eficiente, garantindo melhor compatibilidade com essa abordagem. Dessa forma, a aplicação pode escalar dinamicamente sem sobrecarga desnecessária na comunicação com o banco de dados.

4.1.5 ShadCN

ShadCN forneceu componentes de UI reutilizáveis e personalizáveis, sendo utilizado exclusivamente na camada de frontend do Next.js para construir interfaces consistentes e acessíveis.

4.1.6 Zustand

Zustand gerenciou o estado global do frontend, complementando o gerenciamento de estado servidor/cliente do Next.js com uma solução leve e eficiente para estados efêmeros do cliente.

4.1.7 Stripe

4.2 Modelos de Machine Learning

4.2.1 API Deepgram

4.2.2 API GPT

A API GPT foi integrada primariamente através do Next.js, utilizando o Vercel AI SDK para streaming de respostas e gerenciamento eficiente de prompts.

4.2.3 Segment Anything (SaT)

O modelo **Segment Anything (SaT)** foi utilizado para segmentar as legendas extraídas de vídeos do **YouTube** em parágrafos, com o objetivo de melhorar a legibilidade e organização do texto transscrito. Essa segmentação estruturada facilitou a compreensão e a análise do conteúdo, tornando a experiência do usuário mais intuitiva e agradável.

Durante a fase de desenvolvimento, o modelo foi executado no backend utilizando **FastAPI**. No entanto, devido ao alto custo associado à manutenção de servidores com **GPU**, optou-se por migrar a execução para a infraestrutura **Beam Serverless GPU** em produção. Esse serviço permitiu a execução do modelo de forma escalável e sob demanda, reduzindo os custos operacionais sem comprometer significativamente o desempenho. A principal desvantagem foi uma pequena latência ocasionada pelos *cold starts*, mas isso foi minimizado devido a várias técnicas disponibilizadas pelo serviço.

A **Beam Serverless GPU** será explorada com mais detalhes no próximo capítulo.

4.3 Infraestrutura e Serviços em Nuvem

4.3.1 Open Next.js

Open Next.js otimizou o deploy da aplicação Next.js full-stack, garantindo performance adequada tanto para o frontend quanto para as API Routes do backend.

4.3.2 Google Cloud

Google Cloud Platform hospedou o backend Python complementar, enquanto também fornecia outros serviços de infraestrutura necessários para a aplicação.

4.3.3 Beam Serverless GPU

Beam Serverless GPU foi utilizado especificamente para o backend Python, fornecendo recursos de GPU para processamento de modelos de machine learning mais pesados.

4.3.4 Supabase

Supabase forneceu o banco de dados PostgreSQL e serviços de autenticação, sendo acessado principalmente através do backend Next.js via Drizzle ORM.

4.3.5 Sentry

Sentry monitorou tanto o ambiente Next.js quanto o serviço Python, fornecendo visibilidade completa sobre erros e performance em toda a aplicação.

4.4 Ferramentas de Desenvolvimento

4.4.1 Visual Studio Code (VS Code)

VS Code foi o IDE principal, oferecendo excelente suporte tanto para o desenvolvimento em Next.js/TypeScript quanto para Python.

4.4.2 Cursor

Cursor complementou o ambiente de desenvolvimento com recursos de IA para autocomplete e refatoração, sendo especialmente útil no desenvolvimento full-stack.

4.4.3 v0.dev

v0.dev auxiliou na prototipagem rápida de componentes frontend para o Next.js, acelerando o desenvolvimento da interface do usuário.

4.4.4 OpenAPI TypeScript Code Generator

A geração automática de código TypeScript a partir das especificações OpenAPI do FastAPI foi uma ferramenta crucial no desenvolvimento, criando automaticamente tipos e clients para todas as APIs Python. Isto garantiu uma integração tipo-segura entre o frontend Next.js e o backend Python, reduzindo erros de integração e melhorando a experiência de desenvolvimento.

5 DESENVOLVIMENTO DA APLICAÇÃO

5.1 Visão Geral da Arquitetura

5.2 Melhoria da Legibilidade das Legendas

Para resolver o desafio de aprimorar a legibilidade das legendas, foram exploradas duas abordagens distintas. Inicialmente, testou-se uma solução baseada em LLMs (Large Language Models) combinada com um sistema de validação do resultado. Posteriormente, realizou-se uma segunda implementação, dessa vez utilizando o modelo SaT (Segment Any Text). Essa última abordagem se mostrou mais promissora.

5.2.1 O Problema da Legibilidade

As legendas de vídeos frequentemente apresentam problemas de formatação e segmentação, dificultando a compreensão do conteúdo. Esse problema ocorre porque as transcrições brutas costumam ser geradas como um fluxo contínuo de palavras, sem uma estrutura clara de frases e parágrafos. Não foram construídas tendo em mente a leitura da transcrição como um todo e sim apenas aparecer na tela no momento certo.

A disponibilização da transcrição com boa legibilidade, lado a lado com o conteúdo audiovisual traz diversos benefícios para o processo de aprendizagem. Primeiro, permite que o usuário rapidamente navegue pelo conteúdo, identificando e pulando seções menos relevantes para seu objetivo de estudo. Segundo, quando algum trecho do vídeo não foi bem compreendido, a possibilidade de reler a transcrição daquela parte específica oferece uma abordagem alternativa para entender o conteúdo. Por fim, a integração entre vídeo e transcrição oferece navegação bidirecional: além de realçar automaticamente o texto conforme o vídeo progride, permite saltar para qualquer momento do vídeo com um clique na transcrição, tornando a experiência do usuário mais fluida e interativa.

Na aplicação desenvolvida, o foco foi especificamente na melhoria da segmentação do texto em frases e parágrafos. Embora existam modelos promissores para adicionar pontuação adequada ao texto, como em (Guhr; Schumann; Bahrmann; Böhme, 2021) que utiliza transformers multilíngues para essa tarefa obtendo F1-score médio de 0,94

para detecção de final de sentença em textos em inglês, alemão, francês e italiano, essa funcionalidade será explorada em uma iteração futura do projeto.

