

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Usando Grandes Modelos de Linguagem (GML) para Automatizar o Processo
de Seleção de Artigos no Mapeamento Sistemático em Engenharia de Software

FILIPPE REZENDE DA SILVA

Orientador

Prof. Dr. Carlos Eduardo Ribeiro de Mello

Coorientador

Prof. Dr. Márcio de Oliveira Barros

Rio de Janeiro, RJ - Brasil

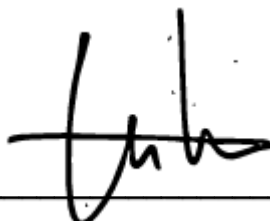
Setembro de 2025

Usando Grandes Modelos de Linguagem (GML) para Automatizar o Processo de Seleção de Artigos no Mapeamento Sistemático em Engenharia de Software

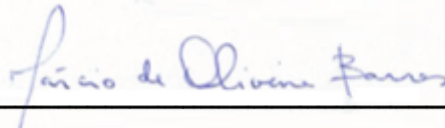
FILIPE REZENDE DA SILVA

PROPOSTA DE DISSERTAÇÃO APRESENTADA COMO REQUISITO PARA OBTENÇÃO DE TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA

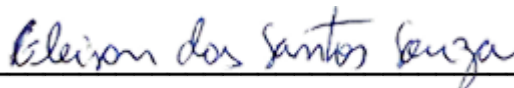
Aprovada por:



Prof. Dr. Carlos Eduardo Ribeiro de Mello - UNIRIO



Prof. Dr. Márcio de Oliveira Barros - UNIRIO



Prof. Dr. Gleison dos Santos Souza - UNIRIO



Prof. Dr. Jorge de Abreu Soares – CEFET - RJ

Rio de Janeiro, RJ - Brasil

Setembro de 2025

Catálogo informatizada pelo(a) autor(a)

- S586 Silva, Filipe Rezende da
Usando Grandes Modelos de Linguagem (GML) para
Automatizar o Processo de Seleção de Artigos no Mapeamento
Sistemático em Engenharia de Software / Filipe Rezende da
Silva, Carlos Eduardo Ribeiro de Mello, Márcio de Oliveira
Barros. -- Rio de Janeiro : UNIRIO, 2025.
60 f
- Orientador: Carlos Eduardo Ribeiro de Mello.
Coorientador: Márcio de Oliveira Barros.
Dissertação (Mestrado) - Universidade Federal do Estado
do Rio de Janeiro, Programa de Pós-Graduação em Informática,
2025.
1. Grandes Modelos de Linguagem (GML). 2. Mapeamento
Sistemático. 3. Seleção de Artigos. I. Mello, Carlos
Eduardo Ribeiro de II. Barros, Márcio de Oliveira III.
Mello, Carlos Eduardo Ribeiro de, orient. IV. Barros,
Márcio de Oliveira, coorient. V. Título.

“Esta é a vida eterna: que te conheçam, o único Deus verdadeiro, e a Jesus Cristo, a quem enviaste.”

Evangelho de Jesus segundo João capítulo 17, versículo 3 na Nova Versão Internacional (NVI).

Agradecimentos

Em primeiro lugar, agradeço a Deus, fonte de toda sabedoria e inspiração. Foi Ele quem sustentou minha caminhada com graça e misericórdia, renovando minhas forças nos momentos de cansaço e abrindo caminhos quando tudo parecia difícil. Reconheço que sem Sua presença e direção, nada disso seria possível, e por isso toda honra e glória pertencem a Ele.

A minha família, agradeço profundamente pelo amor incondicional e pela paciência durante este período desafiador. Em especial, à minha esposa Nathali, companheira fiel e encorajadora em todas as fases desta jornada, que nunca deixou de acreditar em mim, mesmo quando eu duvidava. À minha filha Estela, minha motivação diária para ser alguém melhor, que com sua alegria e pureza de coração me ensinou o verdadeiro sentido do esforço e da perseverança.

Aos meus pais, Maurício e Maria de Fátima, que sempre acreditaram no valor da educação e me ensinaram desde cedo a importância da disciplina e da honestidade, deixo minha eterna gratidão. Agradeço também aos meus irmãos, Fábio e Vinícius, que são inspirações sempre presentes na minha vida, minha família de origem é um alicerce firme que fortalece cada conquista minha.

Por fim, registro minha profunda gratidão aos meus orientadores, Carlos Eduardo e Márcio Barros, e a todos os professores e funcionários do programa de pós-graduação de Informática da Unirio. Cada orientação, cada aula e cada gesto de apoio foi essencial para o amadurecimento desta pesquisa e para minha formação acadêmica. Sou imensamente grato pelo profissionalismo, dedicação e pela paciência em guiar este processo, que culmina nesta dissertação.

Com apreço de Filipe Rezende da Silva

RESUMO

O objetivo desta pesquisa é investigar a aplicação de Grandes Modelos de Linguagem (GML) pré-treinados na seleção de artigos relevantes para mapeamentos sistemáticos da literatura em pesquisa em Engenharia de Software. Avaliamos se parte do esforço do pesquisador pode ser automatizada pelo GML, com o humano atuando como um segundo revisor para verificar a qualidade e a consistência dos artigos selecionados pelo modelo. Selecionamos seis artigos de mapeamento sistemático na área de Engenharia de Software para comparar a lista de artigos selecionados por pesquisadores humanos com a lista de artigos selecionados por diferentes GMLs. Para isso, tanto o resumo de cada artigo quanto a expressão de busca submetida às bases de dados de publicações, juntamente com um prompt projetado para analisar a relação entre os termos, foram submetidos aos GMLs. Por sua vez, os GMLs retornaram uma pontuação de relevância para cada artigo retornado pelas bases de dados de publicações dada a expressão de busca. Nossos principais resultados indicam que a similaridade entre os resumos dos artigos e a expressão de busca pode ser usada pelos GMLs para avaliar em que medida cada artigo é relevante. Ao considerar os 200 artigos mais relevantes selecionados por cada GML para cada um dos seis mapeamentos sistemáticos de referência, encontramos uma cobertura média de 79% dos artigos aceitos e uma economia de tempo de 67% na seleção inicial de artigos.

Palavras-chave: Grandes Modelos de Linguagem (GML), mapeamento sistemático, engenharia de software, seleção de artigos.

ABSTRACT

The objective of this research is to investigate the application of pre-trained Large Language Models (LLM) in the selection of relevant papers for systematic mappings of the literature in Software Engineering research. We evaluate whether part of the researcher's effort can be automated by the LLM, with the human acting as a second reviewer to verify the quality and consistency of the papers selected by the model. We selected six systematic mapping papers in the field of Software Engineering to compare the list of papers selected by human researchers with the list of papers selected by different LLMs. For this, both the abstract of each paper and the search expression submitted to the publication databases, along with a prompt designed to analyze the relationship between the terms, were submitted to the LLMs. On their turn, the LLMs returned a relevance score for each paper returned by the publication databases given the search expression. Our main results indicate that the similarity between the abstracts of the papers and the search expression can be used by the LLMs to evaluate to which extent each paper is relevant. While considering the 200 most relevant papers selected by each LLM for each of the six benchmark systematic mappings, we found an average coverage of 79% of the accepted papers and a time savings of 67% in the initial selection of articles.

Keywords: Large Language Models (LLM), systematic mapping, software engineering, article selection.

Lista de Figuras

Figura 1 - Elaboração do processo de mapeamento sistemático.....	8
--	---

Lista de Tabelas

Tabela 1 - Descrição da estratégia PICO para formulação da pergunta de pesquisa	9
Tabela 2 - Informações sobre os mapeamentos sistemáticos analisados para compor a avaliação experimental	30
Tabela 3 - Expressão de busca dos mapeamentos sistemáticos avaliados	31
Tabela 4 - Resultado da replicação da expressão de busca dos artigos	32
Tabela 5 - Resultado consolidado da replicação da expressão de busca	32
Tabela 6 – Refinamento de busca em cada mapeamento	33
Tabela 7 – Índices dos modelos de incorporação de palavras	34
Tabela 8 - Valores de acurácia, precisão e especificidade pelo modelo de incorporação de palavras	40
Tabela 9 – Grau de cobertura dos selecionados no TOP 200 para os modelos de incorporação de palavras	41
Tabela 10 – Grau de cobertura dos selecionados no TOP 200 para os modelos de GML	42
Tabela 11 - Análise comparativa da incorporação de palavras (IP) – all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS1	45
Tabela 12 - Análise comparativa da incorporação de palavras (IP) – all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS2	45
Tabela 13 - Análise comparativa da incorporação de palavras (IP) – all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS3	46
Tabela 14 - Análise comparativa da incorporação de palavras (IP) – all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS4	47
Tabela 15 - Análise comparativa da incorporação de palavras (IP) – all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS5	47
Tabela 16 - Análise comparativa da incorporação de palavras (IP) – all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS6	48
Tabela 17 – Percentual correspondente de 200 artigos em relação ao total de artigos disponíveis	49
Tabela 18 - Critérios de Ordenação por Relevância das Principais Bases de Dados: ACM Digital Library, IEEE Xplore e Scopus	50

Sumário

1	Introdução.....	1
1.1	Motivação.....	1
1.2	Objetivo.....	3
1.3	Justificativa	5
1.4	Metodologia Geral	5
1.5	Estrutura da Dissertação	7
2	Revisão da Literatura	8
2.1	Fundamentação Teórica.....	8
2.1.1	Mapeamento Sistemático da Literatura	8
2.1.2	<i>Grandes Modelos de Linguagem</i>	11
2.1.3	Aprendizado de Máquina	14
2.1.4	Modelos de Incorporação de Palavras	15
2.1.5	K-Nearest Neighbors (KNN).....	20
2.1.6	Matriz de Confusão	21
2.2	Trabalhos Relacionados	23
2.3	Considerações Finais	27
3	Metodologia de Pesquisa	28
3.1	Proposta de Pesquisa.....	28
3.2	Plano de Experimentos.....	28
3.2.1	Seleção e Replicação de Mapeamentos Sistemáticos	29
3.2.2	Comparação do Resumo com a Expressão de Busca	33
3.2.3	Análise de Cobertura com Número Fixo de 200 Seleccionados...	36
3.2.4	Avaliação de Relevância Usando GML	37
3.3	Considerações Finais	39
4	Resultados e Discussão	40
4.1	Comparação do Resumo com a Expressão de Busca	40
4.2	Cobertura com Número Fixo de 200 Seleccionados.....	41
4.3	Avaliação de Relevância Usando GML	42
4.4	Discussão	48

5	Conclusão.....	52
5.1	Contribuições da pesquisa.....	52
5.2	Limitações da pesquisa	52
5.3	Trabalhos Futuros	53
5.4	Considerações Finais	54

1 Introdução

1.1 Motivação

Segundo Oliveira (2023), Grandes Modelos de Linguagem (GML) são uma classe de modelos de processamento de linguagem natural baseados em algoritmos de aprendizado profundo. O objetivo desses modelos é aprender padrões a partir de textos, representando estruturas, contexto e semântica da linguagem empregada. Desta forma, várias tarefas que envolvem o processamento de linguagem natural podem ser executadas a partir dos GML.

Ainda de acordo com Oliveira (2023), os GML podem executar uma variedade de tarefas, como compreender estruturas de proteínas, escrever código de software, resolver problemas de classificação de texto, responder perguntas, resumir documentos, gerar texto e muito mais. Sua capacidade de resolução de problemas pode ser aplicada a áreas como saúde, finanças e entretenimento. Eles também atendem a uma variedade de aplicações de Processamento de Linguagem Natural (PLN), como tradução, comunicação com usuários como *chatbots*, assistentes inteligentes, e assim por diante.

Dentre as possibilidades de aplicações de GML, nesta pesquisa abordamos o seu uso no processo de mapeamento sistemático da literatura na área de Engenharia de Software. Segundo Petersen (2008), o mapeamento sistemático é uma abordagem de pesquisa que tem como objetivo identificar, avaliar e mapear de forma sistemática e abrangente a extensão, o alcance e a natureza do corpo de literatura sobre um tópico específico.

Embora um estudo de mapeamento sistemático e uma revisão sistemática da literatura partilhem alguns pontos em comum, segundo Petersen (2015), no que diz respeito à pesquisa e seleção de estudos, estes são diferentes em termos de objetivos. As revisões sistemáticas visam sintetizar evidências, como um meio de avaliar e interpretar todas as pesquisas relevantes a uma questão de interesse. Por outro lado, os mapeamentos sistemáticos têm um escopo mais amplo e seu objetivo é a estruturação de uma área de pesquisa

com o fim de fornecer uma visão geral e organizada da área, identificando lacunas, tendências e padrões na literatura existente.

Segundo Kitchenham (2007), um mapeamento sistemático permite que as evidências em um domínio sejam visualizadas em um nível alto de granularidade. Isto facilita a identificação de evidências e/ou a falta delas para direcionar o foco de futuras revisões sistemáticas e identificar áreas para que mais estudos primários sejam realizados.

Logo, a revisão sistemática da literatura (RSL) e o mapeamento sistemático da literatura (MSL) são complementares. Um MSL pode preceder uma RSL visando prover uma visão geral de um tópico de pesquisa. Em relação aos objetivos, RSL agregam os estudos primários em termos de seus resultados e investigam se estes são consistentes ou contraditórios. Por outro lado, os MSL são focados na estruturação de uma área de pesquisa e classificam os estudos primários relevantes em categorias bem definidas. Em relação às questões de pesquisa, RSL agregam resultados relacionados com questões de pesquisa específicas e MSL buscam classificar os estudos primários em subtópicos.

Segundo Scannavino (2017), em relação ao escopo, RSL envolvem menos estudos a serem analisados, mas que devem ser analisados com maior profundidade. Por outro lado, nos MSL o escopo é mais amplo e a análise e síntese dos trabalhos são mais gerais. Em relação ao processo de busca, os mesmos métodos de busca e extração de dados são utilizados, mas os MSL se apoiam na tabulação dos estudos primários em categorias.

De acordo com Petersen (2015), o mapeamento sistemático começa com a escolha das palavras para montar uma *expressão* de busca nas bases de dados, segue para a escolha dos critérios de inclusão e exclusão dos artigos e a leitura parcial dos trabalhos encontrados nas bases. O objetivo é selecionar aqueles trabalhos que estarão no escopo do mapeamento sistemático.

No processo de MSL, a tarefa de leitura e seleção dos artigos é desafiadora por alguns motivos. Primeiro, pelo volume de artigos: dependendo do escopo do mapeamento sistemático, o número de artigos a serem revisados pode ser muito grande. Isso pode sobrecarregar os revisores e tornar o processo

demorado e exaustivo. Segundo, pela variedade de tipos e fontes das bases de dados, já que os artigos a serem revisados podem estar disponíveis como periódicos, conferências, livros e relatórios técnicos. Isso aumenta a diversidade de formatos e estilos de escrita, tornando a leitura e a compreensão dos artigos mais árduas. Um terceiro motivo que torna trabalhosa a leitura e seleção dos artigos que compõem o mapeamento sistemático é a heterogeneidade dos conteúdos. Os artigos podem abordar uma ampla gama de tópicos e aspectos no domínio de interesse. Isso requer que os revisores tenham conhecimento abrangente e especializado sobre o assunto para identificar a relevância dos artigos para incluir no mapeamento.

Waffenschmidt et al. (2019) recomendam que a etapa de leitura seja feita por, no mínimo, dois revisores para amenizar o efeito da interpretação e avaliação subjetiva dos revisores. Desse modo, espera-se evitar possíveis vieses na seleção dos artigos, já que pode haver discrepâncias nas decisões sobre a inclusão ou exclusão de artigos, exigindo consenso entre os revisores. Por isso, é importante ter procedimentos claros e consistentes para a seleção e revisão dos artigos, bem como treinamento adequado para os revisores. A realização de verificações de qualidade, como revisões por pares e verificação da consistência entre os revisores, pode ajudar a mitigar esses desafios.

