Extração de Tópicos de Transcrições de Vídeos do YouTube sobre Desenvolvimento de Software com LLMs

Ian de Barros Seki

Universidade Federal de Uberlândia
Faculdade de Computação
Bacharelado em Sistemas de Informação

Monte Carmelo - MG
2026

Ian de Barros Seki

Extração de Tópicos de Transcrições de Vídeos do YouTube sobre Desenvolvimento de Software com LLMs

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Prof. Dr. Adriano Mendonça Rocha

Monte Carmelo - MG
2026

Para minha família, amigos e mentores, pelo seu apoio inabalável.

Agradecimentos

Sou profundamente grato ao meu orientador, **Prof. Dr. Adriano Mendonça Rocha**, por sua orientação, incentivo e *feedback* rigoroso ao longo deste trabalho. Agradeço também ao **Prof. Dr. Marcelo de Almeida Maia** e ao **Prof. Dr. Carlos Eduardo Carvalho Dantas** pelas valiosas discussões, sugestões e apoio oferecidos em diferentes etapas do projeto.

Meus sinceros agradecimentos ao **PET-SIMC** (Programa de Educação Tutorial Sistemas de Informação de Monte Carmelo) pela oportunidade de aprofundar meus conhecimentos, fortalecer minhas habilidades de pesquisa e integrar uma comunidade acadêmica colaborativa.

Sou grato à minha família e aos meus amigos pelo apoio constante e pela paciência, bem como aos colegas da **FACOM/UFU**, que contribuíram com comentários e revisões úteis em versões anteriores deste trabalho. Agradeço também à **REP Reinicia**, república onde morei, pelo convívio, apoio e companheirismo ao longo dessa trajetória.

"Eu proponho considerarmos a seguinte questão: As máquinas podem pensar?"
-Alan Turing.

Resumo

Este trabalho investiga o uso de *Large Language Models* (LLMs) para a extração de tópicos a partir de transcrições educacionais de vídeos do YouTube na área de desenvolvimento de *software*. Foi projetado um *pipeline* completo composto por seleção de vídeos, coleta de transcrições e geração automática de tópicos com três modelos distintos de LLMs, variando a quantidade de tópicos solicitados (5/7/10) e considerando dois idiomas (português/inglês). A avaliação compara os tópicos gerados pelos modelos com uma lista de referência humana por meio de categorias de correspondência (Muito Alta, Alta, Baixa, Muito Baixa), de modo a capturar alinhamento e granularidade. Também foi incluída uma base de comparação com *Latent Dirichlet Allocation* (LDA).

Os resultados indicam que, no escopo analisado, os LLMs apresentam desempenho superior ao LDA em termos de clareza, coesão e alinhamento semântico, e que solicitar 10 tópicos tende a aumentar as correspondências Muito Alta/Alta ao reduzir omissões. Em português, o GPT-4 apresentou os resultados mais consistentes; em inglês, Gemini e GPT-4 também apresentaram resultados competitivos na configuração com 10 tópicos. Também são discutidas ameaças à validade (viés de avaliador humano único, ruído nas transcrições e sensibilidade ao *prompt*) e apresentados trabalhos futuros, incluindo ajuste fino específico de domínio, análise de tópicos progressiva e expansão nos modelos avaliados.

Palavras-chave: Extração de Tópicos, Transcrições de Vídeos do YouTube, Educação em Desenvolvimento de Software, *Latent Dirichlet Allocation* (LDA), *Large Language Models* (LLMs).

Lista de ilustrações

Lista de tabelas

Lista de siglas

LDA *Latent Dirichlet Allocation*

LLM *Large Language Model*

LDADE *Latent Dirichlet Allocation with Differential Evolution*

NLP *Natural Language Processing*

GPT *Generative Pre-Trained Transformer*

Sumário

Capítulo 1

Introdução

Este capítulo apresenta a motivação e o contexto desta pesquisa, com foco na crescente quantidade de conteúdo educacional sobre desenvolvimento de *software* disponível em plataformas como o *YouTube*. Nesse cenário, muitos desenvolvedores enfrentam dificuldades para localizar rapidamente informações relevantes em vídeos longos que fornecem poucos detalhes sobre seu conteúdo. Este capítulo também destaca a relevância do uso de *Large Language Models* (LLMs) para sintetizar conteúdo e facilitar a navegação por tópicos específicos, apoiando, assim, o aprendizado *online*. Por fim, apresenta a hipótese, os objetivos, as contribuições e a organização desta monografia.

1.1 Motivação

O crescimento exponencial de vídeos educacionais sobre engenharia de *software* e similires, intensificou a necessidade de formas práticas de acessar informações específicas presentes em vídeos longos e, muitas vezes, pouco descritivos. Desenvolvedores recorrem rotineiramente ao *YouTube* para aprender, mas muitos vídeos não apresentam resumos precisos de seu conteúdo, o que resulta em perda de tempo e redução de produtividade. Essas dificuldades demonstram desafios já conhecidos no acesso à informação na web (LAWRENCE; GILES, 1999) e o impacto documentado da pressão de tempo no trabalho em engenharia de *software* (KUUTILA et al., 2020).

Como o conteúdo sobre desenvolvimento de *software* no *YouTube* é produzido em múltiplos idiomas, este estudo considera vídeos em português e em inglês. Para abranger esses contextos linguísticos, configuramos as definições regionais da plataforma para Brasil (vídeos em português) e Estados Unidos (vídeos em inglês) e selecionamos, por relevância, os primeiros resultados retornados para cada consulta técnica. Restringimos o conjunto de dados a vídeos entre 10 e 30 minutos para equilibrar riqueza de conteúdo e viabilidade da avaliação humana, seguindo um roteiro de consultas fundamentado em outro trabalho (ROCHA; MAIA, 2023). Nesse contexto, investigamos se *Large Language Model*s (LLMs) modernos podem gerar listas de tópicos fiéis que ajudem estudantes e interessados a decidir

rapidamente se um vídeo aborda o que eles precisam.

1.2 Problema

O problema central abordado nesta pesquisa é a extração *automática* de tópicos relevantes de transcrições de vídeos sobre desenvolvimento de *software* de uma forma que esteja alinhada ao entendimento humano. A maioria dos vídeos não inclui descrições suficientemente detalhadas; consequentemente, estudantes não conseguem determinar rapidamente se um determinado tópico é abordado durante o vídeo. Portanto, nos questionamos: como podemos automatizar a identificação e a sumarização dos principais tópicos a partir de transcrições de vídeos, de modo a reduzir o tempo de busca e melhorar a efetividade da aprendizagem?

1.3 Hipótese

Nossa hipótese de trabalho é que LLMs (por exemplo, *Generative Pre-Trained Transformer (GPT)* e *Gemini*) podem extrair tópicos de transcrições de vídeos educacionais com alinhamento ao entendimento humano de forma superior ao de uma base de comparação tradicional com *Latent Dirichlet Allocation* (LDA) ajustado por Evolução Diferencial (LDADE) (AGRAWAL; FU; MENZIES, 2018). Especificamente, hipotetizamos que LLMs superam essa linha de base em diferentes idiomas (português e inglês) e em diferentes configurações de quantidade de tópicos.

1.4 Objetivos

O objetivo principal deste trabalho é avaliar a efetividade de LLMs na extração de tópicos de transcrições de vídeos sobre desenvolvimento de *software*.

- ❑ **Objetivo geral:** Investigar a capacidade de LLMs de identificar e extrair tópicos relevantes que estejam alinhados ao julgamento humano no contexto de vídeos sobre desenvolvimento de *software*.
- ❑ **Objetivos específicos:**
  - Comparar os tópicos gerados por LLMs com tópicos identificados manualmente por um avaliador humano utilizando um procedimento baseado em correspondência.
  - Avaliar o alinhamento e a cobertura dos tópicos em três configurações (5, 7 e 10 tópicos) e em dois idiomas (português e inglês).

- Analisar o desempenho de diferentes LLMs (*GPT-3.5*, *GPT-4* e *Gemini*) na extração de tópicos.
- Comparar LLMs com uma análise complementar de uma base de compração tradicional baseada em LDA com ajuste por Evolução Diferencial (LDADE) (AGRAWAL; FU; MENZIES, 2018).

Resumo da avaliação.

Neste estudo, não empregamos métricas clássicas de classificação (por exemplo, precisão, *recall* e *F-measure*). Em vez disso, adotamos uma avaliação baseada em correspondência para as saídas de LLM, enquanto LDA/LDADE é utilizado como uma análise complementar. Para evitar redundância metodológica, o protocolo operacional completo é apresentado no Capítulo 4.

1.5 Contribuições

Este trabalho oferece as seguintes contribuições:

- ❑ Uma metodologia reprodutível para avaliar a extração de tópicos de transcrições do *YouTube* utilizando LLMs, incluindo análise multilíngue (português e inglês) e múltiplas configurações de quantidade de tópicos.
- ❑ Uma comparação sistemática entre *GPT-3.5*, *GPT-4* e *Gemini*, com LDA/LDADE utilizado como base de comparação tradicional complementar (AGRAWAL; FU; MENZIES, 2018).
- ❑ Implementação de uma *extensão de navegador* que automatiza a extração de transcrições do *YouTube*, processa o conteúdo com LLMs e apresenta resumos de tópicos ao usuário.
- ❑ Disponibilização pública dos artefatos e *scripts* do estudo (referenciados no capítulo 4 e no apêndice), promovendo transparência e reutilização.

1.6 Organização da Monografia

Esta monografia está organizada da seguinte forma:

- ❑ **Capítulo 1 Introdução:** apresenta a motivação, o problema de pesquisa, a hipótese, os objetivos e as contribuições.

- **Capítulo 2 Fundamentação Teórica:** revisa conceitos-chave sobre LLMs, *Natural Language Processing* (NLP), extração de tópicos, a abordagem de avaliação baseada em correspondência, a arquitetura *Transformer* e métodos tradicionais, incluindo LDA e a linha de base LDADE.
- **Capítulo 3 Implementação:** descreve a arquitetura e a implementação do protótipo, incluindo os serviços de *backend* e a extensão de navegador.
- **Capítulo 4 Experimentos e Análise dos Resultados:** concentra o protocolo experimental completo (construção do conjunto de dados, *prompting*, configuração multilíngue, procedimentos de avaliação e análises comparativas) tanto para o estudo principal baseado em LLM quanto para a base comparativa LDADE.
- **Capítulo 5 Conclusão:** resume os achados em relação aos objetivos e à hipótese, apresenta limitações e ameaças à validade e propõe direções para trabalhos futuros.

Achados antecipados.

Evidências preliminares sugerem que a configuração com 10 tópicos tende a proporcionar melhor alinhamento com o julgamento humano e que o *GPT-4* alcança, em média, os resultados mais favoráveis entre os modelos avaliados. Esses achados são fundamentados e qualificados nos capítulos subsequentes.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os principais conceitos teóricos que fundamentam esta pesquisa. Inicialmente, introduz os *Large Language Models* (LLMs), enfatizando seu papel em tarefas de processamento de linguagem natural, como sumarização e extração de tópicos a partir de transcrições de vídeos educacionais. Em seguida, discute conceitos centrais de NLP, abordagens de extração de tópicos e a lógica de avaliação baseada em correspondência adotada neste estudo. Além disso, o capítulo resume a arquitetura *Transformer*, estratégias de *pre-training* e *fine-tuning*, processamento multilíngue e o uso de LDA/LDADE como métodos de base comparativa. Por fim, revisa trabalhos relacionados que dão suporte ao problema de pesquisa abordado nesta monografia.

2.1 *Large Language Models* (LLMs)

LLMs são modelos treinados com grandes volumes de dados para realizar tarefas complexas de NLP, incluindo geração de texto, sumarização, tradução, classificação e extração de informações (BROWN et al., 2020; ABDULLAH; MADAIN; JARARWEH, 2022; BOU et al., 2023). Avanços recentes expandiram essa família para incluir modelos fechados e abertos com diferentes escalas, estratégias de treinamento e perfis de aplicação, como sistemas baseados em GPT, famílias do tipo PaLM/Gemini e alternativas abertas, como o LLaMA (CHOWDHERY et al., 2024; TOUVRON et al., 2023). Nesse sentido, LLMs podem ser caracterizados ao longo de múltiplas dimensões, incluindo escala do modelo, adaptabilidade por meio de *fine-tuning* ou instrução orientada e modalidade de entrada. Modelos maiores frequentemente apresentam capacidades emergentes mais fortes, mas também demandam mais recursos computacionais; da mesma forma, variantes multimodais ampliam a gama de aplicações possíveis ao processar não apenas texto, mas também imagens e outros tipos de entrada (BROWN et al., 2020; CHOWDHERY et al., 2024).

Em termos práticos, LLMs têm mostrado potencial em tarefas que exigem abstração semântica e interpretação contextual, o que os torna relevantes para a extração de tópicos a

partir de transcrições de vídeos. Em contextos de engenharia de *software*, eles também têm sido aplicados a atividades como apoio à documentação, rotulação de *issues*, assistência relacionada a código e avaliação de qualidade (TUFANO et al., 2024b; TUFANO et al., 2024a; COLAVITO et al., 2024; EBERT; LOURIDAS, 2023; DANTAS; ROCHA; MAIA, 2023). Ao mesmo tempo, sua adoção envolve limitações importantes. Esses modelos são computacionalmente caros, podem reproduzir vieses presentes nos dados de treinamento, são sensíveis à modelagem do *prompt* e podem gerar conteúdo impreciso ou não relevante, comumente descrito como alucinação (ZUCCON; KOOPMAN; SHAIK, 2023; ABDULLAH; MADAIN; JARARWEH, 2022). Portanto, suas saídas devem ser interpretadas com cautela e, em contextos analíticos, idealmente complementadas por validação humana.

2.1.1 Modelo de *Prompt* e Granularidade dos Tópicos

O Modelo do *prompt* afeta diretamente a especificidade e a granularidade das saídas da LLM. Em cenários de extração de tópicos, solicitar ao modelo um número fixo de tópicos curtos e bem formados ajuda a reduzir variações excessivas de escopo e formulação, melhorando a comparabilidade entre vídeos e entre modelos (WANG et al., 2023). Neste trabalho, as definições do *prompt* quanto ao número de tópicos são especialmente relevantes porque dão suporte à comparação baseada em correspondência adotada posteriormente na avaliação. Ao limitar a saída a quantidades predefinidas de tópicos, o estudo reduz a perca de semântica e respostas excessivamente verbosas, ao mesmo tempo em que ainda permite aos modelos capturar os temas centrais de cada transcrição.

