INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS PICOS
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JAMES RYKELME SALES E MORAIS

USO DE PROCESSAMENTO DE LINGUAGEM NATURAL NA EXTRAÇÃO DE CASOS DE TESTES: UM MAPEAMENTO SISTEMÁTICO

PICOS, PIAUÍ
2025

JAMES RYKELME SALES E MORAIS

USO DE PROCESSAMENTO DE LINGUAGEM NATURAL NA EXTRAÇÃO DE CASOS DE TESTES: UM MAPEAMENTO SISTEMÁTICO

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos.

Orientador: Prof. Dr. Rogério Figueredo de Sousa

PICOS, PIAUÍ
2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Morais, James Rykelme Sales e

M827u Uso de processamento de linguagem natural na extração de casos de testes : um mapeamento sistemático / James Rykelme Sales e Morais. - 2025.
39 f.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos, 2025.
Orientador : Prof Dr. Rogério Figueredo de Sousa.

1. Mapeamento sistemático. 2. Processamento de linguagem natural. 3. Teste de software. 4. Requisitos de software. I.Título.

CDD - 004.21

Elaborado por Silvio de Carvalho Gomes Coutinho CRB 3/1362

JAMES RYKELME SALES E MORAIS

USO DE PROCESSAMENTO DE LINGUAGEM NATURAL NA EXTRAÇÃO DE CASOS DE TESTES: UM MAPEAMENTO SISTEMÁTICO

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos.

Aprovada em 07 de abril de 2025.

BANCA EXAMINADORA:

Prof. Dr. Rogério Figueredo de Sousa(Orientador)
Instituto Federal do Piauí (IFPI)

Prof. M^{e}. João Paulo Lima do Nascimento
Instituto Federal do Piauí (IFPI)

Prof. Dr. Rafael Torres Anchiêta
Instituto Federal do Piauí (IFPI)

PICOS, PIAUÍ
2025

"It's a long, long, long
It's a long way [...]
It's a long and winding road"
Caetano Veloso

RESUMO

Este trabalho propõe um mapeamento sistemático da literatura sobre a aplicação de técnicas de Processamento de Linguagem Natural (PLN) no teste de software, destacando sua relevância para a automação e otimização da extração de casos de teste a partir de requisitos escritos em linguagem natural. Com o aumento da complexidade dos sistemas e a demanda por qualidade, os métodos tradicionais de teste enfrentam desafios significativos, tornando urgente a busca por soluções mais eficientes. O objetivo principal deste estudo é catalogar as principais abordagens e técnicas de PLN utilizadas no contexto dos testes de software. Para isso, será realizada uma revisão abrangente da literatura, visando identificar lacunas entre as práticas atuais de garantia de qualidade e a aplicação do PLN. Além disso, a pesquisa irá analisar os desafios e limitações enfrentados na implementação dessas técnicas, buscando entender como elas podem ser integradas de maneira eficaz nos processos de teste. Este mapeamento sistemático visa não apenas contribuir para a compreensão do estado da arte na intersecção entre PLN e testes de software, mas também servir como um guia para pesquisadores e profissionais que desejam explorar essas abordagens. Assim, o trabalho se propõe a fornecer uma base para futuras pesquisas e aplicações práticas nesta área.

Palavras-chaves: Mapeamento sistemático; Processamento de Linguagem Natural; teste de software; garantia de qualidade; Requisitos de Software; extração de casos de teste.

ABSTRACT

This paper proposes a systematic literature mapping on the application of Natural Language Processing (NLP) techniques in software testing, highlighting their relevance for automating and optimizing the extraction of test cases from requirements written in natural language. As the complexity of systems increases and the demand for quality grows, traditional testing methods face significant challenges, making the pursuit of more efficient solutions imperative. The primary objective of this study is to catalog the main approaches and techniques of NLP used in the context of software testing. A comprehensive literature review will be conducted to identify gaps between current quality assurance practices and the application of NLP. Additionally, the research will analyze the challenges and limitations encountered in implementing these techniques, seeking to understand how they can be effectively integrated into testing processes. This systematic mapping aims not only to contribute to the understanding of the state of the art at the intersection of NLP and software testing but also to serve as a guide for researchers and practitioners interested in exploring these approaches. Thus, the work intends to provide a solid foundation for future research and practical applications in this field.

Keywords: systematic mapping; Natural Language Processing; software testing; quality assurance; Software Requirements; test case extraction.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BERT	Bidirectional Encoder Representations from Transformers
CoT	Chain of Thought
FRDs	Functional Requirement Documents
IA	Inteligência Artificial
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
LLMs	Large Language Models
ML	Machine Learning
MS	Mapeamento Sistemático
MSL	Mapeamento Sistemático da Literatura
PLN	Processamento de Linguagem Natural
QA	Quality Assurance
START	Systematic Tool for Article Review and Tracking

SUMÁRIO

1 INTRODUÇÃO

O ciclo de vida do desenvolvimento de software é tipicamente dividido em fases como planejamento, análise de requisitos, design, codificação, testes, implantação e manutenção. Dentre essas etapas, o teste de software se destaca com um componente essencial para garantir a qualidade do produto final (SOMMERVILLE, 2018). No entanto, com o aumento da complexidade dos sistemas e a crescente demanda por qualidade, os métodos tradicionais do processo de teste têm enfrentado dificuldades (SAHNI, 2022).

Nesse cenário, a integração de técnicas de inteligência artificial (IA), em particular o processamento de linguagem natural (PLN), é proposta como uma alternativa para otimizar o processo de garantia de qualidade. O PLN possibilita a compreensão e o processamento de texto em linguagem humana, o que pode oferecer um grande potencial para a automação e otimização das atividades de teste (GAROUSI; BAUER; FELDERER, 2020). Assim, diversos estudos têm sido realizados buscando compreender e aplicar otimizações baseadas nessa perspectiva.

Conforme destacado por Aghamohammadi, Mirian-Hosseinabadi & Jalali (2021) a medida em que a complexidade do sistema cresce, a quantidade de testes necessários para avaliar a aplicação se amplia. A partir dessa perspectiva, a necessidade de ferramentas para facilitar esse processo torna-se ainda mais pertinente. Nesse contexto, em especial no âmbito da extração de casos de testes a partir de requisitos de software escritos em linguagem humana, diversos estudos são desenvolvidos, no entanto, as diversas abordagens existentes na literatura, podem gerar uma dificuldade na escolha da melhor técnica a ser aplicada.

Além disso, a diversidade existente nas técnicas de PLN aplicadas ao teste de software pode acabar prejudicando o entendimento em relação a alguns pontos como: qual a técnica mais adequada, quais as limitações das abordagens e quão custosa é a aplicação dessas técnicas. Segundo Boukhlif *et al.* (2024), a ausência de uma visão organizada e sistemática sobre as técnicas existentes dificulta a compreensão clara sobre desafios, limitações e benefícios associados ao uso de PLN em testes de software, podendo comprometer a implementação efetiva dessas tecnologias em projetos futuros. Ademais, dentre a vasta variedade de materiais disponíveis, podem estar contidas metodologias que não atendem plenamente às necessidades específicas da área em análise.

Sendo assim, com a grande variedade de abordagens baseadas em PLN propostas na literatura, torna-se essencial a realização de um mapeamento sistemático para catalogar e detectar as estratégias mais recorrentes, além de pontuar as áreas que precisam de maior desenvolvimento. Neste cenário, este estudo visa identificar,

categorizar e analisar as técnicas, abordagens e ferramentas de PLN na geração de casos de teste a partir de artefatos textuais. Também busca-se entender as vantagens apontadas na literatura sobre essa estratégia, além de destacar as limitações encontradas. Assim, este trabalho busca oferecer uma visão abrangente do estado da arte, visando o progresso de estudos que explorem a intersecção entre PLN e garantia de qualidade de software.

