Uso de LLMs no apoio à geração de strings de busca para o desenvolvimento de Estudos Secundários

Maria Luísa de Barros Costa Silva

PÓS-GRADUAÇÃO FACOM-UFMS

Data de Depósito:

Assinatura:_____

Uso de LLMs no apoio à geração de strings de busca para o desenvolvimento de Estudos Secundários[1]

Maria Luísa de Barros Costa Silva

Orientador: *Prof. Dr. Bruno Magalhães Nogueira*

Dissertação apresentada à Faculdade de Computação FACOM - UFMS, para o Exame de Dissertação, como parte dos requisitos necessários para a obtenção do título de Mestre em Ciência da Computação.

UFMS - Campo Grande
Março/2025

[1]Trabalho Realizado com Auxílio da CAPES Proc. No: 88887.843052/2023-00

À CAPES,

Aos meus pais,
Marlene e Ricardo,

À minha irmã,
Fernanda,

À minha namorada,
Giovanna,

Maria Luísa de Barros Costa Silva.

Agradecimentos

Agradeço, primeiramente, à Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro que viabilizou o desenvolvimento desta pesquisa.

Expresso também minha gratidão aos professores e colegas que fizeram parte da minha formação, com destaque especial ao meu orientador, Prof. Dr. Bruno Magalhães Nogueira, cujo tempo e dedicação, ao longo dos meus quatro anos de graduação e dois anos de pós-graduação, foram fundamentais para meu aprendizado e crescimento acadêmico, profissional e pessoal. Agradeço, ainda, ao meu colega de graduação e mestrado, Me. Demetrius Moreira Panovitch, por sua valiosa contribuição e parceria que garantiu o sucesso deste trabalho.

Sou imensamente grata aos meus pais, Marlene e Ricardo, pelo apoio incondicional e pelos cuidados inestimáveis que me permitiram dedicar anos da minha vida à minha formação. À minha irmã, Fernanda, da qual empenho e excelência sempre serviram de inspiração ao longo da minha jornada acadêmica. À minha namorada, Giovanna, que esteve ao meu lado em todos os desafios e percalços, garantindo que eu nunca desistisse.

Por fim, compartilho essa conquista com toda a minha família e, em especial, com meu primo Gabriel. Ainda que ele não tenha podido vivenciar muitas das possibilidades desta vida, sei que agora compartilha, cada uma das conquistas de nossa família como se fossem suas.

Resumo

Estudos Secundários (ESs) são uma metodologia amplamente utilizada no meio científico da Engenharia de Software, desde a introdução do conceito de Engenharia de Software Baseada em Evidências. ESs possuem por objetivo coletar todas as informações disponíveis sobre um conceito ou fenômeno. Uma das etapas necessárias para o desenvolvimento de ESs é a definição e execução da estratégia de busca. A busca automatizada é uma das principais estratégias utilizadas no contexto de busca por estudos acadêmicos, e para realizá-la, o processo de geração e refinamento de *strings* de busca que irão ser aplicadas nas bibliotecas digitais é executado. Nos últimos anos, o domínio de tecnologia textual sofreu expressiva evolução com o avanço dos modelos de linguagem, sobretudo a partir dos *Large language models* (LLMs), que por meio da arquitetura *transformers* e uma grande gama de parâmetros, comportam alto desempenho semântico em conjunto a uma baixa complexidade de utilização. Baseando-se na dificuldade de construção de *strings* de busca, neste trabalho é proposta a criação da SeSGx-LLM. SeSGx-LLM é uma extensão do trabalho de Alves et al. (2022), responsável pela criação da *Search String Generator* (SeSG). A versão proposta neste trabalho possui como objetivo integrar LLMs ao *framework* da SeSG. Em conclusão, foi possível observar que LLMs podem contribuir beneficamente no processo de geração de sinônimos que irão compor as *strings*, sendo o Mistral 7B o modelo mais consistente testado. Em complemento, foi possível observar que o LDA obteve desempenho superior no processo de extração de palavras-chaves.

Palavras-chave: Estudos Secundários, *Large Language Model*, *Deep Learning*, *Strings* de Busca.

Abstract

Secondary Studies (SS) are a widely used methodology in the Software Engineering scientific field, since the introduction of Evidence-Based Software Engineering. The main objective of Secondary Studies is to gather all available information on a concept or phenomenon. One of the steps needed for the conduction of a SS is the definition and execution of a search strategy. One of the main strategies applied is the automated search, in order to perform this strategy, it is necessary to create and refine a search string that will be used in search engines. In the recent years, the textual technology domain has evolved greatly with the advance of Large language models (LLMs), which, through the transformers architecture and an expressive number of parameters, enable a high semantic performance combined with low complexity of use. Based on the difficulty in constructing search strings, this work proposes the creation of SeSG-LLM. SeSG-LLM is a tool based on the Alves et al. (2022) work, the Search String Generator (SeSG). The SeSGx-LLM version aims to integrate Large language models into the SeSG framework. In conclusion, the results demonstrated that LLMs can facilitate the generation of synonyms that will compose the strings, with Mistral 7B exhibiting the most consistent performance among the tested models. Additionally, the findings indicated that LDA demonstrated superior performance in the extraction of keywords.

Keywords: Secondary Studies, Large Language Model, Deep Learning, Search Strings.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AM Aprendizado de Máquina

BERT *Bidirectional Encoder Representations from Transformers*

BSB *Backward Snowballing*

ES Estudo Secundário

EP Estudo Primário

FD Frequência do Documento

FSB *Forward Snowballing*

FT Frequência do Termo

GS *Gold Standard*

IA Inteligência Artificial

LDA *Latent Dirichlet Allocation*

LLM *Large Language Model*

MLM *Masked Language Model*

MT Mineração de Textos

NLI *Natural Language Inference*

NLM *Neural Language Model*

NLP *Natural Language Processing*

NSP *Next Sentence Prediction*

PLM *Pre-trained Language Model*

QA *Question Answering*

QGS *Quasi-Gold Standard*

SLM *Statistical Language Model*

CAPÍTULO

1

Introdução

Uma das maneiras de se reunir informações confiáveis e de qualidade de um domínio específico, é por meio do desenvolvimento de Estudos Secundários (ES) - estudo que visa sintetizar a analisar o estado da arte de dada temática. Para Kitchenham et al. (2015), o propósito de conduzir um ES é identificar todas as evidências científicas disponíveis de determinado assunto ou fenômeno de interesse que foram documentados em Estudos Primários (EP). Um EP pode ser descrito como um conjunto de estudos empíricos que fazem medições diretas de objetos de interesse (Kitchenham et al. (2015)). Portanto, um ES pode ser definido como um conjunto de EPs.

Kitchenham et al. (2015) propuseram diretrizes para o desenvolvimento de ESs que resultassem em dados imparciais. Essas diretrizes são separadas em três grandes fases compostas cada uma por seus subprocessos: Planejamento, Elaboração, Documentação.

A etapa de Elaboração é essencial para a confecção de um ES. Um dos subprocessos dessa etapa é a seleção de estudos ou *screening*, que tem como objetivo buscar e identificar EPs que contenham informações confiáveis do determinado conceito de interesse. Para seguir a natureza sistemática dos Estudo Secundários, se faz necessário definir uma estratégia a ser seguida durante a busca, visando atingir a maior quantidade possível de dados que devem integrar o ES.

Entretanto, definir e seguir uma estratégia de busca é uma tarefa árdua, em termos de tempo e esforço, e está suscetível a erros humanos (Zhang et al. (2011)). Imtiaz et al. (2013) concluíram que a estratégia de busca é a etapa mais problemática e demorada para a elaboração de um ES. Portanto, é perceptível a necessidade de ferramentas que possam apoiar os autores no de-

senvolvimento de Estudos Secundários (Marshall et al. (2014)), sobretudo no subprocesso de seleção de estudos.

A seleção de EPs é composta por diversas tarefas a serem realizadas, como a definição e refinamento de *strings* de busca, identificação de estudos relevantes e aplicação dos critérios de inclusão e exclusão. Inicialmente, defini-se qual a abordagem de busca a ser utilizada, dentre essas abordagens existem: a **busca manual** que consiste em buscar manualmente em periódicos e anais de conferências por textos de interesse, a **busca automatizada** que corresponde na utilização de bibliotecas digitais para identificar EPs, e por fim, o ***Snowballing*** que é a utilização das referências e citações de estudos já considerados como relevantes, para a identificação novos estudos. É possível, e recomendado, o emprego de mais de uma abordagem para a busca de EPs, caracterizando a **busca híbrida** (Mourão et al. (2020)).

Dentre as estratégias de busca, a busca automatizada, também denominada busca em bases (Mourão et al. (2020)), é uma das mais utilizadas. Essa abordagem consiste na utilização de *strings* de busca em bibliotecas digitais, com o fim de consultar banco de dados on-line por textos condizentes com a *string* empregada. *Strings* de busca podem ser definidas como a combinação, por meio de operadores lógicos ("AND" e "OR"), de termos chave e seus sinônimos (Biolchini et al. (2005)).

Contudo, a construção de *strings* de busca está entre uma das maiores dificuldades inerentes ao desenvolvimento de Estudos Secundários (Mergel et al. (2015), Biolchini et al. (2005), Dybå and Dingsøyr (2008), Riaz et al. (2010)). Wohlin (2014b) mostram que existem problemáticas quanto a terminologia, de forma que nem sempre os mesmos termos são usados para definir um mesmo conceito, assim como uma mesma palavra pode possuir significados distintos para autores distintos ou domínios distintos. Devido a isto, além da identificação de palavras-chave, o autor ainda necessita encontrar termos similares ou sinônimos condizentes para compor uma *string* de busca.

Dessa forma, encontrar termos e relacioná-los, procurando atingir todos os estudos existentes de determinado domínio, de forma eficiente, é a razão pela qual a definição e refinamento de *strings* torna-se uma dificuldade aos autores. Devido a essa problemática, surgem diversos estudos com o propósito de criar ferramentas capazes de automatizar ou facilitar a criação de *strings* de busca.

Dentre os estudos com foco em apoiar a geração de *string* de busca, neste trabalho, é destacada a abordagem descrita por Alves et al. (2022), nomeada *Search String Generator* (SeSG). A SeSG, possui como objetivo, gerar *strings* de busca automaticamente, a partir de técnicas de Mineração de Texto, com a finalidade de apoiar estratégias de busca híbridas. Para extração de tópicos, o

latent Dirichlet allocation (LDA) (Blei et al. (2003)) foi aplicado a SeSG, sendo o LDA uma estratégia baseada na distribuição estatística de termos, ou seja, não considera o contexto empregado. Adicionalmente, a SeSG também utiliza o modelo BERT (Devlin et al. (2019)) - que foi responsável por abordar de forma inédita a arquitetura *transformers* - para a geração de sinônimos necessários para a composição da *string* de busca.

Entretanto, em busca de obter evidência quanto a novas alternativas do domínio de Processamento de Linguagem Natural, este trabalho visa propor uma nova ferramenta: *Search String Generator eXtended with Large Language Models* (SeSGx-LLM), que conta com uma completa refatoração da SeSG e com a inclusão de suporte para a utilização de grandes modelos de linguagem (LLMs, do inglês *large language models*).

O grande diferencial da ferramenta SeSGx-LLM, reside na capacidade de produzir *strings* de busca sem que o usuário defina os termos chaves ou sinônimos a serem utilizados. Com base apenas em um conjunto de artigos já identificados como relevantes, a SeSGx-LLM é capaz de retornar ao autor, *strings* que incorporam os termos significativos para o tema pesquisado, assim como os sinônimos destes termos.

A SeSGx-LLM conta com quatro etapas principais: Pré-processamento, Extração de tópicos e enriquecimento, Geração das *strings* de busca, Aplicação das *strings* de busca. A ferramenta utiliza como dado de entrada um conjunto de textos que já foi previamente definido como relevante à pesquisa, chamado *Quasi-Gold Standard* (QGS) (Zhang et al. (2011)). A segunda etapa da ferramenta, como o próprio nome sugere, conta com dois subprocessos: extração de tópicos e enriquecimento. Por tópicos entende-se como um conjunto de palavras-chave que descrevem os textos que servem de insumo à ferramenta. As palavras que compõe um tópico serão então entrada para o enriquecimento. A etapa de enriquecimento se faz necessária para expandir o vocabulário da *string* de busca, visto que os tópicos são oriundos de um conjunto finito e possivelmente não completo de um domínio. São estes os subprocessos que são realizados com LLMs.