2.2 Processamento de Linguagem Natural (NLP)

NLP é a área da computação voltada a permitir que sistemas processem, representem e gerem respostas próxima à linguagem humana (BROWN et al., 2020; DEVLIN et al., 2018). Ela abrange tarefas como análise de sentimentos, tradução automática, sumarização, classificação e identificação de tópicos, todas dependentes da transformação da entrada textual em representações que possam ser analisadas computacionalmente. No contexto deste trabalho, NLP fornece a base conceitual para converter transcrições longas de vídeos em representações semânticas compactas que destacam os principais assuntos discutidos em cada vídeo.

Embora sistemas recentes de NLP tenham se tornado consideravelmente mais expressivos e semelhantes com a linguagem humana, desafios importantes permanecem. Vocabulário específico de um determinado domínio, transcrições ruidosas, variação entre idiomas e expressões dependentes de contexto cultural podem afetar a qualidade da informação extraída, especialmente quando vídeos educacionais misturam jargão técnico com linguagem informal. Por essa razão, a extração de tópicos baseada em transcrições

se beneficia de abordagens que combinem flexibilidade semântica com especificação cuidadosa de *prompts* e procedimentos de avaliação (PIRES; SCHWENK; CORREIA, 2019; CONNEAU et al., 2020).

2.3 Extração de Tópicos

A extração de tópicos tem como objetivo identificar os temas centrais de um texto para que grandes quantidades de conteúdo possam ser organizadas, indexadas e navegadas de forma mais eficiente (WANG et al., 2023; BLEI, 2012). Métodos tradicionais, como LDA, inferem temas latentes a partir de padrões de recorrência de palavras, normalmente representando tópicos como distribuições sobre termos (BLEI; NG; JORDAN, 2003; BLEI; NG; JORDAN, 2001). Essas abordagens são úteis como base comparativa e permanecem relevantes em análise exploratória de texto, mas frequentemente dependem de pressupostos de *bag-of-words*, são sensíveis a escolhas de hiperparâmetros e podem gerar tópicos difíceis de interpretar sob uma perspectiva humana (BLEI, 2012; SRIVASTAVA; SUTTON, 2017; AGRAWAL; FU; MENZIES, 2018).

Em contraste, LLMs podem produzir tópicos em linguagem natural, utilizando informações contextuais e semânticas que vão além da simples recorrência lexical. Trabalhos recentes sugerem que modelagem de tópicos baseada em *prompts* com LLMs pode alcançar maior interpretabilidade e melhor alinhamento com julgamentos humanos em determinados cenários (WANG et al., 2023; SIA; DALMIA; MIELKE, 2020). Ainda assim, a extração baseada em LLM não é isenta de limitações: requer mais recursos computacionais, depende de decisões de engenharia de *prompt* e pode ser menos estável quando a transcrição é muito ruidosa ou quando o domínio é extremamente especializado (ZUCCON; KOOPMAN; SHAIK, 2023; ABDULLAH; MADAIN; JARARWEH, 2022).

2.4 Abordagem de Avaliação

Em vez de utilizar métricas tradicionais de classificação, como precisão, *recall* e *F-measure*, este trabalho adota uma estratégia de avaliação baseada em correspondência para comparar tópicos gerados automaticamente com tópicos produzidos por humanos. Essa escolha reflete o fato de que a extração de tópicos não é um problema comum de classificação de rótulos: diferentes formas de informações podem se referir ao mesmo conteúdo semântico, e variações de granularidade ainda podem preservar um alinhamento confiável. A análise principal concentra-se nas saídas de LLM, enquanto LDADE é utilizado como base comparativa complementar.

2.4.1 Avaliação Baseada em Correspondência

O procedimento de correspondência utilizado neste trabalho baseia-se na correspondência semântica entre tópicos produzidos pelo avaliador humano e tópicos gerados automaticamente. O avaliador constrói um mapeamento de *relação* comparando cada *Evaluator Topic* com todos os *LLM Topics* e registrando toda correspondência semântica identificada, permitindo relações um-para-muitos e muitos-para-um. Cada resultado é resumido como X–Y, em que X representa o número de tópicos produzidos pelo avaliador e Y representa o número de tópicos gerados pelo LLM ligados por meio das relações registradas. Por exemplo, se o tópico 1 do avaliador corresponder aos tópicos {1,3} do LLM e o tópico 2 do avaliador corresponder aos tópicos {1,4,5}, a relação compartilhada por meio do tópico 1 do LLM conecta os conjuntos em uma única correspondência 2–4.

As correspondências resultantes são então agrupadas em quatro níveis de acordo com alinhamento semântico e consistência de escopo, como ilustrado na Figura 1.

GPT 3.5, 10 topics		
Evaluator Topics	GPT Topics	Relation
1	1	(2-4)
1	3	
2	1	
2	4	
2	5	
3	6	(2-1)
4	6	
5	7	(2-1)
6	7	
7	9	(3-1)
8	9	
9	9	
10	10	(1-1)
0	2	(0-2)
0	8	

Figura 1 – Exemplo do mapeamento de relação e da correspondência resultante X–Y.

Classificamos as correspondências em quatro níveis:

- **Correspondências Muito Altas:** casos de alinhamento semântico exato ou quase exato, normalmente envolvendo correspondências um-para-um ou pareamentos limpos, como 1–1 ou 2–2. Esses casos indicam que os tópicos gerados correspondem de perto aos tópicos do avaliador tanto em significado quanto em escopo.
- **Correspondências Altas:** casos com forte sobreposição semântica, mas pequenas diferenças de granularidade, normalmente envolvendo efeitos de divisão ou fusão de tópicos, como 1–2 ou 3–1. Nesses casos, o tema principal é preservado, embora um dos lados o expresse de forma mais ampla ou mais específica.
- **Correspondências Baixas:** casos em que alguma sobreposição existe, mas a correspondência é enfraquecida por diferenças substanciais de escopo, fragmentação

excessiva, agregação excessiva ou deriva semântica parcial, como em exemplos 1–4 ou 4–1.

- **Correspondências Muito Baixas:** casos de incompatibilidade ou ausência de alinhamento confiável, incluindo *commission* (0–Y, saídas não suportadas ou fora do tema) e *omission* (X–0, tópicos relevantes do avaliador não capturados pelo modelo).

Em nível conceitual, os tópicos de LLM são contrastados com tópicos produzidos por humanos e depois resumidos por idioma e quantidade de tópicos (5, 7 e 10). Este capítulo apresenta a lógica do método de avaliação, enquanto o protocolo operacional completo, incluindo procedimentos dos avaliadores e decisões de agregação, é descrito no Capítulo 4.

2.4.2 Interpretação Complementar da Base Comparativa com LDADE

Para a base LDADE, a análise adota a convergência de interpretação entre dois avaliadores como evidência complementar. As categorias são *Agreement*, quando ambos os avaliadores atribuem interpretações semelhantes ao tópico; *Partial*, quando as interpretações se sobrepõem apenas parcialmente; e *Disagreement*, quando os significados atribuídos não são semelhantes. Assim como na análise de LLM, os detalhes de implementação são apresentados no Capítulo 4.

2.5 Transformers e Arquitetura de Redes Neurais

Transformers são a base dos LLMs modernos (VASWANI et al., 2017). Sua inovação central é o mecanismo de *self-attention*, que calcula relações entre *tokens* em uma sequência e permite que o modelo foque em informações contextualmente relevantes independentemente do tamanho. Esse mecanismo é especialmente importante em textos longos e densos em informação, uma vez que pistas relevantes podem estar dispersas em diferentes partes de uma transcrição.

Em comparação com modelos sequenciais anteriores, *Transformers* permitem maior paralelização durante o treinamento e sustentaram as tendências de escalabilidade que caracterizam os LLMs contemporâneos. Ainda assim, esses modelos continuam sendo computacionalmente exigentes e podem enfrentar dificuldades com contextos muito longos, a menos que estratégias arquiteturais ou de *prompting* adicionais sejam adotadas (BROWN et al., 2020; CHOWDHERY et al., 2024).

2.6 Pré-treinamento Não Supervisionado e *Fine-Tuning*

LLMs são comumente pré-treinados em grandes volumes de dados por meio de formas auto-supervisionadas que lhes permitem aprender regularidades linguísticas amplas a partir de texto bruto (DEVLIN et al., 2018). Após essa etapa, os modelos podem ser adaptados a domínios ou tarefas específicas por meio de *fine-tuning*, *instruction tuning* ou procedimentos de alinhamento relacionados. Essa adaptação adicional é relevante em domínios como engenharia de *software* e educação, nos quais vocabulário especializado e convenções comunicativas podem diferir daqueles encontrados em corpora gerais da *web* (HOWARD; RUDER, 2018; EBERT; LOURIDAS, 2023).

O pré-treinamento proporciona competência linguística aprimorada, enquanto o *fine-tuning* e adaptações orientadas por instrução podem melhorar a relevância para a tarefa e a qualidade das saídas. No entanto, esses processos também introduzem compromissos práticos, incluindo custo computacional, dependência de dados precisos e a possibilidade de mudanças indesejadas no comportamento do modelo quando a adaptação é excessivamente estreita ou mal controlada (HOWARD; RUDER, 2018; ABDULLAH; MADAIN; JARARWEH, 2022).

2.7 Processamento Multilíngue e Adaptação Cultural

Modelos de linguagem multilíngues são projetados para capturar regularidades translinguísticas e transferir conhecimento entre idiomas, o que é especialmente útil em estudos que envolvem mais de um contexto linguístico (CONNEAU et al., 2020; PIRES; SCHWENK; CORREIA, 2019). Nesta monografia, o processamento multilíngue é relevante porque o conjunto de dados inclui transcrições em português e em inglês.

Ao mesmo tempo, a capacidade multilíngue não deve ser interpretada como equivalência completa entre idiomas. Variações de terminologia, estilo discursivo e contexto cultural podem afetar a forma como os tópicos são expressos e interpretados. Portanto, a avaliação bilíngue permanece importante para verificar se o comportamento da extração de tópicos se mantém consistente em diferentes condições linguísticas.

2.8 *Latent Dirichlet Allocation* (LDA) e LDADE

LDA é um modelo probabilístico de tópicos que representa documentos como misturas de tópicos latentes e tópicos como distribuições sobre palavras (BLEI; NG; JORDAN, 2003; BLEI, 2012). Devido à sua simplicidade e relevância histórica, é frequentemente utilizado como base comparativa em estudos de modelagem de tópicos. Ainda assim, LDA pode gerar tópicos incoerentes ou de interpretação fraca, e seu desempenho é sensível a

escolhas como número de tópicos e distribuições a priori (BLEI, 2012; AGRAWAL; FU; MENZIES, 2018).

LDADE estende essa base ao combinar LDA com Evolução Diferencial para automatizar o ajuste de hiperparâmetros e reduzir tentativas e erros manuais (AGRAWAL; FU; MENZIES, 2018). Essa abordagem baseada em busca pode melhorar a qualidade dos tópicos em relação a um LDA não ajustado, mas ainda opera sob as limitações representacionais de modelos probabilísticos baseados em distribuições de palavras. Por essa razão, no presente estudo LDA/LDADE é utilizado como linha de base complementar, e não como abordagem analítica principal.

2.9 Trabalhos Relacionados

A modelagem clássica de tópicos é utilizada há muito tempo para extrair temas de grandes volumes textuais. Modelos probabilísticos como LDA representam documentos como misturas de tópicos latentes e tópicos como distribuições sobre palavras, viabilizando a identificação de temas no conjunto analisado (BLEI, 2012; BLEI; NG; JORDAN, 2003). Trabalhos subsequentes propuseram extensões variacionais e autoencodificadoras para melhorar a qualidade e a estabilidade da inferência (SRIVASTAVA; SUTTON, 2017). Em contextos aplicados, modelos de tópicos também têm apoiado extração de palavras-chave e tarefas analíticas em diferentes domínios, incluindo contextos educacionais e institucionais (JUNIOR et al., 2015). No entanto, essas abordagens normalmente dependem de representações *bag-of-words*, são sensíveis a hiperparâmetros e frequentemente geram tópicos difíceis de interpretar ou apenas parcialmente alinhados com julgamentos humanos (AGRAWAL; FU; MENZIES, 2018).

Abordagens neurais anteriores para sumarização e condensação semântica buscaram mitigar algumas dessas limitações. Rush et al. propuseram um modelo neural de atenção para sumarização abstrativa de sentenças, superando linhas de base anteriores em textos no estilo jornalístico (RUSH; CHOPRA; WESTON, 2015). Para conteúdo relacionado ao YouTube, Albeer et al. aplicaram TF–IDF para resumir automaticamente transcrições de vídeos e extrair termos-chave, mostrando que mesmo estratégias relativamente simples de ponderação lexical podem auxiliar a navegação em vídeos longos (ALBEER et al., 2022). Ainda assim, TF–IDF e métodos relacionados permanecem limitados por estatísticas de frequência de palavras, possuem capacidade limitada para lidar com paráfrases e equivalência semântica e frequentemente exigem ajuste de limiares (CHRISTIAN; AGUS; SUHARTONO, 2016).

Mais recentemente, LLMs têm sido investigados como extratores de tópicos e sumarizadores. Zhang et al. compararam *Large Language Models* para sumarização e relataram que saídas geradas por modelos podem se aproximar de referências humanas em qualidade sob determinadas configurações de avaliação (ZHANG et al., 2024a; ZHANG et al.,

2024b). Wu et al. exploraram LLMs como diferentes *role-players* para avaliação de sumarização e encontraram concordância substancial com a avaliação humana (WU et al., 2023). Em modelagem de tópicos, Wang et al. introduziram o *PromptTopic*, uma abordagem baseada em *prompt* que utiliza LLMs para induzir tópicos diretamente a partir de documentos, relatando melhorias em relação a linhas de base tradicionais em medidas relacionadas à coerência e interpretabilidade (WANG et al., 2023). Esses estudos sugerem que LLMs são uma alternativa promissora quando o objetivo não é apenas agrupar palavras estatisticamente, mas também obter rótulos de tópicos semanticamente significativos.