2 JUSTIFICATIVA

Levando em consideração o teste de software como uma das etapas essenciais do processo de desenvolvimento de software e conforme apontado por Leloudas (2023), é possível afirmar que o foco na garantia da qualidade do sistema é fundamental para possibilitar que o software funcione como o esperado e satisfaça as necessidades do usuário, entregando, assim, um sistema confiável e eficiente. Dessa maneira, em um ambiente onde a utilização de software está cada vez mais presente tanto em aspectos do cotidiano social quanto em ambientes corporativos, a garantia de qualidade torna-se ainda mais crucial.

Além disso, em um cenário de crescimento da complexidade do código, a quantidade de testes necessários se amplia (AGHAMOHAMMADI; MIRIAN-HOSSEINABADI; JALALI, 2021). Nessa perspectiva, ao avaliar a etapa da extração de casos de testes a partir da análise de requisitos, o processo manual torna-se gradativamente mais árduo, demandando que os testadores exerçam análises cada vez mais precisas das diversas especificações textuais do sistema. Sob essa visão, para reduzir o esforço crescente na extração manual de casos de teste, pesquisas são desenvolvidas explorando as diversas alternativas tecnológicas para auxiliar com tal impasse. Dentre esses estudos, destaca-se o uso do Processamento de Linguagem Natural (PLN).

No que se refere ao uso de PLN, é importante destacar o potencial oferecido por essa tecnologia para auxiliar na automação da extração de casos de testes em relação às análises de requisitos escritos em linguagem natural. Ferramentas e técnicas provenientes do PLN são empregadas para a melhoria de pontos como: a identificação de padrões, a extração de informações relevantes e a redução de ambiguidades presentes nas especificações.

No entanto, a aplicação do PLN em testes de software ainda apresenta pontos a serem aprimorados como uma melhor identificação de atores no cenário de teste, além da necessidade de uma melhoria para lidar com a ambiguidade da linguagem natural (LIM *et al.*, 2024a). Somada à crescente quantidade de estudos realizados na área, surge a necessidade de uma análise ampla sobre o estado arte da área. Sendo assim, torna-se pertinente a realização de um mapeamento sistemático da literatura (MSL). Nesse contexto, este trabalho busca realizar um MSL com o objetivo de oferecer um panorama da área, abordando as principais tecnologias utilizadas e identificando as possíveis lacunas.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Este trabalho busca oferecer uma compreensão mais clara das técnicas atuais e das potencialidades do uso do PLN em testes de software, através da realização de um mapeamento sistemático da intersecção entre essas duas áreas, a fim de auxiliar aqueles que desejam entender melhor a aplicação do PLN em testes de software.

3.2 OBJETIVOS ESPECÍFICOS

1. Sumarizar técnicas e abordagens do PLN aplicadas para a extração de casos de teste a partir de requisitos escritos em linguagem natural.
2. Analisar os desafios e limitações da aplicação das técnicas de PLN no processo de garantia de qualidade de software.
3. Avaliar o impacto das técnicas de PLN sob a perspectiva do esforço necessário para sua implementação no processo de testes de software.

4 REFERENCIAL TEÓRICO

Nesta seção serão apresentados conceitos-chave, essenciais para o desenvolvimento e entendimento do projeto.

4.1 MAPEAMENTO SISTEMÁTICO DA LITERATURA

De acordo com (KITCHENHAM; CHARTERS, 2007), o mapeamento sistemático (MS) tem como objetivo identificar e classificar conteúdos de um campo de estudo. Ademais, esse método visa estruturar a área de pesquisa, além de categorizar os resultados obtidos (PETERSEN *et al.*, 2008). Além disso, o MS, estudo secundário, se destaca por proporcionar uma visão geral ampla de uma área de pesquisa e pode ajudar a identificar lacunas na análise (WOHLIN *et al.*, 2013). Assim, o MS auxilia nas escolhas relacionadas ao trabalho a ser realizado.

A Figura 1 ilustra as fases do processo de condução de um mapeamento sistemático, oferecendo uma representação visual das principais etapas envolvidas.

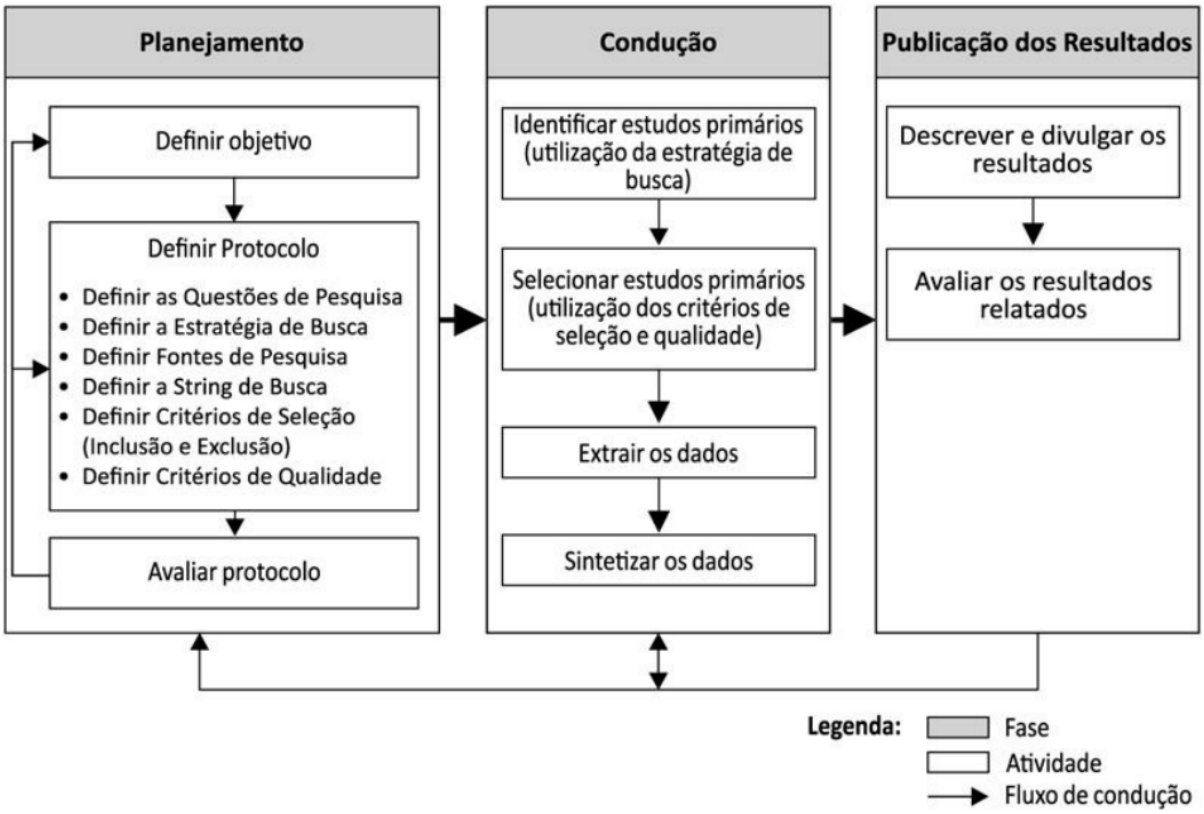


Figura 1 – Fases do Processo de Condução de uma Revisão Sistemática e MS.
Fonte: (SCANNAVINO *et al.*, 2017), p. 19.