Neste contexto, a automação parcial ou completa do processo de seleção de artigos possui grande potencial na redução de custo, tempo, vieses e esforço para os revisores. Isto traz à tona a questão de pesquisa endereçada nesta Dissertação: GML podem trazer benefícios ao processo de mapeamento sistemático da literatura? A aplicação direta de GML a cenários e problemas de natureza semelhante vem apresentando resultados de grande valia para o desenvolvimento de ferramentas de produtividade para cientistas e pesquisadores. Neste trabalho, investiga-se algumas estratégias e abordagens de aplicação de GML a mapeamentos sistemáticos da literatura.

1.2 Objetivo

O objetivo desta pesquisa é investigar a aplicação de *Grandes Modelos de Linguagem* (GML) desempenhando o papel de primeiro ou segundo revisor

no processo de seleção de artigos durante a elaboração de um mapeamento sistemático. Deste modo, a hipótese é que os GML poderão aliviar parte da carga de trabalho do pesquisador.

A etapa de seleção, também chamada de triagem, consiste em analisar os títulos e resumos dos estudos recuperados nas bases de dados com o propósito de identificar aqueles que são de fato relevantes de acordo com critérios previamente definidos. Este processo, embora essencial para a qualidade do mapeamento, é frequentemente trabalhoso, demorado e sujeito à variabilidade entre os revisores humanos.

Neste contexto, a proposta da pesquisa é investigar se modelos de linguagem treinados em larga escala – como o ChatGPT, Gemini, DeepSeek e DistilBERT – podem ser utilizados para realizar a triagem automática de artigos com base na similaridade entre os resumos e a expressão de busca utilizada no protocolo do MSL. A hipótese central é que, ao explorar as capacidades semânticas desses modelos, é possível identificar os artigos mais relevantes com uma acurácia comparável à seleção manual, porém com maior eficiência e escalabilidade.

Neste sentido, esta Dissertação investigará modelos de incorporação de palavras, implementações e ferramentas de GML que possam contribuir para avaliar o desempenho destes modelos de linguagem no apoio ao mapeamento sistemático da literatura. Para isto, um conjunto de experimentos quantitativos será conduzido com o intuito de obter dados para uma análise comparativa entre as diferentes possibilidades de abordagens e ferramentas disponíveis de GML no cenário de Engenharia de Software.

O trabalho busca atingir os seguintes objetivos específicos:

- Avaliar o desempenho da seleção automatizada, com base em métricas amplamente utilizadas, como precisão, sensibilidade, especificidade e matriz de confusão, comparando os resultados de diferentes GML com os obtidos por revisores humanos; e
- Simular, através do experimento quantitativo, diferentes cenários de corte, como a seleção dos N artigos mais próximos semanticamente da

expressão de busca aplicada nas bibliotecas digitais, com o intuito de analisar a cobertura dos estudos realmente relevantes.

1.3 Justificativa

Segundo Christou et al. (2024), o processo de elaboração de um mapeamento sistemático requer a leitura dos trabalhos encontrados em bases de artigos por no mínimo dois revisores. Este requisito impõe um custo alto ao mapeamento sistemático, sobretudo por tratar-se de uma tarefa árdua e que consome um tempo significativo. Neste sentido, o uso de um GML no papel de um dos revisores poderia trazer ganhos para redução deste trabalho.

Segundo Kaddour (2023), os *GML* apresentam resultados consistentes de sucesso na aplicação em diversas áreas, como *chatbots*, biologia computacional, programação de computadores, direito, medicina, raciocínio, robótica e ciências sociais. Além disso, o relatório da McKinsey (2023) sobre impacto da IA generativa prevê grande potencial no uso de GML como assistentes virtuais, trazendo ganhos de produtividade.

Portanto, esta pesquisa está em linha com os principais trabalhos que exploram potenciais aplicações da IA generativa. Além disso, a área de mapeamento sistemático possui esta importante lacuna ainda não abordada através das tecnologias de IA, o que é evidenciado na análise dos trabalhos relacionados no próximo capítulo.

O escopo desta pesquisa mantém seus limites no mapeamento sistemático. Esta escolha deveu-se ao fato deste ser um processo mais abrangente, permitindo assim uma investigação de maior envergadura. Entretanto, há possibilidade de adaptações para a revisão sistemática, uma vez que são similares no que se refere à busca e seleção de estudos, sendo diferentes somente em seus objetivos.

1.4 Metodologia Geral

A metodologia da pesquisa consistiu na avaliação do desempenho de quatro GML na tarefa de seleção de artigos no processo de mapeamento

sistemático. Para isso, comparamos a abordagem com GML contra o método manual. A abordagem desenvolvida com o GML utilizou conceitos de modelos de incorporação de palavras, K-Nearest Neighbors (KNN) e matriz de confusão.

Com o intuito de realizar uma análise comparativa, seis artigos de mapeamentos sistemáticos publicados na área de Engenharia de Software foram selecionados e reproduzidos. Cada artigo reproduziu um mapeamento sistemático, gerando um conjunto de resultados que foram utilizados no experimento proposto nesta pesquisa. Os artigos foram escolhidos por serem recentes (foram publicados no período entre 2021 e 2023) e por apresentarem a relação dos artigos aceitos ao final da seleção. Com isso, é possível supervisionar o resultado, criando uma espécie de gabarito para a revisão sistemática. Esses artigos são particularmente úteis em virtude de apresentarem a relação identificada dos artigos selecionados ao final do protocolo, o que é exatamente o se deseja comparar com a seleção realizada pelo GML. As informações desses artigos encontram-se na Tabela 2 da Seção 3.2.

A metodologia adotada inicialmente buscou identificar se os resumos dos artigos aceitos em cada estudo apresentavam semelhanças e estavam próximos entre si e da expressão de busca. Para esta análise, representações numéricas das palavras da busca, dos títulos e resumos dos artigos foram utilizadas para modelar as relações entre busca e artigos. Diferentes modelos de incorporação de palavras foram investigados e avaliados. A distância euclidiana entre artigos representados como vetores de incorporação de palavras foi empregada para a avaliação da performance dos GML.

Todo o processo de comparação é endereçado como um problema de classificação binária, tendo como principal indicador de análise comparativa a matriz de confusão. Os artigos identificados no mapeamento sistemático são marcados como gabarito para serem confrontados com aqueles identificados pelos GML.

1.5 Estrutura da Dissertação

Este trabalho está dividido em cinco capítulos, onde este é o primeiro. No Capítulo 2 será discutida a literatura sobre os processos de mapeamento sistemático e a fundamentação teórica dos principais conceitos, bem como trabalhos relacionados à aplicação de IA neste contexto.

No Capítulo 3, abordaremos o método de pesquisa, descrevendo sua visão geral, discutindo seus pontos fortes e pontos fracos, juntamente com uma proposição de uma avaliação experimental.

No Capítulo 4 serão apresentados os principais resultados das comparações realizadas entre os modelos de incorporação de palavras, GML e a análise manual, além de uma discussão sobre esses resultados.

Por fim, no Capítulo 5 apresentaremos a conclusão dos resultados encontrados e sugestão de trabalhos futuros e próximos passos para o tema.

2 Revisão da Literatura

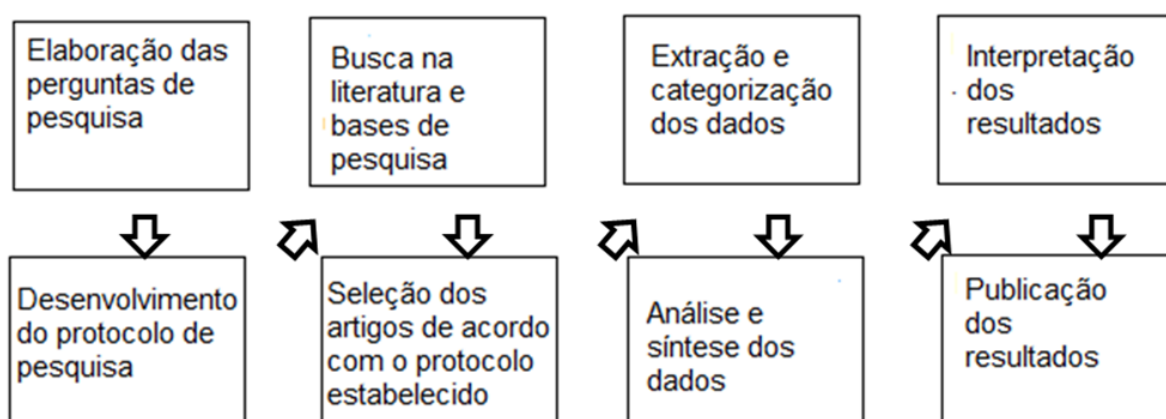
Nesta seção, será apresentada a fundamentação teórica com os principais conceitos utilizados na pesquisa e os trabalhos relacionados.

2.1 Fundamentação Teórica

2.1.1 Mapeamento Sistemático da Literatura

De acordo com Petersen (2015), o processo de elaboração de um mapeamento sistemático envolve várias etapas e requer uma abordagem rigorosa e metodológica. A seguir, serão descritos cada uma das seguintes etapas: (1) elaboração das questões de pesquisa; (2) desenvolvimento do protocolo, (3) busca na literatura; (4) seleção dos artigos; (5) extração dos dados; (6) análise e síntese dos dados; e (7) interpretação e publicação dos resultados. Na Figura 1 é apresentado o processo do mapeamento sistemático.

Figura 1: Elaboração do processo de mapeamento sistemático.



O primeiro passo para elaborar o mapeamento sistemático é formular uma questão de pesquisa clara e específica com o objetivo de mapear e categorizar o conhecimento existente sobre um tópico.

Segundo Kitchenham (2009), a formulação da questão de pesquisa é o primeiro passo na elaboração de um mapeamento sistemático. Neste estágio, os pesquisadores devem identificar o foco da pesquisa e formular uma pergunta clara e específica que orientará o processo. Um método para formular uma

questão de pesquisa é fazer uso do acrônimo PICO, onde “P” indica a população ou problema sendo analisado, “I” indica a intervenção que será realizada, “C” indica a comparação ou controle e “O” indica os resultados esperados (*Outcome*). A Tabela 1 apresenta a descrição da estratégia PICO para formulação da questão de pesquisa que ajuda a direcionar a busca por estudos relevantes e a definir os critérios de inclusão e exclusão.

Tabela 1 - Descrição da estratégia PICO para formulação da pergunta de pesquisa

Acrônimo	Definição	Descrição
P	População ou problema	Um grupo com uma condição particular ou problema de interesse.
I	Intervenção	Intervenção para lidar com o problema ou condição particular da população.
C	Comparação ou controle	Intervenção padrão mais utilizada.
O	<i>Outcome</i> (Desfecho)	Resultado esperado.

Fonte: Adaptado de Santos (2007).

Por exemplo, uma questão de pesquisa utilizando a estratégia PICO poderia ser: qual é o impacto de utilizar modelos de GML (Intervenção) na elaboração de mapeamento sistemático (População) como revisor (Comparação) na seleção de artigos (*Outcome*)?

O segundo passo do processo de realização de um mapeamento sistemático é o desenvolvimento do protocolo, que é essencial para garantir a transparência, a consistência e a objetividade do mapeamento. Neste estágio, os pesquisadores detalham todos os aspectos do plano de pesquisa, incluindo os objetivos, métodos de busca, critérios de inclusão e exclusão de trabalhos primários, estratégias de busca, avaliação da qualidade dos estudos, análise de dados e interpretação dos resultados. O protocolo serve como um guia detalhado do processo de mapeamento sistemático.

Estabelecer um protocolo para o mapeamento sistemático guiará a seleção dos artigos com base em critérios de inclusão e exclusão definidos, assim como as bases de onde os dados serão extraídos. A definição da

metodologia de pesquisa consistirá em delinear o tema e os objetivos, de acordo com as questões de pesquisa que se busca responder por meio da revisão da literatura.

O terceiro passo é a busca por estudos na literatura sobre o tema, visando identificar todas as evidências relevantes disponíveis sobre o tema do mapeamento. É importante documentar todas as etapas da busca, incluindo os termos de pesquisa utilizados e os resultados obtidos, para garantir a transparência e a replicabilidade do processo.

Os pesquisadores realizam uma busca sistemática em bases de dados eletrônicas, como *Engineering Village*, *Scopus* e *Web of Science*, utilizando uma combinação de termos de pesquisa relacionados à questão de pesquisa. A essa combinação de termos de pesquisa dá-se o nome de expressão de busca. A expressão de busca utilizada nas bibliotecas digitais para encontrar trabalhos relevantes sobre o tema de pesquisa é composta por termos chaves e operadores lógicos (AND, OR, NOT) para integrar os termos chaves com seus sinônimos.

O quarto passo na elaboração de um mapeamento sistemático é a seleção dos artigos com base nos critérios de inclusão e exclusão estabelecidos no protocolo. Inicialmente, os títulos e resumos dos artigos identificados durante a busca são examinados para determinar a relevância em relação à questão de pesquisa. Em seguida, os textos completos dos estudos potencialmente relevantes são revisados para confirmar sua inclusão no mapeamento. É uma boa prática que este processo seja conduzido por pelo menos dois revisores independentes para garantir a consistência e minimizar o viés na seleção dos estudos, criando a oportunidade para a aplicação de *GML* nesse estágio de elaboração do mapeamento.

O quinto passo é a extração e categorização dos estudos. Os artigos extraídos dos documentos incluídos no mapeamento sistemático são categorizados e organizados de acordo com as características relevantes do tópico de pesquisa. Isso pode envolver a criação de uma taxonomia ou estrutura

de classificação para categorizar os documentos com base em diferentes aspectos do tópico.

O sexto passo é a análise dos dados e resultados do mapeamento sistemático, que envolve a identificação de padrões, tendências e lacunas no conhecimento existente sobre o tópico, bem como a identificação de áreas de pesquisa futura. O último passo é a interpretação e publicação dos resultados encontrados, descrevendo todos os aspectos do processo, desde a questão de pesquisa até a interpretação dos resultados encontrados.

Peterson (2015) propõe a realização de um estudo de mapeamento sistemático precedendo uma revisão sistemática para fornecer uma base sobre o tema de interesse. O autor relaciona uma série de benefícios de se fazer uma revisão sistemática em um tema que já tenha um mapeamento sistemático, como a existência de uma visão geral sobre uma área, a capacidade de identificar lacunas de investigação, a visualização de tendências de pesquisa, a identificação de trabalhos relacionados e o uso como conjunto de validação com base em referências coletadas, fonte educacional e economia de tempo.

2.1.2 Grandes Modelos de Linguagem

De acordo com a Vaswani et al. (2017), um Grande Modelo de Linguagem (GML) é um algoritmo de aprendizado profundo que pode executar diversas tarefas de Processamento de Linguagem Natural (PLN). Para isso, ele é treinado a partir de conjuntos de dados grandes e volumosos, o que lhe permite reconhecer, traduzir, prever e gerar texto ou outro conteúdo utilizando modelos transformadores.

O GML é baseado em um modelo transformador e funciona recebendo uma entrada, codificando-a e depois decodificando-a para produzir uma previsão de saída. Mas antes de um GML poder receber uma entrada de texto e gerar uma previsão de saída, ele precisa de treinamento, para poder cumprir funções gerais, e ajuste fino, que lhe permite executar tarefas específicas.

De acordo com Elastic (2023), GMLs são pré-treinados usando grandes conjuntos de dados textuais que consistem em trilhões de palavras. A qualidade

deste conjunto de treinamento afeta o desempenho do modelo. Nesta fase, o GML executa um processo de aprendizado não-supervisionado, ou seja, ele processa os conjuntos de dados que lhe são fornecidos sem anotações, aprendendo nesse processo o significado das palavras, as relações entre as palavras e a distinguir palavras com base no contexto.