LLMs (e.g., GPT-3.5 Turbo, Llama3, Mistral, GPT-4o Mini) são modelos de linguagem baseados em aprendizado profundo, que surgiram a partir do aumento da quantidade de parâmetros existentes em modelos de linguagem pré-treinados (PLMs, do inglês *Pre-trained language models*) (e.g., BERT, ELMo) (Zhao et al. (2023)). Esse aumento de parâmetros resultou em habilidades emergentes com alto desempenho em diversas tarefas de Processamento de Linguagem Natural (NLP, do inglês *Natural Language Processing*) (Wei et al. (2022a)).

A partir dessa premissa, este trabalho tem como objetivo explorar e cole-

tar evidências quanto ao uso LLMs no domínio de geração de *strings* para o desenvolvimento de ESs. O objetivo é aplicar as habilidades de LLMs para identificar tópicos latentes de um conjunto de textos, enriquecer os tópicos e gerar as *strings* de busca. Para essa finalidade, foi efetuado um processo de experimentação, a partir do emprego da SeSGx-LLM, para a coleta de métricas que possam medir o desempenho das *strings* de busca geradas com o apoio de LLMs.

1.1 Motivação e Objetivos

Estudos Secundários se tornaram uma metodologia de extrema importância para o âmbito de Engenharia de Software desde a introdução do conceito de Engenharia de Software Baseada em Evidências (Zhang et al. (2011)). Entretanto, o processo de desenvolvimento de ESs demandam tempo e alto esforço humano. Porém, há processos que podem ser aliviados com o apoio de técnicas de Inteligência Artificial (Mergel et al. (2015), Ros et al. (2017), Marcos-Pablos and García-Peñalvo (2020), Grames et al. (2019), Souza et al. (2018), Alves et al. (2022)). Portanto, esse trabalho possui como principal motivação contribuir para uma das etapas de desenvolvimento de Estudos Secundários, a geração de *strings*.

LLMs são modelos caracterizados pela alta quantidade de parâmetros e são pré-treinados em quantidades expressivas de textos. Considerados como estado da arte em NLP, esses modelos possuem alta capacidade em solucionar tarefas complexas com habilidades que não eram observadas em modelos anteriores (Zhao et al. (2023), Paaß and Giesselbach (2023), Wei et al. (2022b)). A natureza de comandos (*prompts*) dos LLMs ainda permite que, facilmente, ocorra a adaptação desses modelos para diversas tarefas, como sumarizar e extrair tópicos de um conjunto de textos de entrada, assim como gerar termos semelhantes aos tópicos extraídos. A possibilidade do uso de LLMs para automatizar a extração de tópicos e geração de sinônimos e termos similares que integram as *strings* de busca é o principal motivador desta pesquisa.

A hipótese deste trabalho é: LLMs tem melhor desempenho, dentro a métricas de precisão, revocação e *f1-score*, enquanto extratores de tópico e enriquecedores de termos no contexto de geração de *strings* de busca, comparado as abordagens previamente implementadas como LDA e BERT.

Com base na hipótese levantada, é proposta uma nova ferramenta, a SeSGx-LLM, que permite aplicar LLMs no contexto de geração de *strings* de busca para o desenvolvimento de ESs. Complementarmente, a SeSGx-LLM permite aplicar as *strings* de busca criadas com o objetivo de testar empiricamente seu desempenho. A partir da SeSGx-LLM, foi realizado um processo experimental

para a coleta de métricas que podem indicar se o emprego de LLMs beneficia ou não o trabalho de autores de ESs, ao diminuir o esforço manual empregado na etapa de seleção dos estudos primários que o compõe.

Para conduzir essa pesquisa foram definidas as seguintes questões de pesquisa:

- **QP-01**: Em quais cenários os LLMs, apresentam maior benefício na geração de *strings* de busca, em termos de precisão e revocação, ao compará-los a métodos tradicionais LDA e BERT?

- **QP-02**: Por meio da capacidade semântica existente em LLMs, esses modelos superam a abordagem estatística do LDA em termos de extração de tópicos latentes?

- **QP-03**: LLMs superam o PLM BERT como enriquecedor de termos para automatizar a geração de *strings* de busca?

- **QP-04**: Dentre as possibilidades de uso de modelos de linguagem (e.g., PLMs e LLMS), e modelos estatísticos (e.g., LDA), quais as melhores combinações de abordagens que minimizam o esforço do pesquisador, medido por meio da precisão, e mantém a completude da pesquisa, medido por meio da revocação?

- **QP-05**: Dentre os LLMs experimentados, qual obteve melhor desempenho na geração de *strings* de busca e qual obteve melhor custo-benefício?

- **QP-06**: LLMs podem auxiliar na geração de *strings* de busca quando o corpus de entrada utilizado é limitado ou especializado?

Ao final do processo experimental realizado nesta pesquisa, foi possível observar que o LDA, mesmo sendo uma abordagem puramente estatística, tem o melhor desempenho enquanto extrator de termos descritores. Entretanto, LLMs são pertinentes na tarefa de enriquecimento, obtendo resultados superiores ao resultados do modelo BERT. O Mistral 7B foi o modelo com maior constância nos resultados, gerando altos valores de revocação, o que é extremamente favorável para a completude dos Estudos Secundários.

A partir deste trabalho é possível afirmar que a SeSGx-LLM, além de beneficiar os autores na diminuição de esforços desprendidos ao automatizar a geração e refinamento de *strings*, é capaz de manter a qualidade da busca, além de apoiar em sua completude, sobretudo em um ambiente de busca híbrida.

1.2 Organização do Texto

Esse texto está divido em seis capítulos, além da Introdução, dispostos da seguinte maneira:

2. Estudos Secundários: contém contexto teórico sobre os conceitos de ESs fundamental para o desenvolvimento do trabalho.

3. Análise de Textos Utilizando Aprendiza de Máquina e Processamento de Linguagem Natural: contém contexto teórico sobre os conceitos de Mineração de Texto aplicados no desenvolvimentos da ferramenta SeSGx-LLM.

4. *Search String Generator eXtended with Large Language Models* (SeSGx-LLM): contém a descrição da ferramenta proposta neste trabalho e a descrição detalhada do seu funcionamento.

5. Experimentação e Resultados: contém a descrição do experimento e os resultados atingidos ao longo do desenvolvimento deste trabalho.

6. Conclusão: síntese dos resultados atingidos nesta pesquisa, limitações e dificuldades enfrentadas, futuros trabalhos e por fim, os disponibilidade dos artefatos gerados.

CAPÍTULO

2

Estudos Secundários

Um Estudo Secundário (ES) é um meio de identificar, avaliar e interpretar todas as pesquisas, e evidências disponíveis que sejam relevantes para uma determinada questão de pesquisa, tópico, ou fenômeno de interesse (Kitchenham (2004)). Os estudos que abordam as evidências inéditas a partir de dados empíricos são nomeados como Estudos Primários (EPs). Portanto, os EPs servem de insumo para os ESs. Dessa forma um ES agrupa as diferentes evidências abordadas em EPs de um determinado domínio, sintetiza-as e documenta seus achados de forma simática e estruturada.

Evidência científica pode ser considerada como um sustentador do conhecimento, e espera-se que o conhecimento seja derivado de alguma evidência interpretada (Kitchenham et al. (2015)). A natureza da interpretação pode se dar de diversas formas (e.g., qualitativa, quantitativa) e por diversos indivíduos. Uma vez que variados, os indivíduos podem reportar suas descobertas de conhecimento a partir de evidências. Como consequência, o elemento de diversidade será inevitavelmente ampliado (Kitchenham et al. (2015)).

Ao lidar com evidências científicas das quais seus valores e qualidade podem variar, se faz necessário a repetição da observação, ou até mesmo, reunir as diversas observações feitas por indivíduos distintos (Kitchenham et al. (2015)). Ao analisarmos a concepção de evidências no meio científico, temos dois conceitos vitais: Estudos Primários (EPs) e Estudos Secundários (ESs). Os Estudos Primários são aqueles que fornecem a evidência através de dados empíricos (Wohlin et al. (2012)). Já os Estudos Secundários representam aqueles que agrupam as evidências observadas que foram documentadas em EPs, a fim de buscar o estado da arte de determinado assunto (Kitchenham et al. (2015)).

Como forma de organizar e sistematizar a elaboração de ESs, Kitchenham et al. (2015) propuseram diretrizes para organizar e compartilhar esse conhecimento de maneira imparcial (Figura 2.1). Essas diretrizes são o conjunto de nove atividades, divididas nas seguintes fases:

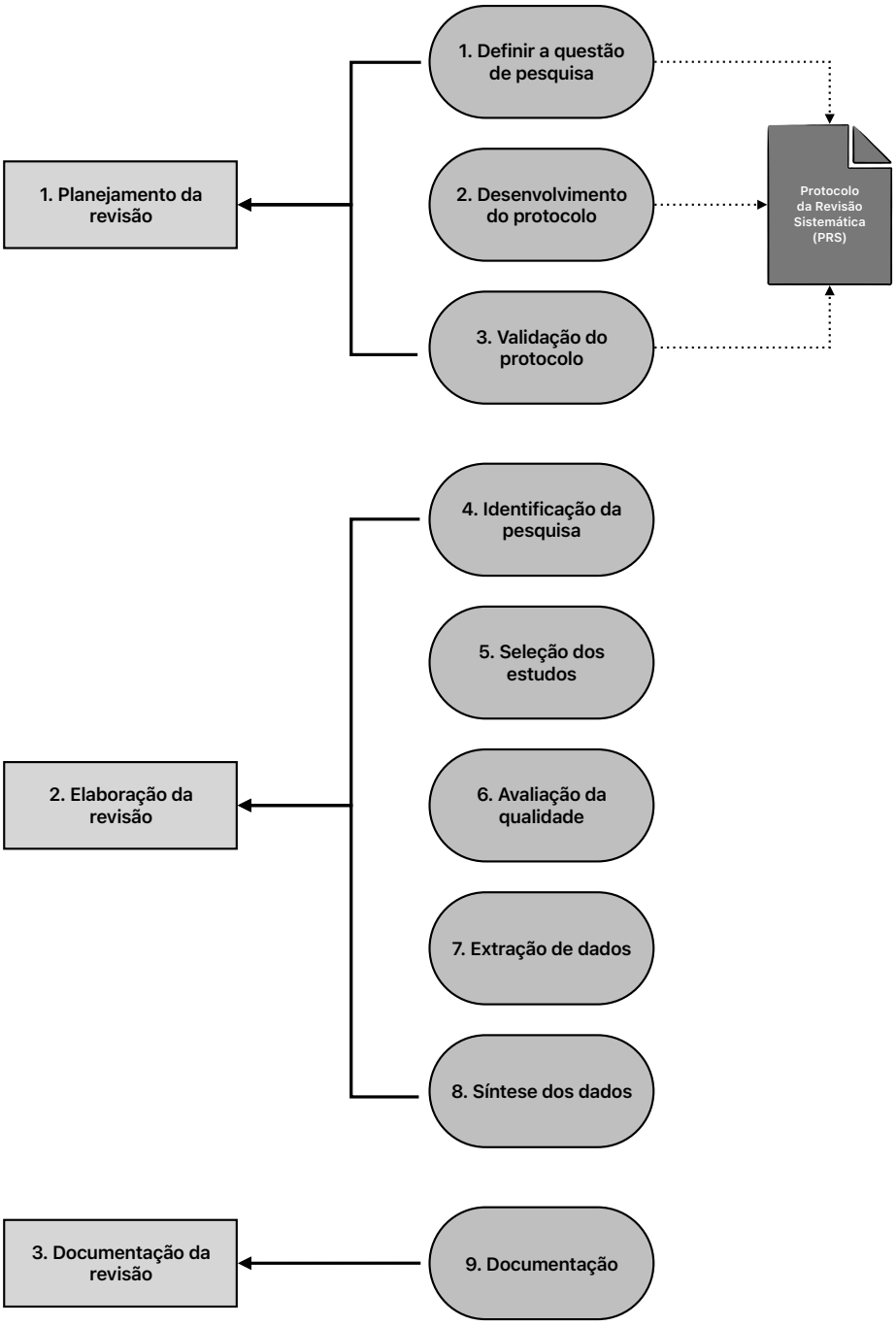

Figura 2.1: Etapas para o desenvolvimento de um Estudos Secundário. Baseado em Kitchenham et al. (2015).