5.2.2 Primeira Abordagem com LLMs

Inicialmente, foi utilizado um modelo de linguagem de grande porte (LLM) para segmentar e melhorar a legibilidade do texto das legendas. A implementação foi feita utilizando a biblioteca `instructor`, que permite a validação do texto gerado através dos modelos de tipagem do `pydantic`, garantindo conformidade com um formato estruturado.

Para evitar alterações indesejadas no conteúdo original, no modelo Pydantic utilizado, adicionou-se o decorador `@field_validator` para executar uma função de validação customizada. Esta função utilizava a normalização de texto para checar se apenas espaços e pontuações foram modificados entre o texto processado e o original.

Quando a validação falhava, a biblioteca `instructor` automaticamente incluia no próximo prompt o texto original, a saída que falhou e o erro específico da validação, facilitando que a LLM corrigisse seus erros. Este processo se repetia automaticamente por até 5 tentativas (definido por `max_retries=5`) para cada texto processado.

Entretanto, essa abordagem apresentou desafios significativos:

- **Latência elevada:** O processo todo acaba sendo muito demorado pois necessita de várias interações devido a limitação da janela de contexto. Tornando a solução pouco eficiente para processar legendas de vídeos de longa duração
- **Custo financeiro elevado:** O uso de LLMs para processamento de texto em larga escala mostrou-se financeiramente custoso, considerando o volume de requisições necessárias.
- **Alterações não desejadas no texto:** Apesar das instruções explícitas no prompt para evitar adições ou remoções de palavras, o modelo ocasionalmente incluía frases como "Aqui está o seu texto com legibilidade aprimorada" ou alterava partes do conteúdo original, omissão de algumas palavras ou frases por exemplo.
- **Falhas no processo:** Mesmo com o mecanismo de validação e o sistema de retentativas implementados, havia casos em que o processo esgotava o número máximo de iterações sem terminar sem erros.

Esses fatores tornaram a abordagem com LLMs inviável para o problema proposto.

5.2.3 Implementação com SaT (Segment Anything Text)

Recentes estudos sobre as limitações fundamentais dos LLMs, particularmente o trabalho “Transformers need glasses!” (Barbero; Banino; Kapturowski; Kumaran;

Araújo; Vitvitskyi; Pascanu; Veličković, 2024), revelaram dois fenômenos críticos que afetam o desempenho destes modelos: *representational collapse* e *over-squashing*.

O primeiro, *representational collapse*, ocorre quando sequências de entrada distintas geram representações praticamente idênticas no token final do modelo. Este fenômeno é especialmente problemático em sequências longas, onde a representação do último token tende a convergir mesmo quando há diferenças significativas no conteúdo. O problema é ainda mais acentuado pelo uso de formatos de ponto flutuante de baixa precisão comumente empregados em LLMs modernos.

Já o último, *over-squashing*, refere-se à perda progressiva de sensibilidade do modelo a tokens específicos na entrada. Este fenômeno é uma consequência direta da arquitetura dos transformers decoder-only, onde o fluxo unidirecional de informação converge no token final. Tokens que aparecem no início da sequência têm mais caminhos para propagar sua informação até o token final, enquanto tokens posteriores têm menos caminhos, resultando em perda de informação importante.

Diante das limitações arquiteturais dos LLMs, foi adotado o SAT (*Segment Any Text*) (Frohmann; Stern; Vulic; Minixhofer; Schedl, 2024), um transformer especializado para segmentação de texto que oferece uma solução mais robusta e eficiente para a melhoria da legibilidade das legendas. Diferente dos LLMs que focam em geração de texto, o SAT utiliza uma arquitetura de classificação binária que apenas identifica os limites entre sentenças, apresentando as seguintes vantagens:

- **Baixa latência:** A segmentação ocorre de maneira rápida e eficiente graças à sua arquitetura otimizada com tokenização por subpalavras com apenas 3 camadas transformer, resultando em um modelo muito menor e especializado para a tarefa.
- **Preservação do conteúdo original:** O modelo não gera ou modifica texto, apenas classifica as posições como limites de sentença ou não. Esta abordagem focada em classificação binária garante tanto a fidelidade ao texto original quanto maior confiabilidade e previsibilidade no processamento das legendas.

5.2.4 Interface e Experiência do Usuário

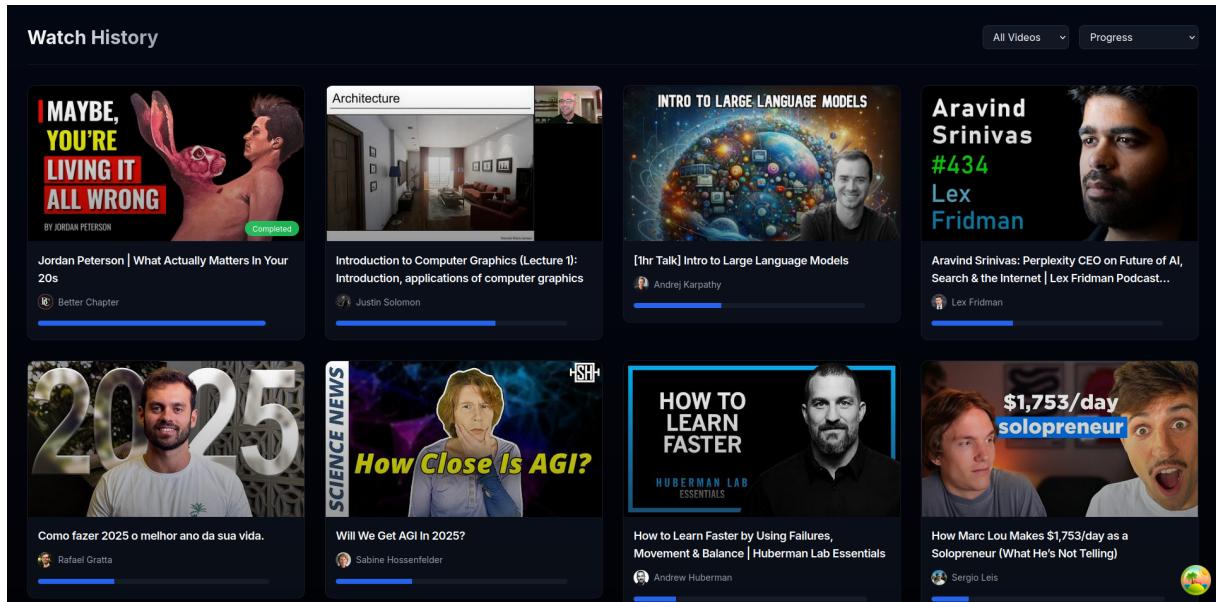


Figura 1 – Logs de execução no Beam Serverless GPU demonstrando o tempo de cold start (18s 573ms) na primeira requisição e tempos de resposta reduzidos (aproximadamente 50-100ms) nas requisições subsequentes