Para que um GML execute uma tarefa específica, como uma tradução, ele deve ser ajustado para essa atividade. A esse processo, se dá o nome de ajuste fino. O ajuste de *prompt*, instrução dada a um GML, cumpre uma função semelhante ao ajuste fino, treinando um modelo para executar uma tarefa específica por meio de *prompts* de poucos disparos ou de disparo zero, ensinando o modelo a prever resultados por meio do uso de exemplos. Segundo Souza (2023), *prompt* é o processo que envolve a elaboração de instruções ou consultas para extrair as respostas desejadas do GML, fornecendo instruções específicas no intuito de melhorar a qualidade e relevância do resultado.

De acordo com Radford (2018), o *prompting* de disparo zero é uma técnica que usa os recursos de generalização dos GMLs para tentar novas tarefas sem treinamento ou exemplos específicos anteriores. Ele usa o extenso pré-treinamento dos GMLs em conjuntos de dados grandes e diversificados, permitindo que eles apliquem seu amplo conhecimento a novas tarefas com base apenas em instruções claras e concisas. Embora essa técnica possa ser altamente eficaz, seu sucesso depende da complexidade da tarefa e da qualidade do *prompt*.

Em geral, o desempenho do GML aumenta à medida que mais dados e parâmetros são adicionados. Quanto mais eles aprendem, melhores ficam. Além do mais, os GMLs podem exibir o que é chamado de aprendizado em contexto. Depois que um GML foi pré-treinado, o *prompt* de poucos disparos permite que o modelo aprenda sem quaisquer parâmetros adicionais.

Segundo Vaswani et al. (2017), um modelo transformador é a arquitetura mais comum de GML. Ele é uma rede neural que aprende o contexto aplicando um conjunto em evolução de técnicas matemáticas, chamadas de atenção ou

auto atenção, para detectar as maneiras sutis como até mesmo elementos de dados distantes em uma série influenciam e dependem uns dos outros.

Um transformador consiste em um codificador e um decodificador. Um modelo transformador processa dados *tokenizando* a entrada e, em seguida, conduzindo simultaneamente equações matemáticas para descobrir relações entre os *tokens*. Isso permite que o computador veja os padrões que um ser humano veria se recebesse a mesma consulta.

Ainda de acordo com Vaswani et al. (2017), os transformadores trabalham com mecanismos de auto atenção, o que lhes permite aprender mais rapidamente do que os modelos tradicionais de memória de longo e curto prazo. A auto atenção é o que possibilita ao modelo transformador considerar diferentes partes da sequência ou todo o contexto de uma frase para gerar previsões.

Os principais componentes do *GML* são as múltiplas camadas de redes neurais, como as camadas recorrentes, camadas *feedforward*, camadas de *embedding* e camadas de atenção, que trabalham em conjunto para processar o texto de entrada e gerar o conteúdo de saída.

A camada de *embedding* cria um tipo especial de representação de palavras distribuídas a partir do texto de entrada. Essa parte do *GML* captura o significado semântico e sintático da entrada em linguagem natural para que o modelo possa compreender o contexto.

A camada *feedforward* é composta de várias camadas totalmente conectadas que transformam os *embeddings* de entrada. Ao fazer isso, essas camadas permitem que o modelo colete abstrações de nível mais alto e entenda a intenção do usuário com a entrada de texto. A camada recorrente interpreta as palavras no texto de entrada em sequência e captura a relação entre as palavras em uma frase.

O mecanismo de atenção possibilita que um modelo de linguagem se concentre em partes únicas do texto de entrada que sejam relevantes para a tarefa em questão, gerando resultados mais precisos.

Os GML são uma ferramenta tecnológica e, como tal, enfrentam vários desafios, como a alucinação, que ocorre quando um GML produz um resultado falso ou que não corresponde à intenção do usuário porque não conseguem interpretar totalmente o significado humano, ainda que a frase esteja sintaticamente correta.

Outro desafio é que os dados usados para treinar modelos de linguagem afetam os resultados produzidos por um determinado modelo. Como tal, se os dados representarem um único grupo demográfico ou não tiverem diversidade, os resultados produzidos pelo GML carecem de diversidade e usuários com intenções maliciosas podem reprogramar a IA de acordo com as suas ideologias ou preconceitos e contribuir para a propagação de desinformação.

Além disso, os GML são treinados em trilhões de conjuntos de dados, e muitos desses dados podem não ter sido obtidos consensualmente, comprometendo a privacidade e os direitos autorais de dados pessoais, produzindo resultados e não dando o crédito devido aos criadores originais do conteúdo.

2.1.3 Aprendizado de Máquina

Segundo Alpaydin (2020), Aprendizado de Máquina (AM) é um ramo da Inteligência Artificial (IA) que se concentra no uso de dados e algoritmos para imitar a maneira como os humanos aprendem, melhorando gradualmente a precisão ao longo do tempo. Foi definido pela primeira vez na década de 1950 como "o campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados" por Arthur Samuel, um cientista da computação e inovador em IA.

O Aprendizado de Máquina envolve alimentar algoritmos de computador com grandes quantidades de dados para que eles possam aprender a identificar padrões e relacionamentos dentro desse conjunto de dados. Os algoritmos começam então a fazer suas próprias previsões ou tomar decisões com base em suas análises. À medida que os algoritmos recebem novos dados, continuam a refinar suas escolhas e melhorar seu desempenho da mesma forma que uma pessoa melhora em uma atividade com a prática. Os quatro tipos de Aprendizado

de Máquina são: supervisionado, não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço.

Em modelos de aprendizado supervisionado, o algoritmo aprende a partir de conjuntos de dados de treinamento rotulados e melhora sua precisão ao longo do tempo. É projetado para construir um modelo que pode prever corretamente a variável de destino quando recebe novos dados que não viu antes.

Aprendizado de máquina não supervisionado é quando o algoritmo procura padrões em dados que não foram rotulados e não têm variáveis de destino. O objetivo é encontrar padrões e relacionamentos nos dados que os humanos possam ainda não ter identificado, como detectar anomalias em *logs*, *traces* e métricas para encontrar problemas no sistema e ameaças à segurança.

Aprendizado semi-supervisionado é um híbrido de supervisionado e não supervisionado. No aprendizado semi-supervisionado, o algoritmo treina em dados rotulados e não rotulados. Ele primeiro aprende com um pequeno conjunto de dados rotulados a fazer previsões ou tomar decisões com base nas informações disponíveis. Em seguida, usa o conjunto maior de dados não rotulados para refinar suas previsões ou decisões, encontrando padrões e relacionamentos nos dados.

Aprendizado por reforço é quando o algoritmo aprende por tentativa e erro, recebendo *feedback* na forma de recompensas ou penalidades por suas ações. O Aprendizado de Máquina é importante porque aprende a executar tarefas complexas usando exemplos, sem programar algoritmos especializados. Em comparação com as abordagens algorítmicas tradicionais, o Aprendizado de Máquina permite maiores automatizações, como prever tendências para melhorar as decisões de negócios, personalize recomendações que melhore a satisfação do cliente, e automatize o monitoramento de aplicações e sistemas complexo de infraestrutura de TI.

2.1.4 Modelos de Incorporação de Palavras

De acordo com Mikolov et al. (2013), um modelo de incorporação de palavras é uma técnica usada no processamento de linguagem natural que

representa palavras como números para que um computador possa trabalhar com elas. Cada palavra precisa receber um formato numérico para que o computador possa processar, sendo uma abordagem para representações numéricas aprendidas de texto, o que pode ser feito por meio de algumas abordagens diferentes. O modelo de incorporação de palavras cria um espaço de alta dimensão, no qual cada palavra é representada por um vetor denso de números. Assim, um computador pode usar esses vetores para compreender as relações entre as palavras e fazer previsões.

Segundo Pilehvar (2020), as incorporações de palavras são um tipo especial de representação de palavras distribuídas que são construídas aproveitando redes neurais, popularizadas após 2013, com a introdução do Word2vec, por Tomas Mikolov e sua equipe no Google, como uma resposta para tornar o treinamento de *embeddings* baseados em redes neurais mais eficiente. Os *embeddings* de palavras são geralmente classificados como modelos preditivos porque eles são utilizados com o objetivo de prever a próxima palavra ou uma palavra que esteja ausente em uma sentença.

Uma abordagem de *embedding* é o *one-hot encoding*, que dá a cada palavra em um corpo de texto um número exclusivo. Esse número é transformado em um vetor binário, que representa a palavra. Outra abordagem é a representação baseada em contagem, que conta o número de vezes que uma palavra aparece em um corpo de texto e atribui um vetor correspondente a ela. Há também a combinação SLIM, que utiliza os dois métodos citados para que um computador possa entender tanto o significado das palavras quanto a frequência com que elas aparecem no texto.

Ainda de acordo com Elastic (2024), a vetorização é o processo de transformar palavras em vetores numéricos. Um vetor denso é um vetor no qual a maioria das entradas não é zero. É o oposto de um vetor esparsos, como o *one-hot encoding*, que tem muitas entradas zero. Esse espaço de alta dimensão é chamado de espaço de *embedding*.

Palavras que têm significados semelhantes ou são usadas em contextos semelhantes recebem vetores semelhantes, o que significa que estão

localizadas próximas umas das outras no espaço de *embedding*. Por exemplo, “chá” e “café” são palavras semelhantes que estariam localizadas próximas uma da outra, enquanto “lar” e “mar” estariam mais distantes porque têm significados diferentes e não são usadas juntas com frequência, embora tenham ortografia semelhante.

Embora existam diferentes métodos de criação de incorporação de palavras, todos eles envolvem o treinamento baseado em uma grande quantidade de dados de texto chamada *corpus*. O *corpus* também pode ser uma camada de *embedding* customizada, projetada para o caso de uso quando outros *corpora* pré-treinados não conseguem fornecer dados suficientes. Durante o treinamento, o modelo aprende a associar cada palavra a um vetor exclusivo, com base nos padrões de uso de palavras no *corpus*. Esses modelos podem ser usados para transformar novas palavras em vetores.

Os modelos de incorporação de palavras podem ser feitos usando uma variedade de parâmetros. A escolha do modelo depende dos requisitos específicos da tarefa. Deve-se considerar o tamanho do conjunto de dados, o domínio dos dados e a complexidade da linguagem para selecionar o modelo mais adequado. A seguir, serão descritos alguns dos modelos de incorporação de palavras mais populares.

O *Word2vec* é um algoritmo baseado em rede neural de duas camadas que recebe um *corpus* de texto e gera um conjunto de vetores. Um exemplo de *Word2vec* comumente usado é “rei – homem + mulher = rainha”. Ao deduzir a relação entre “rei” e “homem” e entre “homem” e “mulher”, o algoritmo pode identificar “rainha” como a palavra apropriada correspondente a “rei”. O *Word2vec* é treinado usando os algoritmos *Skip-Gram* ou *Continuous Bag of Words* (CBOW). O *Skip-Gram* tenta prever palavras de contexto a partir de uma palavra alvo. O *Continuous Bag of Words* funciona de maneira oposta, prevendo a palavra alvo por meio do contexto das palavras ao seu redor.

De acordo com Aoun (2021), o GloVe baseia-se na ideia de que o significado de uma palavra pode ser inferido a partir de sua ocorrência com outras palavras em um *corpus* de texto. O algoritmo cria uma matriz de

ocorrência que captura a frequência com que as palavras aparecem juntas no *corpus*.

Segundo Ziger (2021), o FastText é uma extensão do modelo Word2Vec e é baseado na ideia de representar palavras como um pacote de n-gramas de caracteres, ou unidades de sub palavras, em vez de apenas como palavras individuais. Usando um modelo semelhante ao *Skip-Gram*, o FastText captura informações sobre a estrutura interna das palavras que o ajudam a processar vocabulário novo e desconhecido.

Segundo Elastic (2024), ELMo (*embeddings* de modelos de linguagem) difere dos modelos de incorporação de palavras mencionados anteriormente porque usa uma rede neural profunda que analisa todo o contexto em que a palavra aparece. Isso torna possível captar nuances no significado que outros modelos de incorporação de palavras podem não captar.

Segundo Manning (2008), o TF-IDF (frequência do termo - frequência inversa do documento) é um valor matemático determinado pela multiplicação da frequência do termo (TF) pela frequência inversa do documento (IDF). O TF refere-se à proporção dos termos alvo no documento em comparação com o total de termos no documento. O IDF é um logaritmo da proporção entre o total de documentos e o número de documentos que contêm o termo alvo.

Segundo Elastic (2024), os modelos de incorporação de palavras têm sido usados para melhorar a modelagem da linguagem, que é a tarefa de prever a próxima palavra em uma sequência de texto. Ao representar palavras como vetores, os modelos podem capturar o contexto no qual uma palavra aparece e fazer previsões mais precisas, com a ampla disponibilidade de modelos pré-treinados disponíveis para uso, tornando o processo mais fácil para pesquisadores incorporá-los em seus modelos sem precisar treiná-los do zero.

A construção de modelos de incorporação de palavras pode ser mais rápida do que as técnicas tradicionais de engenharia porque o processo de treinamento de uma rede neural em um grande *corpus* de dados de texto não é supervisionado. Depois de treinado, o modelo poderá ser usado como recurso de entrada para uma ampla gama de tarefas de processamento de linguagem

natural sem a necessidade de recursos computacionais adicionais. Como um modelo de incorporação de palavras é uma representação vetorial densa de palavras, ele representa as palavras com mais eficiência do que as técnicas de vetores esparsos. Isso também possibilita que ele capture melhor as relações semânticas entre as palavras.

Embora os modelos de incorporação de palavras tenham muitas vantagens, também existem desvantagens em seu uso. O treinamento de modelos de incorporação de palavras pode exigir recursos computacionais caros, principalmente ao usar grandes conjuntos de dados ou modelos complexos. Os modelos de incorporação de palavras pré-treinados também podem exigir um espaço de armazenamento significativo, o que pode ser um problema para aplicações com recursos limitados.

Os modelos de incorporação de palavras são treinados em um vocabulário finito, o que significa que podem não ser capazes de representar palavras que não estejam nesse vocabulário. Isso pode ser um problema para idiomas com um grande vocabulário ou para terminologia específica da aplicação.

Se as entradas de dados para um modelo de incorporação de palavras contiverem preconceitos, o modelo de palavras poderá refletir esses preconceitos. Os modelos de palavras são frequentemente considerados uma caixa preta porque seus modelos subjacentes, como as redes neurais do GloVe ou Word2Vec, são complexos e difíceis de interpretar.

A qualidade do modelo de incorporação de palavras depende da qualidade dos dados de treinamento. Embora os modelos captem a relação geral entre as palavras, eles podem perder certas nuances humanas, como o sarcasmo e ironia, que são mais difíceis de reconhecer.

Como os modelos de incorporação de palavras atribuem um vetor a cada palavra, eles podem ter problemas com homógrafos, que são palavras que têm a mesma grafia, mas significados diferentes. Ainda que tenham limitações, os modelos de incorporação de palavras fornecem um caminho eficaz para as

máquinas reconhecerem e capturarem as relações semânticas entre as palavras.

2.1.5 K-Nearest Neighbors (KNN)

De acordo com a IBM (2018), o K-Nearest Neighbours, também conhecido como KNN, é um algoritmo classificador de aprendizado supervisionado não paramétrico, que usa a proximidade para fazer classificações ou previsões sobre o agrupamento de um ponto de dados individual. É normalmente usado como um algoritmo de classificação, partindo do pressuposto de que pontos semelhantes podem ser encontrados próximos uns dos outros.

Segundo Mehta (2021), o KNN, sendo um algoritmo de Aprendizado de Máquina supervisionado, precisa de um conjunto de dados de referência para determinar a categoria dos pontos de entrada que serão classificados. Esse algoritmo classifica o conjunto de dados fornecido em diferentes grupos ou categorias, e quando um novo ponto de dados é inserido, o algoritmo identifica a qual categoria esse ponto pertence com base em várias medidas de similaridade.