1. Planejamento.

2. Elaboração.

3. Documentação.

2.1 Processo de Desenvolvimento de um ES

Nessa seção serão descritos em detalhes as etapas do processo de desenvolvimento de um Estudos Secundário, segundo (Kitchenham et al. (2004)).

2.1.1 Planejamento

Kitchenham et al. (2015) decompõe a fase de Planejamento em três atividades primordiais que resultarão em um protocolo para guiar o desenvolvimento do ES. As três atividades necessária nesta fase são: especificar a questão de pesquisa, desenvolver o protocolo e validar o protocolo.

O protocolo deverá conter os métodos que serão utilizados para efetuar diversas tarefas durante o desenvolvimento do ES (Kitchenham (2004)). O pesquisador deve definir informações como:

- Justificativa para a pesquisa;
- Questões de pesquisa;
- A estratégia de busca a ser aplicada na etapa de identificação da pesquisa;
- Os critérios para seleção de Estudos Primários;
- Os critérios para avaliação de qualidade dos Estudos Primários;
- A estratégia para extração de dados;
- A estratégia para a síntese dos dados extraídos;
- O cronograma do projeto;

O protocolo é indispensável para reduzir a probabilidade de viés do pesquisador, al;em de facilitar uma revisão por outros profissionais, pois permite o *feedback* quanto ao que será elaborado. E por último, servirá de base para as seções de introdução e metodologia do trabalho, que será ao final documentado (Kitchenham et al. (2015)).

2.1.2 Elaboração

A segunda fase foi segmentada em cinco atividades necessárias para a Elaboração, sendo elas:

1. **Identificação da pesquisa**: identificação de possíveis EPs que irão compor o ES, a partir da estratégia de busca definida no protocolo.

2. **Seleção dos estudos**: seleção, com base nos critérios de exclusão e inclusão definidos no protocolo, de quais estudos identificados estão dentro do escopo da pesquisa.

3. **Avaliação da qualidade**: aplicação dos critérios de qualidade definidos no protocolo.

4. **Extração de dados**: extração da informação dos EPs que irão responder as questões de pesquisa do ES.

5. **Síntese dos dados**: consiste em agrupar e resumir os achados dos dados extraídos.

A estratégia de busca é um elemento essencial. Seu objetivo é encontrar EPs dentro do escopo da pesquisa (Kitchenham et al. (2015)). Nessa etapa é preciso dar foco para a completude da busca, visando sempre alcançar a maior número de EPs possível.

Imtiaz et al. (2013) conduziram um estudo terciário (aquele em que ESs são agrupados) com o objetivo de reunir experiências quanto ao desenvolvimento de ESs. A partir de 116 estudos, Imtiaz et al. (2013) concluíram que a estratégia de busca é a etapa mais problemática e demorada para a elaboração de um ES. Os autores abordam também as dificuldades quanto a definição de *strings* de busca que não sejam nem tão específicas, nem tão genéricas, e que façam a correta junção de palavras-chave e sinônimos.

Dada a importância e dificuldade dessa etapa, se faz necessário a pesquisa e desenvolvimento de ferramentas que possam apoiar essa tarefa. Marshall et al. (2014) após conduzirem uma análise comparativa de quatro ferramentas destinadas ao apoio da elaboração de ESs, reforçaram a necessidade de esforço da comunidade para o desenvolvimento de mais ferramentas que possam dar suporte aos autores de ESs.

2.1.3 Documentação

Documentação e publicação do ES é a última fase descrita por Kitchenham (2004). A escrita de um relatório deve ser conduzida e compartilhada em canais coerentes ao tema do ES, de maneira a comunicar os achados à comunidade científica (Kitchenham (2004)). Os autores também sugerem uma estrutura para a escrita do Estudo Secundário.

2.2 Estratégias de Busca

A atividade de identificação da pesquisa contida na fase de Elaboração, como dito anteriormente, é de suma importância para o ES. Para que seja possível identificar EPs que sejam relevantes ao ES, estratégias de busca devem ser definidas na etapa de planejamento e documentadas no protocolo. Estas estratégias estão inseridas nas seguintes categorias: busca automatizada (2.2.1), busca manual (2.2.2), *Snowballing* (2.2.3) e busca híbrida (2.2.4).

2.2.1 Busca Automatizada

A busca automatizada consiste no uso de bibliotecas digitais e indexadores de textos para a procura de Estudos Primários. Para este fim, se faz necessário determinar quais as bibliotecas (e. g., IEEEXplore) e/ou indexadores (e.g., Scopus) que serão utilizadas, e qual a *string* de busca será empregada (Kitchenham et al. (2015)).

String de busca pode ser definida como a combinação de palavras-chave e seus sinônimos e termos similares através de operadores *booleanos*, que podem ser utilizados em mecanismos de busca. Mais sobre esse conceito será abordado na Seção 2.3.

2.2.2 Busca Manual

A busca manual resume-se em analisar manualmente periódicos e anais de conferências para obter os EPs da temática desejada. É necessário definir quais canais serão avaliados e qual o período de publicação que será investigado (Kitchenham et al. (2015)). Este método pode ser extremamente demorado e exaustivo (Zhang et al. (2011)), porém valioso para avaliação de completude que será tratada na Seção 2.4.

2.2.3 Snowballing

Snowballing refere-se a técnica de utilização de citações para encontrar estudos relevantes (Kitchenham et al. (2015)). Inicialmente, determina-se um conjunto de estudos já estabelecidos como relevantes e que serão inclusos no ES, nomeado *start set*. A partir dos estudos inseridos no *start set*, os autores então analisam os estudos que são citados (*Backward Snowballing*, BSB) e os estudos que os citam (*Forward Snowballing*, FSB).

Uma das vantagens do *Snowballing* é já possuir um conjunto de estudos relevantes inserido no contexto desejado. A utilização desses estudos facilita o direcionamento para identificação de novos artigos relevantes (Wohlin (2014a)).

Entretanto, a eficiência dessa estratégia depende profundamente da seleção adequada do *start set*. Para isso, Wohlin (2014a) definiu diretrizes para condução do *Snowballing*, em conjunto com características para um bom *start set*, sendo elas:

- Se os artigos relevantes são provenientes de diferentes comunidades, então é importante que estas estejam representadas no *start set*.
- A quantidade de artigos no *start set* não deve ser tão pequena. Entretanto, esse ponto irá depender da área de pesquisa a ser trabalhada. Uma temática mais específica, naturalmente, requer menos artigos incluídos no ES.
- Caso muitos artigo sejam inicialmente identificados, escolher entre os relevantes que são altamente citados pode ser uma alternativa.
- O *start set* deve ser diverso, cobrindo variados anos de publicação, autores e editores.
- É preferível a utilização de termos sinônimos ou similares das palavras-chave para cobrir diferentes terminologias ao buscar pelo *start set*.

2.2.4 Busca Híbrida

Todas as estratégias mencionadas acima não devem ser vistas apenas como alternativas entre si. Os métodos podem e devem ser combinados a fim de maximizar a melhor abrangência possível da literatura (Alves et al. (2022), Mourão et al. (2020)). A utilização de diversas estratégias de busca durante a atividade de identificação da pesquisa é denominada busca híbrida.

Mourão et al. (2020) conduziram um experimento para avaliar o desempenho de estratégias de busca híbrida. Como resultado, foi constatado que a busca em bibliotecas digitais, quando combinada com a estratégia de *Snowballing*, seja ela realizada de forma paralela ou sequencial, atingiu o melhor equilíbrio entre as métricas de completude, precisão e revocação (ver Seção 2.4).

Os autores ainda observaram que apenas uma iteração do *Snowballing* em conjunto com a busca em bibliotecas digitais forneceu entre 90% e 100% de revocação. Em relação as métricas, foi observado que a utilização de FSB foi responsável por melhorar a precisão, assim como o BSB aprimorou a revocação (Mourão et al. (2020)).

Mesmo com evidências que a utilização de estratégias de busca híbridas pode proporcionar aos pesquisadores maior cobertura do tema a ser pesquisado, com menor esforço (Mourão et al. (2020), Alves et al. (2022)), ainda se

faz necessário a criação e refinamento de *strings* de busca para completar a etapa da busca automatizada.

2.3 Strings de Busca

As *strings* de busca devem representar o tema pesquisado, a fim de retornar textos relevantes das bibliotecas digitais. Consequentemente, a qualidade do desempenho da busca depende da qualidade da *string* (Zhang et al. (2011)).

Para obter uma *string* de busca eficiente é necessário desprender tempo e esforço para a sua criação e refinamento. Idealmente, as *strings* devem retornar todos os Estudos Primários relevantes àquele tema, contendo o mínimo de ruído possível, a fim de reduzir o esforço empreendido na etapa de análise dos textos candidatos. A etapa de análise é também denominada como *screening*.

Na prática, as *strings* de busca são compostas por palavras-chave pertinentes as questões de pesquisa e os sinônimos desses termos. As palavras-chave são necessárias para que, dentro da biblioteca digital, atinja-se textos que estão dentro do domínio esperado. Já os termos sinônimos serão responsáveis por expandir o vocabulário da busca, permitindo alcançar textos relevantes que não necessariamente empregam de maneira explícita as palavras-chave definidas (Wohlin (2014b)).

Entretanto, com a finalidade de se obter todos os artigos relevantes à pesquisa é preciso a aplicação e refinamento iterativo da *string*. Mergel et al. (2015) concluíram que uma das maiores dificuldades na elaboração de ESs é a construção de *strings* de busca. Uma das razões para tal dificuldade reside na falta de formalização de terminologias (Wohlin (2014b)). Dessa forma é necessário expandir o vocabulário utilizado na *string* de busca, adicionando-se termos similares. De mesmo modo, a identificação dos termos representativos e seus sinônimos também é uma tarefa trabalhosa.

O processo de construção das *strings* de busca é realizado observando os seguintes princípios: as palavras-chave identificadas serão concatenadas pelo operador “AND”, visto que essas palavras em conjunto se complementam para representar o domínio da busca. Já os termos sinônimos devem ser concatenados a sua palavra originária pelo operador “OR” considerando que esses termos buscam expandir a busca, trazendo palavras alternativas que irão viabilizar uma maior abrangência.

Dentre as possíveis abordagens para automatizar os processos de construção e desenvolvimento de ESs, a criação e refinamento de *strings* é uma das etapas em enfoque de pesquisas acadêmicas. Para identificar soluções atuais dentro do contexto de geração automatizada de *strings* de busca, foram selecionados estudos publicados até o primeiro semestre de 2024. Com base nessa

busca, destacam-se as seguintes pesquisas:

- Mergel et al. (2015) desenvolveram uma ferramenta (SLR.qub), a partir de técnicas de Mineração de Texto Visual, que sugere novos termos a serem incluídos na *string* aos pesquisadores.
- Ros et al. (2017) propuseram um procedimento para automatizar a seleção de estudos para o ES. Uma das etapas da proposta visa criar *strings* de busca com base em árvores de decisão.
- A partir dos resultados da aplicação de uma *string* inicial (criada manualmente), Marcos-Pablos and García-Peñalvo (2020) recuperaram os resumos dos estudos retornados por essa busca. De forma manual ou utilizando técnicas de SVM (*Support Vector Machine*), esses resumos são rotulados como sendo relevantes ou não. A partir desse rótulo, são calculados termos dos quais os autores podem optar por inserir a fim de refinar sua *string* de busca.
- Grames et al. (2019), utilizando técnicas de Mineração de Texto e redes de co-ocorrência de palavras-chave (do inglês, *keywords co-occurrence networks*), identificam novos termos para serem aplicados na *string*. A abordagem dos autores, entretanto, depende de uma *string* inicial, assim como outras etapas que devem ser efetuadas manualmente pelo pesquisador.
- Souza et al. (2018) realizam a geração e refinamento de *strings* a partir da técnica de busca *Hill Climbing*. A abordagem criada, SBSG (*Search-based String Generation*), necessita que o pesquisador defina inicialmente um conjunto de termos, palavras-chave e sinônimos para que a ferramenta possa funcionar.