Conforme mostrado na Figura 1, o processo de gerenciamento de um estudo secundário como um mapeamento sistemático envolve três fases: Planejamento,

Condução e Publicação dos Resultados. De forma geral, na fase de planejamento é definido o objetivo da execução do trabalho e o protocolo do mapeamento.

Já a fase de condução engloba etapas como identificação e seleção dos estudos primários, tal seleção é executada a partir de critérios previamente definidos, além de ocorrer a extração e síntese dos dados relevantes. Por fim, os resultados são descritos e avaliados na etapa de publicação.

4.2 PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural (PLN) pode ser caracterizado como a subárea da ciência da computação que combina linguística e inteligência artificial, utilizando métodos e recursos computacionais para compreender, processar e gerar dados linguísticos, sendo utilizada para realizar tarefas como análise sintática e semântica, reconhecimento de entidades nomeadas, análise de sentimentos, geração de textos e tradução automática (JURAFSKY; MARTIN, 2008). Em suma, a abrangência do PLN compreende uma vasta variedade de atividades relacionadas à análise de informações em uma linguagem que é característica da comunicação humana.

Dessa forma, o PLN se expande a outras áreas da computação. No contexto da engenharia de software, o PLN é aplicado por meio da integração de técnicas aos processos presentes na área, abordando desde a automação de tarefas como análise de requisitos até a geração automática de casos de teste. Assim, o uso dessas técnicas tem se mostrado eficiente ao otimizar e diminuir o tempo de execução dos processos (NAZIR *et al.*, 2017).

4.3 QUALIDADE DE SOFTWARE

A qualidade de software é caracterizada como o campo da engenharia de software que visa auxiliar no desenvolvimento de sistemas que atendam tanto aos requisitos especificados quanto aos anseios dos usuários, buscando esse objetivo por meio da aplicação de princípios e abordagens sistemáticas ao processo de desenvolvimento (SOMMERVILLE, 2018). Ainda de acordo com (SOMMERVILLE, 2018), o processo de garantia da qualidade envolve práticas contínuas e estruturadas, como a definição clara de requisitos e a realização de testes, visando identificar lacunas na qualidade do sistema, além de incluir gestão de configurações e mudanças, bem como o suporte contínuo. Ademais, ao seguir no ecossistema dos testes, encontram-se conceitos como análise dos requisitos e casos de teste, abordagens intrínsecas a sistemas de alta qualidade.

4.3.1 REQUISITOS DE SOFTWARE

Ao buscar a adoção de um conjunto de tarefas estruturadas para auxiliar na produção de software, conceitos como análise de requisitos ganham destaque. Para (SOMMERVILLE, 2018), no âmbito da engenharia de requisitos, essa análise é essencial para entender e documentar o que o sistema deve fazer. O autor classifica a elicitação e análise de requisitos como um processo iterativo dividido em fases como descoberta de requisitos, classificação e organização, priorização e negociação, além da fase de especificação dos requisitos.

4.3.2 CASOS DE TESTE

Os casos de teste são definidos como especificações das entradas para o teste e dos resultados esperados do sistema (SOMMERVILLE, 2018), sendo essenciais para a descrição da execução e expectativas dos testes. Além disso, de acordo com (PRESSMAN; MAXIM, 2016), no âmbito da engenharia de requisitos, especificamente nas etapas do processo de software, destaca-se que o momento ideal para a criação dos casos de teste é durante a coleta de requisitos.

5 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados e discutidos os trabalhos que possuem propósitos semelhantes ao deste pré-projeto. Em relação à seleção dos trabalhos correlatos, a execução se deu a partir da definição dos termos de busca vinculados ao tema, como: "Processamento de Linguagem Natural", "Testes de Software", "Mapeamento Sistemático"e "Revisão Sistemática", ademais, para a realização da pesquisa foram utilizadas bases acadêmicas, como ACM Digital Library e ScienceDirect. Ainda tratando-se da classificação dos estudos retornados, os materiais foram avaliados a partir de três critérios principais: (1) Qual foi o tipo de estudo secundário realizado? (2) Qual o foco da abordagem de PLN no teste de software? e por último, (3) Qual o tipo de escopo foi predominante no estudo secundário? Além disso, a avaliação desses trabalhos busca discorrer sobre os principais objetivos do estudo, quais bases de dados foram utilizadas e os resultados alcançados.

O estudo conduzido por Garousi, Bauer & Felderer (2020) executa um mapeamento sistemático da literatura buscando resumir o estado da arte em testes de software assistidos por PLN. Em suma, o estudo busca oferecer uma visão ampla do cenário da área. Os artigos foram obtidos a partir das bases de dados do Google Scholar e da Scopus. Inicialmente foram selecionados 95 artigos, no entanto, no conjunto final após a aplicação dos critérios de inclusão e exclusão baseados em pontos como: no que o estudo se concentra, se o estudo tem uma validação relativamente sólida e se a fonte está em inglês, foram selecionados 67 artigos técnicos. Em relação aos resultados, alguns fatores se destacam como: (1) Dentre as 38 ferramentas abordadas nos artigos apenas 4 estão disponíveis para download; (2) Em meio aos 67 artigos selecionados, somente 27 mencionaram o(s) nome(s) da(s) ferramenta(s) de PLN utilizadas nos projetos; (3) Dentre os 67 artigos selecionados, 30 apresentaram uma exposição superficial aos aspectos de PLN.

O estudo realizado por Ahsan *et al.* (2017) executa uma revisão sistemática tendo como objetivo examinar a literatura existente entre os períodos de 2005 a 2016 que abordaram o uso de técnicas de PLN na geração de casos de teste. Os estudos foram obtidos a partir de bases de dados como, ACM, IEEE e SPRINGER. Em relação à seleção, após a aplicação dos critérios de exclusão e inclusão, dos 167 estudos selecionados, somente 16 foram escolhidos, assim, 151 rejeitados. Do critério de seleção, foram considerados alguns pontos como: origem de publicação do estudo, data de publicação do estudo e avaliação da veracidade dos dados do trabalho. Tratando-se dos resultados, em relação às técnicas de PLN utilizadas, 81,25% dos estudos utilizaram de uma a mais de três técnicas de PLN enquanto 18,75% dos estudos não utilizaram uma técnica diretamente, no entanto, utilizaram ferramentas

de PLN. Em resumo, o estudo conseguiu investigar 18 ferramentas, 4 técnicas e 9 algoritmos para geração de casos de teste.

Os estudos analisados apresentam algumas limitações que justificam a realização deste trabalho. Embora a pesquisa conduzida por Garousi, Bauer & Felderer (2020) proporcione uma perspectiva ampla sobre os testes de software auxiliados por PLN, ela aborda um conjunto extenso de estudos que, em sua maioria, apresentam explicações superficiais sobre as técnicas de PLN, além de não focar especificamente na geração de casos de teste a partir de requisitos de software. Por outro lado, a revisão realizada por Ahsan *et al.* (2017), ainda que tenha como foco a criação de casos de teste utilizando PLN, limita-se a um intervalo temporal que se extende somente até 2016, não tratando acerca dos avanços mais recentes da área. Assim, este estudo busca preencher tais lacunas por meio da análise de estudos primários mais atuais que abordem diretamente o uso de técnicas de PLN aplicadas à extração de casos de teste a partir de requisitos de software.

6 METODOLOGIA

Com o propósito de atingir o objetivo desta pesquisa, foi realizado um mapeamento sistemático da literatura (MSL). Por conseguinte, a metodologia adotada foi adaptada a partir da obra de Scannavino *et al.* (2017) e baseada em conceitos definidos por Biolchini *et al.* (2005), Brereton *et al.* (2007), Kitchenham & Charters (2007) e Mafra & Travassos (2006). Neste sentido, sua composição foi dividida em algumas etapas como: planejamento, condução e publicação dos resultados.