O KNN usa a métrica da distância euclidiana para calcular os 'k' vizinhos mais próximos de um novo ponto de entrada, e baseado na quantidade de vizinhos pertencentes a uma categoria ou outra, a classificação é feita. Dependendo do valor de k os resultados são diferentes para cada situação, mas geralmente, quando o k é pequeno, a classificação fica mais sensível a regiões bem próximas, podendo ocorrer o problema de *overfitting*. Com k grande, a classificação fica menos sujeita a ruídos e pode ser considerada mais robusta. Porém, se k for grande demais, pode ser que haja o problema de *underfitting*.

Um cenário de *overfitting* ocorre quando, nos dados de treino, o modelo tem um desempenho excelente; porém, quando utilizamos os dados de teste o resultado é ruim. Neste caso, o modelo aprendeu tão bem as relações existentes no treino, que acabou apenas decorando o que deveria ser feito. Assim, ao receber as informações das variáveis preditoras nos dados de teste, o modelo tenta aplicar as mesmas regras decoradas. Com dados diferentes, estas regras

não têm validade e o desempenho é afetado. Portanto, *overfitting* significa que o modelo treinado não tem capacidade de generalização.

Por outro lado, no cenário de *underfitting* o desempenho do modelo é ruim no próprio treinamento. O modelo não consegue encontrar relações entre as variáveis e o teste nem precisa acontecer. Este modelo já pode ser descartado, pois não terá utilidade. O ideal é encontrar um valor de k que esteja entre alta variância e alto viés, e recomenda-se também escolher um número ímpar para evitar empates na análise de classificação.

Algumas vantagens da utilização do algoritmo *KNN* é que ele não requer tempo de treinamento porque armazena dados de treinamento e seu poder computacional é usado apenas ao fazer previsões. Ele é de fácil implementação e programação, mas devido a sua simplicidade também possui algumas limitações, como os dados de treinamento do modelo não conseguem acompanhar a dimensionalidade em evolução do hiperespaço: as previsões se tornam menos precisas porque a distância entre o ponto de consulta e pontos semelhantes aumenta em outras dimensões.

2.1.6 Matriz de Confusão

De acordo com Morettin (2021), a matriz de confusão é uma ferramenta utilizada para avaliar o desempenho de algoritmos de classificação e identificar possíveis erros de previsão. Ela tem o formato de uma tabela que permite visualizar o desempenho de um modelo de classificação em relação a um conjunto de dados de teste.

Ela mostra a quantidade de previsões corretas e incorretas feitas pelo modelo, divididas em quatro categorias: verdadeiro positivo, que são os casos em que o modelo previu corretamente a classe positiva; verdadeiro negativo, que são os casos em que o modelo previu corretamente a classe negativa; falso positivo, ou número de vezes que o modelo previu incorretamente a classe positiva; e falso negativo, que é o número de vezes que o modelo previu incorretamente a classe negativa.

Para construir uma matriz de confusão é necessário ter um conjunto de dados de teste com rótulos conhecidos. Cada instância do conjunto de teste é classificada pelo modelo e comparada com o rótulo real. Com base nessa comparação, os resultados são contabilizados na matriz. A diagonal principal da matriz representa as previsões corretas, enquanto as células que estão fora dessa diagonal representam as previsões incorretas.

A partir dos valores presentes na matriz, é possível calcular outras métricas de avaliação, como a acurácia, a precisão, o *recall*, a especificidade e a *F1-score*. A acurácia mede a proporção de previsões corretas em relação ao total de previsões. Ela é calculada somando os valores da diagonal principal e dividindo pelo total de instâncias.

A precisão é a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos positivos. Ela mede a capacidade do modelo de classificar corretamente as instâncias da classe positiva.

O *recall*, também conhecido como taxa de verdadeiros positivos ou sensibilidade, é a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos negativos. Ele mede a capacidade do modelo de identificar corretamente as instâncias da classe positiva.

A especificidade, também conhecida como taxa de falso positivo, é a proporção de verdadeiros negativos em relação à soma de verdadeiros negativos e falsos positivos. Ela mede a capacidade do modelo de classificar corretamente as instâncias da classe negativa.

Finalmente, a *F1-score* é uma métrica que combina a precisão e o *recall* em um único valor. Ela é calculada como a média harmônica entre essas duas métricas e é útil quando há um desequilíbrio entre as classes.

Ainda de acordo com o Morettin (2021), a matriz de confusão possui algumas limitações que devem ser consideradas, como se mostrar inadequada quando há um desequilíbrio entre as classes. Nesses casos, a precisão pode ser alta mesmo que o modelo esteja desempenhando mal na classe minoritária e, por isso, a importância de analisar todo o conjunto de métricas e não focar

somente na acurácia. Além da necessidade de equilíbrio, a distribuição dos erros e seus custos não são considerados na matriz, tratando todos os erros de classificação de forma igualitária e não levando em conta a distribuição dos erros de classificação.

Geralmente, é um bom sinal quando um modelo apresenta altas taxas nos parâmetros de acurácia e sensibilidade, o que indica um desempenho preciso nas previsões. Um modelo com um bom valor de acurácia (próximo de 1) e baixos valores de sensibilidade ou especificidade (distante de 1), não está realizando boas previsões, pois está errando bastante na taxa de verdadeiros positivos ou verdadeiros negativos.

2.2 Trabalhos Relacionados

Felizardo (2024a) realizou um estudo de prova de conceito em que avalia como o ChatGPT-4.0 pode apoiar atividades de mapeamento sistemático em Engenharia de Software, particularmente na extração de dados, utilizando o ChatGPT-4.0 para extração de dados de estudos de mapeamento sistemático em contraste com a extração manual. Tiveram como resultado que a precisão do ChatGPT-4.0 foi de 87,83% quando comparada com a extração manual, reconhecendo que a substituição total do processo de extração humana pelo ChatGPT-4.0 não é recomendada. No entanto, é promissor o emprego do ChatGPT para extração semiautomática de dados para síntese de evidências em mapeamento sistemáticos na Engenharia de Software.

Felizardo (2024b) realizou um estudo para avaliar a precisão do uso do ChatGPT-4.0 no processo de revisão sistemática da literatura em Engenharia de Software, principalmente para apoiar a primeira etapa, de seleção e inclusão de artigos selecionados, com base no título, resumo e palavras-chave. No estudo, avaliaram a acurácia da utilização do ChatGPT para seleção dos estudos a serem incluídos em dois estudos de revisão sistemática, em contraste com o método convencional de leitura de título e resumo.

A precisão do ChatGPT no apoio à atividade de seleção inicial foi de 75,3%, no primeiro estudo, com 101 seleções corretas (48 inclusões e 53 exclusões) e 33 seleções incorretas (17 inclusões e 16 exclusões), e de 86,1%

no segundo estudo, com 386 seleções corretas (113 inclusões e 273 exclusões) e 62 seleções incorretas (27 inclusões e 35 exclusões). Seus resultados confirmam que não é aconselhável terceirizar totalmente o processo de seleção para o ChatGPT. Porém, pode ser valioso como ferramenta de apoio, auxiliando pesquisadores iniciantes ou mesmo experientes.

Nashwan (2023) sugere fazer uso dos GML para simplificar o processo de revisão sistemática em duas etapas: a Avaliação de Qualidade (AQ) e o Risco de Viés (RDV). Embora já existam ferramentas para realizar essas avaliações, a utilização de *GMLs* pode contribuir para alcançar uma avaliação automatizada e menos subjetiva do que a realizada por avaliadores humanos. Para isso, o autor considera a utilização do modelo GPT-4 para essa tarefa, embora não apresente resultados nesta direção.

Robinson (2023) foca na utilização de GMLs para realizar um processo de triagem, de inclusão e exclusão de artigos, na revisão sistemática de publicações médicas. O autor propõe o uso de diversos modelos de GML. Ele conclui que, para a área médica, o modelo que obteve melhor resultado foi o modelo próprio de Bio-SIEVE (*Biomedical Systematic Include/Exclude reViewer with Explanations*). O experimento utilizou 7.330 revisões sistemáticas médicas disponíveis no Cochrane Reviews¹.

O treinamento e o desempenho dos *GMLs* foram comparados com a classificação dada por revisores médicos profissionais, utilizando uma amostra de 1711 artigos em três categorias: “Incluir”, “Excluir” e “Informações Insuficientes”. Cada modelo foi treinado por oito dias e o melhor modelo no conjunto de treinamento foi selecionado para comparação. O modelo do Bio-SIEVE obteve uma precisão de 82% no conjunto de testes, enquanto o ChatGPT alcançou uma precisão de 60%. Outro resultado apresentado é que os GML são capazes de excluir efetivamente resumos irrelevantes, fora do tópico, além de terem boa precisão para incluir e excluir.

Guo (2023) utilizou o modelo pré-treinado do *OpenAI GPT API* para realizar a triagem automática pelo título e resumo dos artigos clínicos e comparar

¹ www.cochranelibrary.com

seu desempenho com a rotulagem por dois revisores humanos independentes. Para isso, avaliou-se mais de 24.000 títulos e resumos. O modelo proposto alcançou precisão de 91% para exclusão e 76% para inclusão. Portanto, os resultados do experimento sugerem que o modelo tem potencial para agilizar o processo de revisão de artigos clínicos.

Liu (2023) avaliou como os *GML* podem ajudar na revisão de artigos ou propostas científicas. Para isso, realizou um estudo-piloto utilizando o GPT-4. Os resultados obtidos mostraram-se superiores a outros modelos pré-treinados. A metodologia aprofundou-se na investigação de três tarefas: a identificação de erros chaves, a verificação do cumprimento de exigências mínimas e a recomendação de melhor artigo.

Para investigar a identificação de erros, Liu (2023) desenvolveu treze artigos de Ciência da Computação com erros a fim de avaliar a capacidade do modelo de identificá-los. Os erros incluídos abrangeram desde erros matemáticos, por exemplo, implicação errada da convergência quadrática média, até falácias lógicas, como uma inferência falsa extraída de uma análise de dados. O autor obteve uma taxa de 55% na identificação dos erros, que é comparável à de revisores humanos.

Muitas conferências e periódicos fornecem aos autores uma lista de verificação de itens para auxiliar no *checklist* de cumprimento de exigências mínimas, oferecendo uma orientação aos autores para atender a essas exigências. Na tarefa de verificação de listas, o modelo obteve uma precisão de 86,6% em comparação com a resposta verdadeira. Para recomendação de melhor artigo, o modelo apresentou dificuldades. O autor conclui que os *GML* podem apoiar tarefas específicas de revisão para otimização do tempo dos pesquisadores, mas não para uma avaliação completa.

Em (Marshall e Brereton, 2013), foi realizado um mapeamento sobre ferramentas que apoiam a realização de revisões sistemáticas na área de Engenharia de Software. Foram utilizadas as bases de dados Google Scholar, ACM DL e IEEE Xplore, além da técnica *snowballing* (nos artigos incluídos). A pesquisa incluiu trabalhos entre os anos de 2004 a 2012 e foram identificados

um total de 21 artigos. Nesse estudo não foi reportado de que forma as ferramentas identificadas automatizam as etapas de uma RSL, apenas foram apresentadas as principais abordagens subjacentes utilizadas nos estudos.

Uma revisão sistemática da literatura foi realizada por O'Mara-Eves et al. (2015) com o objetivo de informar aos revisores o estado atual dos métodos de mineração de texto, como o seu uso reduz a carga de trabalho na fase de triagem (seleção), quais os seus potenciais benefícios e que desafios são enfrentados para a implementação de tais tecnologias. Os estudos foram realizados em diversos mecanismos de pesquisa. Foram incluídos estudos de 2005 a dezembro de 2013. Foram identificados alguns artigos que utilizam algoritmos de classificação com uma abordagem de aprendizado ativo. Esse processo é interativo e a precisão das predições feitas pelos algoritmos melhora de acordo com as interações feitas com os revisores.

Em todos os casos, os autores relataram que os sistemas testados levaram a uma redução na carga de trabalho. No entanto, dada a diversidade de abordagens e a falta de sobreposição (replicação) entre as avaliações, é impossível concluir se uma abordagem é melhor que a outra em termos de desempenho. Segundo os autores, apenas seis sistemas diferentes (relatados em 12 artigos) estão disponíveis em alguma ferramenta. Alguns são sistemas de revisão sistemática sob medida, enquanto outros são softwares mais genéricos para a análise preditiva e podem ser usados em RSL. Outro ponto que apontado pelos autores foi que os revisores necessitam realizar algum treinamento para utilizar as ferramentas citadas.

Em (Feng, Chiam e Lo, 2017), foi realizada uma revisão sistemática da literatura com o objetivo de identificar/classificar técnicas e ferramentas de mineração de texto que auxiliam as atividades de revisões sistemáticas. Essa revisão difere da anterior por investigar a adoção de técnicas de mineração de texto no domínio da Engenharia de Software. Os estudos buscados estão compreendidos entre janeiro de 2004 a dezembro de 2016. Foram buscados artigos em 6 bases de dados, além da execução de uma busca manual, foram obtidos um total de 4056 artigos, ao final da seleção restaram apenas 32 estudos relevantes para a pesquisa.

2.3 Considerações Finais

Este capítulo apresentou os fundamentos necessários para a compreensão da pesquisa que será descrita no próximo capítulo, assim como os trabalhos relacionados que a precederam. O próximo capítulo apresentará a metodologia de pesquisa e os detalhes sobre o plano de experimentos.

3 Metodologia de Pesquisa

Neste capítulo, serão apresentados a proposta de pesquisa e o plano de experimentos.

3.1 Proposta de Pesquisa

A proposta de pesquisa consiste na avaliação do desempenho do GML na tarefa de seleção de artigos em um mapeamento sistemático. Para isso, comparamos um método baseado em modelos de incorporação de palavras para o resumo do artigo e o gabarito estabelecido manualmente pelos autores nos mapeamentos sistemáticos originais.

Ao todo, seis mapeamentos sistemáticos publicados na área de Engenharia de Software foram selecionados e reproduzidos. Cada mapeamento sistemático gerou um conjunto de dados, que foram utilizados no experimento de avaliação dos GML e do modelo de incorporação de palavras.

O objetivo desta pesquisa é analisar o desempenho dos modelos em relação à precisão na identificação dos artigos aceitos. Desta forma, podemos testar o desempenho dos GML na tarefa de seleção de artigos. Nas próximas seções, apresentamos a proposta de avaliação experimental e a construção do seu racional.

3.2 Plano de Experimentos

Esta pesquisa consiste em investigar a aplicação de GML e modelos de incorporação de palavras na automatização do processo de seleção de artigos para mapeamentos sistemáticos na área de Engenharia de Software. São comparadas as classificações obtidas por diferentes automatizações da seleção de artigos em uma base controlada de seis artigos de mapeamento sistemático na área de Engenharia de Software, usada como gabarito.

Nesta seção, é apresentado o plano de experimentos realizados para investigar a automatização do processo de seleção de artigos na etapa de

elaboração de mapeamento sistemático, conforme Kitchenham (2009). Os experimentos explorados incluem as seguintes etapas:

1. Seleção e replicação de mapeamentos sistemáticos na área de Engenharia de Software para criação de um gabarito;
2. Comparação do resumo com a expressão de busca utilizando a distância euclidiana entre os vetores de modelos de incorporação de palavras dos resumos e os vetores da expressão de busca utilizada para recuperar os artigos nas bases;
3. Análise de cobertura de artigos com número fixo de 200 selecionados;
4. Avaliação da relevância dos artigos usando *Grandes Modelos de Linguagem* (GML) e um *prompt* desenvolvido para analisar a relação entre os termos do resumo e a expressão de busca.