Paralelamente, diversos estudos examinaram o papel de LLMs em tarefas de engenharia de *software*. Tufano et al. documentaram o uso do ChatGPT em projetos de código aberto, mostrando que desenvolvedores o empregam em atividades como refatoração, documentação, correção de *bugs* e apoio relacionado a testes (TUFANO et al., 2024b). Outros trabalhos discutem oportunidades e riscos mais amplos da IA generativa para profissionais de *software*, incluindo assistência em atividades repetitivas e a contínua necessidade de supervisão humana (EBERT; LOURIDAS, 2023). Colavito et al. investigaram o uso de modelos do tipo GPT para rotulação de *issues*, indicando que LLMs podem capturar semântica refinada em artefatos de repositórios de *software* (COLAVITO et al., 2024). De forma complementar, Dantas et al. avaliaram a legibilidade de trechos de código recomendados pelo ChatGPT, destacando tanto pontos fortes quanto limitações em cenários de engenharia de *software* (DANTAS; ROCHA; MAIA, 2023). Estudos como DevGPT também reforçam que a interação de desenvolvedores com IA conversacional tornou-se um objeto ativo de investigação empírica (XIAO et al., 2024).

Há também um corpo crescente de trabalhos sobre mineração de informações no YouTube. Parte dessa literatura se concentra em sumarização educacional e condensação de transcrições (ALBEER et al., 2022; SUDHAN; VEDHAVIYASSH; SARANYA, 2023; DEVI et al., 2023), enquanto outra parte enfatiza problemas relacionados à moderação, como detecção de *spam* ou cyberbullying (OCALLAGHAN et al., 2021; YAFOOZ; ALDHAQM; ALSAEEDI, 2023). Embora esses estudos demonstrem que a análise em larga escala de dados textuais relacionados ao YouTube é viável, eles não abordam diretamente o problema específico explorado nesta monografia: ajudar aprendizes a identificar rapidamente a cobertura instrucional de vídeos sobre desenvolvimento de *software* por meio da extração de tópicos.

De modo geral, a literatura indica que a modelagem clássica de tópicos e estratégias de ponderação lexical permanecem linhas de base úteis, mas frequentemente ficam aquém em coerência semântica e interpretabilidade humana. Ao mesmo tempo, estudos recentes sugerem que LLMs podem fornecer saídas mais sensíveis ao contexto e mais alinhadas ao julgamento humano em sumarização, indução de tópicos e aplicações em engenharia de *software* (ZHANG et al., 2024a; WU et al., 2023; WANG et al., 2023; TUFANO et al., 2024b; EBERT; LOURIDAS, 2023; COLAVITO et al., 2024; DANTAS; ROCHA; MAIA,

2023). Com base nessa literatura, a presente monografia contribui com uma avaliação bilíngue (português/inglês) da extração de tópicos a partir de transcrições do *YouTube* sobre desenvolvimento de *software*, considerando múltiplas quantidades de tópicos e uma comparação com LDA/LDADE, além da implementação de um protótipo baseado em navegador.

2.10 Resumo do Capítulo

Este capítulo apresentou a fundamentação teórica da monografia, cobrindo LLMs, conceitos centrais de NLP, extração de tópicos, *Transformers*, processamento multilíngue e o papel de LDA/LDADE como métodos de base comparativa. Também introduziu, em nível conceitual, a lógica de avaliação baseada em correspondência adotada neste estudo. Os procedimentos experimentais detalhados são descritos no Capítulo 4, enquanto o Capítulo 3 apresenta o protótipo que operacionaliza o *pipeline* proposto.

Capítulo 3

Implementação

Este capítulo descreve a implementação do protótipo que operacionaliza a abordagem de extração de tópicos discutida nos capítulos anteriores. Enquanto a fundamentação teórica e o estudo empírico se concentram em avaliar LLMs para extração de tópicos, aqui a ênfase está em como essas ideias foram traduzidas em um sistema funcional integrado ao YouTube. O capítulo se concentra nas decisões de engenharia mais diretamente relacionadas à solução proposta; a avaliação comparativa com LDA/LDADE permanece restrita ao Capítulo 4.

A solução implementada é composta por duas partes principais: (i) um *backend* em Python, implementado com Flask, responsável pelo *download* do áudio, transcrição e extração de tópicos; e (ii) uma extensão de navegador, escrita em HTML, CSS e JavaScript, que permite ao usuário acionar o processamento e visualizar os tópicos resultantes diretamente na página do YouTube. As seções a seguir apresentam os requisitos, a arquitetura geral, a implementação do *backend*, a extensão de navegador e um exemplo de uso sob a perspectiva do usuário final.

3.1 Requisitos e Escopo

O objetivo da implementação é oferecer uma ferramenta prática que ajude estudantes e desenvolvedores a avaliar rapidamente a relevância de vídeos educacionais do YouTube com base em tópicos extraídos automaticamente. A partir desse objetivo, foram definidos os seguintes requisitos funcionais:

- **R1 – Aquisição de transcrição:** dado o URL de um vídeo do YouTube, o sistema deve recuperar o áudio correspondente e produzir uma transcrição textual.
- **R2 – Geração de tópicos:** o sistema deve extrair uma lista concisa de tópicos da transcrição utilizando um LLM.

- ❑ **R3 – Seleção de quantidade de tópicos:** o usuário deve poder selecionar a quantidade desejada de tópicos (5, 7 ou 10).
- ❑ **R4 – Integração com navegador:** o usuário deve poder acionar todo o *pipeline* de dentro do navegador, sem sair da página do YouTube.
- ❑ **R5 – Visualização de resultados:** os tópicos resultantes devem ser apresentados de forma legível e estruturada.

Além disso, alguns requisitos não funcionais foram definidos:

- ❑ **NFR1 – Responsividade:** o tempo de processamento deve ser aceitável mesmo para vídeos mais longos, motivando execução assíncrona, transcrição segmentada e extração de tópicos baseada em blocos de texto.
- ❑ **NFR2 – Modularidade:** o *backend* deve ser desacoplado da extensão de navegador, de modo que outras interfaces possam ser adicionadas no futuro.
- ❑ **NFR3 – Privacidade:** o sistema deve evitar o armazenamento de dados do usuário; apenas o áudio temporário e a transcrição necessários para chamadas de API são manipulados durante a execução.
- ❑ **NFR4 – Compatibilidade:** a extensão de navegador deve ser compatível com navegadores baseados em Chromium e seguir a especificação Manifest V3.

3.2 Arquitetura Geral

A Figura 2 ilustra a interface principal da solução. O usuário interage apenas com a extensão de navegador. Quando o botão *Process* é clicado, a extensão envia uma requisição ao servidor Flask local, repassando o URL do vídeo do YouTube e a quantidade desejada de tópicos. A implementação adota um fluxo assíncrono baseado em *jobs*: o *backend* retorna imediatamente um `job_id` e continua o processamento em segundo plano. Esse desenho permite que o usuário feche o *popup* ou troque de aba sem interromper a tarefa.

O *backend* então executa três etapas principais:

1. baixar o fluxo de áudio do vídeo;
2. transcrever o áudio usando a API de speech-to-text da OpenAI; e
3. extrair os principais tópicos da transcrição usando GPT-4.

A extensão monitora a execução consultando periodicamente um *endpoint* de status e atualiza a interface com o estado atual do *job* e o percentual de progresso. Quando o processamento é concluído, os tópicos gerados são exibidos no *popup*. Essa arquitetura

separa a interface do usuário do *pipeline* de processamento, melhorando modularidade e robustez.

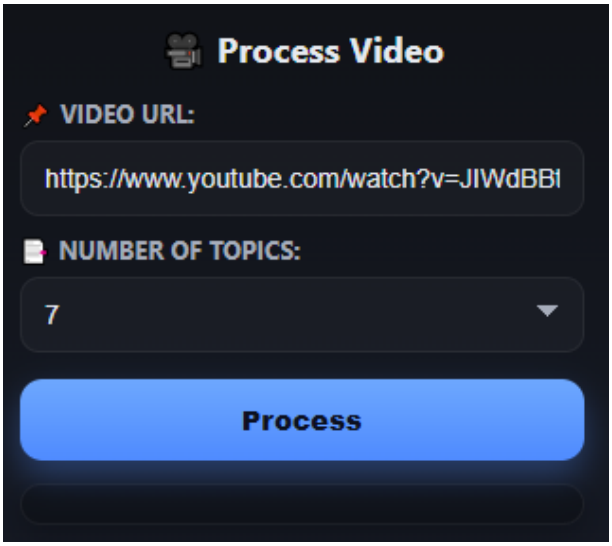

Figura 2 – Estado inicial do *popup* da extensão de navegador.

O projeto está organizado em duas pastas:

- `Back/`: contém os arquivos do *backend* em Python (`main.py`, `transcricao.py`, `topicos.py`);
- `Front/`: contém os arquivos da extensão de navegador (`manifest.json`, `popup.html`, `popup.js`, `style.css`).

Variáveis de ambiente, como a chave da API da OpenAI e parâmetros de transcrição, são definidas em um arquivo `.env` e carregadas em tempo de execução.

3.3 Implementação do *Backend*

O *backend* é responsável por orquestrar todo o *pipeline* de *download*, transcrição e extração de tópicos. Ele foi implementado em Python com Flask e expõe dois *endpoints* centrais: um para iniciar o processamento e outro para recuperar o status do *job*. Em vez de detalhar todas as funções auxiliares, esta seção enfatiza as partes da implementação mais relevantes para a solução proposta.

3.3.1 Serviço assíncrono em Flask

O ponto de entrada do *backend* é o arquivo `main.py`. O serviço adota um fluxo assíncrono baseado em *jobs*. O *endpoint* `/process` recebe o URL do YouTube e a quantidade de tópicos solicitada, cria um *job* e retorna um `job_id`. O *endpoint* `/status/<job_id>`

permite que a extensão consulte o estado atual do processamento e, quando disponível, recupere os tópicos finais.

```python
@app.route("/process", methods=["POST"])
def process_video_async():
    data = request.get_json() or {}
    video_url = (data.get("video_url") or "").strip()
    num_topics = int(data.get("num_topicos", 7))

    if not video_url:
        return jsonify({"error": "Video URL not provided."}), 400

    job_id = str(uuid4())
    _job_set(job_id, status="queued", progress=0,
             video_url=video_url, num_topics=num_topics)

    EXECUTOR.submit(_run_pipeline, job_id, video_url, num_topics)
    return jsonify({"job_id": job_id, "status": "queued"}), 202

@app.route("/status/<job_id>", methods=["GET"])
def job_status(job_id):
    job = _job_get(job_id)
    if not job:
        return jsonify({"error": "job_id not found"}), 404
    return jsonify(job), 200
```

Listing 3.1 – Serviço Flask assíncrono simplificado.

Essa interface é central para o protótipo porque permite desacoplar o processamento de longa duração do ciclo de vida do *popup*. Mecanismos auxiliares, como persistência interna, gerenciamento de *cache* e rotinas de limpeza, também foram implementados, mas não constituem o núcleo conceitual da solução e, por isso, são omitidos aqui por brevidade.

3.3.2 Download do áudio e transcrição

O módulo `transcricao.py` é responsável por baixar o áudio e transcrevê-lo usando a API de speech-to-text da OpenAI. O Listing 3.2 mostra a função que baixa o áudio utilizando `yt-dlp`.

```python
def download_audio_from_youtube(video_url):
    out_dir = Path("work")
    out_dir.mkdir(exist_ok=True)
    output_tmpl = str(out_dir / "audio.%(ext)s")

    base = _yt_dlp_base_cmd() + [
        "-f", "bestaudio/best",
        "--no-playlist",
        "-o", output_tmpl,
```

```
        video_url
    ]

    strategies = [
        ("player_client=default", base + ["--extractor-args", "
            youtube:player_client=default"]),
        ("default,-android_sdkless", base + ["--extractor-args", "
            youtube:player_client=default,-android_sdkless"]),
        ("player_client=web", base + ["--extractor-args", "youtube:
            player_client=web"]),
    ]

    for _, cmd in strategies:
        try:
            _run_cmd(cmd)
            audio_file = max(out_dir.glob("audio.*"), key=lambda p:
                p.stat().st_mtime)
            return str(audio_file)
        except subprocess.CalledProcessError:
            continue

    return None
```

Listing 3.2 – Download de áudio do YouTube (`download_audio_from_youtube`).

Para vídeos mais longos, o áudio é segmentado com `ffmpeg` antes da transcrição. Essa decisão melhora a robustez e reduz o tamanho de cada requisição enviada à API de transcrição. A duração dos segmentos e o paralelismo são configuráveis por meio de variáveis de ambiente, o que torna a solução adaptável a diferentes ambientes de execução.

```
def transcribe_audio(file_path, segment_seconds=8 * 60, workers=1):
    try:
        segment_seconds = int(os.getenv("TRANSCRIBE_SEGMENT_SECONDS"
            , str(segment_seconds)))
        workers = int(os.getenv("TRANSCRIBE_WORKERS", str(workers)))
        model = os.getenv("TRANSCRIBE_MODEL")

        client = openai.OpenAI(api_key=OPENAI_API_KEY)
        segments = _split_audio_ffmpeg(file_path, segment_seconds=
            segment_seconds)

        def _one(seg_path: str):
            with open(seg_path, "rb") as audio_file:
                resp = client.audio.transcriptions.create(
                    model=model,
                    file=audio_file
                )
            return resp.text
```

```
17
18          if workers <= 1:
19              return "\n".join(_one(p) for p in segments)
20
21          texts = [None] * len(segments)
22          with ThreadPoolExecutor(max_workers=workers) as ex:
23              fut_map = {ex.submit(_one, p): i for i, p in enumerate(
                    segments)}
24              for fut in as_completed(fut_map):
25                  texts[fut_map[fut]] = fut.result()
26
27          return "\n".join(texts)
28
29      except Exception:
30          return None
```

Listing 3.3 – Transcrição segmentada com a API speech-to-text da OpenAI.

Após a conclusão do *pipeline*, os arquivos temporários de áudio são removidos. Esse comportamento é consistente com o escopo orientado à privacidade do protótipo, que evita o armazenamento permanente de conteúdo relacionado ao usuário.

3.3.3 Extração de tópicos com GPT-4

O módulo `topicos.py` implementa a lógica de extração de tópicos a partir da transcrição usando GPT-4. Esta etapa é a mais diretamente relacionada à contribuição da monografia, pois operacionaliza o uso de LLMs para extração de tópicos em uma ferramenta real.

