Extração Automatizada de Tecnologias em Trabalhos Acadêmicos de Computação Utilizando LLM

Valéria Silva Santos[1], **Mirelly Barbosa Fontes de Oliveira**[1], **Breno Santana Santos**[2], **Marianne Batista Diniz da Silva**[1]

[1]Universidade Federal de Alagoas (UFAL) – AL – Brasil

[2]Universidade Federal do Rio Grande do Norte (UFRN) – RN – Brasil

{valeria.silva, mirelly.oliveira}@arapiraca.ufal.br

breno1005@hotmail.com, marianne.silva@penedo.ufal.br

Abstract. *Repositories of final projects function as indicators of technological demands in academic training, but manual mapping faces clear operational limitations. Understanding these dynamics is important to support curricular reforms and evaluate the mismatch between teaching and software development practices. This study developed and evaluated an automated method to extract and structure scientometric data from narrative textual abstracts. We propose a sequential Python pipeline integrating prompt engineering with strict data typing and post-extraction regular expressions for exception handling. The validation was conducted through a batch computational experiment over a dataset with 904 historical records from the computing department at UFAL. The results discriminate the chronological frequencies and application domains of the most recurrent programming languages, frameworks, and databases over two decades. It is concluded that integrating language models with low-creativity parameters constitutes a reproducible and scalable framework for computational scientometrics.*

Keywords: *Information Extraction. Large Language Models. Scientometrics. Prompt Engineering.*

Resumo. *Os repositórios de TCCs operam como indicadores das demandas tecnológicas na formação acadêmica, mas seu mapeamento manual enfrenta limitações operacionais. Compreender essas dinâmicas é importante para subsidiar reformas curriculares e avaliar o descompasso entre o ensino e as práticas de mercado. Este trabalho desenvolveu e avaliou um método automatizado para extrair e estruturar dados cienciométricos a partir de resumos textuais. Propõe-se um pipeline sequencial em Python que integra engenharia de prompt com tipagem estrita de dados e expressões regulares pós-extração para o tratamento de exceções. A validação ocorreu por meio de um experimento computacional em lote sobre um conjunto de dados com 904 registros históricos da área de computação da UFAL. Os resultados discriminam as frequências cronológicas e domínios de aplicação das linguagens, frameworks e bancos de dados mais recorrentes ao longo de duas décadas. Conclui-se que a integração de modelos de linguagem com baixa criatividade constitui um arcabouço reprodutível e escalável para a cienciometria computacional.*

Palavras-chave: *Extração de Informação. Modelos de Linguagem de Grande Escala. Cienciometria. Engenharia de Prompts.*

1. Introdução

A rápida e constante evolução das tecnologias da informação impõe às instituições de ensino superior o desafio de manter seus currículos e projetos de graduação alinhados às demandas científicas e de mercado [Goulart et al. 2021]. No ecossistema acadêmico, os Trabalhos de Conclusão de Curso (TCCs) atuam como um reflexo desse alinhamento, materializando a aplicação prática de conceitos teóricos em problemas reais da sociedade e da indústria [Rahman et al. 2025].

Deste modo, o campo da computação testemunhou transições de paradigmas, variando desde a consolidação de arquiteturas web tradicionais e sistemas embarcados até a recente ascensão da Inteligência Artificial (IA) e do Processamento de Linguagem Natural (PLN) [Chen et al. 2024]. O avanço dessas subáreas introduziu novos mecanismos para o tratamento, análise e modelagem automatizada de dados textuais de grande complexidade [do Nascimento 2024].

Com o advento e a popularização dos Grandes Modelos de Linguagem (*Large Language Models* — LLMs), as tarefas de PLN e a extração de informações a partir de *corpus* não estruturados atingiram novos patamares de viabilidade técnica [Khan et al. 2022]. Modelos fundamentais, treinados em volumes massivos de dados, demonstram uma capacidade avançada de mapeamento de contextos e processamento de estruturas linguísticas complexas que antes demandavam arquiteturas altamente especializadas e rígidas [Rosa and da Cruz 2026].

No cenário institucional da Universidade Federal de Alagoas (UFAL), o volume acumulado de produções acadêmicas ao longo dos anos constitui dados cienciométricos. Esses documentos guardam o registro histórico das escolhas tecnológicas dos discentes, como linguagens de programação, *frameworks*, sistemas de gerenciamento de bancos de dados e ferramentas de engenharia de *software*, fornecendo um panorama indireto sobre a evolução do próprio ensino de computação na instituição.