3.2.1 Seleção e Replicação de Mapeamentos Sistemáticos

O objetivo da etapa de seleção e replicação de mapeamentos sistemáticos é criar um conjunto de referência que sirva de base comparativa para os resultados produzidos pelos GMLs. Isto foi realizado a partir dos mapeamentos sistemáticos previamente publicados na área de Engenharia de Software, permitindo comparar os resultados dos mecanismos automatizados de seleção de artigos aqui propostos com os artigos selecionados manualmente pelos autores das pesquisas publicadas.

Para isso, foram selecionados artigos de mapeamento sistemático na área de Engenharia de Software que contêm informações como a expressão de busca, as bases de dados onde os artigos foram pesquisados, o número de artigos retornados em cada base, critérios de inclusão e exclusão dos artigos e relação dos artigos selecionados. Estes mapeamentos sistemáticos foram selecionados com o intuito de replicar, de forma automática, o processo de seleção de artigos realizado por seus autores e comparar os resultados da seleção automática com os resultados da seleção realizada manualmente.

A Tabela 2 apresenta as informações resumidas dos seis artigos que descrevem os mapeamentos sistemáticos selecionados, com as seguintes informações: índice, que indica como os mapeamentos serão referenciados doravante neste trabalho; título do artigo; ano de publicação do artigo; lista de autores do artigo; bases de dados consultadas; número de trabalhos recuperados da consulta e número de trabalhos aceitos.

Tabela 2 – Informações sobre os mapeamentos sistemáticos analisados para compor a avaliação experimental

Índice	Título	Ano	Autores	Bases Consultadas	Trabalhos Analisados sem repetição	Trabalhos Aceitos
MS1	A systematic mapping on quantum software development in the context of software engineering	2021	Paulo Eduardo Zanni Junior, Valter Vieira de Camargo	ACM, IEEE, Scopus, Science Direct, Wiley	698	24
MS2	Sustainability in Software Architecture: A Systematic Mapping Study	2022	Vasilios Andrikopoulos, Rares-Dorian Boza, Carlos Perales, Patricia Lago	ACM, IEEE, Scopus, Science Direct, Wiley	354	32
MS3	Software Configuration Management in Software Product Lines: Results of a Systematic Mapping Study	2022	G. P. Espinel-Mena, J. L. Carrillo-Medina, M. Flores-Calero and M. Urbieto	IEEE, ACM, Scopus	392	23
MS4	Systematic Mapping Study on Security Approaches in Secure Software Engineering	2021	Rafiq Ahmad Khan, Siffat Ullah Khan, Habib Ullah Khan, and Muhammad Ilyas	IEEE, ACM, Springer Link, Science Direct, Wiley, Google Scholar	535	116
MS5	Value-based Software Engineering: A Systematic Mapping Study	2023	Norsaremah Salleh, Emilia Mendes, Fabiana Mendes, Charitha Dissanayake Lekamlage, Kai Petersen	IEEE, ACM, Scopus, Science Direct, Web of Science, SpringerLink	3.170	143
MS6	A Systematic Mapping of the Proposition of Benchmarks in the Software Testing and Debugging Domain	2023	Deuslirio da Silva-Junior, Valdemar V. Graciano-Neto, Diogo M. de-Freitas, Plinio de Sá Leitão-Junior and Mohamad Kassab	IEEE, Scopus, ACM, Engineering Village	732	25

Foram selecionados seis artigos de mapeamento sistemático na área de Engenharia de Software publicados no período entre 2021 e 2023. Esses artigos foram escolhidos por apresentarem a relação dos artigos aceitos ao final da seleção, já que não é comum apresentar a relação nominal dos artigos excluídos

em cada passo do protocolo, mas somente a quantidade. Os trabalhos selecionados apresentaram a lista dos trabalhos selecionados para o mapeamento e o número de trabalhos encontrados pela expressão de busca em cada base de dados.

Em seguida, foi replicado o processo de seleção dos artigos para os seis mapeamentos sistemáticos, aplicando a expressão de busca nas bases consultadas por cada mapeamento sistemático e criando um banco de dados com os títulos e resumos dos artigos selecionados para os mapeamentos. A Tabela 3 apresenta a expressão de busca usada em cada mapeamento sistemático selecionado.

Tabela 3 - Expressão de busca dos mapeamentos sistemáticos avaliados

Índice	Expressão de Busca
MS1	("quantum software" OR "quantum software engineering" OR "quantum programming" OR "quantum programming language" OR "quantum programming languages" OR "quantum computer software" OR "quantum application" OR "quantum applications")
MS2	"software architect*" AND sustainable* AND (requirements OR design OR implementation OR evaluation OR decision)
MS3	"software" AND ("product-line" OR "product line" OR "product family" OR "product-family" OR "product lines" OR "product families" OR "system family" OR "system families" OR "SPL") AND ("configuration management" OR "variability management" OR "variation management")
MS4	((("software security"OR"software privacy"OR"secure software" OR "software protection" OR "software safety") AND("SoftwareEngineering"OR"SoftwareDevelopment lifecycle" OR "SDLC" OR "Software security Model")))
MS5	((`value-based" AND ``software engineering") OR (`value-based software engineering") OR (`value based" AND ``software engineering") OR (`Value based software engineering") OR VBSE) OR ((value OR ``value based" OR ``valuation" OR ``value creation") AND (`economics based" OR ``decision making" OR economics OR ``software project") AND (`software engineering" OR software OR ``software development"))
MS6	("benchmark" OR "benchmarking" OR "dataset" OR "dataset" OR "database" OR "datasets"OR "datasets"OR "benchmarks")AND ("software"OR "program") AND (("fault localization" OR "error localization" OR "defect localization" OR "bug localization" OR "error localisation" OR "defect localisation" OR "bug localisation" OR "fault localisation")OR("software repair"OR"softwarefixing"OR"program repair" OR"programfixing"OR"bugfixing"OR"bug-fixing"OR"automatic repair")OR ("software testing" OR "software test") OR ("software debugging")) AND ("bug" OR "defect" OR "buggy" OR "faulty" OR "failing" OR "failed" OR "bugs" OR "defects")

Com a expressão de busca e o número de artigos recuperados originalmente de cada base, realiza-se a replicação das consultas nas bases para analisar se a avaliação experimental teria cobertura suficiente para encontrar um número de artigos próximo do encontrado pelos pesquisadores na época de cada estudo, assim como encontrar a relação dos artigos selecionados.

A Tabela 4 apresenta a replicação inicial dos resultados da expressão de busca para cada um dos seis artigos em cada base e a variação percentual do total de trabalhos recuperados. Em cada célula da tabela, o número na esquerda é o número original de artigos recuperados pela consulta e o número à direita é o quantitativo da replicação da expressão de busca nas respectivas bases. A Tabela 5 mostra o resultado consolidado dos trabalhos recuperados para cada mapeamento.

Tabela 4 - Resultado da replicação da expressão de busca dos artigos

Índice	ACM	IEEE	Science Direct	Scopus	Wiley / WoS	Spr.Link / Eng. Village	Google Scholar
MS1	70 / 74	115 / 119	80 / 83	416 / 418	17 / 27	-	-
MS2	20 / 22	130 / 138	18 / 20	180 / 187	6 / 6	-	-
MS3	60 / 68	170 / 173	-	162 / 168	-	-	-
MS4	37 / 38	170 / 180	59 / 61	-	36 / 38	16 / 18	217 / 221
MS5	231 / 234	125 / 141	103 / 108	165 / 181	596 / 637	1.950 / 1.972	-
MS6	259 / 260	270 / 302	-	105 / 109	-	98 / 102	-

Tabela 5 - Resultado consolidado da replicação da expressão de busca

Índice	Trabalhos Recuperados (original)	Trabalhos Recuperados (replicação)	Variação Percentual (%)
MS1	698	721	+3,30%
MS2	354	373	+5,37%
MS3	392	409	+4,34%
MS4	535	556	+3,93%
MS5	3.170	3.273	+3,25%
MS6	732	773	+5,60%

Todos os mapeamentos sistemáticos tiveram uma variação percentual abaixo de 6% e cobertura de 100% nos trabalhos selecionados. Levando em consideração que a expressão de busca foi aplicada em um momento posterior ao realizado pelo pesquisador de cada mapeamento, a variação percentual encontrada para trabalho foi considerada aceitável. Assim, trataremos o conjunto de artigos recuperados das bases com a aplicação da expressão de busca como uma proxy do conjunto de artigos analisados manualmente pelos pesquisadores

de cada mapeamento sistemático para chegar à lista de artigos que foram selecionados para compor estes mapeamentos. A base final com o título e resumo dos artigos recuperados estão disponibilizados no arquivo do Github: <https://github.com/filiperezendesilva/Pesquisa-Mestrado/tree/main>.

Para cada mapeamento sistemático foi necessário aplicar refinamentos nos resultados da expressão de busca exibida na Tabela 3, seguindo os critérios de inclusão e exclusão definidos em cada mapeamento. Todos os refinamentos aplicados são critérios objetivos e automatizáveis, ou seja, não foram aplicados critérios de inclusão e exclusão dos autores originais que envolvem alguma subjetividade. A seguir, será detalhado o refinamento de busca em cada mapeamento na Tabela 6.

Tabela 6 – Refinamento de busca em cada mapeamento

Índice	Termos-chaves (metadados textuais)	Período de publicação	Idioma	Crítérios de exclusão
MS1	“software quântico”	2010 a 2021	Inglês	Estudos duplicados e inacessíveis
MS2	arquitetura de software” ou “sustentabilidade	Até 2022	Inglês	Estudos duplicados e inacessíveis
MS3	“gerenciamento de configuração de software” ou “linhas de produtos de software”	Até 2022	Inglês	Estudos duplicados e inacessíveis
MS4	“segurança de software”	Até 2021	Inglês	Estudos duplicados e inacessíveis
MS5	“Engenharia de Software baseada em valor”	Até 2023	Inglês	Estudos duplicados e inacessíveis
MS6	“depuração de software” ou “teste de software”	Até 2023	Inglês	Estudos duplicados e inacessíveis

Ao aplicar os procedimentos descritos em cada busca, chegou-se ao quantitativo de artigos recuperados, conforme demonstrado na Tabela 4

3.2.2 Comparação do Resumo com a Expressão de Busca

Este experimento consistiu em verificar se a similaridade vetorial entre o resumo de um artigo e a expressão de busca usada nas bases de dados pode ser um critério confiável para seleção automática de artigos para um mapeamento sistemático. Foram utilizados modelos de incorporação de palavras para transformar os termos linguísticos utilizados nos resumos e nas expressões

de busca em vetores. Em seguida, os vetores das palavras utilizadas no resumo e na expressão de busca foram consolidados pelos modelos de incorporação de palavras para criar vetores para representar o resumo de cada artigo e a expressão de busca. Para cada vetor de resumo comparou-se com o vetor da expressão de busca e foi feita a distância euclidiana entre os dois vetores, os vetores com a menor distância são os resumos mais relevantes para o modelo e resultados apresentados nesta seção.

Os modelos de incorporação de palavras utilizados foram: *glove-wiki-gigaword-100*, *word2vec-google-news-300*, *fasttext-wiki-news-subwords-300* e o modelo de incorporação de palavras do *all-MiniLM-L6-v2*. A Tabela 7 apresenta os índices que indicam como os modelos de incorporação de palavras serão referenciados doravante neste trabalho.

Tabela 7 – Índices dos modelos de incorporação de palavras

Índice	Modelo de incorporação de palavras
Glove	glove-wiki-gigaword-100
Word2vec	word2vec-google-news-300
FastText	fasttext-wiki-news-subwords-300
all-Mini	all-MiniLM-L6-v2

O all-Mini é uma versão compacta e eficiente de um modelo *transformer* treinado para gerar representações vetoriais de sentenças de maneira rápida e com boa qualidade. Segundo Yin e Zhang (2024), o modelo de incorporação de palavras do *all-MiniLM-L6-v2* é uma variação otimizada do MiniLM, que é uma versão reduzida do modelo BERT ajustado para gerar incorporação eficientes, mantendo um bom desempenho em tarefas de similaridade semântica, como comparação de sentenças, análise de sentimentos e busca semântica. Os modelos tradicionais como Glove, Word2vec e FastText transformam cada palavra em um vetor de n dimensões, no caso, 100 ou 300, e esses vetores são somados e o resultado da frase é um vetor médio que representa toda a frase, nessa média se perde a ordem e estrutura gramatical da frase, o que é uma limitação dos modelos tradicionais.

Os modelos baseados em *Transformers*, como o all-Mini, recebem como entrada a frase inteira, que processa a estrutura da frase, com atenção contextual entre as palavras, criando incorporações contextualizadas para cada *token*, palavra ou subpalavra da frase. Utilizam a média das incorporações dos *tokens* para chegar no vetor único de saída que representará a frase inteira, no entanto cada *token* já está contextualizado ao contexto da frase em apareceu.

Foram selecionados os resumos mais próximos à expressão de busca de cada mapeamento sistemático. Em cada mapeamento, foram selecionados o mesmo número de artigos do mapeamento original. Por exemplo, no mapeamento MS1 foram selecionados os 24 artigos mais próximos da expressão de busca, no mapeamento MS2 foram selecionados os 32 artigos mais próximos da expressão, e assim por diante.

Para calcular a distância entre dois vetores representando um resumo e uma expressão de busca foi aplicada a distância euclidiana normalizada em uma escala de 0 a 1. Para a normalização dos dados foi utilizada a abordagem *min-max*, na qual o valor mínimo é transformado em zero, o valor máximo em 1 e todos os outros valores ficam em um decimal entre 0 e 1, sendo transformados pela equação (1).

$$X' = \frac{X - \max(X)}{(X) - \min(X)} \quad (1)$$

A normalização resulta em uma variável chamada “probabilidade de aceitação”. Esta probabilidade indica as chances de um artigo ser selecionado pelo modelo e cada artigo recebeu um valor de probabilidade. Ela foi definida como 1 menos a distância normalizada, de modo que quanto menor a distância entre o resumo e a expressão de busca, maior a probabilidade de o artigo ser aceito.

Em cada mapeamento sistemático, contamos um verdadeiro positivo (VP) para cada artigo dentre os mais próximos da expressão de busca que foi selecionado pelos autores do mapeamento sistemático. Um falso positivo (FP) é contado quando um dos artigos mais próximos não foi selecionado pelos autores. Os artigos que não figuram entre os mais próximos são contados como

verdadeiro negativo (VN) quando não foram selecionados pelos autores ou falso negativo (FN) quando foram selecionados.

As métricas apresentadas nas colunas da Tabela 8 da Seção 4.1 são calculadas pelas fórmulas abaixo, onde N é o número total de artigos em cada mapeamento. É importante destacar que os dados apresentam um grande desbalanceamento de classes. Por exemplo, no mapeamento MS1 temos 24 artigos selecionados (positivos) e 674 artigos não selecionados (negativos).

$$\text{Acurácia} = \frac{VP+VN}{N} \qquad \text{Precisão} = \frac{VP}{VP+FP} \qquad \text{Especificidade} = \frac{VN}{VP+FP}$$

A acurácia alta, assim como a alta especificidade, são esperadas, dado o número de verdadeiros negativos. A baixa precisão, por outro lado, acusa as limitações do modelo, indicando que o número de artigos selecionados que estão próximos da expressão de busca (VP) é proporcionalmente baixo em relação ao número de artigos próximos da expressão (VP + FP). A falta de proporção entre as classes (um número muito maior de artigos é descartado se comparado ao número de artigos selecionados) faz com que o modelo seja um bom preditor da classe negativa, mas não da classe positiva.