O processo inclui três etapas principais:

1. dividir a transcrição em blocos de texto para evitar *prompts* muito longos;
2. obter tópicos candidatos a partir de cada bloco; e
3. consolidar os candidatos em uma lista final de tópicos, respeitando a quantidade solicitada.

Um *prompt-base* compartilhado é utilizado em todas as chamadas ao GPT-4, seguindo a mesma estrutura adotada no estudo empírico:

```
1 BASE_PROMPT = (
2     "Generate a concise summary by extracting the X most "
3     "important topics from the following text. "
4     "Focus on highlighting important information that "
5     "encapsulates the main points of the content."
6 )
```

Listing 3.4 – Prompt base para extração dos tópicos.

O Listing 3.5 resume a função `extract_topics_with_gpt`.

```
def extract_topics_with_gpt(transcription, num_topicos, workers=2):
    try:
        client = openai.OpenAI(api_key=OPENAI_API_KEY)
        chunks = _chunk_text(transcription, max_chars=12000)

        if len(chunks) == 1:
            prompt = BASE_PROMPT.replace("X", str(num_topicos)) + f"\n\n{transcription}"
            response = client.chat.completions.create(
                model="gpt-4",
                messages=[
                    {"role": "system", "content": "You are an assistant specialized in summarizing educational content."},
                    {"role": "user", "content": prompt},
                ],
                temperature=0.3,
                max_tokens=320,
            )
            return response.choices[0].message.content.strip()

        candidates = []
        with ThreadPoolExecutor(max_workers=workers) as ex:
            futs = [ex.submit(_topics_for_chunk, client, c, num_topicos) for c in chunks]
            for fut in as_completed(futs):
                candidates.extend(fut.result())

        seen, unique = set(), []
        for t in candidates:
            key = t.lower()
            if key not in seen:
                seen.add(key)
                unique.append(t)

        candidates_text = "\n".join(f"- {t}" for t in unique)
        merge_prompt = BASE_PROMPT.replace("X", str(num_topicos)) + f"\n\n{candidates_text}"
        final_resp = client.chat.completions.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": "You are an assistant specialized in summarizing educational content."},
                {"role": "user", "content": merge_prompt},
            ],
```

```
            temperature=0.2,
            max_tokens=320,
        )
        return final_resp.choices[0].message.content.strip()

    except Exception:
        return None
```

Listing 3.5 – Extração e consolidação de tópicos com GPT-4.

Para transcrições mais longas, essa estratégia em duas etapas permite que o protótipo permaneça dentro dos limites de tamanho de *prompt*, preservando ao mesmo tempo a quantidade de tópicos solicitada. Em vez de depender de um único *prompt* muito grande, o *backend* primeiro extrai candidatos locais e depois os consolida em um conjunto final de tópicos. Esse desenho é importante porque traduz a lógica de extração de tópicos do estudo em uma implementação escalável.

3.4 Extensão de Navegador

A extensão de navegador é o componente que expõe o sistema ao usuário final. Seu papel é intencionalmente simples: coletar o URL do vídeo atual, permitir que o usuário escolha a quantidade de tópicos, iniciar o *job* assíncrono e exibir o resultado.

3.4.1 Manifesto e fluxo do *popup*

O arquivo `manifest.json` declara as permissões e os recursos exigidos pela extensão:

```
{
  "manifest_version": 3,
  "name": "YouTube Video Processor",
  "version": "1.0",
  "description": "Extension that processes YouTube videos and
     extracts topics.",

  "permissions": ["tabs", "storage"],

  "host_permissions": [
    "https://www.youtube.com/*",
    "http://127.0.0.1:5000/*"
  ],

  "action": {
    "default_popup": "popup.html",
    "default_icon": "icon.png"
  }
}
```

Listing 3.6 – Trecho de `manifest.json`.

A lógica de processamento permanece no *backend*, enquanto o *popup* armazena localmente o último `job_id`, de modo que o progresso possa ser retomado se a interface for fechada e reaberta. Essa escolha mantém a extensão leve e delega o processamento mais pesado ao serviço de *backend*.

A interface visual é definida em `popup.html` e `style.css`. O *popup* oferece:

- ❏ um campo de texto com o URL do vídeo do YouTube;
- ❏ um seletor para a quantidade de tópicos (5, 7 ou 10); e
- ❏ um botão *Process* que aciona a requisição ao *backend*.

O comportamento do *popup* é implementado em `popup.js`. Quando o usuário clica em *Process*, o script inicia um job com `POST /process`, armazena o `job_id` retornado e consulta periodicamente `GET /status/<job_id>` até que o *backend* informe conclusão ou erro.

```
processButton.addEventListener("click", async () => {
  const videoUrl = videoUrlInput.value.trim();
  const numTopicos = parseInt(numTopicsSelect.value, 10);

  const startResp = await fetch("http://127.0.0.1:5000/process", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ video_url: videoUrl, num_topicos:
        numTopicos }),
  });

  const startData = await startResp.json();
  const jobId = startData.job_id;

  chrome.storage.local.set({ last_job_id: jobId });

  const pollTimer = setInterval(async () => {
    const st = await fetch(`http://127.0.0.1:5000/status/${jobId}`,
        { cache: "no-store" });
    const job = await st.json();

    responseDiv.innerHTML = `Status: <strong>${job.status}</strong>
        (${job.progress}%)`;

    if (job.status === "done") {
      clearInterval(pollTimer);
      renderTopics(job.topics || "");
```

```
25        chrome.storage.local.remove(["last_job_id"]);
26      }
27
28      if (job.status === "error") {
29        clearInterval(pollTimer);
30        responseDiv.innerHTML = `<span class="error">Error: ${job.
              error}</span>`;
31        chrome.storage.local.remove(["last_job_id"]);
32      }
33    }, 2000);
34  });
```

Listing 3.7 – Lógica principal do script de janela popup (`popup.js`).

Quando o processamento termina, os tópicos retornados são exibidos no *popup* como uma lista estruturada. A interface, portanto, atua como uma camada fina de interação sobre o *pipeline* do *backend*.

As Figuras 3 e 4 ilustram a interface final, mostrando os tópicos gerados para um vídeo real sobre desenvolvimento de software enquanto o usuário percorre os resultados.

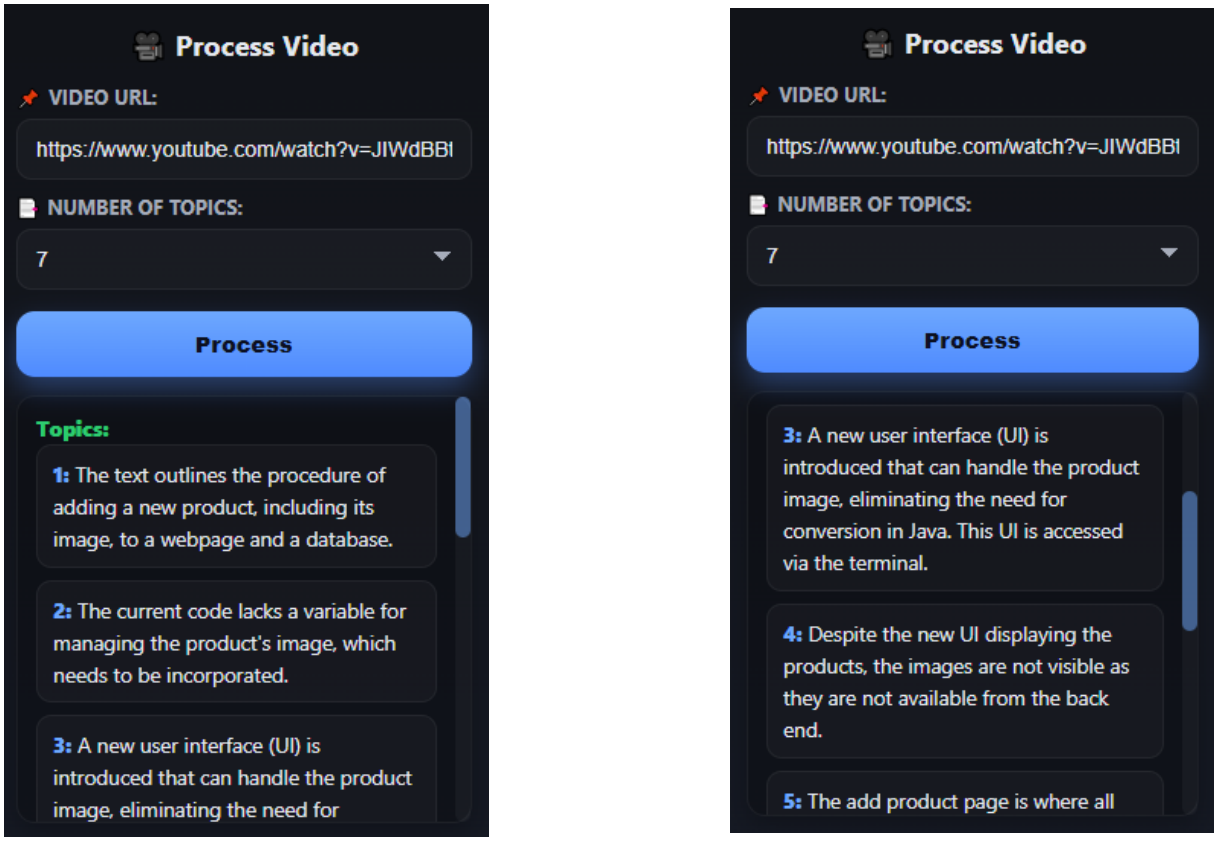

Figura 3 – Lista de tópicos gerados (topo da área rolável e primeiro deslocamento).

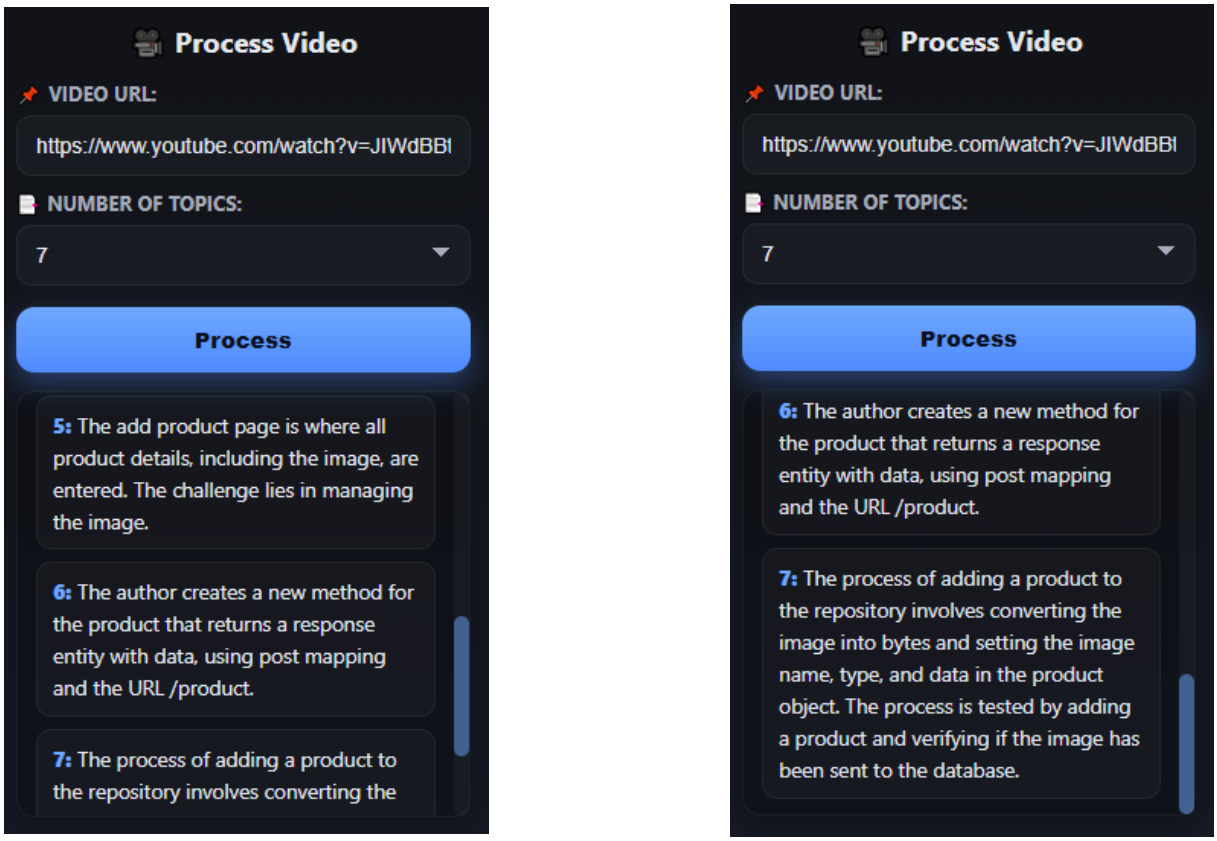

Figura 4 – Continuação dos tópicos gerados (segundo e terceiro deslocamentos).

3.5 Exemplo de uso e relação com o estudo

Para utilizar o protótipo, o usuário deve primeiro iniciar o servidor Flask e depois abrir um vídeo do YouTube no navegador. Quando o ícone da extensão é clicado, o *popup* detecta o URL do vídeo atual e preenche automaticamente o campo de entrada. O usuário seleciona a quantidade desejada de tópicos (5, 7 ou 10) e pressiona *Process*. O *popup* então exibe status e progresso enquanto consulta o *backend*, e os tópicos são mostrados quando o *job* assíncrono é concluído.

O comportamento interno da extensão é consistente com o protocolo experimental apresentado no Capítulo 4: a mesma estrutura de *prompt* é utilizada para solicitar 5, 7 ou 10 tópicos, e o modelo opera sobre a transcrição do vídeo. As principais diferenças são de natureza de engenharia, a saber, o fluxo assíncrono, a transcrição segmentada, o uso de blocos de texto no *prompting* e a integração ao navegador. Nesse sentido, o protótipo pode ser entendido como a operacionalização prática do estudo.