Apesar da riqueza informativa contida nos repositórios institucionais de TCCs, esses dados encontram-se em formatos não estruturados (como arquivos PDF), dispersos em resumos textuais narrativos. O mapeamento e a catalogação desse panorama tecnológico para fins de auditoria curricular ou análise de tendências enfrentam um gargalo operacional [Xu et al. 2024]. A análise manual de centenas de documentos acadêmicos demanda um tempo expressivo por parte de pesquisadores e gestores educacionais, além de ser um processo intrinsecamente sujeito a inconsistências operacionais, ambiguidades de interpretação e fadiga humana [Wang et al. 2020].

Por outro lado, abordagens tradicionais de mineração de texto baseadas em buscas por correspondência de palavras-chave (*keyword matching*) mostram-se insuficientes [Khan et al. 2024]. Elas falham em capturar variações sinônimas, erros de digitação ou o contexto sintático em que uma ferramenta é mencionada (por exemplo, diferenciar se uma tecnologia foi utilizada ou apenas citada como trabalho relacionado) [Asudani et al. 2023]. Diante desse cenário, emerge a necessidade de compreender de que maneira e com qual nível de consistência estrutural os LLMs podem ser integrados a um pipeline automatizado para extrair, categorizar e normalizar competências tecnológicas declaradas em resumos acadêmicos de computação [Dagdelen et al. 2024].

Assim, o objetivo deste trabalho consiste em desenvolver, implementar e ava-

liar um pipeline automatizado baseado em LLM para extrair e estruturar dados cienciométricos sobre tecnologias referenciadas nos resumos de TCCs da área de computação da UFAL. Para isso, viabiliza-se a identificação de linguagens de programação, *frameworks*, bancos de dados e ferramentas auxiliares por meio de uma estratégia de engenharia de *prompt* com tipagem estrita e rotinas pós-extração para garantir a integridade das respostas.

Para a consecução e exposição desta pesquisa, o presente documento está organizado em seções subsequentes. A Seção 2 revisa os Trabalhos Relacionados, estabelecendo um comparativo entre os métodos tradicionais de mapeamento cienciométrico e as abordagens baseadas em modelos de linguagem. A Seção 3 apresenta a abordagem proposta, detalhando a preparação do conjunto de dados, as estratégias de engenharia de *prompt* e as rotinas de tratamento de exceções. A Seção 4 aborda os resultados e discussões, convertendo os dados estruturados pelo pipeline em indicadores estatísticos quantitativos e análises cronológicas. Por fim, a Seção 5 condensa as considerações finais do estudo, sintetizando as contribuições alcançadas e os desdobramentos para trabalhos futuros.

2. Trabalhos Relacionados

O mapeamento automatizado de acervos acadêmicos para fins de análise cienciométrica tem sido investigado sob diferentes abordagens de extração de informação [Nowakowska 2025]. Tradicionalmente, as iniciativas concentram-se no emprego de algoritmos de mineração de texto baseados em correspondência exata de cadeias de caracteres ou em dicionários previamente estruturados [Pandey and Taruna 2021]. Esse padrão metodológico é observado em estudos de mapeamento de repositórios educacionais institucionais, nos quais a identificação de termos apoia-se em buscas indexadas por palavras-chave predefinidas [Casagrande and da Silva 2024]. Contudo, essas arquiteturas exibem limitações para capturar variações sinônimas ou identificar ferramentas que não estejam explicitamente cadastradas no dicionário do sistema [Fujita et al. 2024].

Uma abordagem alternativa para a catalogação de competências tecnológicas baseia-se na aplicação de técnicas de PLN orientadas ao Reconhecimento de Entidades Nomeadas (NER) [Herandi et al. 2024]. No cenário de análise de produções científicas regionais, como nos ecossistemas avaliados no âmbito da Escola Regional de Informática (ERI), o emprego de modelos de aprendizado de máquina supervisionados demonstrou capacidade de categorização de termos textuais, embora sob a restrição de demandar conjuntos de dados massivos previamente anotados por especialistas [Zdravković 2021]. O uso de LLM operando via interface de programação de aplicações *(Application Programming Interface* — API) passou a ser explorado para mitigar essa dependência de rotulagem manual, deslocando a ênfase do pipeline para a engenharia de *prompts* operacionais [do Nascimento 2024].