3.2.3 Análise de Cobertura com Número Fixo de 200 Selecionados

Com todas as distâncias calculadas e normalizadas no intervalo de 0 a 1, os valores foram ordenados de forma decrescente para os modelos de incorporação de palavras explorados, de forma a verificar se os maiores valores seriam os artigos que foram aceitos. A título de comparação foram analisados artigos com os 200 maiores valores de probabilidades de aceitação em cada mapeamento sistemático, calculando-se a **cobertura dos selecionados**, que é a proporção de artigos selecionados que aparecem entre os N artigos com as maiores probabilidades de aceitação, quando os artigos são ordenados de forma decrescente de acordo com essas probabilidades.

Estudos empíricos em mapeamento sistemático assistido por aprendizado ativo têm mostrado que a maioria dos artigos relevantes pode ser localizada nos primeiros 150 a 300 artigos priorizados, especialmente quando modelos de

priorização iterativa são aplicados corretamente (Cormack & Grossman, 2014; Wallace et al., 2013). Assim, a seleção de 200 artigos alinha-se com práticas consolidadas e eficazes observadas na literatura, equilibrando o custo de triagem e o retorno informacional.

Selecionar menos do que 200 artigos comprometeria a cobertura e aumentaria o risco de descarte de estudos relevantes, principalmente em áreas interdisciplinares ou de vocabulário variado. Por outro lado, aumentar esse número para 300 ou mais implicaria em um custo adicional elevado na triagem manual, sem ganhos proporcionais em sensibilidade, que é a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos negativos, nas etapas iniciais, dado que a curva de recuperação de relevantes costuma apresentar retorno decrescente após os primeiros documentos priorizados (van de Schoot et al., 2021).

3.2.4 Avaliação de Relevância Usando GML

Este experimento consistiu em utilizar diferentes GML para avaliar a relevância dos artigos para cada mapeamento sistemático. Nesse processo, o resumo de cada artigo e a expressão de busca foram submetidos ao modelo, juntamente com um *prompt* desenvolvido para analisar a relação entre os termos. O *prompt* escrito para estes GML, foi simples e sem refinamentos disponibilizado abaixo.

```
prompt = (  
    f"Avalie a relevância do seguinte resumo em relação à string de busca:\n"  
    f"String de busca: {search_string}\n"  
    f"Resumo: {truncated_summary}\n\n"  
    f"Retorne uma pontuação de relevância de 0 (totalmente irrelevante) a 1 (altamente  
relevante)."  
  
    )
```

Cada GML gerou uma pontuação de relevância variando entre zero (totalmente irrelevante) a um (altamente relevante) para cada artigo em cada mapeamento sistemático. Os resultados obtidos pelos GML foram comparados com aqueles gerados pelos modelos de incorporação de palavras, tendo como referência o conjunto de artigos aceitos nos seis mapeamentos sistemáticos.

O primeiro GML utilizado para comparar com a base controlada foi o DistilBERT que, segundo Sanh et al. (2019), é uma versão reduzida e mais eficiente do BERT, desenvolvida para ser mais rápida e mais leve, mantendo boa parte do desempenho do BERT. Ele é treinado para ser um modelo "destilado", ou seja, reduzido em termos de tamanho e número de parâmetros, mas mantendo a maior parte das capacidades.

De acordo com Devlin et al. (2019), o modelo BERT original, lançado pelo Google em 2018, é um modelo pré-treinado em grandes volumes de texto e pode ser ajustado para diversas tarefas de processamento de linguagem natural. Ele gera representações de palavras contextualizadas, ou seja, as representações de uma palavra podem variar dependendo do contexto em que ela aparece, utilizando a arquitetura de *transformer* com camadas bidirecionais para modelar o contexto das palavras. Segundo Sanh et al. (2019), em comparação com o BERT, o DistilBERT tem 60% do tamanho e é 60% mais rápido para inferência, mantendo cerca de 97% do desempenho do BERT em diversas tarefas de processamento de linguagem natural.

Para usar o DistilBERT, foi necessário truncar o resumo de cada artigo nas 200 primeiras palavras, pois o código de acesso ao GML dava erro ao ser executado com resumos contendo mais palavras. Considerando que os resumos apresentaram em média 256 palavras, as primeiras 200 palavras correspondem a 78% da informação disponível. O erro de acesso ao GML está relacionado ao número máximo de *tokens* que o modelo do DistilBERT suporta, que é de 512 *tokens* (ou subpalavras).

Além do DistilBERT, foram utilizados os GML ChatGPT 4.0, DeepSeek e Gemini. Cada GML utilizou uma estratégia semelhante para o cálculo da relevância, gerando código em Python, transformando os resumos e a expressão de busca em vetores e calculando a similaridade de cosseno, que é uma métrica que mede o cosseno do ângulo entre dois vetores e indica se os vetores apontam mais ou menos na mesma direção. Porém, mesmo como processos semelhantes, cada GML utilizou critérios distintos detalhados a seguir.

O ChatGPT 4.0 realizou o mapeamento dos textos em vetores utilizando o modelo TF-IDF tradicional através de uma função específica² da biblioteca *scikit-learn* e utilizou a similaridade de cosseno entre os vetores do resumo e expressão de busca.

O DeepSeek separou a expressão de busca em termos individuais usando “OR” e “AND” como separadores e fez a contagem de cada um desses termos encontrados nos resumos, somando todas as ocorrências para obter a relevância total.

O Gemini adotou um procedimento similar ao ChatGPT, utilizando o modelo TF-IDF tradicional e a similaridade de cosseno entre os vetores do resumo e da expressão de busca.

A partir da execução dos experimentos propostos, espera-se identificar a eficácia de diferentes GML para a automação da seleção de artigos em mapeamentos sistemáticos. Os resultados permitirão avaliar quais GML apresentam melhor desempenho no processo de seleção automatizada para o critério já utilizado de selecionar os 200 maiores valores de probabilidades de aceitação em cada mapeamento sistemático.

3.3 Considerações Finais

Este capítulo apresentou a metodologia e o plano de experimentos para a comparação de diferentes GML e modelos de incorporação de palavras com a seleção manual de artigos pelos pesquisadores no contexto de um mapeamento sistemático da literatura. O próximo capítulo vai apresentar os resultados dos experimentos e as conclusões com relação a eles.

² *TfidfVectorizer(stop_words='english')*

4 Resultados e Discussão

Nesta seção serão apresentados os principais resultados dos modelos de incorporação de palavras e dos GML, além de uma discussão sobre a comparação do melhor modelo de incorporação de palavras, *all-MiniLM-L6-v2*, com o modelo do ChatGPT, que apesar de ter sido ligeiramente menor que o DeepSeek, é o mais consistente e chega a superá-lo se desconsiderarmos o mapeamento sistemático MS6, no qual obteve a menor cobertura.

4.1 Comparação do Resumo com a Expressão de Busca

A Tabela 8 apresenta um modelo de matriz de confusão para os mapeamentos sistemáticos com base no modelo de incorporação de palavras *all-MiniLM-L6-v2*. Este modelo apresentou os melhores resultados quando comparado aos demais modelos de incorporação de palavras analisados nesta pesquisa. Nesta tabela, para cada mapeamento é apresentado o número de artigos selecionados, a acurácia, a precisão e a especificidade.

Tabela 8 – Valores de acurácia, precisão e especificidade pelo modelo de incorporação de palavras *all-MiniLM-L6-v2*.

Índice	Selecionados	Acurácia (%)	Precisão (%)	Especificidade
MS1	24	96%	46%	98%
MS2	32	87%	25%	93%
MS3	23	91%	17%	95%
MS4	116	81%	54%	88%
MS5	143	97%	69%	99%
MS6	25	94%	8%	97%

Para o MS1, o modelo de incorporação de palavras com melhor desempenho classificou 11 artigos selecionados como selecionados (verdadeiro positivo) e 13 artigos no falso positivo, além de apresentar um bom desempenho na classificação dos artigos não selecionados, com 684 verdadeiro negativo e somente 13 no falso negativo. Assim, o modelo obteve acurácia de 96%, precisão de 46% e especificidade de 98%.

Para o MS2, o modelo de incorporação de palavras com melhor desempenho classificou 8 artigos selecionados como selecionados (verdadeiro positivo) e 24 artigos no falso positivo, 317 como verdadeiro negativo e 24 no falso negativo. Assim, o modelo obteve acurácia de 87%, precisão de 25% e especificidade de 93%.

Para o MS3, o modelo com melhor desempenho classificou 4 artigos selecionados como selecionados (verdadeiro positivo) e 19 artigos no falso positivo, 367 como verdadeiro negativo e 19 no falso negativo. Assim, o modelo obteve acurácia de 91%, precisão de 17% e especificidade de 95%.

Para o MS4, o modelo com melhor desempenho classificou 63 artigos selecionados como selecionados e 53 artigos no falso positivo, 387 como verdadeiro negativo e 53 no falso negativo. Assim, o modelo obteve acurácia de 81%, precisão de 54% e especificidade de 88%. Para o MS5, o modelo com melhor desempenho classificou 99 artigos selecionados como selecionados (verdadeiro positivo) e 44 artigos no falso positivo, 3.086 como verdadeiro negativo e 44 no falso negativo. Assim, o modelo obteve acurácia de 97%, precisão de 69% e especificidade de 99%. Para o MS6, o modelo com melhor desempenho classificou 2 artigos selecionados como selecionados e 23 artigos no falso positivo, 725 como verdadeiro negativo e 23 no falso negativo. Assim, o modelo obteve acurácia de 94%, precisão de 8% e especificidade de 97%.

4.2 Cobertura com Número Fixo de 200 Selecionados

A Tabela 9 apresenta os resultados de cobertura dos selecionados considerando os primeiros 200 artigos para os seis mapeamentos sistemáticos e os modelos de incorporação de palavras Glove, Word2vec, FastText e all-Mini. Pelo grau de cobertura, é demonstrado que o modelo *all-Mini* apresenta resultados superiores aos outros modelos de incorporação de palavras, obtendo uma média de cobertura dos selecionados de 79% nos seis mapeamentos, enquanto os demais modelos ficaram na faixa de 43% a 51%.

Um destaque é o modelo de incorporação de palavras *Word2Vec*, que teve um bom resultado no MS3, com 87% de cobertura, demonstrando que pode ser eficaz em casos específicos. Já o *Glove* e o *FastText* ficaram com coberturas

mais baixas, principalmente no MS5. Os modelos clássicos de incorporação de palavras (*Word2Vec*, *GloVe* e *FastText*) tiveram utilidade limitada, enquanto o *all-MiniLM-L6-v2* lidera o *ranking* geral com dois resultados de altíssima cobertura, 100% no MS1 e 97% para o MS2, e consistência de cobertura para os seis mapeamentos.

Tabela 9 – Cobertura dos selecionados nos primeiros 200 artigos para os diferentes modelos de incorporação de palavras.

Índice	all-Mini	Glove	Word2vec	FastText
MS1	100%	54%	50%	46%
MS2	97%	62%	56%	62%
MS3	70%	74%	87%	65%
MS4	83%	41%	52%	41%
MS5	75%	10%	23%	14%
MS6	48%	24%	36%	32%
Média de cobertura	79%	44%	51%	43%

4.3 Avaliação de Relevância Usando GML

A Tabela 10 apresenta a cobertura dos artigos selecionados no TOP 200 para os seis mapeamentos sistemáticos e os GML ChatGPT, DeepSeek, Gemini e DistilBERT. O destaque é o modelo do DeepSeek, que obteve a maior média de cobertura para os seis artigos (67%), utilizando um critério simples que foi a contagem dos termos da expressão de busca presentes nos resumos. Seu menor grau de cobertura foi de 47% e seu maior foi de 91%.

Tabela 10 – Cobertura dos selecionados no TOP 200 para os diferentes GML.

Índice	ChatGPT	DeepSeek	Gemini	DistilBERT
MS1	79%	88%	71%	38%
MS2	66%	47%	47%	69%
MS3	78%	91%	65%	35%
MS4	69%	74%	60%	32%
MS5	67%	48%	35%	5%
MS6	24%	52%	40%	36%
Média de cobertura	64%	67%	53%	36%

O modelo ChatGPT apresentou bom desempenho geral, com 64% de cobertura e maior consistência, tendo menos variação de cobertura entre os artigos e, ao desconsiderar da média o MS6, chega a obter uma cobertura média de 72%, superando o DeepSeek, que alcança 70% de cobertura ao descartar o mesmo artigo.

O modelo Gemini apresentou um desempenho geral regular, com 53% de cobertura média e maior cobertura de 71% no MS1, ficando abaixo do DeepSeek em todos os artigos e conseguindo um empate para o MS2 de 47% de cobertura.

O modelo DistilBERT apresentou desempenho inferior até mesmo aos modelos de incorporação de palavras clássicos *Glove*, *Word2Vec* e *FastText*, mostrando que modelos menores podem não captar nuances para análise de relevância dos termos, além de ter sido prejudicado com a limitação de informação do resumo em 200 palavras.

Os principais resultados indicaram uma similaridade entre os resumos dos artigos aceitos. No entanto, devido à quantidade significativamente maior de artigos não aceitos, a matriz de confusão mostrou que o método é mais eficiente em rejeitar artigos não selecionados do que aceitar os selecionados, o que é observado pelos altos valores de especificidade, que mede a capacidade do modelo de classificar corretamente as instâncias da classe negativa.

Por exemplo, para o MS6, o fato de a acurácia ser alta (94%) indica que o modelo, em geral, acerta bastante. No entanto, a precisão baixa (8%) revela que, quando o modelo diz que algo é positivo, ele quase sempre está errado. No entanto, a especificidade do modelo foi de 97%. Essa discrepância acontece em situações em que os casos negativos são muito mais frequentes do que os positivos. Ou seja, o modelo classifica quase tudo como negativo. Com isso, ele acerta muito nos verdadeiros negativos, mas erra ao tentar identificar os verdadeiros positivos.

Portanto, apesar da impressão inicial de que o modelo é eficaz por conta da alta acurácia e alta especificidade, a baixa precisão revela que ele falha onde mais importa: detectar corretamente os casos positivos, artigos selecionados pelos autores originais. Somente dois mapeamentos sistemáticos tiveram

precisão maior que 50% (MS4 e MS5), acertando mais da metade dos artigos selecionados. Esses dois casos são os que tiveram a maior quantidade de artigos selecionados em seus mapeamentos sistemáticos originais, 116 e 143, respectivamente.

Os resultados comparativos para cada mapeamento sistemático são apresentados nas Tabelas 11 a 16, onde são apresentadas as análises comparativas da distância euclidiana do resumo à expressão de busca utilizando o método de incorporação de palavras *all-Mini-LM-L6-v2* e a análise de relevância dos resumos em relação à expressão de busca utilizando o GML ChatGPT para todos os seis mapeamentos.

Para essa comparação serão realizadas análises intermediárias ao TOP 200, para verificar o desempenho de cobertura desses modelos em números menores ao TOP 200, além dos parâmetros de grau de acerto e percentual de artigos analisados. O **grau de acerto** é a proporção de artigos selecionados que estão entre os N artigos com as maiores probabilidades de aceitação. Por outro lado, o **percentual de artigos analisados** é a proporção de artigos que foram analisados em relação ao total de artigos disponíveis para análise.

Para o MS1, usando o modelo de incorporação de palavras foi possível obter 100% de cobertura dos 24 artigos selecionados ao analisar os 200 resumos mais próximos da expressão de busca, um resultado bem superior ao do GML, que apresentou 79% de cobertura na mesma amostragem, como pode ser observado na Tabela 11. O total de 200 artigos ranqueados corresponde a um total de 28% da base de artigos do MS1: 200 de 721 artigos no total.

É possível observar que no TOP 50 a cobertura do modelo de incorporação de palavras obteve um percentual de cobertura de 63% dos artigos selecionados manualmente pelos autores originais. Além disso, a partir do TOP 100 não houve mais aumento considerável na cobertura, tanto para a incorporação de palavras quanto para o GML. Assim, no contexto do MS1, o TOP 100 seja mais adequado e já utiliza 14% dos artigos analisados, diminuindo pela metade o percentual a ser analisado em relação ao TOP 200.