A partir das etapas, baseando-se no objetivo definido, foram constituídas algumas atividades essenciais para a condução do projeto, como: delimitação do protocolo, que incluiu a definição de aspectos como questões de pesquisa, string de busca, bases de dados, critérios de seleção e filtros. Além disso, foram realizadas as buscas, a seleção de estudos, a extração e a apresentação dos dados sintetizados.

6.1 QUESTÃO DE PESQUISA

Segundo Petersen *et al.* (2008), a definição das questões de pesquisa configura-se como uma das etapas essenciais para a condução do MS, uma vez que a partir delas, etapas subsequentes como a busca, a triagem e a análise dos estudos são realizadas.

Dessa forma, o estudo em questão teve como foco responder às seguintes questões de pesquisa (QP):

- **QP1:** Quais são as principais técnicas, abordagens ou ferramentas de PLN utilizadas?
- **QP2:** Quais são as vantagens reportadas sobre a aplicação de PLN na derivação de casos de teste?
- **QP3:** Quais são as limitações apontadas na extração de casos de teste com o auxílio do PLN?

6.2 STRING DE BUSCA E FONTES DIGITAIS DE PESQUISA

Em relação à construção da string de busca, visando identificar quais combinações de termos relacionados e palavras-chave retornariam uma maior quantidade de estudos, foi realizada uma pesquisa inicial, denominada busca piloto, baseada nos conceitos de Brereton *et al.* (2007), que a define como uma busca preliminar que visa avaliar a string mais adequada para cada estudo de revisão. Ademais, a definição da

string de busca teve como base as questões de pesquisa e a delimitação do objetivo do estudo.

Fundamentada nesse ponto, a aplicação desta consulta se deu a partir da utilização de palavras-chave escritas em inglês e em português. No que concerne à composição da expressão de pesquisa, os conceitos-chave semelhantes foram organizados em domínios por meio do operador lógico OR, enquanto o operador booleano AND foi utilizado para unir os diferentes domínios principais. Após a organização e a aplicação da busca piloto, a expressão foi iterativamente aprimorada, resultando na seguinte string de busca:

> ("Natural Language Processing"OR NLP OR "Processamento de Linguagem Natural"OR PLN) AND ("software testing"OR "test case extraction"OR "test case generation"OR "test automation"OR "testes de software"OR "extração de casos de teste"OR "geração de casos de teste"OR "automação de testes") AND ("requirements"OR "software requirements"OR "requisitos de software"OR "linguagem natural"OR "natural language") AND ("techniques"OR "approaches"OR "tools"OR "técnicas"OR "abordagens"OR "ferramentas"OR "advantages"OR "vantagens"OR "limitations"OR "limitações"OR "challenges"OR "desafios")

Tratando-se das bases de pesquisa científicas, foram selecionadas as fontes bibliográficas a seguir: IEEExplore[1], Scopus[2] e Web of Science[3]. As bases foram escolhidas levando em consideração a grande variedade de estudos publicados, além de possuírem mecanismos de pesquisa avançada que aceitam expressões booleanas.

6.3 CRITÉRIOS DE SELEÇÃO E FILTROS

Com a finalidade de selecionar somente trabalhos pertinentes, diretamente relacionados ao escopo do estudo, fez-se necessário seguir algumas estratégias, incluindo a utilização de critérios para a seleção dos trabalhos, como:

- **Critério de Inclusão (CI):** define a inclusão de um estudo no MS.
- **Critério de Exclusão (CE):** define a exclusão de um estudo no MS.

Somada aos critérios de seleção, as publicações obtidas foram submetidas a dois filtros. No primeiro, foram analisados o título, o resumo (*abstract*), as palavras-chave e o ano de publicação; já no segundo, as publicações que passaram pelo primeiro filtro tiveram o seu conteúdo lido por completo. As tabelas abaixo detalham os critérios de seleção adotados nas etapas de filtragem:

[1] <https://ieeexplore.ieee.org/>
[2] <https://www.scopus.com/>
[3] <https://www.webofscience.com/>

Tabela 1 – Critérios de Inclusão do 1º Filtro

Código	Descrição
CI-01	Estudos primários.
CI-02	Estudos que descrevam o uso de Processamento de Linguagem Natural na garantia da qualidade de software, em específico no contexto de casos de teste.
CI-03	Estudos disponíveis via acesso CAFE Periódicos CAPES.
CI-04	Estudos publicados entre 2022 e 2025.

Fonte: Elaborado pelo autor.

Tabela 2 – Critérios de Exclusão do 1º Filtro

Código	Descrição
CE-01	Estudos secundários (mapeamentos sistemáticos ou revisões bibliográficas).
CE-02	Estudos publicados em literatura cinzenta.
CE-03	Estudos em que as palavras-chave da string de busca não estejam presentes no título, no *abstract* ou na lista de palavras-chave da publicação.
CE-04	Estudos com conteúdo pago.
CE-05	Estudos que não estejam escritos nos idiomas inglês ou português.
CE-06	Estudos duplicados.

Fonte: Elaborado pelo autor.

Tabela 3 – Critérios de Inclusão do 2º Filtro

Código	Descrição
CI-01	Estudos que propõem abordagens e técnicas de PLN na extração de casos de teste explicitamente a partir de requisitos de software.

Fonte: Elaborado pelo autor.

Tabela 4 – Critérios de Exclusão do 2º Filtro

Código	Descrição
CE-01	Estudos em que é possível identificar que a extração do caso de teste com auxílio do PLN não foi realizada a partir de requisitos de software.
CE-02	Estudos onde não é possível identificar qual técnica ou abordagem de PLN foi implementada.

Fonte: Elaborado pelo autor.

6.4 SELEÇÃO DOS ESTUDOS

Após a elaboração e aplicação das fases anteriores, que incluíram a etapa de pesquisa a partir da utilização da string de busca formada por termos combinados e escritos em inglês e em português, sendo aplicada nas bases definidas e levando em consideração os estudos retornados, foi realizado o *download* com foco nas referências dos trabalhos. As buscas foram feitas entre os meses de fevereiro e março de 2025. A Tabela 5 apresenta o número de estudos retornados por base.

Seguindo o fluxo do processo do mapeamento, após a execução da pesquisa,

Tabela 5 – Bases de Pesquisa e Quantitativo de Estudos

Base de Pesquisa	**Número de Estudos**
IEEE	173
SCOPUS	433
Web of Science	90
Total	696

Fonte: Elaborado pelo autor.

foi realizada a etapa de seleção dos estudos retornados, aplicando-se os filtros e os critérios de inclusão e exclusão previamente definidos. Para auxiliar esse processo de estruturação e triagem, utilizou-se o software StArt (*State of the Art through Systematic Review*)[4], desenvolvido pelo Laboratório de Pesquisa em Engenharia de Software (LaPES) da Universidade Federal de São Carlos (UFSCar). A Tabela 6 apresenta o número de estudos selecionados em cada filtro.

Tabela 6 – Quantidade de estudos nas diferentes etapas de seleção

Etapa	**Quantidade**
Total de estudos na busca inicial	696
Após aplicação do Filtro 1	39
Após aplicação do Filtro 2	10
Total final	**10**

Fonte: Elaborado pelo autor.

A lista final, que contém o total de 10 estudos selecionados foram tabulados e estão explicitados na Tabela 7.