Investigações comparativas sobre o desempenho de LLMs em tarefas de extração estruturada indicam que a formulação de restrições semânticas e a definição de personas nos *prompts* reduzem a variabilidade das respostas textuais. Em análises cienciométricas de escopo regional, a consistência dos dados gerados por modelos de linguagem depende diretamente de estratégias de controle de qualidade na entrada do pipeline [Neves et al. 2024]. Todavia, a literatura aponta que o tratamento de falhas sintáticas ou omissões de chaves na saída dos modelos ainda constitui um gargalo para a automação

completa dos fluxos de processamento em lote [Younesi et al. 2025].

O presente trabalho diferencia-se das abordagens identificadas na literatura ao propor um pipeline que integra a extração via API sob temperatura zero a uma rotina de normalização pós-extração baseada em expressões regulares. Enquanto os trabalhos relacionados focam na validação da acurácia semântica dos modelos de linguagem, esta pesquisa aborda o acoplamento do extrator a estruturas tabulares. A função desenvolvida intercepta e trata de forma automatizada eventuais quebras de integridade na estrutura JSON, permitindo o processamento contínuo de acervos institucionais volumosos sem a necessidade de intervenção humana ou supervisão manual entre as etapas.

3. Abordagem Proposta

A abordagem proposta foi estruturada por meio de um pipeline automatizado de processamento de dados desenvolvido na linguagem Python. A construção das requisições fundamentou-se na engenharia de *prompts*, que compreende a elaboração estratégica de instruções destinadas a guiar os modelos de Inteligência Artificial na geração de respostas específicas, precisas e contextualmente aderentes [do Nascimento 2024]. A Figura 1 ilustra o pipeline implementado.

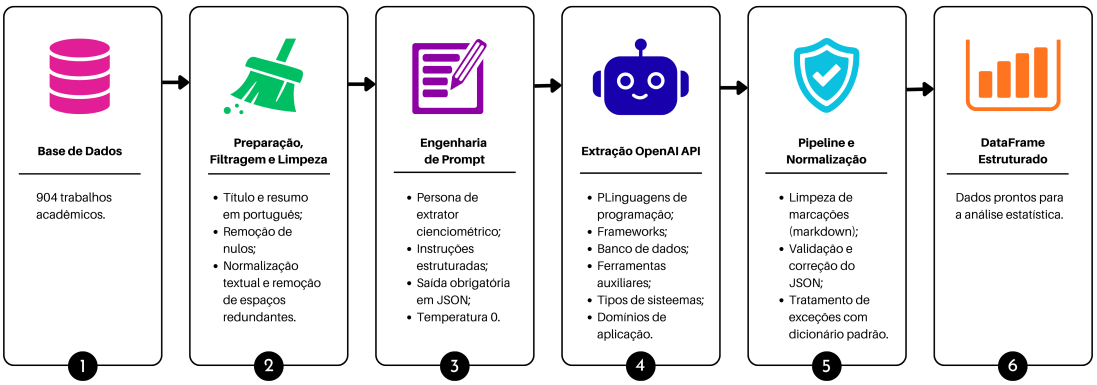


Figura 1. Pipeline Automatizado de Extração Cienciométrica

3.1. Preparação, Filtragem e Limpeza dos Dados

O conjunto de dados inicial totalizava 904 registros históricos de produções acadêmicas vinculadas aos cursos da área de computação da UFAL. Para viabilizar a extração automatizada, aplicou-se uma etapa preliminar de filtragem baseada em critérios de completude. Foram descartados todos os registros que não apresentavam, simultaneamente, título e resumo redigidos em língua portuguesa.

Os campos referentes às nomenclaturas dos cursos e os registros nulos foram submetidos a tratamentos de normalização textual, incluindo a eliminação de espaços em branco redundantes. Visando otimizar a camada de entrada do modelo de linguagem e reduzir o consumo de *tokens* nas requisições, as variáveis textuais de cada trabalho foram concatenadas. Esse procedimento resultou em uma estrutura de texto única e padronizada para cada documento.