Tabela 11 - Análise comparativa do modelo de incorporação de palavras (IP) all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS1

Maiores probabilidades de aceitação	Cobertura do IP	Cobertura do GML	Grau de Acerto do IP	Grau de Acerto do GML	Percentual de Artigos Analisados
TOP 5	8%	0%	40%	0%	1%
TOP 10	17%	13%	40%	30%	1%
TOP 24	46%	29%	46%	29%	3%
TOP 50	63%	54%	30%	26%	7%
TOP 100	88%	75%	21%	18%	14%
TOP 150	88%	79%	14%	13%	21%
TOP 200	100%	79%	12%	10%	28%

Para o MS2, usando o modelo de incorporação de palavras foi possível obter 97% de cobertura dos 32 artigos selecionados ao analisar os 200 resumos mais próximos da expressão de busca, resultado superior ao do GML, que apresentou 66% de cobertura na mesma amostragem, como pode ser observado na Tabela 12. O percentual de artigos selecionados no TOP 200 corresponde a um total de 54%, que são 200 de 373 artigos no total.

Para o MS2, somente a partir do TOP 100 houve uma cobertura maior que 50% para o modelo de incorporação de palavras. Interessante notar que até o TOP 10 o GML chegou a ter uma cobertura maior que o modelo de incorporação de palavras.

Tabela 12 - Análise comparativa do modelo de incorporação de palavras (IP) all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS2

Maiores probabilidades de aceitação	Cobertura do IP	Cobertura do GML	Grau de Acerto do IP	Grau de Acerto do GML	Percentual de Artigos Analisados
TOP 5	0%	3%	0%	20%	1%
TOP 10	3%	9%	10%	30%	3%
TOP 32	25%	13%	25%	13%	9%
TOP 50	47%	19%	30%	12%	13%
TOP 100	72%	38%	23%	12%	27%
TOP 150	91%	47%	19%	10%	40%
TOP 200	97%	66%	16%	11%	54%

Para o MS3, usando o modelo de incorporação de palavras foi possível obter 70% de cobertura dos 23 artigos selecionados ao analisar os 200 resumos mais próximos da expressão de busca, resultado inferior ao do GML, que apresentou 78% de cobertura na mesma amostragem, como pode ser observado

na Tabela 13. Os 200 artigos ranqueados correspondem a um total de 49% da base de artigos do MS3: 200 de 409 artigos no total.

Deve ser dado destaque para o grau de acerto do TOP 100 do modelo de incorporação de palavras, que apresentou um valor de 43%, empatando com o método de relevância do GML, sendo superado pelo GML a partir do TOP 100. Ou seja, nos 100 primeiros artigos elencados como mais relevantes, ambos os métodos tiveram o mesmo desempenho.

Tabela 13 - Análise comparativa do modelo de incorporação de palavras (IP) all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS3

Maiores probabilidades de aceitação	Cobertura do IP	Cobertura do GML	Grau de Acerto do IP	Grau de Acerto do GML	Percentual de Artigos Analisados
TOP 5	9%	9%	40%	40%	1%
TOP 10	13%	22%	30%	50%	2%
TOP 23	17%	22%	17%	22%	6%
TOP 50	30%	30%	14%	14%	12%
TOP 100	43%	43%	10%	10%	24%
TOP 150	57%	61%	9%	9%	37%
TOP 200	70%	78%	8%	9%	49%

Para o MS4, usando o modelo de incorporação de palavras foi possível obter 83% de cobertura dos 116 artigos selecionados ao analisar os 200 resumos mais próximos da expressão de busca, um resultado bem superior ao do GMLn que apresentou 69% de cobertura na mesma amostragem. Os 200 artigos ranqueados correspondem a um total de 36% do percentual de artigos analisados, 200 de 556 artigos no total.

Deve ser dado destaque para o grau de acerto do TOP 5 do modelo de incorporação de palavras, que apresentou um valor de 100%, indicando que os 5 artigos com maiores probabilidades de aceitação são artigos aceitos pelos autores do mapeamento sistemático em questão. Enquanto o TOP 5 do método pelo GML ficou em 60%, como pode ser observado na Tabela 14.

Tabela 14 - Análise comparativa do modelo de incorporação de palavras (IP) all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS4

Maiores probabilidades de aceitação	Cobertura do IP	Cobertura do GML	Grau de Acerto do IP	Grau de Acerto do GML	Percentual de Artigos Analisados
TOP 5	4%	3%	100%	60%	1%
TOP 10	6%	5%	70%	60%	2%
TOP 50	29%	25%	68%	58%	9%
TOP 100	49%	37%	57%	43%	18%
TOP 116	54%	44%	54%	44%	21%
TOP 150	67%	55%	52%	43%	27%
TOP 200	83%	69%	48%	40%	36%

Para o MS5, usando o modelo de incorporação de palavras foi possível obter 75% de cobertura dos 143 artigos selecionados ao analisar os 200 resumos mais próximos da expressão de busca, um resultado superior ao do GML, que apresentou 67% de cobertura na mesma amostragem. Os 200 artigos ranqueados correspondem a um total de 6% do percentual de artigos analisados: 200 de 3.273 artigos no total.

Deve ser dado destaque para o grau de acerto do TOP 10 de ambos os modelos, que apresentaram um valor de 100%, indicando que os 10 artigos com maiores probabilidades de aceitação e relevância foram aceitos pelos autores do mapeamento sistemático em questão, como pode ser observado na Tabela 15.

Tabela 15 - Análise comparativa do modelo de incorporação de palavras (IP) all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS5

Maiores probabilidades de aceitação	Cobertura do IP	Cobertura do GML	Grau de Acerto do IP	Grau de Acerto do GML	Percentual de Artigos Analisados
TOP 5	3%	3%	100%	100%	0%
TOP 10	7%	7%	100%	100%	0%
TOP 50	31%	32%	90%	92%	2%
TOP 100	57%	50%	81%	72%	3%
TOP 143	69%	60%	69%	60%	4%
TOP 150	70%	62%	67%	59%	5%
TOP 200	75%	67%	54%	48%	6%

Para o MS6, usando o modelo de incorporação de palavras foi possível obter 48% de cobertura dos 12 artigos selecionados ao analisar os 200 resumos mais próximos da expressão de busca, o que corresponde a um total de 26% do percentual de artigos analisados: 200 de 773 artigos no total. Este resultado foi

superior ao do GML, que apresentou 24% de cobertura na mesma amostragem, como pode ser observado na Tabela 16.

Tabela 16 - Análise comparativa do modelo de incorporação de palavras (IP) all-Mini-LM-L6-v2 e relevância do GML ChatGPT para o artigo MS6

Maiores probabilidades de aceitação	Cobertura do IP	Cobertura do GML	Grau de Acerto do IP	Grau de Acerto do GML	Percentual de Artigos Analisados
TOP 5	4%	8%	20%	40%	1%
TOP 10	4%	8%	10%	20%	1%
TOP 25	8%	12%	8%	12%	3%
TOP 50	16%	16%	8%	8%	6%
TOP 100	20%	20%	5%	5%	13%
TOP 150	36%	24%	6%	4%	19%
TOP 200	48%	24%	6%	3%	26%

Os principais resultados indicam que há similaridade nos resumos dos trabalhos aceitos e a expressão de busca e, assim, o processo de seleção de artigos no campo da Engenharia de Software pode ser parcialmente automatizado. O trabalho do pesquisador seria montar a expressão de busca do estudo de mapeamento sistemático que pretende fazer e realizar a análise dos primeiros 200 artigos retornados como segundo revisor.

É possível chegar a essa conclusão, porque cinco dos seis artigos analisados apresentaram uma cobertura superior a 70% e uma média de cobertura geral de 79% usando o modelo de incorporação de palavras e a distância euclidiana do resumo em relação à expressão de busca, no TOP 200, que demonstrou ser um número amostral aceitável. O artigo 6, que obteve pior resultado, teve uma cobertura de 48% nos trabalhos aceitos.

4.4 Discussão

O trabalho avança no estado da arte na exploração da automação da triagem baseada em resumo e expressão de busca, simulando o cenário real enfrentado por revisores humanos, na geração de benchmark com métricas claras e padronizadas com método comparativo reprodutível e prático e oferece evidência empírica oriunda de múltiplos estudos reais, ao trabalhar com 6 mapeamentos sistemáticos reais da Engenharia de Software em relação aos trabalhos relacionados.

Os duzentos artigos com maior probabilidade de aceitação em média representam 33% do percentual de trabalhos analisados, o que representa um ganho de tempo da ordem de 67% em comparação com a verificação manual de todos os artigos, conforme demonstrado na Tabela 17. Esta tabela apresenta o percentual de artigos analisados pelo pesquisador depois das ações da automação proposta neste trabalho, que é a proporção de artigos que foram analisados em relação ao total de artigos recuperados das bibliotecas digitais.

Tabela 17 – Percentual correspondente de 200 artigos em relação ao total de artigos disponíveis

Índice	Proporção de 200 artigos em relação ao total de artigos disponíveis	Percentual de Artigos Analisados
MS1	200 / 721	28%
MS2	200 / 373	54%
MS3	200 / 409	49%
MS4	200 / 556	36%
MS5	200 / 3.273	6%
MS6	200 / 773	26%
Média	33%	

Conclui-se que os 200 artigos com maior probabilidade de aceitação possuem uma cobertura média de 79% no modelo de incorporação de palavras, que é um modelo de fácil implementação e de código aberto, sendo superior ao modelo do GML ChatGPT, que obteve um resultado médio de cobertura de 64% no TOP 200.

Portanto, o uso dos primeiros 200 artigos aumenta a eficiência do processo de triagem assistida, mantendo um bom nível de cobertura e sem sacrificar a praticidade. Em contextos específicos, esse número pode ser ajustado para mais ou menos, de acordo com o escopo da revisão e a densidade de literatura relevante, mas como diretriz geral, o número 200 é justificado por evidência empírica e ampla adoção na prática científica.

Os modelos de incorporação de palavras e GML utilizados nesta pesquisa se diferem do ranqueamento clássico por relevância das principais bases de dados, que se baseiam principalmente em ocorrência literal de palavras. Os métodos propostos neste trabalho focam na similaridade semântica entre os resumos e a expressão de busca, utilizando representações vetoriais de

palavras e, com isso, é mais provável capturar sinônimos e contextos semelhantes do que utilizar uma abordagem baseada em contagem de palavras.

Os detalhes exatos dos algoritmos de ranqueamento disponíveis nas bibliotecas digitais não são totalmente públicos e podem variar entre bases. No entanto, os mecanismos conhecidos e, de forma resumida, os critérios de ordenação por relevância são apresentados para as principais bibliotecas digitais na Tabela 18: ACM Digital Library, IEEE Xplore e Scopus.

Tabela 18 – Critérios de Ordenação por Relevância das Principais Bases de Dados: ACM Digital Library, IEEE Xplore e Scopus

Critério	ACM Digital Library	IEEE Xplore	Scopus
Correspondência de termos	Alta (título, resumo, palavras-chave)	Alta (título, resumo, palavras-chave)	Alta (título, resumo, palavras-chave)
Peso do título	Muito alto	Muito alto	Alto
Peso do resumo	Alto	Alto	Alto
Palavras-chave dos autores	Médio	Médio	Alto
Citações recebidas	Baixo impacto	Moderado impacto	Alto impacto
Recência da publicação	Moderado	Moderado	Moderado
Transparência do modelo	Parcial	Parcial	Parcial

As bases como ACM, IEEE e Scopus, em sua *maioria, usam métodos de recuperação de informação clássicos, como: TF-IDF (Term Frequency-Inverse Document Frequency) ou BM25, que é uma variante do TF-IDF. O TF-IDF é uma técnica amplamente utilizada em recuperação de informação e processamento de linguagem natural para avaliar a importância de uma palavra em um documento dentro de um conjunto de documentos (corpus), combinando dois conceitos: frequência do termo no documento e raridade do termo. Um termo terá um peso alto se ele aparecer com frequência em um documento e for raro no conjunto de documentos. Esta foi uma limitação apresentada pelo modelo do GML do ChatGPT, que também utilizou essa técnica, por focarem na frequência de palavras, esse método não entende o significado das palavras — ele trata textos como simples conjuntos de termos, sem considerar sinônimos, contexto, ou relações semânticas entre os termos. As bibliotecas digitais utilizam variantes ao algoritmo base de TF-IDF, a saber:*

- **ACM Digital Library:** Apresenta um algoritmo de ranqueamento adaptativo que considera o comportamento temporal dos usuários, ajustando a relevância com base em padrões de acesso e citações ao longo do tempo. Maiores informações em: <https://library-guides.ucl.ac.uk/ACM-Digital-Library/viewing-results>
- **IEEE Xplore :** Utiliza uma combinação de correspondência de termos e análise de metadados, incluindo o uso do IEEE Thesaurus para melhorar a precisão na recuperação de documentos relevantes; Detalhamentos da IEEE Xplore Digital Library estão disponíveis em: <https://library.ashoka.edu.in/wp-content/uploads/2022/07/IEEE-Xplore-Searching-and-Saving-Searches-User-Guide.pdf>
- **Scopus:** Emprega um sistema de ranqueamento que integra contagem de citações, correspondência de termos e análise de metadados, além de utilizar técnicas de aprendizado de máquina para aprimorar a relevância dos resultados. Fonte: <https://elsevier.libguides.com/scopus/metrics>

Outras bases de dados sobre publicações científicas, como *Semantic Scholar* ou o sistema de recomendação da Google Scholar, já estão começando a utilizar métodos semelhantes ao proposto neste trabalho, utilizando modelos de incorporação de palavras para ranquear e sugerir artigos. Assim, a pesquisa aqui reportada apresenta um método moderno e alinhado com práticas de Processamento de Linguagem Natural avançado e recuperação de informação baseada em significado, e não apenas em correspondência textual. Isto faz com que se diferencie substancialmente dos algoritmos ainda utilizados nas principais bases científicas, que continuam priorizando contagem de palavras, mesmo quando incorporam métricas como citações ou recência.

Na próxima seção, serão discutidas as contribuições da pesquisa, suas limitações, possibilidades de trabalhos futuros e considerações finais em relação aos principais resultados encontrados nesta pesquisa.

5 Conclusão

5.1 Contribuições da pesquisa

Os resultados indicam que a similaridade entre os resumos dos trabalhos aceitos e a expressão de busca pode orientar o GML a retornar os 200 artigos mais relevantes, levando a uma economia média de 67% de tempo na seleção inicial de artigos se considerarmos os seis mapeamentos sistemáticos utilizados como benchmark neste trabalho. Modelos de incorporação de palavras, como all-Mini, Glove, Word2Vec e FastText, também foram avaliados para medir a similaridade vetorial entre resumos e a expressão de busca, fornecendo uma abordagem complementar para capturar a sensibilidade semântica, comparável aos GML.

Na base controlada de seis artigos de mapeamento sistemático em Engenharia de Software o método de incorporação de palavras do all-Mini obteve uma cobertura de 79% dos trabalhos selecionados, que é um modelo de fácil implementação e de código aberto, sendo superior ao modelo do GML do ChatGPT que obteve um resultado médio de cobertura de 64% dos trabalhos aceitos nos 200 artigos mais relevantes. O modelo DeepSeek foi um pouco melhor que o ChatGPT, atingindo 67% de cobertura.

A combinação dessas abordagens — GML orientados por análise de relevância e vetores de incorporação semântica — mostrou-se promissora na redução do esforço manual envolvido na triagem inicial, contribuindo para maior escalabilidade e replicabilidade dos mapeamentos sistemáticos. Além disso, a utilização de múltiplos modelos forneceu uma visão mais robusta da relevância, permitindo comparação cruzada.