Partindo da triagem final dos estudos, a etapa seguinte é a de extração dos dados. Conforme a metodologia proposta na obra de Scannavino *et al.* (2017) a utilização de formulários de extração de dados exerce a função de auxiliar na coleta das informações necessárias para responder as questões de pesquisa, comportando-se como uma etapa prévia a sintetização dos dados. Dessa forma este estudo adotou o formulário contido na Tabela 8.

[4] <https://www.lapes.ufscar.br/resources/tools-1/start-1>

Tabela 7 – Estudos incluídos na etapa final da seleção

Título	**Referência**	**Base**
Natural Language Processing Technology Based on Artificial Intelligence in Software Testing	(LIU, 2024)	IEEE
Domain-Specific Software System Testing by Using Intelligent Approaches	(CHAUHAN; BALAKRISHNAN, 2024)	IEEE
Naive Bayes Classification Model for Precondition-Postcondition in Software Requirements	(FADHLURROHMAN *et al.*, 2023)	IEEE
Intelligent Requirement-to-Test-Case Traceability System via NLP and ML	(SAWADA *et al.*, 2023)	IEEE
Automated Test Case Generation for Satellite FRD Using NLP and LLM	(S *et al.*, 2024)	IEEE
Leveraging Pre-Trained LLMs for On-Premises Automated Test Case Generation	(YIN; MOHAMMED; BOYAPATI, 2024)	IEEE
Automatic Generation of Acceptance Test Cases from Use Case Specifications	(WANG *et al.*, 2022)	IEEE
Automatic Creation of Acceptance Tests by Extracting Conditionals from Requirements	(FISCHBACH *et al.*, 2023)	SCOPUS
Leveraging Conditional Statement to Generate Acceptance Tests Automatically	(ZHAO *et al.*, 2023)	SCOPUS
Test case informat ion extraction from requirements specifications using NLP-based unified boilerplate approach	(LIM *et al.*, 2024b)	WEB OF SCIENCE

Fonte: Elaborado pelo autor.

Tabela 8 – Formulário de extração de dados

Descrição	**Tipo**	**Valores**
Quais as principais técnicas ou abordagens utilizadas?	Descritiva / Seleção Múltipla	Ex.: análise sintática, análise semântica, redes neurais, etc.
Quais ferramentas ou frameworks foram empregados?	Descritiva	Ex.: NLTK, spaCy, BERT, etc.
O processo foi automatizado?	*Boolean*	*True* *False*
Quais vantagens foram reportadas?	Descritiva	-
Quais limitações e desafios foram apontados?	Descritiva	-
O estudo utilizou técnicas complementares de *machine learning* ou inteligência artificial aplicadas ao PLN?	*Boolean*	*True* *False*

Fonte: Elaborado pelo autor.

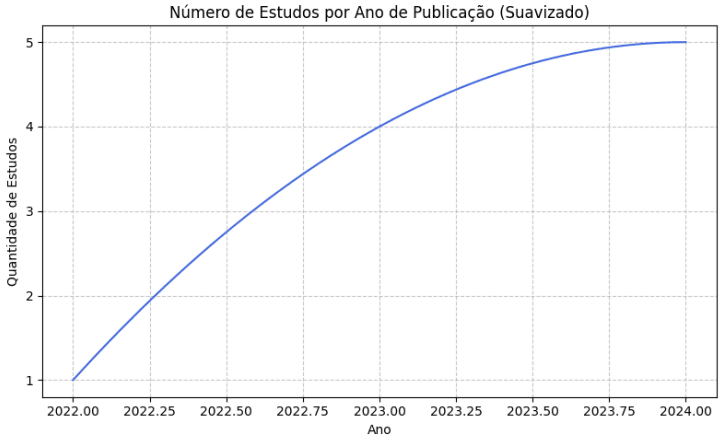
7 RESULTADOS E DISCUSSÕES

Neste capítulo é fornecida uma perspectiva detalhada sobre o uso de PLN para geração de casos de testes tendo como base requisitos escritos em linguagem natural, esse tópico busca fornecer respostas às questões de pesquisa com base nos estudos primários previamente selecionados como relevantes. Na Seção 7.1 são apresentados as principais técnicas, abordagens e ferramentas de PLN utilizadas no contexto de desenvolvimento dessas estruturas de validação que apareceram nos estudos incluídos, buscando responder à **QP1**. Na Seção 7.2 são explicitadas as vantagens reportadas nos estudos sobre a aplicação de PLN na derivação de casos de teste, como resposta à **QP2**. A Seção 7.3 aborda as limitações apontadas na literatura sobre a geração de casos de testes auxiliados pelo PLN, como resposta à **QP3**.

Ademais, este trabalho reúne e apresenta informações atualizadas sobre o tema, buscando evidenciar os avanços mais recentes no que diz respeito ao uso de PLN aplicadas à geração automatizada de casos de teste.

No gráfico da figura 2, pode-se observar a distribuição temporal dos estudos incluídos neste Mapeamento Sistemático. É possível notar que, com o passar dos anos, o número de publicações neste âmbito tem crescido. Enquanto que em 2022 houve apenas 1 estudo e, em 2023, 4 publicações, no ano de 2024 foram realizados 5 estudos, mais do que o ano anterior, Já em 2025, até o momento da realização deste trabalho (março de 2025), não foram identificados estudos relevantes, o que pode ser atribuído ao fato de que o ano ainda está em andamento, sendo possível que novas publicações venham a ser realizadas nos meses seguintes.

Figura 2 – Distribuição dos estudos por ano de publicação



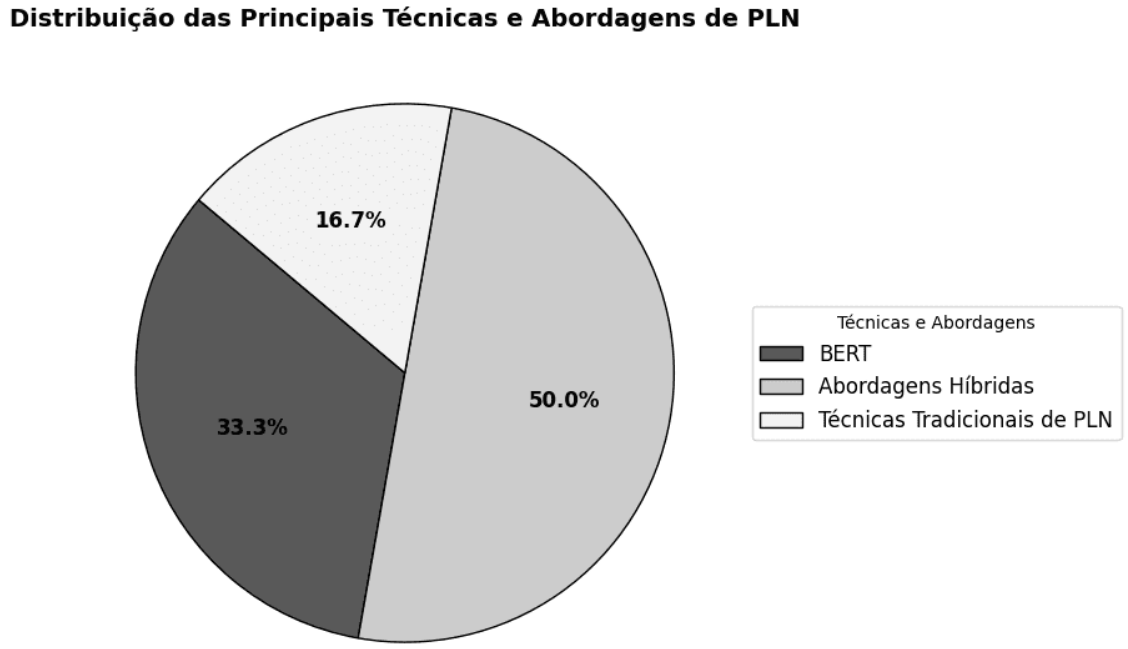
Fonte: Elaborado pelo autor (2025).