3.2. Engenharia de *Prompt* e Arquitetura de Extração

O processo de extração concentrou-se na identificação de entidades específicas: linguagens de programação, *frameworks*, sistemas de gerenciamento de bancos de dados, ferramentas auxiliares, tipologias de sistemas e domínios de aplicação. A aplicação de técnicas estruturadas de engenharia de *prompt* desempenhou papel determinante no controle de qualidade do pipeline, influenciando diretamente a precisão das respostas fornecidas pelo modelo [Neves et al. 2024].

As extrações ocorreram mediante a integração com a API da OpenAI, baseada no modelo GPT-4o mini. O modelo foi parametrizado com temperatura igual a zero, configuração necessária para restringir a variabilidade estocástica das respostas e inibir respostas de natureza inventiva por parte do algoritmo.

Adicionalmente, definiu-se no *prompt* uma persona operacional de extrator cienciométrico especializado. Sob essa diretriz, impôs-se a restrição de coletar exclusivamente tecnologias explicitamente declaradas ou diretamente evidenciadas nos resumos, reduzindo a incidência de alucinações. Para garantir o acoplamento com o restante do pipeline, determinou-se a obrigatoriedade do formato JSON como estrutura de saída. O Quadro 1 apresenta a estrutura do prompt de sistema e do prompt de usuário utilizados nas requisições à API, incluindo o esquema exigido como saída.

```
SYSTEM_PROMPT:
"Você é um extrator de informações para um estudo científico
sobre trabalhos acadêmicos de Computação. Extraia apenas
informações presentes explicitamente no texto ou fortemente
evidentes pelo contexto. Não invente tecnologias. Responda
exclusivamente em JSON válido, sem comentários."

USER_PROMPT:
"Analise o texto abaixo e extraia informações tecnológicas.

Retorne exatamente este JSON:
{
  "linguagens": [],
  "frameworks_bibliotecas": [],
  "bancos_dados": [],
  "ferramentas": [],
  "tipo_sistema": "",
  "dominio_aplicacao": ""
}

Regras:
- Use listas vazias quando nenhuma tecnologia for mencionada.
- Padronize nomes: JavaScript, Python, Java, React, Node.js,
  PostgreSQL, MySQL etc.
- tipo_sistema deve ser curto: Sistema Web, Aplicativo Mobile,
  Modelo de Machine Learning, Estudo Experimental, Revisão
  Bibliográfica, Jogo Digital.
- dominio_aplicacao deve ser curto: Educação, Saúde, Segurança,
  Governo, Agricultura, Finanças, Computação, Não identificado.
- Não inclua explicações fora do JSON.

Texto:
{texto}"
```

Quadro 1. Estrutura do prompt de sistema e de usuário utilizados no pipeline de extração

3.3. Pipeline e Normalização

Dado que o propósito do pipeline consiste em alimentar um *DataFrame* estruturado para posterior análise estatística, eventuais falhas de sintaxe nas respostas da API inviabilizariam o fluxo contínuo de execução. Para mitigar esse risco de interrupção, desenvolveu-se uma rotina de normalização pós-extração baseada em expressões regulares. Esta rotina intercepta a resposta do modelo e realiza a limpeza de blocos de marcação *markdown* eventualmente inseridos de forma redundante na saída.

Em cenários nos quais a estrutura JSON sofria perda de integridade ou o modelo omitia o preenchimento de chaves obrigatórias, a função tratava a exceção de execução. Nesses casos, o pipeline injetava de forma automatizada um dicionário padrão preenchido com listas vazias, preservando a continuidade do fluxo e impedindo a perda de dados no processamento em lote.

4. Resultados e Discussão

A avaliação do pipeline proposto foi realizada por meio de um experimento computacional executado em lote sobre o conjunto de dados selecionado. A conversão das respostas estruturadas em formato JSON para variáveis quantitativas permitiu mapear a frequência acumulada e as distribuições temporais das entidades tecnológicas identificadas. Esta transição de dados textuais para representações gráficas fundamenta a análise do extrator e viabiliza a interpretação dos fluxos cienciométricos gerados [Casagrande and da Silva 2024].