Um fator comum entre as pesquisas atuais que abordam a automatização da criação de *strings* de busca, é a necessidade dos autores previamente decidirem os termos e sinônimos, ou até mesmo uma *string* inicial. Dada a essa característica, Alves et al. (2022) propuseram uma ferramenta para contornar essa necessidade, a *Search String Generator* (SeSG). A SeSG necessita apenas de um conjunto de artigos já selecionados pelos autores para gerar *strings*, concatenando os tópicos extraídos dos documentos de entrada e seus sinônimos.

2.4 Análise de Completude

Por definição, um ES deve buscar reunir todos os EPs publicados que tenham relação com o tema desejado (Kitchenham et al. (2015)). Se faz necessá-

rio então, medir a completude do ES, e isso pode ser feito através das métricas: precisão e revocação.

A precisão (Equação 2.1) refere-se a proporção entre a quantidade de estudos relevantes encontrados (R_{enc}) e o total de estudos encontrados (N_{total}) (Kitchenham et al. (2015)).

$$\text{Precisão} = \frac{R_{enc}}{N_{total}} \tag{2.1}$$

A revocação (Equação 2.2) quantifica a proporção entre os estudos relevantes encontrados (R_{enc}) e o total de estudos relevantes existentes (R_{total}) (Kitchenham et al. (2015)).

$$\text{Revocação} = \frac{R_{enc}}{R_{total}} \tag{2.2}$$

Idealmente, busca-se a maior revocação e precisão possíveis. Temos que, Revocação = 1 indica que encontramos 100% dos estudos relevantes ao ES. E Precisão = 1 significa que 100% dos estudos encontrados são relevantes. Isso sugere que os autores não desprenderam esforços analisando estudos que não seriam incluídos. Todavia, é improvável que tal cenário ocorra, tornando necessário balancear as duas métricas de completude.

Petitti (2000) define como resultado de busca ótimo, o equilíbrio entre precisão e revocação. Dessa maneira, é possível aplicar a "medida de eficácia" (do inglês, *"effectiveness measure"*), estabelecida por Van Rijsbergen (1979): o F-Score, também chamado de F-Measure. O F-Score consiste na média harmônica entre precisão e revocação, assim como é mostrado na Equação 2.3.

$$F_\beta\text{-}Score = \frac{(1+\beta^2)\cdot \text{Precisão} \cdot \text{Revocação}}{(\beta^2 \cdot \text{Precisão}) + \text{Revocação}} \tag{2.3}$$

Pode-se balancear o F-Score utilizando $\beta = 1$, dessa maneira damos o mesmo peso para a revocação e precisão. Essa variação do F-Score é nomeada $F1$-Score (Equação 2.4).

$$F_1\text{-}Score = \frac{2 \cdot \text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{2.4}$$

Entretanto, é possível notar que o cálculo da revocação pode ser um problema. Visto que, é impossível saber de fato quantos EPs relevantes a um tema existem (Kitchenham et al. (2015)). Por essa razão, são definidos os conceitos de *Quasi-Gold* e *Gold standards* na seção a seguir.

2.5 Quasi-Gold Standard e Gold Standard

O *Gold Standard* (GS) representa o conjunto de artigos que devem estar indispensavelmente inseridos no ES (Zwakman et al. (2018)). A quantidade de estudos nesse conjunto, é o que foi representado por R_{total} na Seção 2.4. Entretanto, esse conjunto e seu real tamanho não são de fato conhecidos. Em função disso, Zhang et al. (2011) definiram o conceito de *Quasi-Gold Standard* (QGS), que consiste em um subconjunto do GS. Isto é, um o subconjunto de artigos relvante que de fato é conhecido.

Dentre as utilidades desse subconjunto tem-se a utilização do QGS para avaliação da completude e para o refinamento das *strings* (Kitchenham et al. (2015)), ilustradas na Figura 2.2. Uma das dificuldades durante a etapa de busca por EPs é avaliar o quão completo o ES deve ser. Por essa razão, Zhang et al. (2011) estabeleceram uma estratégia de busca que utiliza a revocação do QGS como base para continuar ou finalizar as iterações da busca automatizada. Ou seja, enquanto a busca não atingir determinado valor de revocação, se faz necessário repetir a busca automatizada.

Como complemento, Kitchenham et al. (2015) propuseram o uso do QGS para o refinamento de *strings* (Figura 2.2). A partir do QGS estabelecido previamente por busca manual, é possível extrair os termos que irão compor a *string* de busca inicial. Com uma primeira *string* definida, a busca automatizada deve ser executada iterativamente. A cada nova execução, a *string* empregada deve ser revisitada, com a finalidade de refinar os termos utilizados, até que um dado valor de revocação do QGS seja atingido.

Esse processo pode ainda ser acrescido dos passos necessário para se constituir uma busca híbrida (Alves et al. (2022)). Após a busca automatizada atingir o valor de revocação desejado pelos autores, os estudos relevantes que foram encontrados irão servir de *start set* para a técnica de *Snowballing*. É importante evidenciar que, a utilização do próprio QGS como *start set* é desencorajada, visto que esse conjunto pode enviesar a seleção de novos estudos (Alves et al. (2022)).

2.6 Considerações Finais

Estudos Secundários são de suma importância para a comunidade científica. O objetivo dessa pesquisa reside em expandir as evidências quanto as técnicas e algoritmos, relacionados a Mineração de Texto, para melhor apoiar autores de ESs na etapa de criação e refinamento de *strings* de busca.

Em virtude dos avanços significativos dos modelos de linguagem nos últimos anos, busca-se atrelar o poder semântico das LLMs ao extenso processo

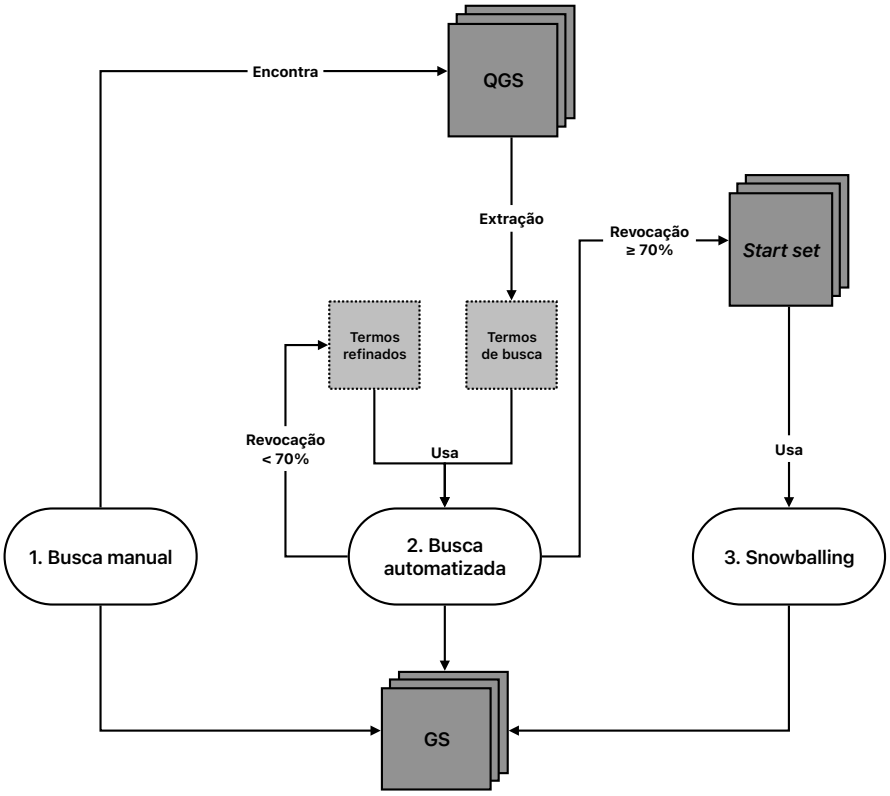

Figura 2.2: Processo para a realização de busca híbrida apoiada por *Snowballing*. Baseado em Alves et al. (2022).

de geração de *strings*. A partir de *prompts* de comando é possível extrair tanto as palavras-chave quanto os sinônimos que irão compor as *strings*, assim como o processo automatizado permite diminuir o esforço com as iterações de refino. No próximo capítulo, serão abordados os conceitos de Aprendizagem de Máquina e Processamento de Linguagem Natural que capacitam a construção da SeSGx-LLM.

CAPÍTULO

3

Análise de Textos Utilizando Aprendizado de Máquina e Processamento de Linguagem Natural

Uma das consequências do uso exponencial de computadores das últimas décadas é a geração de dados de maneira digital (Weiss et al. (2010)). A partir desse comportamento, surge o conceito de Mineração de Dados, responsável por encontrar padrões e informações valiosas no imenso volume de dados existentes. Entretanto, existem diversos tipos de dados (e.g., textos, imagens, vídeos, áudios) e para cada tipo existem determinados algoritmos a serem aplicados.

Para Aggarwal (2015), o objetivo da Mineração de Texto (MT) é encontrar conhecimento a partir de dados textuais. Com este fim, Rezende (2003), com base nos estudos de outros autores (Fayyad et al. (1996), Weiss and Indurkhya (1998)), determinou um processo de três etapas (Pré-processamento, Extração de padrões e Pós-processamento), que ao fim geram conhecimento que poderá ser aplicado ao domínio necessário.

Com base no processo definido por Rezende (2003), Alves et al. (2022) criaram uma ferramenta (SeSG) capaz de gerar e refinar *strings* de busca. A ferramenta conta com dois algoritmos essenciais, o *latent Dirichlet allocation* (LDA) e o BERT. A partir da SeSG, neste trabalho é proposta uma nova versão, a SeSGx-LLM, que permite a substituição do LDA e BERT pela aplicação de *Large language models* na etapa de extração de tópicos e enriquecimento.

Um das grandes evoluções em relação ao campo de aprendizado de máquina e mineração textual é a utilização de redes neurais profundas que permitiram relacionar as palavras com o seu contexto, evoluindo abordagens tradicionais que consideram apenas a frequência de aparição de termos. Em complemento, ao escalar essas redes neurais, tanto em quantidade de parâmetros quanto em quantidade de dados de treinamento, foi possível observar uma melhora significativa de desempenho em tarefas como raciocínio lógico, matemático e geração textual, iniciando uma nova era no campo de inteligência artificial, sobretudo no âmbito de modelos de linguagem. Portanto, a SeSGx-LLM conta com abordagens de aprendizado de máquina considerados estado da arte no domínio de PLN, e visa contrastar as abordagens mais tradicionais, LDA e BERT, com o fim de atestar experimentalmente o desempenho de LLMs quando aplicados ao domínio de geração de *strings*.

Nesse capítulo serão apresentadas as etapas do processo de MT, os algoritmos e o *framework* utilizados na SeSGx-LLM, e por fim, os conceitos necessários para o entendimento da proposta deste trabalho.

3.1 Etapas da Mineração de Texto

De acordo com Rezende (2003), o processo de MT em si, é composto por três etapas: Pré-processamento, Extração de padrões e Pós-processamento. Entretanto, Rezende (2003) adicionou ainda duas etapas referentes ao conhecimento do domínio. Uma etapa que precede o processo de MT: Identificação do problema, e uma etapa sucessora: Utilização do conhecimento (Figura 3.1).