A segmentação de texto foi implementada como uma funcionalidade sob demanda, onde o usuário decide quando deseja realizar o processamento. Para cada vídeo, é realizada uma verificação automática para determinar se já existe uma versão segmentada da transcrição armazenada no banco de dados. Caso a versão segmentada já exista, ela é automaticamente carregada e o dropdown exibe "Read Text" como opção selecionada. Caso contrário, o dropdown mostra "Normal Subtitle" e o usuário pode acionar o processo de segmentação selecionando a opção "Read Text".

Durante o processamento, uma requisição para o servidor em beam serverless GPU é realizada e em paralelo um toast notification é exibido para manter o usuário informado sobre o progresso da operação. Ao término do processo, o sistema exibe uma notificação de sucesso ou, em caso de falha, uma mensagem de erro apropriada.

Após o recebimento do texto segmentado, ocorre então um processo de pós processamento, para que nesse texto segmentado, seja reeinserido as informações sobre o tempo que foram perdidas no processo anterior.

What fusion is and how it differs from fission

Well Jump Right In I think the thing we could do for our viewers and listeners that would be most useful to begin with is to tell them is for you to tell them what Fusion Energy is and how that differs from standard nuclear energy like just like a rationale for the pursuit of Fusion Energy and a place and placing of it in the proper context with regard to our pursuit of Advanced Energy and reliable energy supplies right so Fusion is the process of fusing together the most abundant and the lightest element hydrogen uh into heavier elements um so it actually changes the element and this is the process that powers the universe because it powers us All Stars including our own sun and you can think of a star our own Sun a big conversion Factory it's like a standard burner in this sense that it takes the huge masses of hydrogen that the sun is made out of um and in the center of it where the conditions are meet the requirement for Fusion it converts the hydrogen into helium uh and by that process uh releases staggering amounts of energy per reaction um so um you know usually when I comment in public about Fusion it's like so Fusion makes life possible in the universe because it's the it's the radiant heat that comes from stars that makes life possible in a place like the planet Earth um so it is the you think of it it's the quintessential or fundamental energy source of the universe that's that's the the starting point so it distinguishes why is it such an effective energy source it's because it changes the element right so what happens is that if you take the mass of the starting particles of this before you fuse them together they have larger mass than the particles that result from this and you go but how can this be because we all learned in school that you know Mass cannot be destroyed or created but this is what Einstein realized was that in fact mass and energy are the same thing and then when you convert them in these processes you end up with energy uh and it's it's hard to imagine how much of a different process this is than either fission or standard chemical reactions which is basically what we run the world on today in terms of in terms of comparing it to chemical energy the average energy released per reaction or per mass of particle is about 10 million times larger that's it's amazing right so this is this is why stars on our own Sun can last for 10 billion years I mean there's an enormous amount of hydrogen in the Sun but if it was running on a chemical process like burning hydrogen like you would think of in a fuel cell something like that it would only last for a few thousand years it lasts for 10 billion years that's the difference between them um and with respect to fission it's actually there's a relation there in the sense that it changes the elements as well too but it's literally the opposite process fission the name implies splits of parts or fissions the most unstable heaviest elements that exist like uranium uh and again by this

(a) Transcrição antes da segmentação

What fusion is and how it differs from fission

Well Jump Right

In I think the thing we could do for our viewers and listeners that would be most useful to begin with is to tell them is for you to tell them what Fusion Energy is and how that differs from standard nuclear energy like just like a rationale for the pursuit of Fusion Energy and a place and placing of it in the proper context with regard to our pursuit of Advanced Energy and reliable energy supplies right

So Fusion is the process of fusing together the most abundant and the lightest element hydrogen uh into heavier elements

Um so it actually changes the element and this is the process that powers the universe because it powers us All Stars including our own sun and you can think of a star our own Sun a big conversion Factory

It's like a standard burner in this sense that it takes the huge masses of hydrogen that the sun is made out of um and in the center of it where the conditions are meet the requirement for Fusion

It converts the hydrogen into helium uh and by that process uh releases staggering amounts of energy per reaction

Um I comment in public about Fusion it's like so Fusion makes life possible in the universe because it's the it's the radiant heat that comes from stars that makes life possible in a place like the planet Earth

Um so it is the you think of it

It's the quintessential or fundamental energy source of the universe

That's that's the the starting point so it distinguishes why is it such an effective energy source

It's because it changes the element right

So what happens is that if you take the mass of the starting particles of this before you fuse them together they have larger mass than the particles that result from this and you go

(b) Transcrição após segmentação com sat-3l-sm

Figura 2 – Comparação antes e após a segmentação de parágrafos.

5.2.5 Desafios de Implantação em Produção

A implantação de modelos de machine learning que requerem GPU apresenta desafios significativos, principalmente relacionados a custos e infraestrutura. Apesar do modelo utilizado ser relativamente pequeno, ocupando apenas alguns gigabytes, sua execução utilizando somente CPU é impraticável para uma boa experiência dos usuários, tornando o uso de GPU indispensável. No entanto, manter um servidor dedicado com GPU seria financeiramente inviável para este projeto, dado que servidores assim são bem custosos.

Como alternativa, optou-se por utilizar o Beam Serverless GPU, um serviço que permite o pagamento apenas pelo tempo efetivo de uso do recurso. Esta abordagem

oferece uma solução mais econômica, especialmente para cargas de trabalho intermitentes como é o caso do uso nessa aplicação.