Deste modo, a distribuição quantitativa apresentada na Figura 2 indica a concentração de ferramentas vinculadas aos ecossistemas de Ciência de Dados e Desenvolvimento Web. Essa amostragem inicial demonstra a capacidade do modelo de linguagem de discriminar ferramentas de subáreas distintas. Constata-se que esses dois agrupamentos tecnológicos representam os principais eixos de aplicação prática adotados no escopo da instituição.

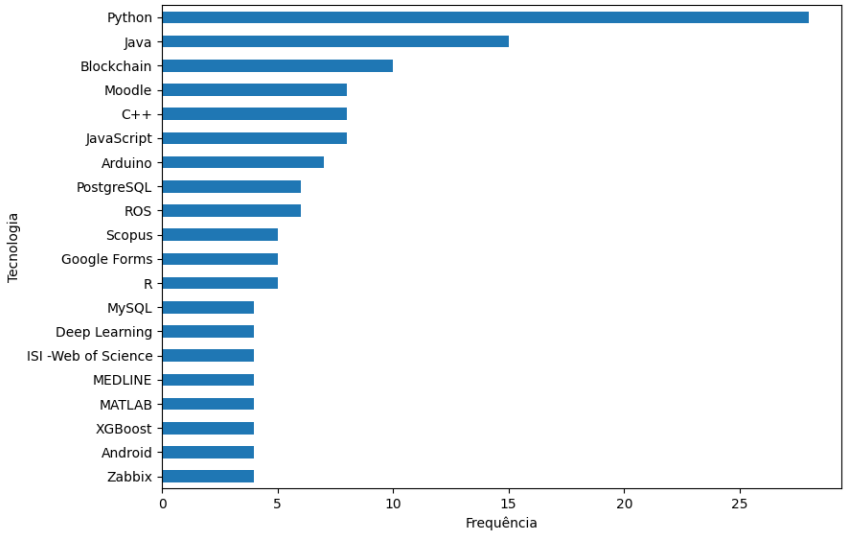


Figura 2. Frequência das 20 Tecnologias mais Citadas nos TCCs

Para investigar a capacidade do pipeline na captura de variações cronológicas e

identificar se essa distribuição reflete um padrão histórico contínuo, os dados foram mapeados em uma série temporal. A Figura 3 destaca a evolução das menções atribuídas às cinco tecnologias que registraram maior frequência acumulada entre os anos de 2006 e 2025.



Figura 3. Evolução Temporal das Principais Tecnologias nos TCCs

Os indicadores cronológicos consolidados na Figura 3 podem evidenciar uma variação nos paradigmas de desenvolvimento de software no ambiente universitário. Linguagens de uso geral, como Java, exibem uma distribuição contínua ao longo de toda a série histórica. Em contrapartida, o ecossistema Python demonstra uma trajetória de crescimento a partir de 2015, registrando maior volume de citações nos anos de 2021 e 2024. Essa variação quantitativa correlaciona-se com o aumento de projetos institucionais voltados para a modelagem preditiva.

Adicionalmente, o pipeline realizou a extração dos contextos práticos nos quais essas ferramentas foram inseridas pelos discentes. A frequência dos domínios de aplicação identificados a partir dos resumos textuais está apresentada na Figura 4.