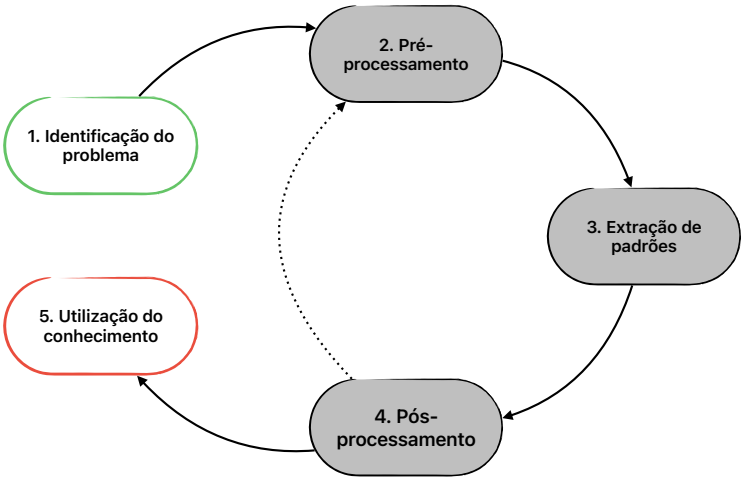

Figura 3.1: Fases do processo de Mineração de Texto. Baseado em Rezende (2003).

3.1.1 Identificação do Problema

Para que seja possível extrair informações úteis dos dados a serem minerados é necessário o estudo do domínio da aplicação e a definição de metas a serem atingidas. Rezende (2003) define quatro questões a serem respondidas nessa etapa:

- Quais as principais metas do processo de mineração?
- Quais critérios de desempenho são importantes?
- O conhecimento a ser extraído deve ser compreensível aos seres humanos ou um modelo caixa-preta pode ser utilizado?
- Qual deve ser a relação entre a simplicidade e precisão do conhecimento extraído?

As metas, objetivos e restrições levantadas nessa etapa serão úteis em todo o decorrer do processo de MT.

3.1.2 Pré-Processamento

Os dados disponíveis que serão minerados, normalmente não estão prontos para uso. Isto é, os dados podem não estar dispostos da maneira apropriada para a aplicação do algoritmo escolhido. Podem existir também, limitações de *hardware* (memória e tempo de execução), tornando necessário a utilização de métodos para tratar todas as adversidades do conjunto de dados.

As transformações que podem ocorrer nos dados, segundo Rezende (2003), são:

- **Extração e Integração**: os dados podem ser encontrados em diversas fontes e dispostos de diferentes maneiras (e.g., planilhas, bancos de dados). Será preciso então concatenar todas as bases de maneira a se obter apenas uma fonte, organizada no formato de atributo-valor.
- **Transformação**: os dados, dependendo de seu domínio, podem necessitar transformações a fim de se ajustar ao algoritmo a ser aplicado. Por exemplo, normalização de valores contínuos.
- **Limpeza**: é preciso corrigir os dados do conjunto que apresentarem inconsistências (e.g, erros de digitação) advindas do processo de coleta. Como o objetivo da MT é encontrar padrões, é indispensável se atentar a qualidade dos dados que serão utilizados no processo.

- **Seleção e redução de dados**: As limitações de *hardware* podem inviabilizar a aplicação de determinados algoritmos. Para isso, é pode ser preciso reduzir a quantidade de dados. Isso pode ser feito reduzindo o número de exemplos, reduzindo o número de atributos utilizados ou reduzindo o número de valores de um atributo. Porém, é importante manter a as características originais do conjunto de dados.

Portanto, a etapa de Pré-Processamento consiste em preparar os dados da melhor forma para que seja possível extrair, com qualidade, conhecimento ao final do processo de MT.

3.1.3 Extração de Padrões

A etapa de Extração de Padrões corresponde à escolha e aplicação dos algoritmos que serão utilizados para atingir os objetivos traçados na etapa de Identificação do Problema. É necessário decidir a natureza do problema, se contém atividades de predição ou descritivas, para assim decidir qual algoritmo melhor se encaixa na necessidade do projeto.

Uma vez que o algoritmo é definido, pode-se então executar de fato a extrações de padrões. É importante notar que o processo de MT é iterativo, ou seja, pode ser necessário revisitar etapas anteriores no intuito de alcançar o melhor resultado possível.

3.1.4 Pós-Processamento

Muitas vezes os algoritmos aplicados para a Extração de Padrões produzem enormes quantidades de padrões, dos quais nem todos são importantes para o objetivo traçado na etapa inicial. Por isso, é fundamental analisar quais as informações necessárias que de fato são valiosas ao usuário final. Além disso, é essencial atrelar medidas de desempenho (e.g., precisão, erro, sensitividade, cobertura) aos padrões encontrados. Em síntese, o propósito dessa etapa é trazer sentido para o que foi encontrado pelos algoritmos, e dispô-los de forma que o usuário possa entender e utilizar esse conhecimento.

3.1.5 Utilização do Conhecimento

A última etapa refere-se ao aproveitamento do que foi descoberto ao final do processo de MT. O conhecimento extraído por muitas vezes pode ser utilizado de modo a apoiar a tomada de decisões.

3.2 Processamento de Linguagem Natural

O domínio de Processamento de Linguagem Natural (PLN) surgiu para que fosse possível extrair padrões e conhecimento de textos. O principal objetivo de modelos de linguagem é predizer as probabilidades dos termos próximos ou faltantes (Zhao et al. (2023)). Zhao et al. (2023) dividiram a literatura de modelos de linguagem em quatro categorias: modelos de linguagem estatísticos (SLM, do inglês *statistical language models*), modelos de linguagem neurais (NLM, do inglês *neural language models*), modelos de linguagem pré-treinados (PLM, do inglês *pre-trained language models*) e por fim, grandes modelos de linguagem (LLM, do inglês *large language models*).

Os SLMs são modelos baseados em aprendizado estatístico e são conhecidos por ter grande dificuldade devido a problemática dimensionalidade (Zhao et al. (2023)). Em resposta as dificuldades e limitações atingidas pelos SLMs, ocorre o incentivo à pesquisa que gerou os modelos de linguagem baseados em redes neurais (e.g., word2vec (Church (2017))). Então, no ano de 2017, Vaswani et al. (2017) publicam o artigo que gerou um grande avanço para o domínio de IA como um todo, documentando a arquitetura *transformers* (do inglês, *transformers*). A arquitetura *transformers* e o conceito de atenção definidos por Vaswani et al. (2017), permitiu um salto de evolução que foi caracterizado pelo surgimento dos PLMs (e.g., BERT, GPT-2, BART). Os PLMs também trouxeram como inovação o conceito de atenção ao contexto bidirecional, gerando vetores de *tokens* capazes de serem representados em um espaço dimensional. Dessa forma o modelo é capaz de adicionar entendimento semântico para o cálculo de probabilidade dos termos. Conforme os PLMs foram evoluindo, esses modelos tiveram suas quantidades de parâmetros sendo cada vez mais aumentada, dando assim origem a quarta e última categoria: os LLMs.

Ao escalar a quantidade de parâmetros presentes nas redes neurais dos PLMs, os pesquisadores conseguiram identificar habilidades emergentes que aumentaram a capacidade linguística dos modelos de linguagem. Assim, foi iniciada um nova era no campo de NLP. No presente trabalho, temos por objetivo explorar as habilidades inéditas dos LLMs, e como comparativo o PLM BERT também será alvo do experimento prático realizado. A seguir, serão descritos todos os conceitos referente aos modelos de linguagem utilizados que motivaram esta pesquisa.

3.2.1 Latent Dirichlet Allocation

O latent Dirichlet allocation (LDA) é um modelo Bayesiano no qual cada item do conjunto de dados é modelado como uma mistura finita de um con-

junto de tópicos (Blei et al. (2003)). O funcionamento do LDA é baseado na ideia de que um conjunto de documentos (ou *corpus*) é composto por um conjunto de tópicos, que por sua vez, é constituído por um conjunto de palavras e termos existentes no vocabulário do *corpus*. Dessa forma, cada tópico é associado ao conjunto de dados, e assume-se que cada um desses tópicos está relacionado ao *corpus* em proporções distintas. Ou seja, cada tópico será representativo de um subconjunto do *corpus*.

No domínio de geração de *strings* de busca, o LDA é aplicado para a extrair os tópicos latentes presentes no QGS. Esses tópicos são compostos então pelas palavras-chaves que integrarão a *string*. Por se tratar de um método estatístico, o LDA não considera o contexto em que os termos estão inseridos, apenas sua distribuição entre os documentos existentes. Por esse motivo, levanta-se a hipótese de que determinados modelos de linguagens podem obter um desempenho superior a um modelo puramente estatístico, ao considerar o contexto semântico em que os termos estão dispostos.

3.2.2 Arquitetura Transformers

Um dos pontos de evolução dos modelos de linguagem recorrentes era sua difícil capacidade de paralelização (Vaswani et al. (2017)). Devido sua natureza de computação sequencial, modelos recorrentes foram alvo de pesquisa científica em prol da melhoria das limitações de *hardware* relacionadas ao treinamento desses modelos. Como resultado dessa evolução, o conceito de atenção começou a ser aplicado em modelos sequenciais, fundamentando também modelos convolucionais como ConvS2S (Gehring et al. (2017)) e o ByteNet (Kalchbrenner et al. (2016)). Os modelos convolucionais já apresentavam a utilização de cálculo paralelo, entretanto nesses modelos o número de operações não é constante, crescendo linearmente no ConvS2S e logaritmicamente no ByteNet (Vaswani et al. (2017))).

Por outro lado, na arquitetura Transformers, as operações possuem complexidade constante devido ao uso das médias dos pesos de atenção. No entanto, para mitigar essa limitação, foi introduzido o mecanismo de múltiplas cabeças de atenção. O conceito de atenção ou auto-atenção, pode ser definida como o mecanismo que busca relacionar as distintas posições de uma sequência, a fim de computar a representação dessa sequência (Vaswani et al. (2017)). Este processo é realizado a partir da Equação demonstrada em 3.1 na qual, Q e K são projeções das *embeddings* dos *tokens* e a multiplicação entre Q e K^t é um produto escalar que tem por função representar a relação entre as diferentes projeções de *embeddings*. Em mais alto nível, tais multiplicações significam encontrar a relação entre as palavras da entrada do modelo. Para evitar a dimensionalidade dos números encontrados pelo produto escalar de

$Q \cdot K$, esse resultado é divido por $\sqrt{d_k}$, no qual d_k é a dimensão de K. Dessa forma é possível aplicar a função $softmax$, assim obtém-se uma espécie de matriz de ativação, na qual os valores mais altos representam as *embeddings* que de fato têm alguma similaridade. As projeções de V então são multiplicadas por essa matriz de ativação, para que seja possível transmitir ao restante da arquitetura os valores ativados.

$$\text{Atenção}(Q,K,V) = softmax(\frac{QK^t}{\sqrt{d_k}})V \tag{3.1}$$

A arquitetura *transformers* é baseada no conceito codificador-decodificador (do inglês, *encoder-decoder*). Em cada uma dessas estruturas há camadas de multi-cabeças de atenção que permitem extrair diversas relações do conjunto de entrada. A disposição de múltiplas cabeças de atenção permite que a arquitetura seja altamente paralelizável, gerando um dos grandes benefícios que essa arquitetura introduziu no domínio de modelos de linguagem.

3.2.3 BERT

Bidirectional Encoder Representations from Transformers é um modelo de linguagem baseado na arquitetura *transformers* (Devlin et al. (2019)). Apresentado pela equipe de pesquisa Google, o BERT é um conhecido algoritmo de aprendizado profundo no domínio de PLN. O *framework* do BERT consiste em duas etapas: pré-treinamento e *fine-tuning*. Por Zhao et al. (2023), o BERT foi categorizado com um PLM, sendo um dos modelos de linguagem pioneiros em analisar o contexto bidirecional dos *tokens*.

A etapa de pré-treinamento é composta por duas atividades. A primeira refere-se ao processo denominado por *Masked Language Model* (MLM) (Modelo de Linguagem Mascarada, em tradução livre para o português) (Devlin et al. (2019)). O MLM consiste na ocultação aleatória de determinados *tokens* dos textos de entrada. O objetivo é que o modelo possa predizer qual o *token* que foi ocultado, baseando-se no contexto ao seu redor, tanto nos *tokens* que estão a sua esquerda, quanto aos que estão a sua direita, daí surge o nome *Bidirectional* (Devlin et al. (2019)).