No entanto, uma das principais limitações de arquiteturas serverless é o chamado *cold start* - o tempo necessário para inicializar os containers quando não há instâncias ativas. Para mitigar este problema, o Beam Serverless GPU oferece otimizações como a persistência do modelo em disco, eliminando a necessidade de download do modelo a cada inicialização do container.

Em testes realizados em ambiente de produção, observou-se que o primeiro processamento após um cold start levava aproximadamente 17 segundos para completar a inicialização e segmentação do texto. Contudo, as requisições subsequentes eram processadas quase instantaneamente, pois aproveitavam o container já inicializado.

5b764967-b9de-4c01-9daa-14b6ea62aa4 endpoint/deployment/script:segment_text	Complete	2/5/25 19:07	2/5/25 19:07	103ms	42ms
d4e3b209-398c-4ee6-97d1-e294775ffd4f endpoint/deployment/script:segment_text	Complete	2/5/25 19:07	2/5/25 19:07	51ms	50ms
1f470137-ef99-4567-aa01-b8a468a3c0b6 endpoint/deployment/script:segment_text	Complete	2/5/25 19:07	2/5/25 19:07	104ms	45ms
dd4d16ea-5d4-42b3-9e11-a491096cf568 endpoint/deployment/script:segment_text	Complete	2/5/25 19:07	2/5/25 19:07	61ms	48ms
b562e123-c634-4c20-a1f4-1afb5ef58d22 endpoint/deployment/script:segment_text	Complete	2/5/25 19:07	2/5/25 19:07	18s 573ms	350ms

Figura 3 – Logs de execução no Beam Serverless GPU demonstrando o tempo de cold start (18s 573ms) na primeira requisição e tempos de resposta reduzidos (aproximadamente 50-100ms) nas requisições subsequentes

ADICIONAR AQUI INFORMAÇÃO DE TEMPO—> o qto demora pra segmentar um texto de X horas

5.3 Geração de Capítulos

A geração automática de capítulos em vídeos do YouTube é um processo essencial para melhorar a naveabilidade e compreensão do conteúdo. A divisão em blocos menores de conteúdo é fundamental para o processo de aprendizagem, pois permite que o estudante absorva o material em partes gerenciáveis, facilitando a retenção e o entendimento.

Além disso, a estruturação em capítulos é crucial para diversas funcionalidades da aplicação:

- Permite que usuários selecionem capítulos específicos para discussão no bate-papo com vídeo
- Possibilita a geração de quizzes personalizados para cada segmento do conteúdo
- Facilita a navegação através da tabela de conteúdo
- Viabiliza o rastreamento preciso do progresso do usuário através do histórico de visualização

Para implementar esta funcionalidade, a implementação foi baseada na segmentação do texto em parágrafos e na identificação de mudanças de tópico através de um modelo de linguagem de grande escala (LLM).

5.3.1 Algoritmo e Implementação

O algoritmo segue um fluxo de processamento em três etapas principais:

1. **Estruturação do texto:** O conteúdo transscrito do vídeo é segmentado em parágrafos coerentes, respeitando pausas naturais e mudanças de contexto no discurso. Essa segmentação melhora a compreensão e a análise subsequente do texto.
2. **Numeração dos parágrafos:** Cada parágrafo recebe um identificador numérico no início, o que permite um referenciamento direto na etapa seguinte.
3. **Identificação de mudanças de tópico:** Utilizamos um LLM para analisar o texto e determinar os parágrafos que marcam a transição entre tópicos distintos. O modelo retorna a lista de números dos parágrafos que representam pontos de mudança relevantes no conteúdo.

5.3.2 Integração com LLMs

Para a identificação das transições de tópico, utilizamos function calling para interagir com o LLM. Essa abordagem permite enviar a transcrição segmentada e numerada, solicitando ao modelo que indique os pontos de mudança mais relevantes. A função chamada retorna uma lista de parágrafos que representam as mudanças de tópico, possibilitando a geração automática de capítulos estruturados.

Previamente, foi tentado um método em que a function calling solicitava apenas o nome dos tópicos de transição e seu tempo correspondente. No entanto, observou-se um aumento significativo nas alucinações do modelo. Ao utilizar a numeração dos parágrafos, conseguimos reduzir esse problema, pois conhecendo a posição do parágrafo na transcrição, podemos determinar com maior precisão a sua posição temporal no vídeo.

Essa abordagem garante uma segmentação eficiente e adaptável, permitindo uma melhor compreensão do conteúdo dos vídeos e melhorando a experiência do usuário ao consumir informações em formato de vídeo.

There are no chapters available!

Generate Chapters

The magnetic field of Earth has a hole. And

The hole is changing. It's not dangerous, at least NASA doesn't think it is, but it could affect satellites and other spacecraft. Is it an indication that Earth's magnetic field is about to flip? I've had a look. This weak spot of the magnetic field is called the South Atlantic Anomaly. As the name suggests, it's largely located over the South Atlantic Ocean,

But also extends over parts of South America, including the eastern coasts of Brazil and Argentina. It's not a new discovery but it was in the news

Recently because the hole is changing. This anomaly was first noticed in 1958, when Scientists from early American satellite missions—most notably those involving the Explorer series—observed that the Van Allen radiation belts dip unusually close to Earth's surface in this region. The Van Allen belts are two doughnut-shaped regions of energetic charged particles, primarily electrons and protons, that are trapped by Earth's magnetic field. In latitude, they are closer to the equator than the regions where the aurora comes down, and at a really high altitude, beginning at more than 600 kilometres. NASA has kept close track of this anomaly

Because if the Van Allen belts move down, then satellites and also the International Space Station are exposed to higher levels of radiation. This radiation can affect both the functioning of electronic devices as well as that of humans. The magnetic field

Figura 4 – Antes da geração de capítulos. Aviso de que vídeo não possui capítulos e botão para geração

The South Atlantic Anomaly

The magnetic field of Earth has a hole.

And the hole is changing, at least NASA doesn't think it is, but it could affect satellites and other spacecraft.

Is it an indication that Earth's magnetic field is about to flip?

I've had a look. This weak spot of the magnetic field is called the South Atlantic Anomaly.