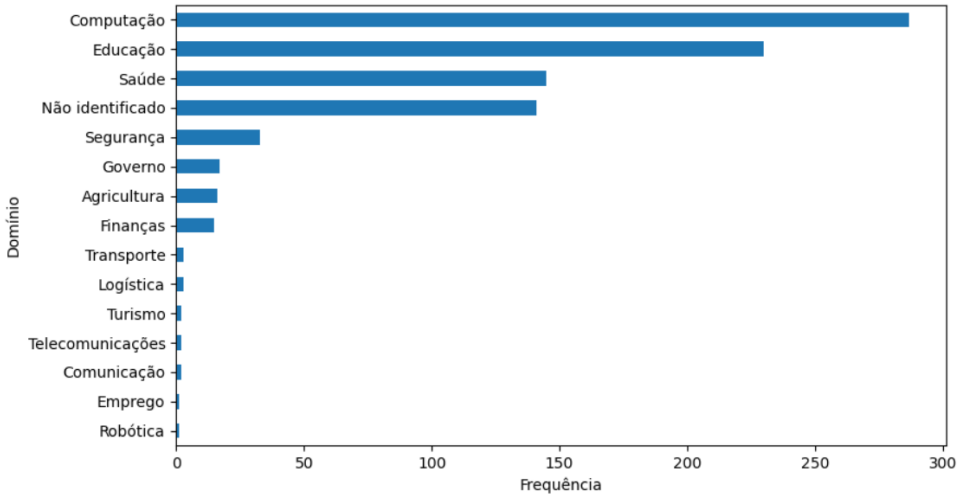


Figura 4. Frequência de Domínios de Aplicação

O comportamento dos dados ilustrados na Figura 4 pode indicar maior frequência nas categorias associadas a ”Computação”de maneira geral e ”Educação”, seguida de ”Saúde”. Esses resultados demonstram a aplicação de ferramentas para a resolução de problemas de caráter social, institucional ou administrativo.

Por fim, o pipeline avaliou a natureza técnica e a abordagem metodológica declarada nas produções. A Figura 5 quantifica a distribuição dos trabalhos de acordo com a tipologia da solução proposta.

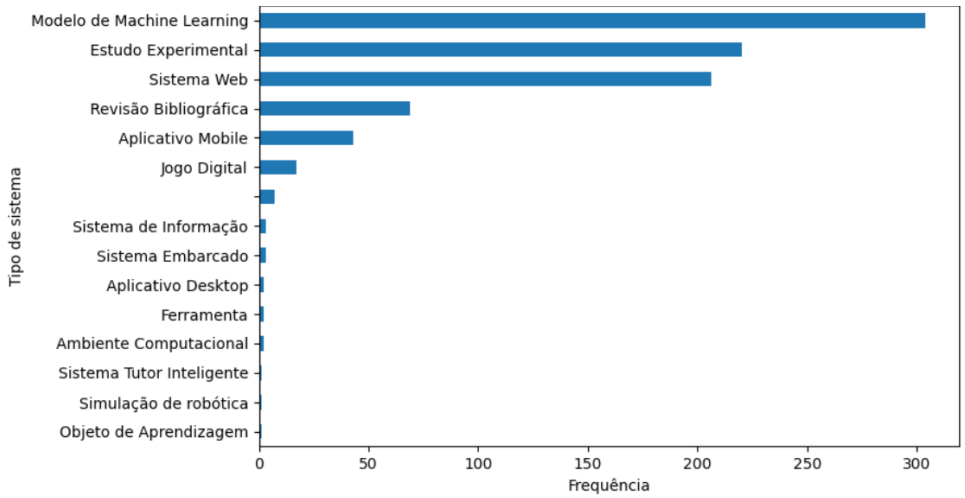


Figura 5. Distribuição dos Trabalhos por Tipo de Sistema

Conforme exposto na Figura 5, a implementação de ”Modelos de *Machine Learning*”destaca-se como a tipologia de maior frequência, seguida por ”Estudos Experimentais”e ”Sistemas Web”. O volume de projetos voltados a algoritmos de aprendizado de máquina complementa a transição identificada na análise cronológica (Figura 3). A correspondência entre esses indicadores gerados pelo modelo demonstra a análise do pipeline automatizado na tarefa de extração cienciométrica, indicando que as subáreas de Inteligência Artificial e Ciência de Dados consolidaram-se como opções para o desenvolvimento de trabalhos finais na instituição.

5. Conclusão

Este trabalho apresentou o desenvolvimento e a avaliação de um pipeline automatizado baseado em LLM para a extração de entidades tecnológicas em resumos acadêmicos. Os resultados obtidos indicam que a integração de LLMs via interface de programação de aplicações atua como ferramenta complementar aos procedimentos tradicionais de auditoria manual e aos métodos restritos a buscas por correspondência de palavras-chave.

O pipeline implementado demonstrou que a imposição de restrições sintáticas por meio da engenharia de *prompt*, associada à parametrização de temperatura igual a zero, mitiga a variabilidade das respostas e fornece a padronização requerida para a ingestão de dados em fluxos cienciométricos estruturados. A aplicação do pipeline sobre os dados textuais da UFAL viabilizou o mapeamento do panorama contemporâneo de ferramentas e a identificação cronológica de transições de paradigmas de software ao longo de duas décadas.