A segunda atividade da etapa de pré-processamento é denominada pelos autores como *Next Sentence Prediction* (NSP) (Predição da Próxima Sentença, em tradução livre para o português). A fim de obter bons resultados em tarefas como *Resposta a perguntas* (QA, do inglês *Question Answering*) e *Natural Language Inference* (NLI) (Inferência de Linguagem Natural, em tradução livre para o português), é necessário que o modelo tenha bom entendimento da relação entre duas frases (Devlin et al. (2019)).

A segunda etapa, o *fine-tuning*, depende da necessidade de quem irá utilizar

o modelo. Devido a arquitetura *transformers* do BERT é possível adicionar as camadas anteriores a camada final do modelo para adaptá-lo a tarefa de interesse.

Os dados utilizados para o pré-treinamento do BERT são provenientes de dois conjunto de dados: BookCorpus (800M de palavras) e Wikipédia em inglês (2500M de palavras). Devlin et al. (2019), inicialmente, geraram dois modelos $\text{BERT}_{\text{base}}$ (110M de parâmetros) e $\text{BERT}_{\text{large}}$ (340M de parâmetros).

Dentro da SeSG, Alves et al. (2022) implementaram o $\text{BERT}_{\text{base}}$ para a predição de termos sinônimos aos termos extraídos como tópicos dos documentos de entrada. Termos similares são codificados com *embeddings* similares. Portanto, sinônimos podem ser encontrados pelo cálculo da distância de cossenos entre os vetores.

BERT é um modelo de linguagem reportado em 2018. No decorrer dos últimos seis anos, novas técnicas no âmbito de PLN surgiram. Um dos marcos mais importantes foi o lançamento do ChatGPT (Zhao et al. (2023)), desenvolvido pela OpenAI, responsável por popularizar o uso de LLMs em aplicativos orientados à conversas. O surgimento e a popularização de LLMs levam a necessidade de pesquisa quanto a essa tecnologia e o benefício que pode ser alcançado ao aplicá-la ao contexto de ESs, sobretudo no *framework* da SeSG.

3.2.4 Large Language Models

Para Zhao et al. (2023) os LLMs são o atual estágio dos modelos de linguagem. A evolução e ascensão dos PLMs, muito baseada na publicação do estudo que apresentou o BERT Devlin et al. (2019), iniciou um ciclo de pesquisas que exploraram os conceitos de pré-treinamento e ajuste fino (do inglês, *fine-tuning*), dando origem então a novos modelos como BART (Lewis (2019)) e GPT-2 (Radford et al. (2019)) (Zhao et al. (2023)). A partir desse novos resultados, busca-se então entender os limites dessa arquitetura quando expostas a condições de maior dimensão de parâmetros e dados de treinamento. O redimensionamento desse modelos resultou no que hoje é denominado *Large Language Models*.

Para o contexto deste trabalho temos que o desempenho exibido pelos LLMs em relação a percepção contextual, e facilidade de uso proporcionada pelo uso de *prompts* de comando fazem com que esses modelos sejam atrativos como abordagem alternativa para geração de tópicos e identificação de sinônimos, duas tarefas que dependem veemente do contexto de utilização. Na tabela 3.1 são apresentados alguns modelos de linguagem e seus tamanhos de acordo com a quantidade de parâmetros.

Para distinguir PLMs e LLMs, Zhao et al. (2023) definiu então três diferenças principais. Primeiramente, LLMs apresentam habilidades que não são ob-

Tabela 3.1: Modelos de linguagem.

Modelo	**Qtd. de Parâmetros**	**Categorização**
ELMo	93M	PLM
$\text{BERT}_{\text{base}}$	110M	PLM
$\text{RoBERTa}_{\text{base}}$	125M	PLM
$\text{BERT}_{\text{large}}$	340M	PLM
GPT-2	1.5B	PLM
Mistral	7B	LLM
Llama2	70B	LLM
Llama3	8B|70B	LLM
Gemma2	2B|9B|27B	LLM
GPT-3	175B	LLM

servadas em modelos menores (PLMs), como: aprendizagem contextual, capacidade de seguir instruções, raciocínio passo a passo. Outra grande diferença, é a capacidade de se utilizar LLMs através de interfaces de prompt (Zhao et al. (2023)). E por fim, o desenvolvimento de LLMs, segundo Zhao et al. (2023), extingue a distinção entre engenharia e pesquisa, dado que, treinar LLMs demanda muita prática na manipulação de conjuntos de dados de larga escala e conhecimento de treinamento paralelo. Sendo assim, é necessário que os pesquisadores resolvam complicados problemas de engenharia, não existindo mais, a clara distinção entre as duas áreas.

As habilidades que são observadas apenas em LLMs foram nomeadas como habilidades emergentes (Wei et al. (2022a), Zhao et al. (2023)). Para Zhao et al. (2023), as habilidades emergentes observadas em LLMs são:

- **Aprendizagem contextual**: habilidade de gerar a saída esperadas apenas através de instruções e demonstrações da tarefa, sem a necessidade de treinamento adicional.
- **Capacidade de seguir instruções**: a partir do *instruction tuning* - ajuste fino (do inglês, *fine-tuning*) com dados de multitarefas formatados em descrições de linguagem natural - LLMs demonstraram a capacidade de seguir instruções concedidas pelo usuário.
- **Raciocínio passo a passo**: habilidade de executar tarefas complexas que demandam múltiplos passos de raciocínio, como por exemplo, problemas matemáticos.

Devido a popularização de modelos caracterizados como LLMs, surge a necessidade de explorá-los, com a finalidade de se gerar evidências nos mais variados campos do conhecimento. Para estudar a aplicação desses modelos no domínio de ESs, utilizamos três LLMs nesse estudo: Mistral 7B, Llama3 8B, GPT-3.5 Turbo e GPT-4o Mini.

3.2.5 Mistral 7B

O Mistral 7B é um LLM *open-source* de 7 bilhões de parâmetros (Jiang et al. (2023)). Uma as principais vantagens em utilizar esse modelo é seu tamanho compacto e alta performance quando o comparamos com outros modelos *open-source*, como o Llama1 13B e o Llama2 7B, segundo Jiang et al. (2023).

Os autores atribuem a performance do Mistral 7B a duas técnicas que foram empregadas: *grouped-query attention* (GQA) e *sliding window attention* (SWA). O GQA possibilita acelerar a velocidade de inferência e reduz os requisitos por memória durante o processo de decodificação (Jiang et al. (2023)). Já o SWA permite que o modelo tenha acesso as camada empilhadas do *transformers* para que seja possível possuir informações para além do tamanho da janela (Jiang et al. (2023)).

O Mistral 7B foi escolhido para este trabalho devido o seu satisfatório desempenho em *benchmarks* e o fato de ser um modelo de código aberto, algo vital para aplicações acadêmicas, assim como interessante para futuros usuários.

3.2.6 Llama3 8B

O Llama3 8B é o modelo *open-source* mais recente lançado pela empresa Meta. Baseado em uma estrutura otimizada da arquitetura *transformers*, o Llama3 8B conta com técnicas como o ajuste fino supervisionado (do inglês, *supervised fine-tuning*) e aprendizado por reforço com *feedback* humano para que seja possível alinhar o desempenho do modelo com as preferências humanas de prestatividade e segurança (Dubey et al. (2024)).

Assim como o Mistral 7B, o Llama 3 também aplica a técnica de GQA. O modelo ainda conta com o oito mil *tokens* de comprimento de contexto e foi pré-treinado em uma base de dados com mais de 15 trilhões de *tokens* (Dubey et al. (2024)).

Esse modelo foi adicionado a esse trabalho por motivos análogos ao Mistral 7B, por se tratar de um modelo de código aberto. Além disso, trata-se de um dos modelos mais recentes disponíveis, lançado publicamente em Abril de 2024. Por meio de *benchmarks* realizados pelos autores do modelo, o Llama3 8B inclusive demonstrou pontuações superiores ao Mistral 7B, como é possível observar na Tabela 3.2.

3.2.7 GPT-3.5 Turbo

O terceiro modelo escolhido para este trabalho é o modelo GPT-3.5 Turbo da OpenAI. Por se tratar de um modelo proprietário, seus detalhes (e.g., tamanho do modelo, processo de treinamento e detalhes da arquitetura) não estão

Tabela 3.2: *Benchmarks* publicados pela Meta (Dubey et al. (2024)) comparando os modelos Llama3 8B e Mistral 7B.

Benchmark	**Llama3 8B**	**Mistral 7B**	**Descrição**
MMLU	**66.7**	63.6	Conhecimento geral (Hendrycks et al. (2021a)).
AGIEval English	**47.8** ±1.9	42.7 ±1.9	Capacidades gerais dos modelos em tarefas pertinentes à cognição humana (Zhong et al. (2023)).
BIG-Bench Hard	**64.2** ±1.2	56.8 ±1.2	Conhecimento de mundo e capacidade de resolução de problemas (Suzgun et al. (2022)).
ARC-Challange	**79.7** ±2.3	78.2 ±2.4	Matemática e raciocínio (Clark et al. (2018)).
DROP	**59.5** ±1.0	53.0 ±1.0	Matemática, raciocínio e resolução de problemas (Dua et al. (2019)).
MATH	**20.3** ±1.1	13.1 ±0.9	Matemática (Hendrycks et al. (2021b)).
WorldSense	**45.5** ±0.3	44.9 ±0.3	Matemática, raciocínio e resolução de problemas (Benchekroun et al. (2023)).
CommonSenseQA	**75.0** ±2.5	71.2 ±2.6	Raciocínio e entendimento de senso comum (Talmor et al. (2019)).
PiQA	81.0 ±1.8	**83.0** ±1.7	Raciocínio e entendimento de senso comum do mundo físico (Bisk et al. (2019)).
SiQA	**49.5** ±2.2	48.2 ±2.2	Raciocínio e entendimento de senso comum de interações sociais (Sap et al. (2019)).
OpenBookQA	45.0 ±4.4	**47.8** ±4.4	Raciocínio e entendimento de senso comum (Mihaylov et al. (2018)).
Winogrande	75.7 ±2.0	**78.1** ±1.9	Raciocínio e entendimento de senso comum (Sakaguchi et al. (2019)).

disponíveis ao público. Entretanto, é altamente provável que o modelo possua uma arquitetura baseada nos blocos decodificadores do *transformers*.

A OpenAI é uma das empresas que popularizou o uso de LLMs através da ferramenta ChatGPT, uma ferramenta capaz de gerar chats com os modelos LLM pertencentes a empresa. Hoje em dia, o uso gratuito ilimitado da ferramenta conta com o modelo GPT-3.5 Turbo.

A escolha desse modelo foi baseada no conhecido desempenho que os modelos da OpenAI possuem. O GPT-3.5 Turbo, assim como outros modelos GPT, podem ser utilizados por meio de uma API disponibilizada pela OpenAI, entretanto trata-se de um serviço pago. Cada modelo possui possui um preço por utilização, a depender do tamanho da entrada e saída do modelo em *tokens*. Foi observado que o GPT-3.5 Turbo possui ótimo custo-benefício para o nosso propósito. Mais informações sobre o custos do experimento desse trabalho são apresentados na Seção 5.1.

3.2.8 GPT-4o Mini

Assim como o modelo GPT-3.5 Turbo, o GPT-4o Mini é de autoria da OpenAI. Anunciado em Julho de 2024, o GPT-4o Mini foi apresentado como um modelo de alto desempenho e com o melhor custo-benefício da empresa. O modelo ainda apresentará suporte para variados tipos de dados de entrada como áudios, vídeos e imagens. Com uma janela de contexto de 128 mil *tokens* e com dados atualizados até Outubro de 2023, o GPT-4o Mini foi apresentado como sendo superior ao GPT-3.5 Turbo, com custo semelhante.

Devido a data de lançamento desse modelo, o GPT-4o Mini foi o último modelo a ser acrescentado nesta pesquisa. Sua escolha foi baseada nos mesmo motivos da escolha do GPT-3.5 Turbo. Além de apresentar um ótimo custo-benefício, o modelo possui *benchmarks* positivos, sendo um modelo competitivo comparado com outros modelos proprietários de mesmo tamanho.