As the name suggests, it's largely located over the South Atlantic Ocean, but also extends over parts of South America, including the eastern coasts of Brazil and Argentina.

It's not a new discovery but it was in the news recently because the hole is changing.

This anomaly was first noticed in 1958, when scientists from early American satellite missions—most notably those involving the Explorer series—observed that the Van Allen radiation belts dip unusually close to Earth's surface in this region.

The Van Allen belts are two doughnut-shaped regions of energetic charged particles, primarily electrons and protons, that are trapped by Earth's magnetic field.

In latitude, they are closer to the equator than the regions where the aurora comes down, and at a really high altitude, beginning at more than 600 kilometres.

NASA has kept close track of this anomaly because if the Van Allen belts move down, then satellites and also the International Space Station are exposed to higher levels of radiation.

This radiation can affect both the functioning of electronic devices as well as that of humans.

The magnetic field anomaly however doesn't affect aircraft, because these fly at much lower altitudes.

This hole in the magnetic field seems to be splitting apart into two distinct patches.

A few years ago, Scientists from NASA's Goddard Space Flight Centre used data from the European Space Agency's Swarm satellite missions and the CubeSat ELM-2 mission to track how the hole has been developing.

Figura 5 – Após geração de capítulos.

5.3.3 Tabela de Conteúdos

The screenshot shows a video player interface with a dark theme. At the top, there's a dropdown for 'English - Default' and some other controls. The main video area shows a portrait of Aravind Srinivas with the text '#434 Lex'. Below the video, there's a 'Video Chatbot' section with a message input field. To the right, a 'Video Progress' section shows a circular progress bar at 27% and a list of chapters and topics. Some items in the list have checkboxes next to them.

Figura 6 – Tabela de conteúdos gerada automaticamente a partir dos capítulos identificados

5.3.4 Últimos Vídeos Vistos

The screenshot shows a 'Watch History' page with a grid of video thumbnails. The top row includes videos by Jordan Peterson, Justin Solomon, Andrej Karpathy, and Lex Fridman. The second row includes videos by Rafael Gratta, Sabine Hossenfelder, Andrew Huberman, and Sergio Leis. Each thumbnail shows the video title, a small image, and some descriptive text below it.

Figura 7 – Histórico de visualização mostrando os últimos vídeos assistidos pelo usuário

5.4 Transcrição

A transcrição precisa de vídeos do YouTube é essencial para gerar textos mais confiáveis do que as legendas automáticas, que apresentam qualidade inferior especialmente para línguas diferentes do inglês.

5.4.1 Primeira implementação com Whisper

No fluxo original utilizando o Whisper, o processo consistia em:

1. **Download otimizado do áudio:** - Download apenas da stream de áudio via pytube-fix, sem baixar o vídeo - Seleção da segunda pior qualidade de áudio, equilibrando precisão e velocidade - Progresso do download transmitido em tempo real do backend para o frontend via SSE
2. **Pré-processamento específico para Whisper:** - Divisão do áudio em segmentos de 10 minutos - Cortes preferencialmente em momentos de silêncio - Envio paralelo dos segmentos para transcrição - Necessidade decorrente das limitações do Whisper com arquivos longos - Sem feedback de progresso durante a transcrição

5.4.2 Migração para o Deepgram

Com a adoção do Deepgram, o fluxo foi otimizado:

1. **Download otimizado do áudio mantido:** - Mesmo processo de download apenas do áudio via pytube-fix - Mantida a seleção da segunda pior qualidade para eficiência - Feedback em tempo real do progresso via SSE - Eliminação da etapa de divisão do áudio
2. **Transcrição direta:** - Envio do arquivo de áudio completo - Processamento único sem necessidade de paralelização - Estimativa de tempo restante baseada na métrica de 29.8 segundos por hora de áudio do modelo nova-2 - Capacidade nativa de lidar com longas durações

5.4.3 Vantagens adicionais do Deepgram

O Deepgram oferece recursos avançados ainda não implementados:

- Marcação temporal no nível da palavra (word-level timestamps)
- Diarização: identificação e segmentação de diferentes interlocutores
 - Possibilita diferenciação visual por cores no texto
 - Permite interações direcionadas no chat com falantes específicos
- Programa para startups com crédito inicial de US\$ 200

5.4.4 Interface e Experiência do Usuário

Usuário tem que clicar lá naquele botão de legenda, depois vai pra esse modal, e depois entao apertando no OK, aparece um widget que primeiro irá mostrar o progresso do download e depois irá mostrar uma esmitimativa de tempo pra transcrição.

Finalmente o transcript é adicionado em language selector a o texto é atualizado.

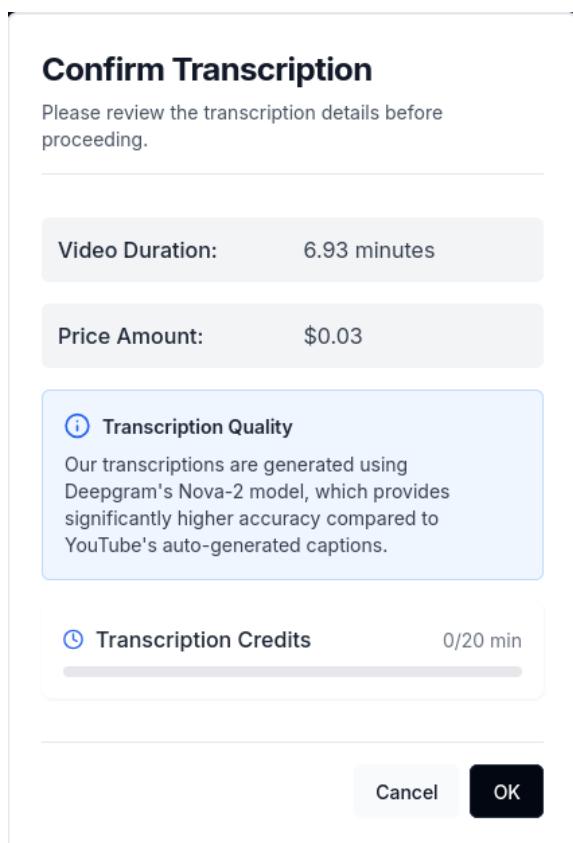


Figura 8

5.5 Geração de Quizzes

A geração de quizzes para cada capítulo é uma funcionalidade essencial, pois permite ao usuário verificar sua compreensão do conteúdo antes de avançar para a próxima seção. Essa abordagem auxilia na retenção do conhecimento, garantindo que os conceitos fundamentais tenham sido absorvidos de maneira eficaz.