7.1 QUESTÃO DE PESQUISA 01

QP1: Quais são as principais técnicas, abordagens ou ferramentas de PLN utilizadas?

Ao observar os trabalhos selecionados, pode-se definir as principais técnicas e abordagens de PLN utilizadas com base na frequência de sua aparição nos estudos. Desse modo, 4 estudos utilizaram o modelo BERT como técnica principal de PLN. Logo em seguida vieram abordagens híbridas, ou seja que combinaram PLN com outras técnicas do âmbito de aprendizado de máquina ou de modelos de linguagem ampla, tais abordagens híbridas incluíram o uso de Naive Bayes, BiLSTM, GPT, T5 e LLaMA, estando presentes em 6 estudos. Tratando-se dos trabalhos que utilizaram abordagens linguísticas tradicionais de PLN, como: tokenização, reconhecimento de entidades nomeadas (NER), rotulagem de partes do discurso (POS), entre outras, resultaram em um total de 2 estudos. O gráfico da figura 3 ilustra a distribuição das ferramentas e das abordagens de PLN utilizadas nos estudos. Já a Tabela 9 descreve as ferramentas e metodologias implantadas em cada estudo.

Figura 3 – Distribuição das ferramentas e abordagens de PLN utilizadas nos estudos.



Fonte: Elaborado pelo autor (2025).

Tabela 9 – Ferramentas e metodologias dos estudos

Título	**Metodologia/Ferramenta**
Natural Language Processing Technology Based on Artificial Intelligence in Software Testing	O estudo explora o uso da tecnologia de PLN baseada em BERT (Bidirectional Encoder Representations from Transformers), tal tecnologia foi utilizada na extração de informações-chaves dos requisitos, por meio de uma análise semântica profunda com foco na geração de casos de teste. Além disso, o BERT foi utilizado na análise de informações de texto em códigos e arquivos de log, com foco em identificar potenciais defeitos e problemas.
Domain-Specific Software System Testing by Using Intelligent Approaches	O artigo propõe o desenvolvimento de um projeto de arquitetura de ferramenta específica de domínio para automatizar a geração de casos de teste,integrando IA (por exemplo, GPT, T5) e PLN.
Naive Bayes Classification Model for Precondition-Postcondition in Software Requirements	O artigo desenvolve um modelo de classificação que prevê as pré-condições e pós-condições dos requisitos de software como base para geração de casos de teste, dentre as etapas apresentadas: se encontra a preparação dos dados (limpeza, divisão e etc) pré-processamento dos dados e a implementação do modelo em si que utilizou o algoritmo Naive Bayes para treinar o conjunto de dados e classificar os textos, além da utilização do NTLK e do Scikit-learn para implementar o modelo.

Continua na próxima página

Título	Metodologia/Ferramenta
Intelligent requirement-to-test-case traceability system via Natural Language Processing and Machine Learning	O artigo emprega técnicas de PLN combinadas a Machine Learning (ML) visando gerar recomendações para mapeamento de requisitos para teste, cobertura de teste e análise de lacuna de cobertura, além de gerar e recomendar casos de teste quando uma lacuna de cobertura é detectada. Para a implementação da abordagem, foram escolhidos algumas tecnologias como: Python3 além das bibliotecas e frameworks scikit-learn, NLTK, TensorFlow, Pandas, SciPy, PyTorch e NumPy, além da integração do BERT para gerar sugestões de casos de teste.
Automated Test Case Generation for Satellite FRD Using NLP and Large Language Model	O estudo desenvolve um sistema que gera casos de teste de forma automática utilizando PLN, assim ele interpreta e analisa de forma autônoma *Functional Requirement Documents* (FRDs), para extrair detalhes importantes para gerar os casos de teste. Além disso, há a integração de *Few-Shot Prompting* e *Chain of Thought* (CoT) prompting com *Large Language Models* (LLMs) buscando facilitar o processo de aprendizado dinâmico e permitindo que o sistema se adapte às várias configurações e cenários de teste.

Continua na próxima página

Título	Metodologia/Ferramenta
Leveraging Pre-Trained LLMs for On-Premises Automated Test Case Generation	O artigo, apresentou um estudo empírico, investigando o uso de LLMs pré-treinados para gerar casos de teste a partir de requisitos de linguagem natural. Sendo assim, dentre os modelos de linguagem analisados temos o LLaMA-2 (nas versões 7B, 13B e 70B), o LLaMA-2-chat (também nas versões 7B, 13B e 70B) e o LLaMA-3 70B Instruct.
Automatic Generation of Acceptance Test Cases From Use Case Specifications: An NLP-Based Approach	O estudo emprega uma abordagem chamada *Use Case Modeling for System-level, Acceptance Tests Generation* (UMTG), que em resumo utiliza técnicas de PLN para a extração automática de informações de requisitos escritos em linguagem natural, a tecnologia UMTG busca auxiliar na geração de casos de teste de aceitação com base nas especificações do sistema. Assim a abordagem UMTG utiliza cinco técnicas diferentes de PLN como: a Tokenização, o Reconhecimento de Entidades Nomeadas (NER), Etiquetagem de Partes do Discurso (POS Tagging), Rotulagem de Papéis Semânticos (SRL) e Detecção de Similaridade Semântica.

Continua na próxima página

Título	Metodologia/Ferramenta
Automatic creation of acceptance tests by extracting conditionals from requirements: NLP approach and case study	É apresentado no estudo a ferramenta CiRA(*Conditionals in R equirements A rtifacts*) ela fornece recursos para gerar casos de teste de aceitação automaticamente, conseguindo detectar e extrair declarações condicionais em um formato refinado a partir de requisitos em linguagem natural, servindo como base para a geração dos casos de teste. Dentre as ferramentas de apoio apresentados temos o uso de BERT, que é usado para gerar embeddings contextuais das palavras e Selenium, Robot Framework e Tricentes que são usados para converter os casos de teste manuais gerados em casos de teste automático.
Leveraging Conditional Statement to Generate Acceptance Tests Automatically via Traceable Sequence Generation	No estudo em questão é proposto uma nova abordagem chamada de *Traceable Sequence Generation* (TSG) baseado em Seq2Seq ,essa abordagem busca implementar a geração automática de casos de testes de aceitação a partir de requisitos de software escritos em linguagem natural, apoiando-se em declarações condicionais extraídas dos requisitos. Dentre as ferramentas descritas no estudo, temos: o BERT e BiLSTM como codificadores para processar o texto dos requisitos

Continua na próxima página

Título	Metodologia/Ferramenta
Test case information extraction from requirements specifications using NLP-based unified boilerplate approach	O estudo utilizou ferramentas como NLTK e WordNet nas etapas de pré-processamento e extração dos dados dos requisitos. Também foi utilizado técnicas como pipeline de PLN, busca de strings na identificação dos atores, e no tipo do requisito. Somado a um boilerplate unificado.

Fonte: Elaborado pelo autor.

7.2 QUESTÃO DE PESQUISA 02

QP2: Quais são as vantagens reportadas sobre a aplicação de PLN na derivação de casos de teste?

Dentre as vantagens apresentadas nos trabalhos avaliados podemos trazer alguns benefícios principais, como a redução do tempo e do esforço manual necessário para a criação dos casos de teste, também a ampliação na abrangência dos casos de teste em relação aos requisitos de software, além de uma melhoria na eficácia na detecção de defeitos. Ademais, a automação embasada no uso do PLN possibilitou em alguns contextos a eliminação do viés humano, fazendo com que houvesse uma geração de casos de teste mais eficientes. A Tabela 10 detalha as vantagens da aplicação do PLN no contexto da geração de casos de teste, com base nos resultados encontrados nos estudos avaliados.