3.2.9 Aprendizado por Contexto

Um *prompt*, ou comando, pode ser definido como um conjunto de instruções, em linguagem natural, que descrevem aquilo que o usuário deseja que o modelo gere (Liu et al. (2021)). O avanço e popularização dos LLMs motivaram pesquisas para o completo entendimento de qual seria melhor forma de construir um *prompt* para que seja possível extrair o máximo desempenho desse modelos.

Em modelos de linguagem é comum a realização de técnicas de ajuste fino (*fine-tuning*) para especializar o modelo de acordo com a necessidade do usuário. Essas técnicas, muitas vezes, demandam mais esforço, dados e tempo in-

Tabela 3.3: Comparativo e sumarização dos modelos utilizados nessa pesquisa. *Open-source* (OS), Quantidade de parâmetros (Qtd. params).

Modelo	**OS**	**Qtd. params.**	**Observações**
Mistral	Sim	7B	GQA e SWA
Llama3	Sim	8B	GQA, ajuste fino supervisionado e aprendizado por reforço com feedback humano
GPT-3.5 Turbo	Não	Sem informação oficial	Detalhes não publicados, API paga disponível
GPT-4o Mini	Não	≈8B (Não oficial, baseado em Abacha et al. (2024))	Modelo mais atualizado (dentre os escolhidos), maior custo-benefício da API

vestidos em treinar o modelo. Porém, um resultado similar pode ser atingindo de maneira mais direta em LLMs devido sua natureza orientada a comandos.

A partir de técnicas de aprendizado por contexto (*in-context learning*), os modelos podem ser ajustados à preferência do usuário, essas são: *Few-Shot* (FS), *One-Shot* (1S) e *Zero-Shot* (0S) (Brown et al. (2020)). FS consiste em adicionar ao prompt algumas demonstrações da tarefa que o usuário deseja que o modelo desempenhe. Analogamente, 1S consiste na adição de apenas um único exemplo da tarefa a ser desempenhada. E por fim, 0S representam os *prompts* que não contêm nenhum tipo de exemplo cedido pelo usuário.

Brown et al. (2020) demonstraram que, o modelo GPT-3, quando utilizado 1S ou FS, obteve desempenho maior quando comparado com o modelo 0S. O estudo foi feito a partir do modelo sem que fosse realizado nenhum procedimento de ajuste fino ou atualização de gradiente, apenas demonstrações foram inseridas nos comandos texto para interação com o modelo.

3.3 Considerações Finais

Baseado nos conceitos e modelos apresentados nesse Capítulo, surge a ferramenta SeSGx-LLM. Os avanços no domínio de Processamento de Linguagem Natural permitiram expandir a busca por meios de automatização das etapas de desenvolvimento de ESs. A estrutura de comandos ainda permite que LLMs possam ser vistas como abordagens genéricas altamente adaptáveis a diversos contextos. A partir dessas características, por meio deste trabalho, busca-se evoluir a ferramenta SeSG aumentando a diversidade de abordagens que possam ser testadas experimentalmente.

No próximo capítulo será elucidado as evoluções atingidas pela SeSGx-LLM, assim como detalhado seu funcionamento.

CAPÍTULO

4

SeSGx-LLM: Search String Generator Extended with Large Language Models

A complexidade inerente ao desenvolvimento de Estudos Secundários fomentou a pesquisa por automatização. Entretanto, Alves et al. (2022) identificaram na literatura a falta de estudos que visavam automatizar a etapa de geração e refinamento de *strings*. Por essa razão, foi desenvolvida uma ferramenta que, com apenas um conjunto de estudos pré-selecionados (QGS), é capaz de gerar *strings* de busca aos autores. Com base no *framework* desenvolvido por Alves et al. (2022), propomos neste trabalho a SeSGx-LLM (*Search String Generator Extended with LLM*). A SeSGx-LLM reproduz os passos do processo de Mineração de Texto descritas por Rezende (2003) e permite, de forma modular, escolher LLMs a serem utilizados, assim como os algoritmos e estratégias a serem aplicados em cada um dos processos da ferramenta.

Além da possibilidade de habilitar mais abordagens para compor o processo automatizado de geração de *strings* de busca, a SeSGx-LLM também evoluiu pontos vulneráveis que a SeSG apresentava. A SeSGx-LLM conta com uma arquitetura modular e independente de estratégia, permitindo facilidade de acoplamento de novas abordagens. Os processos presentes na ferramenta foram repensados de maneira a maximizar a possibilidade de uso de computação paralela e memória cache para trazer agilidade ao processo, permitindo expandir a quantidade de métricas levantadas, gerando maior confiabilidade ao processo experimental. Por fim, a ferramenta também foi acoplada a uma interface de linha de comando, aumentando sua usabilidade e a um banco de

dados para melhor persistir os dados gerados.

Como podemos observar na Figura 4.1, o processo empregado na SeSGx-LLM é constituído por quatro etapas. Estas são:

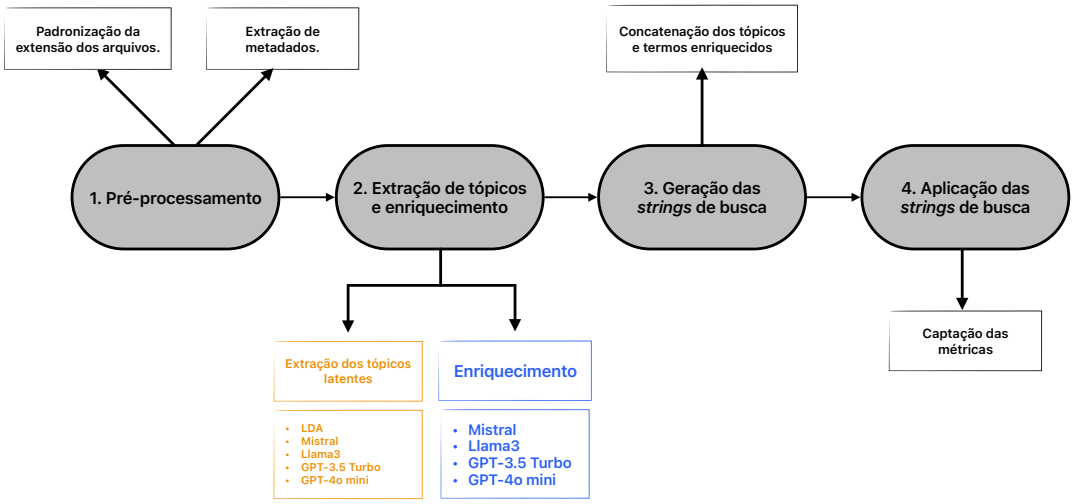

Figura 4.1: Processo executado pela ferramenta SeSGx-LLM.

1. Pré-processamento.
2. Extração de tópicos e enriquecimento.
3. Geração das *strings* de busca.
4. Aplicação das *strings* de busca.

A SeSGx-LLM faz parte do resultado de uma refatoração da ferramenta SeSG desenvolvida por Alves et al. (2022). O objetivo dessa refatoração foi aumentar a manutenibilidade da ferramenta, otimizando seus processos de maneira a preservar o *framework* original. Para que isso fosse possível, a arquitetura da ferramenta foi repensada de maneira a obtermos um pacote Python que opera de maneira agnóstica a qualquer algoritmo, apenas definindo as etapas a serem implementadas pela aplicação cliente. A partir do pacote que a SeSGx-LLM foi implementada, é possível a implementação de diversos algoritmos e técnicas para que essas possam ser igualmente experimentadas.

Como resultado da refatoração, também é enfatizada a otimização da ferramenta em termos de tempo de execução e melhor uso de *hardware*. Por meio de técnicas de programação paralela, o tempo de execução da ferramenta foi drasticamente reduzido. Em média, uma execução da ferramenta pré refatoração poderia levar mais de 24 horas e atualmente temos uma execução sendo concluída em menos de 2 horas.

4.1 Pré-Processamento

Os dados de entrada que serão utilizados como insumo da ferramenta são os estudos que integram o GS. Contudo, os textos podem estar disponíveis em diferentes formatos (e.g., PDF, linguagem de marcação), por esse motivo se faz necessário realizar a padronização dos dados. Para a SeSGx-LLM o formato utilizado é o texto simples (.txt), que permitirá a manipulação dos textos. Outro passo dessa etapa é a remoção de *stopwords*, palavras que não agregam significado semântico as frases e aparecem com alta frequência (e.g., "e", "a", "os") (Rajaraman and Ullman (2011)). A ferramenta SeSGx-LLM atualmente, possui suporte para textos escritos apenas em Inglês.

Os textos utilizados como entrada da ferramenta nada mais são do que um conjunto de palavras (*tokens*). Cada *token* pode ser visto com um termo simples, também nomeado de *unigram*. Entretanto, é possível buscar por termos compostos que tenham sentido semântico, estes são conhecidos como *n-grams*, onde n representa o número de *tokens* presentes nesse termo. Por exemplo, "inteligência artificial" é um temo composto *n-gram* com $n = 2$, ou seja, é composto por mais de um elemento mas possui apenas um significado semântico (Manning (2009)). Alves et al. (2022) definiram que, em uma *string* de busca, é incomum que existam termos compostos por mais de três palavras. Portanto, foi limitado que o número de *tokens* em um *n-gram* pode variar somente entre um e três ($n\_gram\_range = [1,3]$).

É necessário ainda realizar a remoção de termos infrequentes. Uma maneira para realizar essa filtragem é contar o número de documentos (estudos inserido no QGS) em que há ocorrência do termo, isto é conhecido como a Frequência do Documento (FD) de um termo. Alves et al. (2022) determinaram que o intervalo recomendável de FD mínimo está entre 10% a 40%. Porém, esse valor pode ser empiricamente ajustado.

A etapa de pré-processamento também é responsável por criar uma estrutura que relaciona cada termo com cada documento através da Frequência do Termo (FT). Essa associação é estruturada como um modelo de espaço vetorial denominado *bag-of-words*. A *bag-of-words*, por sua vez, é representada por uma matriz atributo-valor que servirá de entrada para a etapa de Extração de tópicos e enriquecimento. Tanto a definição do número de *tokens* em um *n-gram*, como a definição do intervalo adequado de FD mínimo, são necessárias para a execução do algoritmo que irá produzir a *bag-of-words*.

Vale ressaltar que, a criação de uma *bag-of-words* e os parâmetros relacionados a sua criação estão diretamente ligados com a utilização do LDA. Ou seja, os passos a serem realizados no pré-processamento da ferramenta dependem diretamente das estratégias a serem aplicadas nas próximas etapas.

Cada algoritmo tem sua particularidade a ser seguida.

É importante evidenciar que, a SeSGx-LLM, como dito anteriormente, utiliza os textos dos artigos inseridos no GS e seus metadados (i.e., título, resumo e palavras-chave) como insumo. Para a análise experimental da ferramenta, considera-se já ter conhecimento do GS, isto é, todos os artigos relevantes do determinado assunto de interesse do ES. Isto é justificado pela necessidade de extrairmos as métricas de completude atingida por cada *string* de busca, para que assim seja possível identificarmos o desempenho das estratégias aplicadas e da ferramenta em si.

Contudo, a geração das *strings* de fato, é realizada a partir apenas de um QGS selecionado aleatoriamente pela ferramenta. Por padrão foi definido que o QGS seria composto por um terço dos estudos do GS. Com o QGS aleatório selecionado é possível simular um situação real em que os autores tem acesso apenas a alguns dos artigos relevantes à sua pesquisa. O pré-processamento também tem por responsabilidade fazer a seleção aleatória do QGS e recuperar os metadados (título, resumo e palavras-chave) do QGS selecionado. Apenas os metadados servirão de insumo para a próxima etapa: Extração de Tópicos e Enriquecimento.