3.6 Limitações atuais e possíveis melhorias

A versão atual do protótipo apresenta algumas limitações:

- depende de ferramentas externas (`yt-dlp` e `ffmpeg`) e da disponibilidade das APIs

de speech-to-text da OpenAI e do GPT-4;

- ❏ executa o *backend* localmente, o que simplifica o desenvolvimento, mas ainda exige configuração manual por parte do usuário;
- ❏ apenas o GPT-4 é exposto na extensão, embora o estudo empírico tenha comparado múltiplas famílias de LLM;
- ❏ a extensão atualmente oferece suporte apenas ao YouTube; outras plataformas exigiriam integrações adicionais.

Como trabalhos futuros, o *backend* poderia ser implantado em um servidor remoto, LLMs adicionais poderiam ser disponibilizados como opções selecionáveis na interface, e recursos de navegação mais ricos, como marcações de tempo por tópico, poderiam ser incorporados.

Apesar dessas limitações, a implementação demonstra a viabilidade de integrar a extração de tópicos baseada em LLM ao fluxo cotidiano de estudantes e desenvolvedores, aproximando o estudo empírico de uma ferramenta concreta que pode ser utilizada na prática.

Capítulo 4

Experimentos e Análise dos Resultados

Este capítulo apresenta o *pipeline* experimental e a análise dos resultados. Inicialmente, descrevemos o processo completo utilizado para extrair tópicos de vídeos sobre desenvolvimento de *software* com (LLMs). Em seguida, apresentamos resultados quantitativos comparando tópicos gerados por LLM com tópicos identificados por humano, organizados por idioma (português/inglês) e pela quantidade de tópicos solicitada (5/7/10). Também incluímos uma base de comparação com LDA, implementada com ajuste baseado em Evolução Diferencial (LDADE), além de uma discussão comparativa. O capítulo é concluído com ameaças à validade e um resumo dos principais achados. Todos os materiais necessários para inspecionar e reproduzir os experimentos (por exemplo, transcrições, *prompts*, saídas brutas e *scripts* de análise) estão disponíveis em um registro no Zenodo.[1]

4.1 Metodologia

Esta seção detalha o *pipeline* de oito etapas utilizado para selecionar vídeos, obter transcrições diretamente do YouTube, gerar tópicos com diferentes LLMs e diferentes quantidades de tópicos, e compará-los com uma referência humana, incluindo uma análise complementar com LDADE como base de comparação. É importante destacar que a fonte de transcrição utilizada neste capítulo experimental é a legenda automática do YouTube, o que é metodologicamente independente do fluxo de implementação posterior da extensão de navegador descrito no Capítulo 3.

O estudo avalia três configurações de quantidade de tópicos: 5, 7 e 10 tópicos. Esses valores foram escolhidos porque representam níveis compacto, intermediário e mais detalhado de granularidade dos tópicos, mantendo ao mesmo tempo a viabilidade operacional do processo de avaliação manual. Como cada saída de modelo precisou ser comparada com uma referência produzida por humanos por meio de um procedimento baseado em correspondência, testar um número substancialmente maior de configurações de quantidade de

[1] Zenodo DOI: 10.5281/zenodo.17901591. Página do registro: <https://zenodo.org/records/17901591>.

tópicos teria aumentado o tempo e o esforço necessários para a avaliação manual. Assim, as quantidades selecionadas refletem um equilíbrio entre variedade analítica e viabilidade da avaliação.

Seguimos um processo de oito etapas para avaliar a efetividade de LLMs na extração de tópicos de transcrições de vídeos (Figura 5).

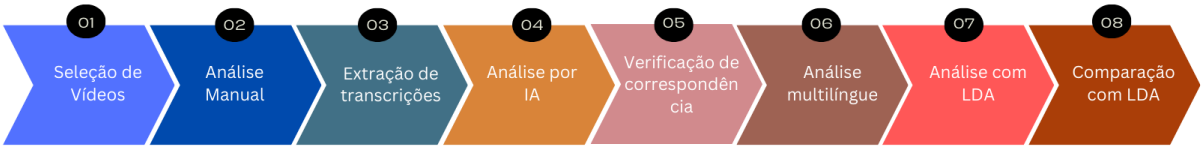

Figura 5 – Fluxograma da metodologia em oito etapas.

1. **Seleção de vídeos.** Selecionamos 20 vídeos do YouTube (10 em inglês e 10 em português) relacionados a desenvolvimento de *software*, utilizando consultas predefinidas (Tabela 1) adaptadas de (ROCHA; MAIA, 2023) para garantir diversidade temática.

2. **Análise manual.** Um avaliador humano assistiu a cada vídeo e produziu uma lista concisa de seus principais tópicos. Essa lista serve como referência para a comparação com LLM.

3. **Extração de transcrições.** Obtivemos as transcrições automáticas do YouTube e realizamos uma limpeza leve (por exemplo, remoção de marcas de tempo e artefatos) para preparar a entrada para os modelos.

4. **Análise por IA.** Solicitamos a três LLMs (*GPT-4*, *GPT-3.5* e *Gemini*) que extraíssem os principais 5, 7 e 10 tópicos de cada transcrição, impondo rótulos sucintos e bem formados para aumentar a comparabilidade. Para os vídeos em inglês, utilizamos o *prompt* (com $X \in \{5, 7, 10\}$): *"Generate a concise summary by extracting the X most important topics from the following text. Focus on highlighting important information that encapsulates the main points of the content."* Para os vídeos em português, utilizamos uma tradução direta do mesmo *prompt* (novamente com $X \in \{5, 7, 10\}$): *"Gere um resumo conciso extraindo os X tópicos mais importantes do texto a seguir. Foque em destacar informações importantes que encapsulem os principais pontos do conteúdo."*

5. **Verificação de correspondência.** Para cada vídeo, o mesmo avaliador comparou a lista de tópicos gerada pelo LLM com a lista produzida manualmente e atribuiu um de quatro níveis de correspondência: *Muito Alta*, *Alta*, *Baixa* ou *Muito Baixa*. Os critérios de avaliação baseados em correspondência e a notação X–Y utilizada

para resumir cada resultado de correspondência são definidos no Capítulo 2, Seção 2.4 e Subseção 2.4.1; aqui, eles são aplicados operacionalmente a cada vídeo e configuração.

6. **Análise multilíngue.** Repetimos o mesmo procedimento para vídeos em português e inglês, adaptando apenas o idioma do *prompt* e preservando a mesma estrutura e restrições.

7. **Análise com LDA.** Como base de comparação complementar, aplicamos o *pipeline* LDADE com 5 e 10 tópicos por vídeo. Em seguida, dois outros avaliadores independentes interpretaram individualmente cada tópico gerado pela base de comparação, escrevendo o que acreditavam que cada tópico baseado em palavras-chave representava. Quando uma interpretação clara não era possível, o tópico era marcado como indeterminado.

8. **Comparação com LDA.** Após ambos os avaliadores concluírem suas interpretações individuais, o avaliador principal comparou as duas interpretações lado a lado para cada tópico da base de comparação e classificou sua relação como *Agreement* (interpretação semelhante), *Partial* (interpretação parcialmente semelhante) ou *Disagreement* (interpretação não semelhante). Essa comparação entre avaliadores foi utilizada como uma análise adicional de base de comparação em relação aos resultados centrados em LLM.

Tabela 1 – Consultas utilizadas para busca de vídeos (adaptadas de (ROCHA; MAIA, 2023)).

APIs	**Domínios**	**Consultas**
Spring	Desenvolvimento Web	How to upload image java spring
Java Persistence API (JPA)	Persistência	How to implement CRUD operations java JPA
Selenium	Testes Web	How to search a product item on an e-commerce web application java selenium
JavaFX	Interface Desktop	How to implement menu java javafx
TensorFlow	Aprendizado de Máquina	How to build a neural network java tensorflow
OpenCV	Visão Computacional	How to classify image java opencv
JUnit	Testes Unitários	How to implement matchers test java junit
Jackson	Processamento de Dados Estruturados	How to process JSON data java Jackson
LibGDX	Motor de Jogos	How to implement animation java libgdx
OpenGL	Computação Gráfica	How to draw 2D objects java opengl

4.2 Comparação segundo idioma e quantidade de tópicos

Esta seção agrega os padrões de correspondência entre tópicos humanos e tópicos de LLM, agrupados por modelo, idioma e quantidade de tópicos. Primeiro, apresentamos

percentuais resumidos por modelo e, em seguida, analisamos com mais detalhe os resultados em inglês e em português.

As Tabelas 2, 3 e 4 apresentam percentuais agregados por categoria de correspondência (Muito Alta, Alta, Baixa, Muito Baixa) para cada modelo, separados por idioma (PT/EN) e quantidade de tópicos (5/7/10). Os percentuais por categoria são calculados somando todas as ocorrências dos padrões de correspondência correspondentes. Para complementar esses resumos numéricos, as Figuras 6–9 fornecem uma comparação visual compacta entre idiomas e quantidades de tópicos.

	5 tópicos (PT)	5 tópicos (EN)	7 tópicos (PT)	7 tópicos (EN)	10 tópicos (PT)	10 tópicos (EN)
Muito Alta (%)	43.75%	38.77%	44.82%	48.33%	50%	48.52%
Alta (%)	29.15%	32.64%	32.75%	24.99%	31.07%	26.46%
Baixa (%)	0%	2.04%	1.72%	1.67%	0%	2.94%
Muito Baixa (%)	27.06%	26.52%	20.72%	25%	18.9%	22.05%
Total (%)	100%	100%	100%	100%	100%	100%

Tabela 2 – Tabela comparativa para GPT-4.

	5 tópicos (PT)	5 tópicos (EN)	7 tópicos (PT)	7 tópicos (EN)	10 tópicos (PT)	10 tópicos (EN)
Muito Alta (%)	25.58%	36.73%	47.28%	32.72%	45%	51.39%
Alta (%)	39.53%	28.56%	23.63%	32.72%	28.34%	27.78%
Baixa (%)	0%	4.08%	1.82%	5.46%	5%	0%
Muito Baixa (%)	34.87%	30.6%	27.26%	29.09%	21.67%	20.78%
Total (%)	100%	100%	100%	100%	100%	100%

Tabela 3 – Tabela comparativa para GPT-3.5.

	5 tópicos (PT)	5 tópicos (EN)	7 tópicos (PT)	7 tópicos (EN)	10 tópicos (PT)	10 tópicos (EN)
Muito Alta (%)	30%	39.58%	36.84%	43.54%	38.46%	50.70%
Alta (%)	36%	27.21%	33.31%	27.41%	30.76%	23.94%
Baixa (%)	2%	0%	1.75%	1.61%	0%	0%
Muito Baixa (%)	32%	33.33%	28.06%	27.4%	30.79%	25.27%
Total (%)	100%	100%	100%	100%	100%	100%

Tabela 4 – Tabela comparativa para Gemini.

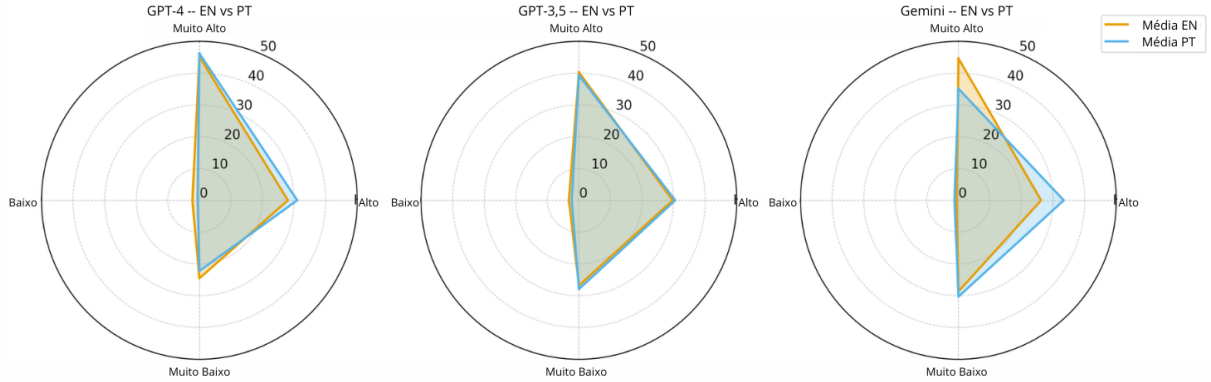

Figura 6 – Gráficos radar comparando a distribuição média das avaliações de correspondência (Muito Alta, Alta, Baixa, Muito Baixa) para GPT-4, GPT-3.5 e Gemini em inglês (EN) e português (PT).

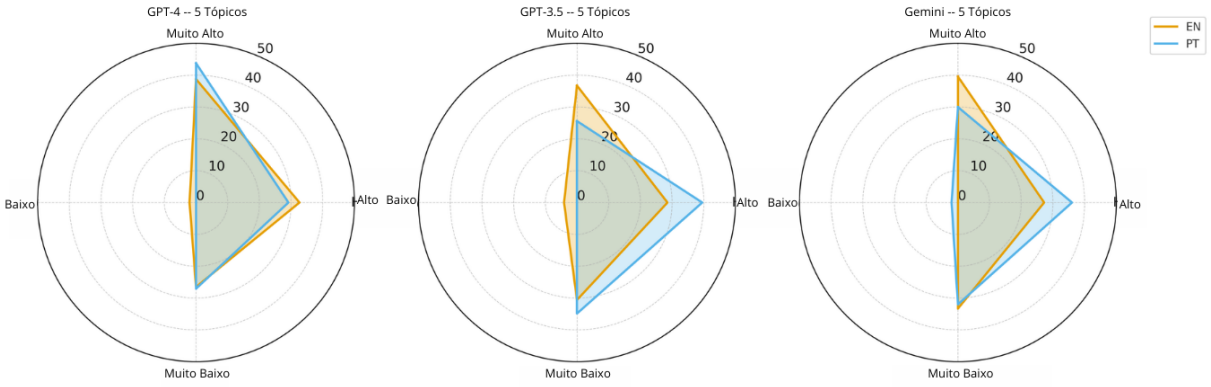

Figura 7 – Gráficos radar comparando a distribuição de correspondências Muito Alta, Alta, Baixa e Muito Baixa entre inglês (EN) e português (PT) para GPT-4, GPT-3.5 e Gemini quando são solicitados 5 tópicos.