5.2 Limitações da pesquisa

A pesquisa propõe do pesquisador atuar como segundo revisor no processo de seleção de artigos, analisando os resultados obtidos pelo modelo de incorporação de palavras ou pelo modelo de linguagem (GML) em relação à

expressão de busca. Essa etapa faz parte da automatização do mapeamento sistemático da literatura. O número de artigos a serem analisados pode ser definido conforme os objetivos do mapeamento. A sugestão apresentada neste estudo é que esse número seja de 200 artigos. No entanto, os resultados mostraram um grau de acerto significativo já no TOP 10. Além disso, foi observada uma boa cobertura de artigos relevantes no TOP 100, o que pode ser útil para quem deseja trabalhar com um volume menor de artigos.

5.3 Trabalhos Futuros

Como trabalhos futuros sugere-se inverter os papéis, o pesquisador atuar como primeiro revisor, e os modelos de linguagem atuarem como um segundo revisor, ou até mesmo, a revisão por pares ser feita por 2 modelos diferentes, até mesmo como uma estratégia para tratar uma possível alucinação dos modelos na seleção de artigos.

Outra sugestão é utilizar mapeamentos sistemáticos em outras áreas do saber, diferente da Engenharia de Software, para avaliar se os resultados serão próximos aos reportados no capítulo 4, além de expandir a análise e investigar diferentes técnicas de *prompting* no processo de seleção de artigos, fazendo uso de *RAG* com *GML*.

Utilizar o *GML* em outras etapas do mapeamento sistemático ou revisão sistemática, como a própria geração automática do estudo, também é possibilidade de trabalho futuro no artigo.

Segundo (Silva et al. 2025) artigo recente foi produzido um artigo de revisão científica usando ferramentas de IA sobre nanocamadas formadas por moléculas que se organizam na interface ar-água e que podem ser transferidas para superfícies sólidas, usadas em sensores e dispositivos ópticos. Além disso, especialistas nessa área foram chamados para validar a qualidade do texto final. A maioria dos especialistas responderam que a versão gerada pela IA era superior ao que eles esperariam de um aluno de doutorado ou pós-doutorando.

5.4 Considerações Finais

Segundo Gao (2023), a *Retrieval Augmented Generation* (RAG) pode ser vista como um sistema que combina a capacidade de encontrar informações relevantes (recuperação), interpretá-las, complementá-las (aumento) e gerar respostas completas e bem formuladas (geração), sendo uma estrutura de Inteligência Artificial para recuperar fatos de uma base de conhecimento externa para complementar o treinamento de GML com informações atualizadas e fornecer aos usuários uma visão sobre o processo gerador de GML.

A RAG é o processo de otimizar a saída de um GML, de forma que ele faça referência a uma base de conhecimento confiável fora das suas fontes de dados de treinamento antes de gerar uma resposta e estendendo os recursos do GML para domínios específicos ou para a base de conhecimento interna de uma organização sem a necessidade de treinar novamente o modelo, sendo uma abordagem econômica para melhorar a produção do GML, de forma que ele possa ser relevante, preciso e útil em vários contextos.

Este estudo concentrou-se exclusivamente na etapa de recuperação da arquitetura RAG (Retrieval-Augmented Generation), por meio do uso de modelos semânticos de incorporação de palavras para identificar artigos relevantes com base na similaridade entre resumos e a expressão de busca. Como diretriz para pesquisas futuras, propõe-se a extensão do processo com a integração da etapa de geração. Esses modelos poderiam ser empregados para interpretar o conteúdo dos resumos recuperados, justificar a relevância dos artigos selecionados, responder a perguntas específicas relacionadas ao tema de pesquisa e gerar sínteses contextuais. Essa ampliação permitiria uma abordagem mais qualitativa, explicativa e orientada ao conteúdo, contribuindo para uma seleção mais interpretativa dos estudos no contexto de mapeamentos sistemáticos.

Referências Bibliográficas

Alpaydin, E. (2020). *Introduction to Machine Learning* (4th ed.). The MIT Press.

Mehta, A. K-Nearest Neighbors (KNN) Algorithm for Machine Learning. Disponível em: <https://ai.plainenglish.io/k-nearest-neighbors-knn-algorithm-for-machine-learning-ab6f17df4a7f>. Acessado em: jun/2024.

Aoun, P.H.C., Nascimento, A. S., Adenilton J.S. (2021). Evaluation of Dimensionality Reduction and Truncation Techniques for Word Embeddings. DOI: <http://dx.doi.org/10.5753/eniac.2018.4477>. Acessado em: ago/24.

Christou, E., Parmaxi, A., Zaphiris, P. (2024). A systematic exploration of scoping and mapping literature reviews. Universal Access in the Information Society.

Cormack, G.V., & Grossman, M.R. (2014). Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, 153–162. Disponível em: <https://doi.org/10.1145/2600428.2609601>. Acessado em: mai/25.

Datacamp (2024). *Prompting de disparo zero: Exemplos, teoria e casos de uso*. Disponível em: <https://www.datacamp.com/pt/tutorial/zero-shot-prompting>. Acessado em ago/2024.

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171-4186. <https://arxiv.org/abs/1810.04805>. Acessado em fev/2025.

Elastic (2023). “O que é machine learning?”. Disponível em: <https://www.elastic.co/pt/what-is/machine-learning>. Acessado em jul/2024.

Elastic (2023), “O que são grandes modelos de linguagem (LLMs)?”. Disponível em: <https://www.elastic.co/pt/what-is/large-language-models>. Acessado em: jul/2024.

Elastic (2024), “O que são embeddings de palavras?”. Disponível em: <https://www.elastic.co/pt/what-is/word-embedding>. Acessado em: jul/2024.

Felizardo, K. R., Steinmacher, I., Lima, M.Ss, Deizepe, A., Conte, T.U. e Barcellos, M. P. (2024). “Data extraction for systematic mapping study using a large language model - a proof-of-concept study in software engineering”.

Disponível em: <https://conf.researchr.org/details/esem-2024/esem-2024-emerging-results-vision-and-reflection-papers/3/Data-extraction-for-systematic-mapping-study-using-a-large-language-model-a-proof-o>. Acessado em: jul/2024

Felizardo, K. R., Lima, M. S., Deizepe, A., Conte, T.U., Steinmacher, I. (2024). “ChatGPT application in Systematic Literature Reviews in Software Engineering: an evaluation of its accuracy to support the selection activity”. Disponível em: <https://conf.researchr.org/details/esem-2024/esem-2024-technical-track/2/ChatGPT-application-in-Systematic-Literature-Reviews-in-Software-Engineering-an-eval>. Acessado em: jul/2024.

Felizardo, K. R., Martins R. M., Höhn, E. N., Revisão Sistemática (RS) - Pesquisa Científica em Engenharia de Software, Mini-Curso ESELAW 2009, Notas do Curso.Fernandes, Anita M.R, et all. (2012). Mapeamento Sistemático para Diagnóstico de Processo de Software. Brasil. Disponível em: <https://www.aedb.br/seget/arquivos/artigos12/441623.pdf>. Acessado em dez/2023.

Feng, L.; Chiam, Y. K.; Lo, S. K. Text-mining techniques and tools for systematic literature reviews: A systematic literature review. In: IEEE. Asia-Pacific Software Engineering Conference (APSEC), 2017 24th. [S.l.], 2017. p. 41–50.

Galvão, M.C.B e Ricarte, I.L.M. (2019) “Revisão Sistemática da Literatura: Conceituação, Produção e Publicação”, Brasil. DOI: <https://doi.org/10.21728/logeion.2019v6n1.p57-73>. Acessado em out/2023.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai Yi, S., Jiawei, W.M., Wang, H. (2023). “Retrieval-Augmented Generation for Large Language Models: A Survey”, China. Disponível em: <https://arxiv.org/abs/2312.10997>. Acessado em jul/2024.

Guo, E., Gupta, M., Deng, J., Park, Y.J., Paget, M., Naugler, C. (2023) “Automated Paper Screening for Clinical Reviews Using Large Language Models”, Canada. Disponível em: <https://arxiv.org/abs/2305.00844>. Acessado em set/2023.

IBM (2018). What is the k-nearest neighbors algorithm? Disponível em: https://www.ibm.com/topics/knn?mhsrc=ibmsearch_a&mhq=KNN. Acessado em jun/2024.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., McHardy, R. (2023) “Challenges and Applications of Large Language Models”, England. Disponível em: <https://arxiv.org/abs/2307.10169>. Acessado em set/2023.

Kitchenham, B., 2007, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01, Departament of Computer Science Keele University, Keele.

Kitchenham, B., Brereton, P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1), 7-15. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0950584908001390>. Acessado em mar/2024.

Kuhrmann, M., Méndez, F., Daniel M. e Daneva, M. (2017). On the Pragmatic Design of Literature Studies in Software Engineering: An Experience-Based Guideline. *EUA*. DOI: 10.1007/s10664-016-9492-y. Acessado em nov/2023.

Liu, R. and Shah, N. (2023) “Reviewer GPT? An Exploratory Study on Using Large Language Models for Paper Reviewing”, *EUA*. Disponível em: <https://arxiv.org/abs/2306.00622>. Acessado em set/2023.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press.

Marshall, C.; Brereton, P. Tools to support systematic literature reviews in software engineering: A mapping study. In: IEEE. 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. [S.l.], 2013. p. 296–299. Disponível em: <https://ieeexplore.ieee.org/document/6681371/> Acessado em mai/2024.

McKinsey & Company. O estado da inteligência artificial em 2023: o ano do crescimento explosivo da IA Generativa. 2023. Disponível em: <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year/pt-BR>. Acessado em: set/2024.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. Disponível em: <https://arxiv.org/abs/1301.3781>. Acessado em fev/2025.

Miwa, M., Thomas, J., O'Mara-Eves, A., Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *Journal of Biomedical Informatics*, 51, 242–253. Disponível em: <https://doi.org/10.1016/j.jbi.2014.06.005>. Acessado em: mai/25.

Mourão, E., Pimentel, J.F., Murta, L., Kalinowski, M., Mendes, Emilia, C. e Wohlin, C. (2019). On the Performance of Hybrid Search Strategies for

Silva, J. C. M. C., Gouveia, R. P., Zielinski, K. M. C., Oliveira, M. C. F., Amancio, D. R., Bruno, O. M., & Oliveira Jr., O. N. (2025). AI-assisted tools for scientific review writing: Opportunities and cautions. *ACS Applied Materials & Interfaces*, 17(34), 47795–47805. <https://doi.org/10.1021/acsami.5c08837>

Systematic Literature Reviews in Software Engineering, Brazil. Disponível em: <https://doi.org/10.1016/j.infsof.2020.106294>. Acessado em nov/2023.

Nashwan, A. J. and Jaradat, J. H. (2023) “Streamlining Systematic Reviews: Harnessing Grandes Modelos de Linguagem for Quality Assessment and Risk-of-Bias Evaluation”, Qatar. Disponível em: <https://www.cureus.com/articles/178248-streamlining-systematic-reviews-harnessing-large-language-models-for-quality-assessment-and-risk-of-bias-evaluation#!/>. Acessado em set/2023.

Oliveira, L. (2023). O Guia Definitivo para Grandes Modelos de Linguagem (GMLs): Benefícios, Desafios e Estratégias de Uso, Brasil. Disponível em: <https://awari.com.br/o-guia-definitivo-para-large-language-models-GMLs/>. Acessado em out/2023.

O'Mara-Eeves, A. Thomas, J., McNaught, J., Miwa, M. e Ananiadou, S. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, BioMed Central, v. 4, n. 1, p. 5, 2015.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. In *EASE* (p. 68–77). Disponível em: http://robertfeldt.net/publications/petersen_ease08_sysmap_studies_in_se.pdf. Acessado em dez/2023.

Petersen, K., Vakkalanka, S. e Kuzniarz, L. (2015). “Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update”, Suécia. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2015.03.007>. Acessado em nov/2023.

Pilehvar, M.T. e Collados, J.C. (2020). *Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning*, Editora: Morgan & Claypool Publishers.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pretraining*. OpenAI Blog.

Robinson, A, Thorne, W., Wu, B.P., Pandor, A., Essat, M., Stevenson, M. e Song, X. (2023) “Bio-SIEVE: Exploring Instruction Tuning Large Language Models for Systematic Review Automation”, England. Disponível em: <https://arxiv.org/abs/2308.06610>. Acessado em set/2023.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing. 2019. Recuperado de <https://arxiv.org/abs/1910.01108>. Acessado em fev/2025.

Santos, C. M. C, Pimenta, C., A. M., Nobre, Moacyr, R.C. (2007). A Estratégia PICO para a construção da pergunta de pesquisa e busca de evidências. DOI: <https://doi.org/10.1590/S0104-11692007000300023> . Acessado em mar/2024.

Scannavino, K. R. F., Nakagawa, E. Y., Fabbri, S.C.P.F., Ferrari, F.C. (2017). Revisão Sistemática da Literatura em Engenharia de Software: teoria e prática. Rio de Janeiro: Elsevier. Acesso em: 29 set. 2024.

Siddaway, A. P.; Wood, A. M.; Hedges, L. V. How to do a systematic review: a best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses. *Annual Review of Psychology*, v. 70, n. 1, p. 747–770, 2019. DOI: <https://doi.org/10.1146/annurev-psych-010418-102803>. Acessado em out/2023.

Silva, G. A. e Neto, P. A. S. e Moura, R. Santos. Um Estudo sobre a Seleção Automática de Trabalhos em Revisões e Mapeamentos Sistemáticos da Literatura. 2019. Disponível em: http://repositorio.ufpi.br:8080/xmlui/bitstream/handle/123456789/2179/disserta%C3%A7%C3%A3oGleisonAndrade_final.pdf?sequence=1. Acessado em mai/2025.

Souza, A. (2023). GenAI - O que é Engenharia de Prompt?. Brasil. Disponível em: <https://medium.com/blog-do-zouza/genai-o-que-%C3%A9-engenharia-de-prompt-6d416afe1323>. Acessado em out/2023.

van de Schoot, R., de Bruin, J., Schram, R., Zahedi, P., de Boer, J., Weijdem, F., ... & Flügge, M. L. (2021). An open-source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*, 3(2), 125–133. Disponível em: <https://doi.org/10.1038/s42256-020-00287-7>. Acessado em mai/25.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is All You Need*. Advances in Neural Information Processing Systems, 30, disponível em: <https://arxiv.org/abs/1706.03762>. Acessado em fev/2025.

Waffenschmidt S, Knelangen M, Sieben W, Bühn S, Pieper D. *Single screening versus conventional double screening for study selection in systematic reviews: a methodological systematic review*. BMC Medical Research Methodology, vol. 19, Art. 132, 2019.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2013). Active learning for biomedical citation screening. *Journal of the American Medical Informatics Association*, 20(6), 969–980. Disponível em: <https://doi.org/10.1136/amiajnl-2012-001461>. Acessado em mai/25.

Wallace, B. C., Trikalinos, T. A., Lau, J., Brodley, C. E., & Schmid, C. H. (2010). Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, 11, 55. Disponível em: <https://doi.org/10.1186/1471-2105-11-55>. Acessado em mai/25.

Wohlin C. (2014). Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering, Suécia. Disponível em: <http://dx.doi.org/10.1145/2601248.2601268>. Acessado em nov/2023.

Yin, C. e Zhang Z. (2024). A Study of Sentence Similarity Based on the All-minilm6-v2 Model With "Same Semantics, Different Structure" After Fine Tuning. Disponível em: <https://www.atlantis-press.com/article/126004096.pdf>. Acessado em mai/2025.

Ziger, G. e Pereira, F. R. (2021). Comparação de técnicas de word embedding na análise de sentimentos. Brasil. Disponível em: https://repositorio.ifsc.edu.br/bitstream/handle/123456789/2056/Gustavo_Ziger_TCC_2021.pdf?sequence=1&isAllowed=y. Acessado em ago/2024.