Foram desenvolvidas duas interfaces gráficas distintas para os quizzes, adaptadas a diferentes tipos de perguntas:

- Interface para questões de *Verdadeiro ou Falso*.
- Interface para questões do tipo *Pergunta e Resposta (Q&A)*.

Em ambos os casos, a interação com o quiz é feita de maneira dinâmica, utilizando o Vercel AI SDK. A interface do usuário é atualizada em tempo real conforme a *Large*

Language Model (LLM) gera as respostas. Esse fluxo interativo melhora a experiência do usuário, proporcionando uma sensação de engajamento e imersão no processo de aprendizado.

Para viabilizar essa funcionalidade, utilizamos *function calling*, um mecanismo que permite especificar um *schema* que a LLM deve obedecer ao processar os pedidos. Esse conceito foi previamente abordado na fundamentação teórica, mas, de maneira resumida, ele garante que as respostas sigam um formato estruturado, permitindo maior controle sobre a saída gerada pela IA.

Além disso, para ambos os tipos de quiz, o usuário tem a opção de personalizar o comportamento da LLM por meio da edição do *prompt* específico. Essa personalização pode ser feita ao clicar no botão de três pontos dentro da interface do quiz, permitindo que o usuário defina diretrizes mais precisas para a geração das questões. Esse recurso adiciona flexibilidade e adaptação ao sistema, tornando-o mais versátil para diferentes necessidades e preferências de aprendizado.

5.5.1 Questões discursivas

Question	Answer	AI Answer
What is Fusion Energy?		✖ ✓ Fusion Energy is the process of fusing together hydrogen atoms to form heavier elements, releasing a significant amount of energy in the process. It powers stars, including our sun, and is considered the fundamental energy source of the universe.
How does Fusion Energy differ from standard nuclear energy?		✖ ✓ Fusion Energy differs from
Why is the pursuit of Fusion Energy important?		✖ ✓ The pursuit of Fusion Energy
What are the advantages of Fusion Energy over chemical energy?		✖ ✓ Fusion Energy releases about
What is the significance of Einstein's theory in understanding Fusion Energy?		✖ ✓ Einstein's theory highlights
+		

Figura 9 – Exemplo de interface para perguntas e respostas.

A interface de perguntas e respostas (Q&A) apresenta uma tabela com três colunas: a primeira exibe a pergunta gerada, a segunda, chamada “Resposta”, permite ao usuário inserir sua resposta manualmente, e a terceira, “AI Answer”, que é inicialmente

oculta por um efeito de *blur*. A visibilidade dessa coluna pode ser alternada através de um botão, permitindo que o usuário visualize ou oculte a resposta gerada pela IA. O objetivo desse design é incentivar o usuário a tentar responder antes de consultar a resposta fornecida pela IA, promovendo um aprendizado mais ativo e eficaz.

5.5.2 Questões de Verdadeiro ou Falso

The screenshot shows a dark-themed user interface for a True or False quiz. At the top left, there is a battery icon with the text "5 credits remaining". Below it, a section titled "True or False" contains the following questions:

- Fusion energy is the process of fusing hydrogen into heavier elements, which powers the universe.**
 - True**
 - False**

Correct
- Fusion energy is the same as fission energy.**
 - True**
 - False**

Correct
- The energy released from fusion reactions is significantly greater than that from chemical reactions.**
 - True**
 - False**

Incorrect. The energy released per reaction in fusion is about 10 million times larger than that from standard chemical reactions.
- The sun can last for billions of years because it runs on chemical processes.**
 - True**
 - False**

Incorrect. The sun lasts for billions of years because it runs on fusion processes, not chemical processes.
- Harnessing fusion energy on Earth would have consequences similar to those of fossil fuel energy.**
 - True**
 - False**

Incorrect. Harnessing fusion energy would have very different consequences compared to fossil fuel energy, as it is a fundamentally different energy source.

Figura 10 – Exemplo de interface para questões de verdadeiro ou falso.

Na interface de Verdadeiro ou Falso, as perguntas são apresentadas como afirmações textuais, e abaixo de cada uma há dois botões que permitem ao usuário marcar a resposta como verdadeira ou falsa. Após a seleção um texto adicional surge abaixo, indicando o resultado da resposta: caso esteja correta, a confirmação aparece em verde; caso esteja errada, um texto em vermelho explica o motivo do erro. Esse feedback imediato auxilia na compreensão dos conceitos e melhora a fixação do aprendizado.

5.6 Prompt por capítulo

5.7 Bate-Papo com Vídeo

5.7.1 Visão Geral

O componente de bate-papo com vídeo foi desenvolvido como uma ferramenta interativa que permite aos usuários fazer perguntas e obter respostas contextualizadas sobre o conteúdo de vídeos. O sistema fundamenta-se na análise de legendas e capítulos para fornecer respostas precisas e relevantes, mantendo um gerenciamento eficiente de tokens para otimizar a interação com o modelo de linguagem.

5.7.2 Interface do Usuário

A interface foi projetada com foco na usabilidade e eficiência. No centro da tela, encontra-se o player de vídeo integrado, acompanhado por um painel expansível para seleção de contexto. Abaixo, localiza-se o campo de chat para interação e o histórico da conversa. Um elemento crucial da interface é o botão de exclusão de mensagens, que permite ao usuário limpar todo o histórico de conversa e o contexto selecionado, possibilitando um novo início de interação.

O sistema de seleção de contexto foi implementado de forma flexível, oferecendo duas modalidades principais: seleção através de checkboxes individuais e entrada textual com sintaxe simplificada (por exemplo, "1-3,5"para selecionar os capítulos 1, 2, 3 e 5).