Como limitação do cenário experimental conduzido, constata-se que a análise restrita aos resumos pode omitir ferramentas secundárias ou bibliotecas auxiliares que os discentes deixam de reportar nesse componente textual. Além disso, mudanças nas práticas de escrita ao longo do tempo podem ocorrer e, junto a elas, algumas tecnologias importantes no escopo dos trabalhos podem ter não sido incluídas na análise. Estas restrições delimitam o alcance do mapeamento cienciométrico às tecnologias centrais dos projetos.

Para o desenvolvimento de trabalhos futuros, propõe-se a expansão do pipeline de processamento para realizar a leitura integral do corpo dos documentos, com ênfase nas seções de introdução e metodologia. Adicionalmente, planeja-se a execução de testes comparativos com modelos de código aberto executados localmente, visando mensurar o balanço entre o custo computacional demandado e a precisão obtida na extração das entidades.

Referências

Asudani, D. S., Nagwani, N. K., and Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: a review. *Artificial Intelligence Review*, pages 1 – 81.

Casagrande, V. M. and da Silva, T. P. (2024). Uma introdução à análise de dados usando pandas, matplotlib e seaborn. In *Escola Regional de Informática de Mato Grosso (ERI-MT)*, pages 196–203. SBC.

Chen, Y., Wang, H., Yu, K., and Zhou, R. (2024). Artificial intelligence methods in natural language processing: A comprehensive review. *Highlights in Science, Engineering and Technology*.

Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., and Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications*, 15.

do Nascimento, J. R. (2024). Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de LLM no TCMRio. Trabalho de conclusão de curso, Universidade Federal do Rio Grande do Norte.

Fujita, M., Tartarotti, R., Dal'Evedove, P., and Cruz, M. C. A. (2024). How do authors choose keywords for their theses and dissertations in repositories of university libraries? an introspection-based enquiry. *Coll. Res. Libr.*, 85.

Goulart, V. G., Liboni, L., and Cezarino, L. (2021). Balancing skills in the digital transformation era: The future of jobs and the role of higher education. *Industry and Higher Education*, 36:118 – 127.

Herandi, A., Li, Y., Liu, Z., Hu, X., and Cai, X. (2024). Skill-llm: Repurposing general-purpose llms for skill extraction. *arXiv preprint arXiv:2410.12052*.

Khan, M. Q., Shahid, A., Uddin, M. I., Roman, M., Alharbi, A., Alosaimi, W., Almalki, J., and Alshahrani, S. M. (2022). Impact analysis of keyword extraction using contextual word embedding. *PeerJ Computer Science*, 8:e967.

Khan, N., Elizondo, D. A., Deka, L., and Molina-Cabello, M. A. (2024). Natural language processing tools and workflows for improving research processes. *Applied Sciences*.

Neves, B. B., Sousa, T., Coutinho, D., Garcia, A., and Pereira, J. A. (2024). Explorando o potencial e a viabilidade de llms open-source na análise de sentimentos. In *Congresso Brasileiro de Software: Teoria e Prática (CBSoft)*, pages 89–98. SBC.

Nowakowska, M. (2025). A comprehensive approach to preprocessing data for bibliometric analysis. *Scientometrics*, 130:5191 – 5225.

Pandey, R. and Taruna, S. (2021). Prevalent exact string-matching algorithms in natural language processing: A review. *Journal of Physics: Conference Series*, 1854.

Rahman, M., Sarwar, H., Tin, T. T., and Kader, M. A. (2025). A systematic literature review on final year capstone project in an engineering curriculum. *Edelweiss Applied Science and Technology*.

Rosa, M. S. and da Cruz, F. M. (2026). Modelos de linguagem de grande escala para extração de informações estruturadas de documentos financeiros: Uma avaliação comparativa. In *Inovações e Avanços em Inteligência Artificial*, pages 117–138. Atena Editora.

Wang, Z., Nayfeh, T., Tetzlaff, J., O'Blenis, P., and Murad, M. (2020). Error rates of human reviewers during abstract screening in systematic reviews. *PLoS ONE*, 15.

Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., and Chen, E. (2024). Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18:186357. arXiv:2312.17617.

Younesi, A., Samani, Z. N., and Fahringer, T. (2025). Autostreampipe: Llm assisted automatic generation of data stream processing pipelines. *ArXiv*, abs/2510.23408.

Zdravković, M. (2021). Supervised ml-based approach for auto-tagging of scientific literature. In *2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH)*.