Tabela 10 – Vantagens destacadas nos estudos analisados

Título	Vantagens
Natural Language Processing Technology Based on Artificial Intelligence in Software Testing	Dentre as vantagens da tecnologia BERT citadas no estudo, estão: A alta cobertura dos casos de teste gerados pela tecnologia. A notável precisão e eficiência na detecção de defeitos, no qual no experimentos realizados a precisão da detecção se manteve acima de 82,4% além de possuir um ciclo de regressão máximo de apenas 599ms

Continua na próxima página

Título	Vantagens
Domain-Specific Software System Testing by Using Intelligent Approaches	O design de arquitetura proposto pelo artigo apresenta algumas vantagens como na parte de automação a arquitetura foi projetada para automatizar a geração de casos de testes, no trato com a ambiguidade da linguagem, a arquitetura sugere um modelo aprimorado que integra IA e PLN e para lidar com mudanças frequentes nos requisitos, a arquitetura reprocessa os novos dados de entrada, ajustando os casos de teste.
Naive Bayes Classification Model for Precondition-Postcondition in Software Requirements	O modelo proposto, combinando as duas bibliotecas NLTK e Scikit-learn, permitiu uma forma de validação cruzada entre algumas implementações do algoritmo Naive Bayes, o que resultou em resultados positivos para cada uma das abordagens. Em relação ao Scikit-learn ele se destacou pelo desempenho em lidar com dados pré-rebalanceados, já o NLTK obteve resultados superiores com dados pós-rebalanceados.
Intelligent requirement-to-test-case traceability system via Natural Language Processing and Machine Learning	O estudo trouxe uma série de melhorias, como, uma redução no tempo de processamento para o mapeamento de requisitos ao utilizar a combinação da similaridade de cosseno com a distância de Jaccard por meio da biblioteca NLTK. Outra melhoria que pode ser destacada a geração automática de casos de testes utilizando o modelo BERT, o que promoveu uma maior produtividade.

Continua na próxima página

Título	Vantagens
Automated Test Case Generation for Satellite FRD Using NLP and Large Language Model	O estudo sugere que com a adoção de técnicas avançadas de LLM, houve um aumento na precisão dos casos de teste, também demonstrou uma redução significativa no tempo e no esforço necessários para a construção do caso de teste quando comparado com abordagens tradicionais.
Leveraging Pre-Trained Large Language Models (LLMs) for On-Premises Comprehensive Automated Test Case Generation: An Empirical Study	Dentre os benefícios abordados no estudo podemos trazer: A redução do esforço manual, a melhora na eficiência e na escalabilidade do processo de *Quality Assurance* (QA). Além da abordagem diminuir significativamente o tempo necessário gasto na criação dos casos de teste.
Automatic Generation of Acceptance Test Cases from Use Case Specifications	O artigo apresentou alguns beneficios em relação a abordagem UMTG, apontado que houve uma redução do esforço manual para a criação dos casos de teste, também garantiu uma alta cobertura dos requisitos pelos testes, além de constatar que a abordagem teve uma boa desenvoltura para lidar com contexto de especificações frequentemente atualizadas.
Automatic creation of acceptance tests by extracting conditionals from requirements: NLP approach and case study	A utilização da ferramenta CiRA trouxe uma série de vantagens, dentre os principais benefícios citados no estudo, podemos listar: a eficiência na geração de casos de teste, além da redução do viés humano, com base na capacidade do CiRA de criar testes fundamentados em heurísticas.

Continua na próxima página

Título	Vantagens
Leveraging Conditional Statement to Generate Acceptance Tests Automatically via Traceable Sequence Generation	A abordagem TSG apresentou algumas vantagens, como a automatização do processo de testes de aceitação reduzindo tempo e esforço gastos para a geração de casos de teste e também apresentou casos de teste com uma alta cobertura de requisitos em comparação com métodos tradicionais e outros modelos baseados em PLN.
Test case information extraction from requirements specifications using NLP-based unified boilerplate approach	A abordagem catalisa a geração de casos de teste, pois reduz esforço humano, tempo e custo empregados nesse processo, também é capaz de processar requisitos positivos e negativos.

Fonte: Elaborado pelo autor.

7.3 QUESTÃO DE PESQUISA 03

QP3: Quais são as limitações apontadas na extração de casos de teste com o auxílio do PLN?

Tratando-se dos 10 estudos avaliados, somente 8 reportaram as limitações contidas na abordagem discorrida. Sendo assim, dentre os principais desafios identificados nos estudos temos: Problemas atrelados a alta dependência da qualidade dos dados de entrada, necessitando de uma avaliação manual nas saídas geradas, e em alguns contextos é exigido uma complexidade técnica do profissional executor do processo. Outras questões apontadas envolvem o desafio de gerenciar e lidar com requisitos mal formulados, além disso, em algumas situações há uma alta demanda de recursos computacionais. Essas limitações evidenciam lacunas e pontos de melhoria nas abordagens e ferramentas avaliadas. A Tabela 11 detalha as limitações encontradas nos estudos analisados.

Tabela 11 – Limitações reportadas nos estudos analisados

Título	**Limitações**
Natural Language Processing Technology Based on Artificial Intelligence in Software Testing	A tecnologia BERT apresenta alguns desafios como a grande dependência na qualidade dos dados de entrada, além de exigir um alto conhecimento do profissional que irá lidar com o modelo, somada ao consumo elevado de recursos computacionais
Naive Bayes Classification Model for Precondition-Postcondition in Software Requirements	A validade do modelo proposto no artigo foi ameaçada por algumas limitações. Sendo a amostra restrita, limitada de requisitos de software analisados como o principal ponto limitante.
Intelligent requirement-to-test-case traceability system via Natural Language Processing and Machine Learning	Dentre as limitações apontadas pelo estudo, uma relacionada ao modelo de linguagem BERT, ganha destaque, o artigo reporta que mesmo com a geração automática das histórias de teste, cenários e etapas, as saídas do BERT ainda requerem avaliações manuais e ajustes, pois não alcaçam o nível de expertise dos engenheiros de teste.
Automated Test Case Generation for Satellite FRD Using NLP and Large Language Model	Tratando-se das limitações apontadas no estudo, os principais pontos levantados foram em relação aos LLMs como o GPT-4 e o Llama-2-7ob por questões como: Problemas de confidencialidade, falta de transparência ao utilizar modelos de código fechado, alta demanda por recursos computacionais, em modelos de código aberto somado a forte restrição do formato de dados de entrada

Continua na próxima página

Título	Limitações
Leveraging Pre-Trained LLMs for On-Premises Automated Test Case Generation	A limitação principal apontada no estudo sobre o aproveitamentode LLMs pré-treinados para gerar casos de teste, foi a generalização dos resultados, no qual os casos de teste para produtos em diferentes áreas tiveram uma alta variação em termos de estilo, detalhe, números e granularidade.
Automatic Creation of Acceptance Tests by Extracting Conditionals from Requirements	Dentre as limitações apontadas no estudo, podemos listar: A forte dependência da qualidade dos requisitos, com dificuldade para lidar com requisitos mal formulados, também apresenta limitações não sendo capaz de lidar com sentenças compostas ou condicionais que se estendam por mais de uma sentença, além disso, apresenta problemas com requisitos que envolvam cadeias de evento.
Leveraging Conditional Statement to Generate Acceptance Tests Automatically	Em relação às limitações apontadas pelo estudo, podemos listar o principal desafio descrito no estudo, que seria em relação a complexidade na geração de teste em contextos onde as declarações condicionais envolvem múltiplos antecedentes ou efeitos, podendo resultar em inconsistências nos casos de teste gerados.
Test case information extraction from requirements specifications using NLP-based unified boilerplate approach	Alta dependência de palavras-chave no requisito, desconsiderando UML e outras informações. Somado a identificação errônea de atores, quando há ambiguidade na linguagem.