5.7.3 Gerenciamento de Contexto e Tokens

O sistema opera sob parâmetros específicos de controle: um limite de 12.000 tokens para mensagens do sistema e máximo de 120.000 tokens para a conversa completa. Para manter a eficiência, o sistema preserva apenas os cinco pares mais recentes de mensagens ativas.

Para vídeos extensos sem capítulos, um sistema de proteção automática entra em ação. O usuário é notificado através de um aviso sobre o truncamento da legenda, recebendo sugestões para a geração de capítulos. Quando o contexto selecionado excede os limites estabelecidos, o sistema realiza ajustes automáticos, comunicando as alterações através de notificações toast.

5.7.4 Implementação Técnica

5.7.4.1 Streaming e Processamento

O núcleo da implementação utiliza o Vercel AI SDK, responsável pelo gerenciamento de streams de dados em tempo real, processamento de respostas incrementais e manipulação do estado da conexão. Esta escolha tecnológica permite uma experiência

fluída e responsiva durante a interação com o vídeo.

5.7.4.2 Perspectivas de Evolução

O sistema foi arquitetado prevendo futuras expansões, com especial atenção para a implementação de um sistema agêntico de seleção de contexto, integração com RAG (Retrieval-Augmented Generation), processamento de diarização e análise semântica avançada.

5.7.5 Benefícios e Características

A implementação atual se destaca pela precisão nas respostas, alcançada através de uma contextualização eficiente. O sistema mantém a coerência conversacional mesmo em diálogos extensos, sem comprometer a qualidade das interações. A interface responsiva e adaptável, combinada com o gerenciamento inteligente de recursos, proporciona uma experiência robusta para análise e discussão de conteúdo em vídeo.

O equilíbrio entre funcionalidade, eficiência e experiência do usuário resulta em uma ferramenta versátil e eficaz para a exploração de conteúdo audiovisual através de conversação natural. A persistência do contexto inicial, combinada com o gerenciamento dinâmico das mensagens, garante respostas consistentes e relevantes ao longo de toda a interação.

5.8 Desafios e Soluções

5.8.1 Obtenção dos dados do youtube

5.8.2 Utilização de GPU's em produção

6 CONCLUSÃO

O desenvolvimento do VideoLearnAI demonstrou o potencial significativo da integração entre modelos de linguagem natural e tecnologias web modernas para criar experiências educacionais mais efetivas. A plataforma conseguiu transformar o consumo passivo de vídeos em um processo de aprendizagem ativa, oferecendo ferramentas que promovem maior engajamento e compreensão do conteúdo.

6.1 Objetivos Alcançados

Os objetivos inicialmente propostos foram alcançados através da implementação bem-sucedida das cinco funcionalidades principais:

1. **Melhoria da Legibilidade das Legendas:** A implementação do modelo SAT (Segment Any Text) proporcionou uma solução eficiente e precisa para a segmentação de texto, superando as limitações encontradas na abordagem inicial com LLMs. Esta funcionalidade demonstrou ser fundamental para melhorar a experiência de leitura e compreensão do conteúdo.
2. **Geração de Capítulos:** O sistema de geração automática de capítulos, combinando segmentação de texto com análise por LLMs, mostrou-se eficaz na organização estruturada do conteúdo, facilitando a navegação e o acesso a informações específicas.
3. **Legendas de maior qualidade:** A geração de legendas com o serviço Deepgram ao invés das legendas geradas automaticamente pelo youtube, resultou em um processo mais eficiente e preciso de transcrição, com benefícios adicionais como marcação temporal no nível da palavra e capacidade de diarização dos locutores.
4. **Quizzes Interativos:** Superando as limitações do consumo passivo de vídeo, os questionários interativos com feedback em tempo real promovem engajamento ativo através de exercícios práticos e reflexivos, maximizando a absorção do conteúdo.

5. **Bate-Papo Contextual:** O sistema de chat contextualizado proporcionou uma forma natural e eficiente de interação com o conteúdo do vídeo, com gerenciamento adequado de contexto, transparente para o usuário.

6.2 Trabalhos Futuros

O desenvolvimento do VideoLearnAI abriu diversas possibilidades para aprimoramentos e expansões futuras:

6.2.1 Diarização e Análise de Múltiplos Falantes

- Aprimorar o texto da transcrição utilizando a diarização do Deepgram através de elementos visuais distintivos (legendas coloridas, avatares, badges) para cada interlocutor
- Integrar a identificação de falantes às demais funcionalidades, enriquecendo o contexto no chat, quizzes.

6.2.2 Otimizações Técnicas

- Implementação de RAG (Retrieval-Augmented Generation) para:
 - Melhorar a precisão das respostas no chat
 - Possibilitar análises avançadas de debates, incluindo fact-checking, qualidade argumentativa e identificação de falácias
 - Integração com diarização para análise específica por interlocutor
- Redução adicional dos tempos de cold start em funções serverless
- Aprimoramento do sistema de cache para conteúdos frequentemente acessados

6.2.3 Expansão de Funcionalidades

- Sistema de repetição espaçada utilizando os quizzes gerados
- Elementos de gamificação:
 - Sistema de ranking por vídeo
 - Métricas de engajamento e aprendizado
 - Recompensas por participação ativa
- Suporte a mais formatos de conteúdo além do YouTube
- Ferramentas colaborativas para estudo em grupo
- Sistema de exportação de notas e resumos
- Integração com plataformas de ensino existentes

6.2.4 Melhorias na Experiência do Usuário

- Interface adaptativa para diferentes dispositivos e contextos de uso
- Mais opções de personalização visual
- Suporte expandido para múltiplos idiomas

A aplicação desenvolvida nesse trabalho demonstrou a viabilidade de combinar tecnologias avançadas de IA com princípios de aprendizagem ativa. Embora em fase inicial, a aplicação apresenta potencial significativo para transformar como as pessoas aprendem através de conteúdo audiovisual educativo.