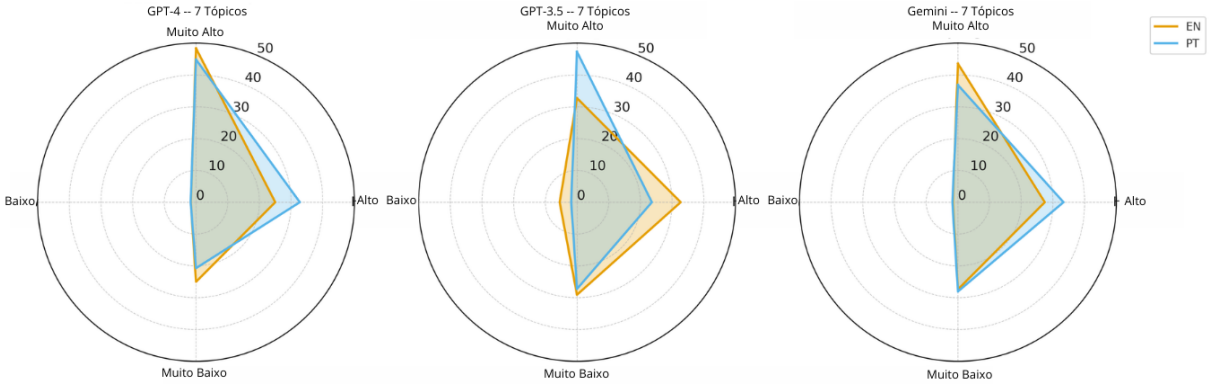

Figura 8 – Gráficos radar comparando a distribuição de correspondências Muito Alta, Alta, Baixa e Muito Baixa entre inglês (EN) e português (PT) para GPT-4, GPT-3.5 e Gemini quando são solicitados 7 tópicos.

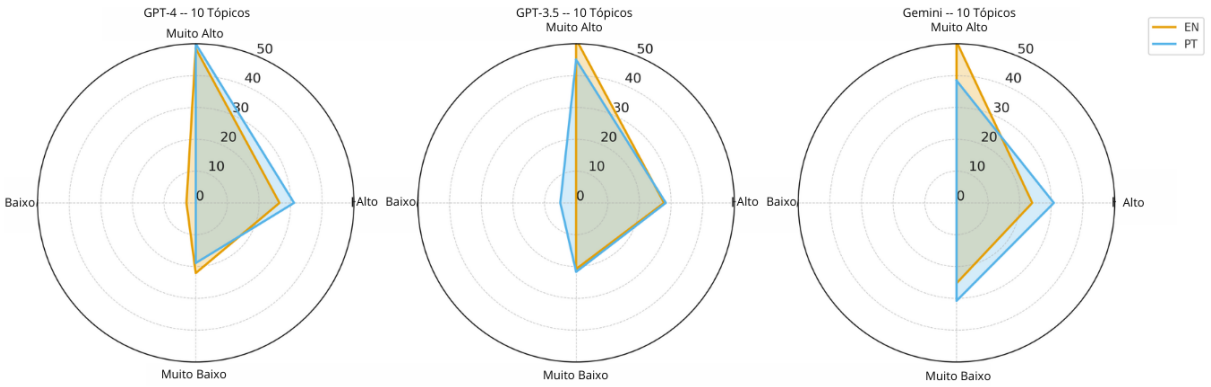

Figura 9 – Gráficos radar comparando a distribuição de correspondências Muito Alta, Alta, Baixa e Muito Baixa entre inglês (EN) e português (PT) para GPT-4, GPT-3.5 e Gemini quando são solicitados 10 tópicos.

Resumo visual por meio de gráficos radar.

As Figuras 6 à 9 fornecem um resumo visual compacto de como os tópicos gerados por cada modelo se distribuem nas quatro categorias de correspondência em inglês (EN) e português (PT). De modo geral, o GPT-4 apresenta inclinação consistente em direção aos eixos *Muito Alta* e *Alta*, com componentes *Muito Baixa* comparativamente menores nas diferentes quantidades de tópicos, o que está de acordo com a predominância agregada de tópicos bem alinhados relatada nas Tabelas 2–4. O GPT-3.5 segue um padrão semelhante, embora com maior sensibilidade à quantidade de tópicos. O Gemini tende a apresentar a maior região de *Muito Baixa* e maior variabilidade entre idiomas, particularmente quando mais tópicos são solicitados (Figuras 8–9). Em todos os modelos, o eixo *Baixa* permanece próximo de zero, sugerindo que as correspondências raramente caem em casos limítrofes: quando os modelos se alinham ao avaliador, tendem a se enquadrar em *Muito Alta* ou *Alta*, e quando falham, a discordância geralmente é grande o suficiente para ser classificada como *Muito Baixa*.

4.2.1 Análise em inglês

Esta subseção examina os padrões detalhados de correspondência para os vídeos em inglês, destacando como a quantidade de tópicos e a escolha do modelo influenciam alinhamentos exatos, parciais e fracos.

Análise complementar (inglês).

Para os vídeos em inglês, aumentar a quantidade de tópicos de 5 para 10 melhora sistematicamente a cobertura e o alinhamento (Tabela 5 e Figuras 7–9). Essa melhora é visualmente refletida pela expansão dos polígonos radar em direção a *Muito Alta*/*Alta* e pela redução do componente *Muito Baixa*, especialmente com 10 tópicos (Figura 9). Com 10 tópicos, as correspondências exatas 1–1 atingem o pico no *Gemini* (49.29%), com *GPT-3.5* (47.22%) e *GPT-4* (44.11%) logo atrás (Tabela 5). Quando os pares 2–2 e 3–3 também são considerados, a categoria "Muito Alta" representa aproximadamente metade ou mais dos alinhamentos na maioria das configurações, indicando que quantidades maiores de tópicos ajudam os modelos a convergir para a estrutura conceitual do avaliador.

Diferença de granularidade. Correspondências "Altas" (1–2 e 2–1) continuam frequentes, especialmente com 10 tópicos (Tabela 5). A maior parte dos casos 1–2 corresponde a um tema humano sendo decomposto de forma sensata em subtópicos pelo LLM (por exemplo, separando configuração de ambiente, detalhes de implementação e etapas de teste). Do ponto de vista do usuário, esses alinhamentos parciais ainda são úteis, pois expõem subcomponentes que podem apoiar navegação mais refinada ou filtragem do conteúdo do vídeo.

	GPT 4.0			GPT 3.5			Gemini		
(X-Y)	**5 tópicos**	**7 tópicos**	**10 tópicos**	**5 tópicos**	**7 tópicos**	**10 tópicos**	**5 tópicos**	**7 tópicos**	**10 tópicos**
1-1	17 (34.69%)	23 (38.33%)	30 (44.11%)	17 (34.69%)	17 (30.9%)	34 (47.22%)	18 (37.5%)	25 (40.32%)	35 (49.29%)
2-2	2 (4.08%)	6 (10%)	3 (4.41%)	1 (2.04%)	0	3 (4.17%)	1 (2.08%)	2 (3.22%)	1 (1.41%)
3-3	0	0	0	0	1 (1.82%)	0	0	0	0
1-2	3 (6.12%)	5 (8.33%)	11 (16.18%)	2 (4.08%)	9 (16.36%)	13 (18.05%)	3 (6.38%)	10 (16.13%)	14 (19.72%)
2-1	7 (14.28%)	5 (8.33%)	3 (4.41%)	6 (12.24%)	3 (5.45%)	2 (2.78%)	8 (16.67%)	5 (8.06%)	3 (4.22%)
2-3	0	0	1 (1.47%)	0	1 (1.82%)	0	0	0	0
3-2	1 (2.04%)	2 (3.33%)	0	1 (2.04%)	3 (5.45%)	0	0	0	0
1-3	0	1 (1.67%)	2 (2.94%)	0	0	1 (1.39%)	1 (2.08%)	0	0
2-4	0	0	0	0	0	1 (1.39%)	0	0	0
3-1	5 (10.20%)	2 (3.33%)	0	5 (10.2%)	2 (3.64%)	3 (4.17%)	0	2 (3.22%)	0
4-2	0	0	0	0	0	0	0	0	0
5-3	0	0	1 (1.47%)	0	0	0	1 (2.08%)	0	0
2-5	0	0	1 (1.47%)	0	1 (1.82%)	0	0	0	0
4-1	1 (2.04%)	1 (1.67%)	0	0	2 (3.64%)	0	0	1 (1.61%)	0
6-3	0	0	1 (1.47%)	1 (2.04%)	0	0	0	0	0
8-5	0	0	0	1 (2.04%)	0	0	0	0	0
1-6	0	0	0	0	0	0	0	0	1 (1.41%)
0-1	6 (12.24%)	7 (11.67%)	6 (8.82%)	6 (12.24%)	5 (9.09%)	3 (4.17%)	3 (6.25%)	6 (9.68%)	1 (1.41%)
0-2	0	2 (3.33%)	2 (2.94%)	0	3 (5.45%)	5 (6.94%)	1 (2.08%)	2 (3.22%)	4 (5.63%)
0-3	0	0	1 (1.47%)	0	0	1 (1.39%)	0	1 (1.61%)	3 (4.22%)
0-4	0	0	1 (1.47%)	0	0	1 (1.39%)	2 (4.17%)	0	2 (2.74%)
1-0	4 (8.16%)	3 (5%)	4 (5.88%)	7 (14.28%)	5 (9.09%)	4 (5.5%)	3 (6.25%)	3 (4.84%)	1 (1.41%)
2-0	0	1 (1.67%)	1 (1.47%)	1 (2.04%)	1 (1.82%)	0	3 (6.25%)	2 (3.22%)	4 (5.63%)
3-0	2 (4.08%)	2 (3.33%)	0	0	1 (1.82%)	1 (1.39%)	1 (2.08%)	2 (3.22%)	0
4-0	1 (2.04%)	0	0	1 (2.04%)	1 (1.82%)	0	2 (4.17%)	0	1 (1.41%)
5-0	0	0	0	0	0	0	1 (2.08%)	0	0
6-0	0	0	0	0	0	0	0	1 (1.61%)	0
7-0	0	0	0	0	0	0	0	0	1 (1.41%)
Total	49	60	68	49	55	72	48	62	71

Tabela 5 – Inglês / X–Y: X representa os tópicos identificados pelo avaliador; Y representa os tópicos identificados pelo LLM.

Modos de falha. A cauda longa de correspondências Muito Baixas (0–k e k–0) se torna menor com 10 tópicos, mas não desaparece (Tabela 5). Os erros residuais incluem: (i) **dispersão de tópicos**, em que um tema humano minoritário é espalhado por muitos tópicos finos do LLM, reduzindo a cobertura de outros temas; (ii) rótulos de **estrutura genérica** (por exemplo, "Introduction", "Next steps") que refletem a retórica do vídeo em vez do conteúdo técnico; e (iii) **ruído da transcrição**, que pode levar os modelos a detalhes irrelevantes. Esses problemas são relativamente raros, mas ainda importantes na interpretação de incompatibilidades.

Implicações práticas. Para conteúdo em inglês, a configuração com 10 tópicos é uma escolha razoável quando cobertura e navegabilidade são prioridades (Tabela 5 e Figura 9). Na prática, uma etapa leve de pós-processamento pode filtrar termos estruturais genéricos e fundir rótulos quase duplicados resultantes de padrões 1–2/2–1. Em comparação com a base de comparação com LDA (Seção 4.3), que frequentemente produz agrupamentos de palavras-chave que exigem interpretação manual, modelos baseados em transformer já fornecem rótulos coerentes e orientados ao usuário, mais próximos da intenção humana.

4.2.2 Análise em português

Esta subseção analisa os padrões de correspondência para os vídeos em português, enfatizando como os modelos se comportam em um cenário não inglês e como a quantidade de tópicos afeta a qualidade do alinhamento.

	GPT 4.0			GPT 3.5			Gemini		
(X-Y)	**5 tópicos**	**7 tópicos**	**10 tópicos**	**5 tópicos**	**7 tópicos**	**10 tópicos**	**5 tópicos**	**7 tópicos**	**10 tópicos**
1-1	21 (43.75%)	23 (39.65%)	37 (50%)	9 (20.93%)	24 (43.64%)	26 (43.33%)	15 (30%)	20 (35.09%)	23 (35.38%)
2-2	0	3 (5.17%)	0	2 (4.65%)	1 (1.82%)	1 (1.67%)	0	1 (1.75%)	2 (3.08%)
3-3	0	0	0	0	1 (1.82%)	0	0	0	0
1-2	2 (4.16%)	5 (8.62%)	10 (13.51%)	4 (9.3%)	1 (1.82%)	12 (20%)	6 (12%)	8 (14.03%)	8 (12.3%)
2-1	7 (14.58%)	7 (12.07%)	7 (9.46%)	5 (11.63%)	6 (10.9%)	4 (6.67%)	6 (12%)	5 (8.77%)	6 (9.23%)
2-3	0	3 (5.17%)	0	0	0	0	0	0	1 (1.54%)
3-2	0	0	0	1 (2.32%)	0	0	3 (6%)	2 (3.5%)	0
3-4	0	0	0	0	1 (1.82%)	0	0	0	0
1-3	0	1 (1.72%)	3 (4.05%)	0	1 (1.82%)	0	0	1 (1.75%)	3 (4.61%)
2-4	0	0	1 (1.35%)	0	1 (1.82%)	0	0	0	1 (1.54%)
3-1	5 (10.41%)	3 (5.17%)	2 (2.7%)	7 (16.28%)	3 (5.45%)	1 (1.67%)	2 (4%)	3 (5.26%)	1 (1.54%)
4-2	0	0	0	0	0	0	1 (2%)	0	0
1-4	0	1 (1.72%)	0	0	0	2 (3.33%)	0	1 (1.75%)	0
3-6	0	0	0	0	0	1 (1.67%)	0	0	0
4-1	0	0	0	0	1 (1.82%)	0	0	0	0
5-2	0	0	0	0	0	0	1 (2%)	0	0
1-5	0	0	0	0	0	0	0	0	1 (1.54%)
5-1	2 (4.16%)	0	0	1 (2.32%)	0	1 (1.67%)	0	1 (1.75%)	1 (1.54%)
1-6	0	0	1 (1.35%)	0	0	0	0	0	0
0-1	3 (6.25%)	1 (1.72%)	3 (4.05%)	7 (16.28%)	5 (9.09%)	2 (3.33%)	4 (8%)	4 (7.02%)	2 (3.08%)
0-2	1 (2.08%)	3 (5.17%)	4 (5.4%)	1 (2.32%)	3 (5.45%)	2 (3.33%)	2 (4%)	3 (5.26%)	1 (1.54%)
0-3	0	0	0	0	0	1 (1.67%)	0	0	2 (3.08%)
0-4	0	0	0	0	0	2 (3.33%)	0	0	1 (1.54%)
0-5	0	0	1 (1.35%)	0	0	0	0	0	0
0-6	0	0	0	0	0	0	0	0	2 (3.08%)
1-0	5 (10.41%)	5 (8.62%)	3 (4.05%)	1 (2.32%)	3 (5.45%)	1 (1.67%)	3 (6%)	2 (3.5%)	6 (9.23%)
2-0	1 (2.08%)	1 (1.72%)	2 (2.7%)	3 (6.98%)	4 (7.27%)	4 (6.67%)	4 (8%)	6 (10.53%)	2 (3.08%)
3-0	1 (2.08%)	2 (3.49%)	0	2 (4.65%)	0	0	1 (2%)	0	2 (3.08%)
4-0	0	0	0	0	0	0	1 (2%)	0	0
5-0	0	0	0	0	0	0	1 (2%)	0	0
Total	48	58	74	43	55	60	50	57	65

Tabela 6 – Português / X–Y: X representa os tópicos identificados pelo avaliador; Y representa os tópicos identificados pelo LLM.