Fonte: Elaborado pelo autor.

8 CONSIDERAÇÕES FINAIS

Este estudo apresentou um mapeamento sistemático da literatura acerca do uso do processamento de linguagem natural na extração de casos de testes a partir de requisitos de software escritos em linguagem natural. Para alcançar o objetivo do estudo, buscou-se respostas para três questões de pesquisa, para elucidar a primeira QP foi investigado quais técnicas, abordagens e ferramentas do PLN são mais exploradas no contexto dos casos de teste, visando fornecer um estado da arte do tema sob análise. Em relação a segunda QP buscou-se esclarecer quais as reais vantagens de se aplicar tais estratégias de PLN no âmbito de teste de software. Já, no que se refere a terceira QP, foi analisado buscando-se estruturar quais limitações eram reportadas nos estudos selecionados.

Sendo assim, após a implementação dos filtros e dos critérios de inclusão e exclusão, foram selecionados 10 estudos. Com base nos trabalhos escolhidos, foi possível responder às três questões de pesquisa. A sistematização dos estudos mostrou que, dentre as principais abordagens e técnicas, o BERT ganha destaque, sendo a técnica principal em quatro estudos. Sobre as vantagens da utilização do PLN no âmbito investigado, foram constatados alguns pontos, como a otimização do tempo para a criação dos casos de teste, uma maior cobertura dos requisitos pelos casos de teste, além de uma redução do esforço manual por parte do executor do teste. Sobre as limitações, em suma, a questão mais apontada foi a necessidade de uma alta qualidade dos dados de entrada. Deste modo, ao obter e apresentar as respostas para as três QPs, o projeto cumpriu seu objetivo principal.

Dessa forma, o trabalho resultou numa base organizada para futuras investigações, podendo auxiliar na escolha de abordagens que melhor se adequem ao contexto do pesquisador. O estudo identificou e estruturou os benefícios e lacunas presentes nos trabalhos. Firmando um panorama sobre as técnicas, abordagens, ferramentas nos últimos 3 anos. Baseando-se nos resultados alcançados e em relação aos trabalhos futuros, pretende-se seguir com o mapeamento para os próximos anos a frente de 2025, podendo incorporar outras bases, além de bases com conteúdo pago. Assim, o prosseguimento do estudo pode explorar a grande potencialidade envolvida no PLN como auxilio para a geração de casos de teste.

REFERÊNCIAS

AGHAMOHAMMADI, A.; MIRIAN-HOSSEINABADI, S.-H.; JALALI, S. Statement frequency coverage: A code coverage criterion for assessing test suite effectiveness. **Information and Software Technology**, v. 129, p. 106426, 01 2021.

AHSAN, I. *et al.* A comprehensive investigation of natural language processing techniques and tools to generate automated test cases. In: **Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing**. New York, NY, USA: Association for Computing Machinery, 2017. (ICC '17). ISBN 9781450347747. Disponível em: <https://doi.org/10.1145/3018896.3036375>.

BIOLCHINI, J. *et al.* Systematic review in software engineering. 01 2005.

BOUKHLIF, M. *et al.* Natural language processing-based software testing: A systematic literature review. **IEEE Access**, v. 12, 2024.

BRERETON, P. *et al.* Lessons from applying the systematic literature review process within the software engineering domain. **Journal of Systems and Software**, v. 80, p. 571–583, 04 2007.

CHAUHAN, P.; BALAKRISHNAN, V. Domain-specific software system testing by using intelligent approaches. In: **2024 International Conference on Computational Intelligence and Network Systems (CINS)**. [S.l.: s.n.], 2024. p. 1–5.

FADHLURROHMAN, D. H. *et al.* Naive bayes classification model for precondition-postcondition in software requirements. In: **2023 International Conference on Data Science and Its Applications (ICoDSA)**. [S.l.: s.n.], 2023. p. 123–128.

FISCHBACH, J. *et al.* Automatic creation of acceptance tests by extracting conditionals from requirements: Nlp approach and case study. **Journal of Systems and Software**, v. 197, 2023.

GAROUSI, V.; BAUER, S.; FELDERER, M. Nlp-assisted software testing: A systematic mapping of the literature. **Information and Software Technology**, v. 126, p. 106321, 2020. ISSN 0950-5849. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584920300744>.

JURAFSKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. [S.l.: s.n.], 2008. v. 2.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. v. 2, 01 2007.

LELOUDAS, P. **Introduction to Software Testing**. Berkeley, CA: Apress, 2023.

LIM, J. W. *et al.* Test case information extraction from requirements specifications using nlp-based unified boilerplate approach. **Journal of Systems and Software**, v. 211, p. 112005, 2024. ISSN 0164-1212. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121224000487>.

LIM, J. W. *et al.* Test case information extraction from requirements specifications using nlp-based unified boilerplate approach. **JOURNAL OF SYSTEMS AND SOFTWARE**, v. 211, MAY 2024. ISSN 0164-1212.

LIU, Y. Natural language processing technology based on artificial intelligence in software testing. In: **2024 3rd International Conference on Artificial Intelligence and Computer Information Technology (AICIT)**. [S.l.: s.n.], 2024. p. 1–6.

MAFRA, S.; TRAVASSOS, G. **Estudos primários e secundários apoiando a busca por evidência em engenharia de software**. [S.l.], 2006.

NAZIR, F. *et al.* The applications of natural language processing (nlp) for software requirement engineering - a systematic literature review. In: . [S.l.: s.n.], 2017. p. 485–493. ISBN 978-981-10-4153-2.

PETERSEN, K. *et al.* Systematic mapping studies in software engineering. **Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering**, v. 17, 06 2008.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: AMGH, 2016.

S, S. *et al.* Automated test case generation for satellite frd using nlp and large language model. In: **2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)**. [S.l.: s.n.], 2024. p. 1–8.

SAHNI, B. Quality assurance / software testing – traditional to modern. **Journal of Mathematical Computer Applications**, p. 1–3, 09 2022.

SAWADA, K. *et al.* Intelligent requirement-to-test-case traceability system via natural language processing and machine learning. In: **2023 IEEE 9th International Conference on Space Mission Challenges for Information Technology (SMC-IT)**. [S.l.: s.n.], 2023. p. 78–83.

SCANNAVINO, K. R. F. *et al.* **Revisão Sistemática da Literatura em Engenharia de Software: teoria e prática**. [S.l.]: Elsevier, 2017.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

WANG, C. *et al.* Automatic generation of acceptance test cases from use case specifications: An nlp-based approach. **IEEE Transactions on Software Engineering**, v. 48, n. 2, p. 585–616, 2022.

WOHLIN, C. *et al.* On the reliability of mapping studies in software engineering. **Journal of Systems and Software**, v. 86, p. 2594–2610, 10 2013.

YIN, H.; MOHAMMED, H.; BOYAPATI, S. Leveraging pre-trained large language models (llms) for on-premises comprehensive automated test case generation: An empirical study. In: **2024 9th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)**. [S.l.: s.n.], 2024. v. 9, p. 597–607.

ZHAO, G. *et al.* Leveraging conditional statement to generate acceptance tests automatically via traceable sequence generation. In: . [S.l.: s.n.], 2023. p. 394 – 405.