REFERÊNCIAS

- BARBERO, F.; BANINO, A.; KAPTUROWSKI, S.; KUMARAN, D.; ARAÚJO, J. G.; VITVITSKYI, A.; PASCANU, R.; VELIČKOVIĆ, P. Transformers need glasses! Information over-squashing in language tasks. **arXiv preprint arXiv:2406.04267**, [S.I.], 2024.
- BELTAGY, I.; PETERS, M. E.; COHAN, A. Longformer: The long-document transformer. In: XIV PREPRINT ARXIV:2004.05150, 2020. **Anais...** [S.I.: s.n.], 2020.
- BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J. D.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A. et al. Language models are few-shot learners. **Advances in neural information processing systems**, [S.I.], v.33, p.1877–1901, 2020.
- DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES, 2019., 2019. **Proceedings...** [S.I.: s.n.], 2019. p.4171–4186.
- FROHMANN, M.; STERNER, I.; VULIC, I.; MINIXHOFER, B.; SCHEDL, M. Segment Any Text: A Universal Approach for Robust, Efficient and Adaptable Sentence Segmentation. **arXiv preprint arXiv:2406.16678**, [S.I.], 2024.
- GAO, Y.; XIONG, Y.; GAO, X.; JIANG, K.; SHEN, J.; REN, X.; HAN, J. Retrieval-augmented generation for large language models: A survey. **arXiv preprint arXiv:2312.10997**, [S.I.], 2023.
- GUHR, O.; SCHUMANN, A.-K.; BAHRMANN, F.; BÖHME, H. J. FullStop: Multilingual Deep Models for Punctuation Prediction. In: SWISS TEXT ANALYTICS CONFERENCE 2021, 2021, Winterthur, Switzerland. **Proceedings...** CEUR Workshop Proceedings, 2021.

KAPLAN, J.; MCCANDLISH, S.; HENIGHAN, T.; BROWN, T. B.; CHESS, B.; CHILD, R.; GRAY, S.; RADFORD, A.; WU, J.; AMODEI, D. Scaling laws for neural language models. **arXiv preprint arXiv:2001.08361**, [S.I.], 2020.

KASNECI, E.; SESSLER, K.; KÜCHEMANN, S.; BANNERT, M.; DEMENTIEVA, D.; FISCHER, F.; GASSER, U.; GROH, G.; GÜNNEMANN, S.; HÜLLERMEIER, E. et al. ChatGPT for good? On opportunities and challenges of large language models for education. **Learning and Individual Differences**, [S.I.], v.103, p.102274, 2023.

KOJIMA, T.; GU, S. S.; REID, M.; MATSUO, Y.; IWASAWA, Y. Large language models are zero-shot reasoners. **Advances in Neural Information Processing Systems**, [S.I.], v.35, p.22199–22213, 2022.

LEWIS, P.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; KÜTTLER, H.; LEWIS, M.; YIH, W.-t.; ROCKTÄSCHEL, T. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. **Advances in Neural Information Processing Systems**, [S.I.], v.33, p.9459–9474, 2020.

LIU, N. F.; BOSSELUT, A.; SRINIVASAN, D.; CHOI, Y.; HAJISHIRZI, H.; KHASHABI, D. Lost in the middle: How language models use long contexts. **arXiv preprint arXiv:2307.03172**, [S.I.], 2023.

LIU, P.; YUAN, W.; FU, J.; JIANG, Z.; HAYASHI, H.; NEUBIG, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Computing Surveys**, [S.I.], v.55, n.9, p.1–35, 2023.

LIU, X.; ZHANG, F.; HOU, Z.; MIAN, L.; WANG, Z.; ZHANG, J.; TANG, J. Self-supervised learning: Generative or contrastive. **IEEE Transactions on Knowledge and Data Engineering**, [S.I.], v.35, n.1, p.677–694, 2021.

LU, Y.; BARTOLO, M.; MOORE, A.; RIEDEL, S.; STENETORP, P. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. **arXiv preprint arXiv:2104.08786**, [S.I.], 2022.

OpenAI. GPT-4 Technical Report. **arXiv preprint arXiv:2303.08774**, [S.I.], 2023.

OpenAI. **Pricing**. Accessed: 2023-12-01, <https://openai.com/pricing>.

RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I. Language models are unsupervised multitask learners. **OpenAI blog**, [S.I.], v.1, n.8, p.9, 2019.

RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M.; ZHOU, Y.; LI, W.; LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, [S.I.], v.21, p.1–67, 2020.

- SENNRICH, R.; HADDOW, B.; BIRCH, A. Neural machine translation of rare words with subword units. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 54., 2016. **Proceedings...** [S.I.: s.n.], 2016. p.1715–1725.
- VASWANI, A.; SHAZER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2017. **Anais...** [S.I.: s.n.], 2017. p.5998–6008.
- WANG, X.; WEI, J.; SCHUURMANS, D.; LE, Q.; CHI, E.; ZHOU, D. Self-consistency improves chain of thought reasoning in language models. **arXiv preprint arXiv:2203.11171**, [S.I.], 2023.
- WEI, J.; WANG, X.; SCHUURMANS, D.; BOSMA, M.; ICHTER, B.; XIA, F.; CHI, E.; LE, Q.; ZHOU, D. Chain of thought prompting elicits reasoning in large language models. **Advances in Neural Information Processing Systems**, [S.I.], v.35, p.24824–24837, 2022.
- WHITE, J.; FU, Q.; HAYS, S.; SANDBORN, M.; OLEA, C.; GILBERT, H.; ELNASHAR, A.; SPENCER-SMITH, J.; SCHMIDT, D. C. Prompt engineering for large language models: A survey. **arXiv preprint arXiv:2307.10169**, [S.I.], 2023.
- XU, Y.; SARTHI, S.; AGARWAL, A.; GUPTA, A.; SAXENA, A.; ARALIKATTE, R.; BATRA, D.; PARIKH, D.; MISRA, I.; AWADALLAH, A. Retrieval-augmented generation for knowledge-intensive nlp tasks. **arXiv preprint arXiv:2305.14002**, [S.I.], 2023.
- YAO, S.; ZHAO, J.; YU, D.; DU, N.; SHAFRAN, I.; NARASIMHAN, K.; CAO, Y. ReAct: Synergizing reasoning and acting in language models. **arXiv preprint arXiv:2210.03629**, [S.I.], 2023.

Apêndices

APÊNDICE A – Um Apêndice

Anexos

ANEXO A – Um Anexo