Análise complementar (português).

Para os vídeos em português, o *GPT-4* se destaca, alcançando 50% de correspondências 1–1 com 10 tópicos (Tabela 6). Essa vantagem também é aparente nas comparações radar, nas quais o GPT-4 mostra uma inclinação mais forte em direção a *Muito Alta/Alta* e um componente *Muito Baixa* reduzido em quantidades maiores de tópicos (Figuras 8–9). Quando os padrões 2–2 e 3–3 também são considerados, a categoria "Muito Alta" se torna dominante, enquanto as correspondências Muito Baixas diminuem à medida que a quantidade de tópicos cresce (Tabelas 2 e 6). O *GPT-3.5* apresenta desempenho compe-

titivo e se beneficia de quantidades maiores, enquanto o *Gemini* melhora com 10 tópicos, mas ainda exibe a maior cauda de tópicos pouco alinhados.

Padrões de granularidade. Assim como em inglês, as correspondências 1–2 e 2–1 capturam diferenças benignas de granularidade e representam uma parcela substancial dos alinhamentos (Tabela 6). Elas normalmente ocorrem quando o modelo divide de maneira sensata um único tema do avaliador em subtópicos, como separar configuração de ambiente de implementação, testes e integração. Para uso posterior, agrupar esses tópicos semelhantes pode melhorar a capacidade de leitura rápida sem perder precisão.

Erros típicos. Três problemas recorrentes aparecem em português: (i) **tópicos estruturais** que refletem a estrutura retórica do vídeo (por exemplo, "Objetivo do vídeo", "Conclusão", "Agradecimentos"), especialmente no *Gemini* para temas como Selenium; (ii) **dispersão** de um tema humano minoritário em muitos rótulos finos (1–5 ou 1–6), o que dilui a cobertura de outros temas relevantes; e (iii) **subcobertura** ocasional (k–0), quando múltiplos tópicos do avaliador são espalhados em rótulos amplos e genéricos do LLM. Aumentar para 10 tópicos ajuda a mitigar os problemas (ii) e (iii), dando aos modelos mais espaço para separar conceitos.

Implicações práticas. Para canais ou públicos em português, o *GPT-4* com 10 tópicos é uma escolha forte quando recordação e navegabilidade são importantes (Tabela 6 e Figura 9). Na prática, uma pequena lista de bloqueio pode ser usada para remover termos estruturais da saída, e rótulos irmãos decorrentes de correspondências 1–2/2–1 podem ser fundidos em títulos mais descritivos. Em comparação com a base de comparação com LDA (Seção 4.3), que exibe uma alta taxa de *Disagreement* em português (Tabela 7), as saídas de LLM são mais imediatamente acionáveis como descritores para o usuário final e se alinham melhor com julgamentos humanos em diferentes APIs e tarefas.

4.3 Avaliação de LDA pelos avaliadores

Esta seção apresenta uma análise adicional de base de comparação utilizando LDADE, baseada na consistência de interpretação entre avaliadores.

Para cada vídeo, a base de comparação gerou 5 e 10 tópicos baseados em palavras-chave. Dois avaliadores independentes interpretaram individualmente cada tópico gerado. Quando um tópico não podia ser interpretado de maneira confiável, ele era marcado como indeterminado.

Após a etapa independente, o avaliador principal comparou diretamente ambas as interpretações tópico por tópico. *Agreement* indica interpretações semelhantes, *Partial* indica interpretações parcialmente semelhantes e *Disagreement* indica interpretações não semelhantes.

As Tabelas 7 e 8 apresentam os resultados da comparação entre avaliadores para os tópicos gerados por LDA em português e inglês, respectivamente, nas configurações com

5 e 10 tópicos.

Tabela 7 – Interpretações dos avaliadores para tópicos gerados por LDA (português).

Tema	Tópicos	Agreement	Partial	Disagreement
JavaFX	10	2	2	6
	5	3	0	2
JPA	10	2	2	6
	5	2	0	3
JSON	10	2	3	5
	5	1	0	4
JUnit	10	3	3	4
	5	0	2	3
LibGDX	10	0	3	7
	5	0	2	3
OpenCV	10	0	4	6
	5	0	1	4
OpenGL	10	0	0	10
	5	1	1	3
Selenium	10	2	0	8
	5	1	1	3
Spring	10	2	0	8
	5	1	0	4
TensorFlow	10	1	0	9
	5	1	0	4
Total	150	**23**	**18**	**109**
Percentual (%)	100%	**15.33%**	**12.00%**	**72.67%**

Tabela 8 – Interpretações dos avaliadores para tópicos gerados por LDA (inglês).

Tema	Tópicos	Agreement	Partial	Disagreement
OpenGL	10	1	0	9
	5	1	0	4
JSON	10	5	1	4
	5	2	0	3
JPA	10	2	2	6
	5	2	1	2
JavaFX	10	4	2	4
	5	4	0	1
OpenCV	10	5	3	2
	5	2	0	3
LibGDX	10	4	2	4
	5	3	1	1
JUnit	10	5	1	4
	5	4	0	1
TensorFlow	10	2	0	9
	5	0	0	5
Spring	10	1	0	9
	5	1	0	4
Selenium	10	1	0	9
	5	0	0	5
Total	150	**59**	**13**	**78**
Percentual (%)	100%	**39.33%**	**8.67%**	**52.00%**

4.4 Discussão dos resultados

Esta seção sintetiza os achados entre idiomas e entre modelos, relacionando quantidades de tópicos, padrões de alinhamento e a base de comparação com LDA ao uso prático da extração de tópicos baseada em LLM.

Efeito da quantidade de tópicos (5/7/10).

Aumentar a quantidade de tópicos para dez melhora consistentemente o alinhamento entre modelos e idiomas. Para GPT-4 (EN), a proporção de correspondências *Muito Alta* cresce de 38.77% (5 tópicos) para 48.52% (10 tópicos); em português, o alinhamento 1–1 sozinho atinge 50% com 10 tópicos (Tabelas 5 e 6). Em paralelo, os casos *Muito Baixa* (0–k/k–0) diminuem, e parte das incompatibilidades anteriores se converte em correspondências *Alta* (1–2 / 2–1), indicando cobertura mais ampla dos subtemas com deriva de granularidade aceitável.

Comparação entre modelos.

Em inglês, *Gemini* e *GPT-3.5* alcançam as maiores taxas 1–1 com 10 tópicos (49.29% e 47.22%), com *GPT-4* logo atrás (44.11%) (Tabela 5). Em português, *GPT-4* lidera (50% em 1–1 com 10 tópicos), enquanto *GPT-3.5* e *Gemini* melhoram com a quantidade maior, mas permanecem ligeiramente abaixo (Tabela 6). De modo geral, as diferenças entre os modelos diminuem à medida que a quantidade de tópicos aumenta, embora a

comparação entre esses três modelos deva ser interpretada dentro dos limites da amostra e do conjunto de modelos selecionados.

Padrões de erro.

Com apenas cinco tópicos, os casos de correspondência Muito Baixa (notadamente 0–Y, X–0.) são mais frequentes, refletindo omissões ou rótulos fora do alvo. Com dez tópicos, esses casos tendem a migrar para Alta (1–2, 2–1), o que é preferível para recordação e navegação do usuário, pois os subcomponentes do tema humano passam a ser capturados.

LDA versus LLMs.

A base de comparação com LDADE apresenta alta taxa de *Disagreement*, especialmente em português (72.67%), e apenas uma taxa moderada de *Agreement* em inglês (39.33%) (Tabelas 7 e 8). Em contraste, os LLMs produzem substancialmente mais correspondências *Muito Alta/Alta* (Tabelas 2, 3 e 4). Como as análises com LLM e LDADE utilizam procedimentos de avaliação diferentes (correspondência entre tópicos humanos versus convergência de interpretação entre avaliadores), essa comparação deve ser interpretada como evidência metodológica complementar, e não como uma equivalência estrita de métrica um-para-um. Mesmo sob essa interpretação cautelosa, a análise da base de comparação ainda sugere que os rótulos gerados por LLM estão mais alinhados semanticamente com julgamentos humanos e são mais diretamente utilizáveis para navegação em vídeos educacionais no cenário avaliado.

4.5 Ameaças à Validade

Esta seção resume ameaças internas, de construto e externas que podem afetar a interpretação e a generalização dos resultados, juntamente com estratégias de mitigação.

4.5.1 Validade Interna

Subjetividade relacionada ao avaliador. A comparação com LLM contou com um avaliador, enquanto a interpretação da base de comparação (LDADE) utilizou dois avaliadores independentes seguidos de uma verificação direta de convergência entre as análises. Embora isso reduza a subjetividade na análise da base de comparação, ainda pode haver ambiguidade interpretativa. *Mitigação:* diretrizes escritas e comparação lado a lado das interpretações dos avaliadores; trabalhos futuros devem relatar métricas formais de concordância entre avaliadores manuais para análise das LLMs (por exemplo, Cohen's κ).

Ruído da transcrição. As legendas automáticas do YouTube podem conter erros de reconhecimento e pontuação. *Mitigação:* limpeza leve e remoção de artefatos; um estudo posterior pode comparar transcrições manuais versus automáticas.

4.5.2 Validade de Construto

Métrica baseada em correspondência. Nossa avaliação enfatiza padrões de alinhamento (por exemplo, 1–1, 1–2) em vez de métricas da área de Recuperação de Informação. Embora capture assimetrias de granularidade e de omissão/*commission*, ela ainda é uma aproximação de utilidade. *Mitigação:* percentuais agregados e tabelas de contingência detalhadas (Seções 4.2 e 4.3).

Efeito da quantidade de tópicos. Solicitar 5/7/10 tópicos altera a granularidade e, consequentemente, a distribuição das correspondências. *Mitigação:* relatar as três quantidades e analisar explicitamente esse efeito.

Configuração da base de comparação. A qualidade de LDA depende de hiperparâmetros, pré-processamento e escolhas de otimização. Neste estudo, a base de comparação foi executada como LDA com ajuste baseado em Evolução Diferencial (LDADE), sob um *pipeline* consistente e com 5 e 10 tópicos. *Mitigação:* pré-processamento fixo e protocolo de avaliação fixo; trabalhos futuros podem testar estratégias adicionais de ajuste e outras bases de comparação clássicas.

4.5.3 Validade Externa

Tamanho e escopo da amostra. Os resultados são baseados em 20 vídeos em um contexto de desenvolvimento de *software* centrado em Java (Tabela 1). A generalização para outros domínios ou idiomas pode ser limitada. *Mitigação:* configuração bilíngue (PT/EN) e APIs diversificadas; trabalhos futuros devem expandir conjuntos de dados e temas.

Cobertura de plataforma e idioma. Os achados estão ligados a transcrições do YouTube e a dois idiomas. *Mitigação:* estender para outras plataformas e idiomas (por exemplo, MOOCs, espanhol).

Cobertura de modelos. A comparação experimental inclui três famílias de LLM (*GPT-4*, *GPT-3.5* e *Gemini*), o que fornece uma visão útil, mas ainda limitada, do comportamento atual dos modelos. Outras arquiteturas, provedores ou modelos de pesos abertos podem exibir padrões de alinhamento diferentes. *Mitigação:* trabalhos futuros devem ampliar o conjunto de modelos avaliados para reduzir vieses específicos de provedor ou modelo e fortalecer a robustez comparativa.

Mudanças nos modelos ao longo do tempo. LLMs hospedados podem sofrer atualizações ao longo do tempo, o que pode afetar a reprodutibilidade dos resultados. *Mitigação:* os modelos e *prompts* utilizados neste trabalho foram documentados e dispo-

nibilizados com os materiais do estudo; adicionalmente, trabalhos futuros podem priorizar versões fixas ou modelos com pesos abertos.

4.6 Conclusão

De modo geral, os resultados indicam que a extração de tópicos baseada em LLM alcança alinhamento semântico mais próximo com tópicos produzidos por humanos do que a análise complementar de base de comparação com LDADE no cenário avaliado. Aumentar a quantidade de tópicos para dez geralmente melhora as correspondências *Muito Alta*/*Alta* e reduz incompatibilidades, ao mesmo tempo em que diminui as diferenças de desempenho entre os modelos. Entre as configurações testadas, *GPT-4* e *Gemini* apresentam desempenho forte nos dois idiomas, com *GPT-4* mostrando resultados mais consistentes em ambos. Em vez de estabelecer uma regra geral, esses achados fornecem evidências de que LLMs podem produzir listas de tópicos úteis para navegação em conteúdo educacional em vídeo dentro do escopo deste estudo.

Capítulo 5

Conclusão

Este capítulo conclui a monografia retomando o problema prático apresentado no Capítulo 1: desenvolvedores de *software* dependem cada vez mais de vídeos educacionais longos que frequentemente oferecem poucos detalhes descritivos, dificultando a identificação rápida de quais tópicos são abordados. Dentro do escopo analisado, os resultados indicam que LLMs podem gerar listas de tópicos com alinhamento razoável ao julgamento humano e, portanto, podem servir como recursos úteis de navegação para a aprendizagem *online*. Entre as configurações avaliadas, solicitar dez tópicos tendeu a oferecer o melhor equilíbrio entre cobertura e granularidade útil, e o *GPT-4* alcançou os resultados mais consistentes, particularmente em português. Em contraste, a análise complementar utilizando LDA/LDADE como base de comparação mostrou menor convergência de interpretação entre avaliadores, o que reforça a percepção de que LLMs podem oferecer rótulos mais coerentes semanticamente e mais alinhados ao julgamento humano nesse contexto educacional baseado em transcrições.

5.1 Principais Contribuições

As principais contribuições deste trabalho são:

- **Avaliação de LLMs para extração de tópicos:** Este trabalho fornece evidências de que LLMs podem extrair tópicos relevantes de transcrições de vídeos educacionais em português e inglês. A análise comparativa em relação a uma referência humana indica alinhamento substancial, especialmente quando são solicitados dez tópicos por vídeo (seção 4.2).
- **Metodologia de avaliação baseada em correspondência:** Adotamos uma metodologia baseada em correspondência com quatro níveis (*Muito Alta*, *Alta*, *Baixa*, *Muito Baixa*), que captura diferenças de granularidade (por exemplo, 1–2, 2–1) e padrões de omissão/*commission* (por exemplo, 0–1, 1–0). Isso fornece uma visão mais detalhada da qualidade do alinhamento em diferentes configurações.

- ❑ **Comparação com métodos tradicionais (LDA/LDADE):** Como base de comparação complementar, o LDA com ajuste por Evolução Diferencial (LDADE) apresentou menor convergência de interpretação entre avaliadores, com taxas mais altas de *Disagreement*, particularmente em português (seção 4.3). Esse contraste sugere vantagens de LLMs em coerência e compreensão contextual para conteúdo educacional no cenário avaliado.

- ❑ **Avaliação multilíngue:** A configuração bilíngue (português/inglês) indica que LLMs permanecem úteis em diferentes idiomas, com ganhos mais consistentes quando a quantidade de tópicos é ampliada para dez.

De modo geral, os resultados reforçam a utilidade da extração de tópicos baseada em LLMs para transcrições de vídeos educacionais dentro do escopo analisado, ao mesmo tempo em que indicam que o desempenho depende de fatores como quantidade de tópicos, idioma e escolha do modelo.

5.2 Trabalhos Futuros

Com base nas limitações discutidas em seção 4.5, apontamos as seguintes direções para pesquisas futuras:

- ❑ **Adaptação específica de domínio:** Aplicar *fine-tuning* ou *instruction tuning* em LLMs com conjuntos de dados educacionais de engenharia de software (por exemplo, tutoriais de APIs e explicações guiadas de código) para aumentar ainda mais a precisão e a estabilidade dos rótulos de tópicos.

- ❑ **Sinais multimodais:** Incorporar pistas visuais (slides, código exibido em tela, diagramas) e metadados do vídeo com modelos multimodais para complementar os sinais obtidos apenas da transcrição.

- ❑ **Extração em tempo real:** Explorar a extração de tópicos *online*/*streaming* para aulas ou transmissões ao vivo, permitindo recursos imediatos de navegação para aprendizes.

- ❑ **Métricas de avaliação:** Complementar a abordagem baseada em correspondência com medidas baseadas em tarefas (por exemplo, tempo para localizar um tópico) e métricas de similaridade semântica; incluir múltiplos avaliadores na etapa de correspondência com LLM e relatar medidas formais de concordância entre avaliadores.

- ❑ **Exploração da quantidade de tópicos:** Realizar uma análise mais ampla para determinar qual quantidade de tópicos é mais apropriada para essa tarefa, aumentando gradualmente o número de tópicos solicitados e examinando como alinhamento, granularidade e usabilidade variam entre as configurações. Isso pode ajudar

a identificar uma faixa mais adequada de tópicos para navegação em vídeos educacionais.

- ❑ **Cobertura e eficiência dos modelos:** Ampliar o conjunto de LLMs avaliados para reduzir vieses específicos de modelo e obter uma perspectiva comparativa mais ampla. Além disso, estudos futuros podem incorporar uma análise temporal do tempo de resposta entre modelos, investigando a relação entre qualidade de alinhamento, latência e usabilidade prática.

- ❑ **Integração com LMS:** Integrar a extração de tópicos a Sistemas de Gestão da Aprendizagem (*Learning Management Systems*) para apoiar curadoria de conteúdo, indexação e trilhas de aprendizagem personalizadas.

- ❑ **Evolução de engenharia do protótipo:** Implantar o *backend* em um ambiente remoto, disponibilizar LLMs adicionais como opções selecionáveis na interface e incorporar marcações temporais por tópico para melhorar a navegação refinada em vídeos longos.

5.3 Contribuições Acadêmicas e Divulgação

- ❑ **Pôster:** *Topic Extraction from Software Development-Related Video Transcriptions using LLMs* (Título em Português: *Extração de Tópicos a Partir de Transcrições de Vídeo Relacionados ao Desenvolvimento de Software por meio de LLMs*). XVIII Workshop de Teses e Dissertações em Ciência da Computação (WTDCC), FACOM TechWeek 2024, Universidade Federal de Uberlândia (UFU), Uberlândia–MG, Brasil, 24–25 Outubro 2024. **Menção Honrosa** (categoria de Graduação). Autores: Ian B. Seki; Marcelo de Almeida Maia; Carlos Eduardo Carvalho Dantas; Adriano Mendonça Rocha.

Em síntese, este trabalho fornece evidências empíricas de que a extração de tópicos baseada em transcrições com LLMs pode apoiar a navegação em contextos de aprendizagem voltados ao desenvolvimento de *software*. Ao mesmo tempo, os achados devem ser interpretados dentro dos limites do conjunto de dados, dos modelos e da metodologia avaliados. Esses resultados motivam avanços futuros em avaliação comparativa, análise da quantidade de tópicos e integração do protótipo.

Referências

ABDULLAH, M.; MADAIN, A.; JARARWEH, Y. Chatgpt: Fundamentals, applications and social impacts. Milan, Italy, p. 1–8, 2022. 4 citações na página 15, 16, 17 e 20.

AGRAWAL, A.; FU, W.; MENZIES, T. What is wrong with topic modeling? and how to fix it using search-based software engineering. **Information and Software Technology**, Elsevier, v. 98, p. 74–90, 2018. 4 citações na página 12, 13, 17 e 21.

ALBEER, R. A. et al. Automatic summarization of youtube video transcription text using term frequency-inverse document frequency. **Indonesian Journal of Electrical Engineering and Computer Science**, 2022. 2 citações na página 21 e 22.

BLEI, D.; NG, A.; JORDAN, M. Latent dirichlet allocation. **Journal of Machine Learning Research**, v. 3, p. 601–608, 2001. Citação encontrada na página 17.

BLEI, D. M. Probabilistic topic models. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 4, p. 7784, apr 2012. ISSN 0001-0782. 3 citações na página 17, 20 e 21.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **Journal of Machine Learning Research**, v. 3, p. 993–1022, 2003. 3 citações na página 17, 20 e 21.

BOU, G. et al. BOU-Guard: Uma abordagem para detecção de conteúdo impróprio na internet. **FACOM Universidade Federal de Uberlândia (UFU)**, Monte Carmelo, Brasil, 2023. Citação encontrada na página 15.

BROWN, T. et al. Language models are few-shot learners. **Advances in Neural Information Processing Systems**, p. 1877–1901, 2020. 3 citações na página 15, 16 e 19.

CHOWDHERY, A. et al. Palm: scaling language modeling with pathways. **J. Mach. Learn. Res.**, JMLR.org, v. 24, n. 1, mar 2024. ISSN 1532-4435. 2 citações na página 15 e 19.

CHRISTIAN, H.; AGUS, M. P.; SUHARTONO, D. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). **ComTech**, v. 7, n. 4, 2016. Citação encontrada na página 21.

COLAVITO, G. et al. Leveraging gpt-like llms to automate issue labeling. **arXiv preprint arXiv:2401.12345**, 2024. 3 citações na página 16, 22 e 23.

CONNEAU, A. et al. Unsupervised cross-lingual representation learning at scale. **arXiv preprint arXiv:1911.02116**, 2020. 2 citações na página 17 e 20.

DANTAS, C.; ROCHA, A.; MAIA, M. Assessing the readability of chatgpt code snippet recommendations: A comparative study. Association for Computing Machinery, New York, NY, USA, p. 283–292, 2023. 3 citações na página 16, 22 e 23.

DEVI, S. et al. Abstractive summarizer for youtube videos. In: **Proceedings of the International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022)**. [S.l.]: Atlantis Press, 2023. p. 431–438. ISBN 978-94-6463-136-4. ISSN 2352-538X. Citação encontrada na página 22.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. 2 citações na página 16 e 20.

EBERT, C.; LOURIDAS, P. Generative AI for software practitioners. **IEEE Software**, v. 40, p. 30–38, 07 2023. 4 citações na página 16, 20, 22 e 23.

HOWARD, J.; RUDER, S. Universal language model fine-tuning for text classification. In: **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. [S.l.: s.n.], 2018. p. 328–339. Citação encontrada na página 20.

JUNIOR, A. C. et al. Topic modeling for keyword extraction: using natural language processing methods for keyword extraction in portal min@s. **REVISTA DE ESTUDOS DA LINGUAGEM**, v. 23, n. 3, p. 695–726, 2015. ISSN 2237-2083. Citação encontrada na página 21.

KUUTILA, M. et al. Time pressure in software engineering: A systematic review. **Inf. Softw. Technol.**, Butterworth-Heinemann, USA, v. 121, n. C, maio 2020. ISSN 0950-5849. Disponível em: <https://doi.org/10.1016/j.infsof.2020.106257>. Citação encontrada na página 11.

LAWRENCE, S.; GILES, C. L. Searching the web: General and scientific information access. **IEEE Communications Magazine**, v. 37, n. 1, p. 116–122, 1999. Citação encontrada na página 11.

OCALLAGHAN, D. et al. Network analysis of recurring youtube spam campaigns. **Proceedings of the International AAAI Conference on Web and Social Media**, v. 6, n. 1, p. 531–534, Aug. 2021. Citação encontrada na página 22.

PIRES, T.; SCHWENK, H.; CORREIA, T. Multilingual bert: Monolingual performance in a multilingual model. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2019. p. 4996–5001. 2 citações na página 17 e 20.

ROCHA, A. M.; MAIA, M. A. Mining relevant solutions for programming tasks from search engine results. **IET Software**, p. 1–17, 2023. 3 citações na página 11, 37 e 38.

RUSH, A. M.; CHOPRA, S.; WESTON, J. A neural attention model for abstractive sentence summarization. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**. [S.l.], 2015. p. 379–389. Citação encontrada na página 21.

SIA, S.; DALMIA, A.; MIELKE, S. J. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In: WEBBER, B. et al. (Ed.). **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Online: Association for Computational Linguistics, 2020. p. 1728–1736. Citação encontrada na página 17.

SRIVASTAVA, A.; SUTTON, C. Autoencoding variational inference for topic models. In: **International Conference on Learning Representations**. [S.l.: s.n.], 2017. 2 citações na página 17 e 21.

SUDHAN, R.; VEDHAVIYASSH, D. R.; SARANYA, G. Learning to summarize youtube videos with transformers: A multi-task approach. Chennai, India, p. 1–6, 2023. Citação encontrada na página 22.

TOUVRON, H. et al. Llama: Open and efficient foundation language models. **ArXiv**, abs/2302.13971, 2023. Citação encontrada na página 15.

TUFANO, M. et al. Evaluating chatgpt for code summarization and quality assessment. In: **International Conference on Software Maintenance and Evolution**. [S.l.: s.n.], 2024. Citação encontrada na página 16.

TUFANO, R. et al. Unveiling chatgpt's usage in open source projects: A mining-based study. **Proceedings of the International Conference on Mining Software Repositories**, 2024. 3 citações na página 16, 22 e 23.

VASWANI, A. et al. Attention is all you need. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. p. 5998–6008. Citação encontrada na página 19.

WANG, H. et al. Prompting large language models for topic modeling. In: IEEE. **2023 IEEE International Conference on Big Data (BigData)**. [S.l.], 2023. p. 1236–1241. 4 citações na página 16, 17, 22 e 23.

WU, N. et al. Large language models are diverse role-players for summarization evaluation. In: LIU, F. et al. (Ed.). **Natural Language Processing and Chinese Computing**. Cham: Springer Nature Switzerland, 2023. p. 695–707. ISBN 978-3-031-44693-1. 2 citações na página 22 e 23.

XIAO, T. et al. DevGPT: Studying Developer-ChatGPT Conversations. In: **Proceedings of the International Conference on Mining Software Repositories (MSR 2024)**. [S.l.: s.n.], 2024. Citação encontrada na página 22.

YAFOOZ, W. M. S.; AL-DHAQM, A.; ALSAEEDI, A. Detecting kids cyberbullying using transfer learning approach: Transformer fine-tuning models. In: _____. **Kids Cybersecurity Using Computational Intelligence Techniques**. Cham: Springer International Publishing, 2023. p. 255–267. ISBN 978-3-031-21199-7. Citação encontrada na página 22.

ZHANG, T. et al. Benchmarking Large Language Models for News Summarization. **Transactions of the Association for Computational Linguistics**, v. 12, p. 39–57, 01 2024. ISSN 2307-387X. 3 citações na página 21, 22 e 23.

_____. Utilizing gpt-3 for summarization. In: **Proceedings of the 2024 Conference on Neural Information Processing**. [S.l.: s.n.], 2024. 2 citações na página 21 e 22.

ZUCCON, G.; KOOPMAN, B.; SHAIK, R. Chatgpt hallucinates when attributing answers. In: **Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region**. New York, NY, USA: Association for Computing Machinery, 2023. (SIGIR-AP '23), p. 4651. ISBN 9798400704086. 2 citações na página 16 e 17.

Apêndices

APÊNDICE **A**

Artefatos e materiais

Os artefatos utilizados neste trabalho estão disponíveis publicamente no Zenodo.
Link de acesso (DOI): <https://doi.org/10.5281/zenodo.17901591>
Página do registro: <https://zenodo.org/records/17901591>