Em resumo, utiliza-se todos os textos do GS para avaliação da completude das *strings* geradas. Entretanto, para a construção das *strings* de fato, a SeSGx-LLM, utiliza como dado de entrada apenas os títulos e resumos dos textos de um QGS. Os textos por completo não são utilizados como dado de entrada devido ao tamanho que textos acadêmicos podem ter, além da possibilidade de do QGS ser composto por muitos estudos dependendo do domínio de interesse. Dado que os títulos e resumos possuem as informações mais latentes do estudos, não há perda significativa em se utilizar apenas esses metadados para a geração das *strings* de busca.

4.2 Extração de Tópicos e Enriquecimento

Nessa etapa duas atividades são realizadas: Extração de tópicos e o Enriquecimento dos tópicos extraídos. Alves et al. (2022) utilizaram originalmente para essas etapas o *latent Dirichlet allocation* (LDA) (Blei et al. (2003)) e o modelo *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al. (2019)) respectivamente. Para esse trabalho propomos a utilização de LLMs para as duas atividades. Entretanto, para fins de comparação, tanto o LDA quanto o BERT foram mantidos como possíveis algoritmos a serem utilizados na SeSGx-LLM, inclusive com a possibilidade de combinação entre as estratégias (e.g., extração de tópicos com LDA e enriquecimento com LLMs).

A primeira atividade dessa etapa trata-se da Extração de tópicos. No con-

texto deste trabalho, tópicos são definidos como um conjunto de palavras que descrevem um conjunto de textos. Ou seja, os tópicos representam os termos mais latentes, as palavras-chave, que integrarão a *string* de busca.

Como descrito na Seção 3.2.1, o LDA possui um funcionamento baseado na ideia de que um conjunto de documentos é proveniente de um conjunto de tópicos, dos quais cada tópico é definido como a distribuição de um vocabulário fixo de termos. Dessa forma, associa-se os tópicos ao conjunto de documentos e considera-se que cada tópico se relaciona em diferentes proporções com cada documento (Blei and Lafferty (2009)). Sendo assim, o LDA é embasado na distribuição estatística dos termos que compõem o *corpus*, ignorando o contexto em que esse termos estão inseridos. Como forma de contrapor a falta de relevância dado ao contexto dos termos pelo LDA, neste trabalho é proposta a utilização dos LLMs.

Para o uso do LDA, a SeSGx-LLM utiliza como insumo a *bag-of-words* gerada no Pré-processamento, onde foram removidas as *stopwords* e gerada a relação de frequência de cada termo. O algoritmo então computa a distribuição de palavras para cada tópico, e a distribuição de cada tópico por documento. Os tópicos mais latentes então representam as temáticas retratadas na coleção de documentos. A SeSGx-LLM considera esses tópicos como as palavras chaves que devem integrar a *string* de busca, com a finalidade de atingir mais estudos referentes aos tópicos extraídos.

Já para o uso dos LLMs, a SeSGx-LLM executa um processo em quatro etapas para a extrair os tópicos. Primeiro, calcula-se as *embeddings* dos metadados do QGS selecionado pela SeSGx-LLM, utilizando o pacote Setence-Transformers [1] e o modelo "all-MiniLM-L6-v2", por se tratar de um modelo que oferece um bom balanço entre agilidade e qualidade (Reimers and Gurevych (2019)). A segunda etapa realizada diz respeito a redução de dimensionalidade das *embeddings*, esse processo é realizado como forma de preparação para a seguinte etapa, o agrupamento. Para redução da dimensionalidade foi utilizado o algoritmo UMAP (*Uniform Manifold Approximation and Projection*), capaz de preservar mais da estrutura global e com melhor desempenho em tempo de execução (McInnes et al. (2018)).

A terceira etapa é o agrupamento das *embeddings* reduzidas, esse processo é realizado para que os textos similares, e consequentemente descritos pelos mesmos tópicos, possam ser dispostos juntos no *prompt* que será oferecido ao LLM. Para o agrupamento foi optado pela utilização do algoritmo *K-means* (MacQueen (1967),Lloyd (1982)). A partir do resultado do agrupamento, os metadados dos *clusters* identificados são agrupados em um *prompt* que será apresentado ao LLM, que fica responsável por extrair o tópico descritor da-

[1] https://sbert.net

quele grupo. Nota-se que, no contexto desta pesquisa um tópico é definido por um conjunto de palavras. O processo de extração de tópicos por LLMs pode ser observado nas Figuras 4.2 e 4.3.

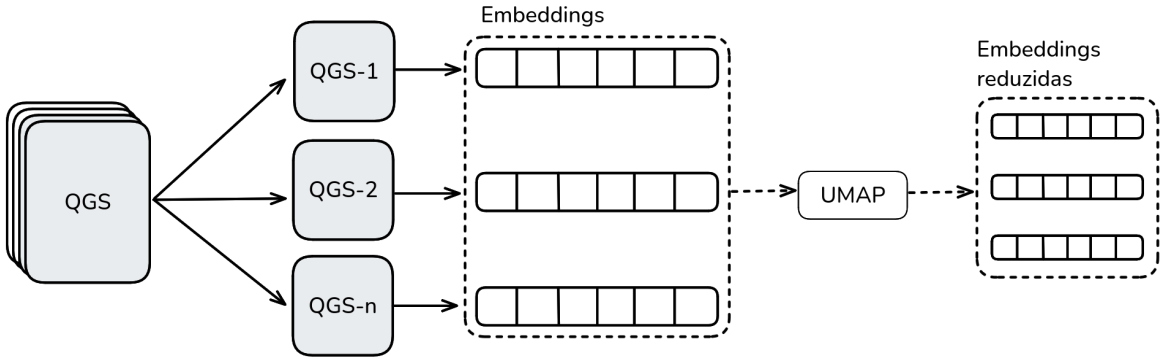

Figura 4.2: Representação do processo de extração de tópicos. Parte 1.

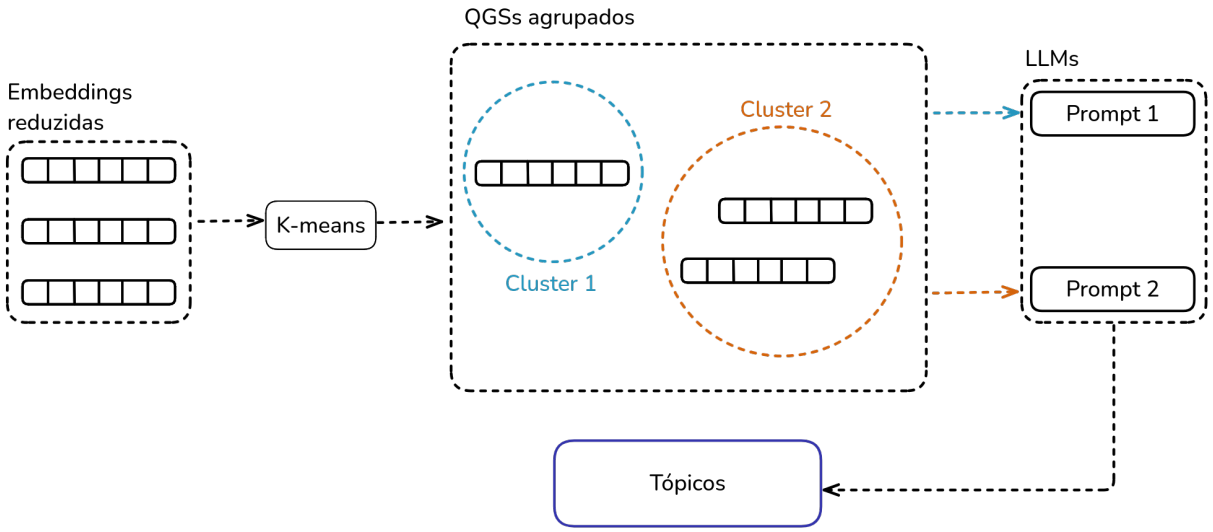

Figura 4.3: Representação do processo de extração de tópicos. Parte 2.

Com os tópicos extraídos, estes devem ser enriquecidos. Ou seja, ocorre a geração de termos similares para cada palavra pertencente a cada tópico encontrado, que serão incorporados nas *strings*. Esse processo é conduzido pois os tópicos extraídos tanto pelo LDA quanto pelos LLMs são referentes ao conjunto do QGS, um conjunto limitado que ainda não representa o conjunto total dos estudos relevantes. Assim sendo, os tópicos são oriundos de um vocabulário limitado presente apenas nesse conjunto, podendo então prejudicar a revocação da *string*. Os tópicos enriquecidos serão utilizados para aumentar a cobertura das *strings* de busca geradas.

Em conjunto a isso, Wohlin (2014b) demonstram que existem problemáticas quanto a terminologia, de forma que nem sempre os mesmos termos são usados para definir um mesmo conceito, assim como uma mesma palavra pode possuir significados distintos para autores distintos ou domínios distintos. Dessa maneira, para aumentar a probabilidade de se encontrar todos

os artigos relevantes, faz-se necessário expandir os termos que descrevem a temática da pesquisa, a fim de se atingir o máximo de terminologia que poderiam ser utilizadas.

Nessa etapa originalmente, Alves et al. (2022) aplicaram o modelo, que até o momento de sua publicação, era considerado o estado da arte em modelos de linguagem: o BERT. A partir da técnica de *Masked Language Model* (MLM), a palavra do tópico a ser enriquecida era mascarada para que o modelo pudesse predizer um termo que melhor se encaixasse naquele contexto.

Em contraponto, para a geração de termos sinônimos utilizando LLMs, os termos identificados como latentes são apresentados ao modelo, por meio de um *prompt* de comando, em conjunto do contexto em que aquele termo é utilizado no texto. Desta forma o modelo é capaz de gerar sinônimos considerando o contexto em que o termo originário é empregado assim como é feito com o BERT.

4.3 Geração das Strings de Busca

As duas atividades executadas na etapa anterior irão gerar os termos que serão concatenados com a finalidade de criar diferentes *strings* de busca. A primeira atividade irá produzir diferentes tópicos, conjunto de termos, que representam os textos dados como entrada. A segunda atividade produz os termos enriquecidos (sinônimos e similares) de cada termo dos tópicos extraídos.

A SeSGx-LLM necessita de um conjunto de hiperparâmetros para a formulação de *strings* da ferramenta (Tabela 4.1). Dentre os seis hiperparâmetros presentes na ferramenta, três são responsáveis por definir a estrutura da *string* de busca, estes são:

- **Número de tópicos** ($n\_topicos$): Representa quantidade de tópicos que serão associados os termos do conjunto de entrada.
- **Número de palavras por tópico** ($n\_palavras\_por\_topico$): Representa quantas palavras do tópico serão inseridas na *string*.
- **Quantidade de termos similares** ($n\_enriquecimentos$): Representa quantos termos similares de cada palavra do tópico serão inseridas na *string*.

Com os tópicos e termos similares gerados, a *string* de busca pode ser finalmente formada. O processo de Geração das *Strings* de Busca é realizado seguindo os mesmos princípios apresentados na Seção 2.3. De maneira a ilustrar esse princípios dentro do ecossistema da SeSGx-LLM, tem-se que: os termos enriquecidos serão concatenados à palavra de origem pelo operador

Tabela 4.1: Descrição dos hiperparâmetros da SeSGx-LLM.

Nome	**Descrição**	**Etapa**
n_palavras_por_topico	Número de palavras por tópico.	Geração de *strings*
n_enriquecimento	Número de enriquecimentos por palavra.	Geração de *strings*
n_topicos	Número de tópicos presentes na *string*.	Pré-processamento (LDA)
kmeans_clusters	Número de *clusters*. Parâmetro específico do *K-means*.	Extração de tópicos (LLM)
umap_vizinhos	Número de vizinhos. Parâmetro especifico do UMAP.	Extração de tópicos (LLM)
min_fd	Frequência no documento (FD) mínima.	Pré-processamento (LDA)

"OR". Isto é feito pois juntar esses termos tem por objetivo expandir a busca. Já os termos de um mesmo tópico serão concatenados entre si pelo operador "AND", visto que as palavras de um mesmo tópico descrevem um conjunto de documentos. E por fim, os tópicos são concatenados pelo operador "OR", uma vez que cada tópicos representa um subconjunto do GS. Esta estrutura está representada na Figura 4.4.

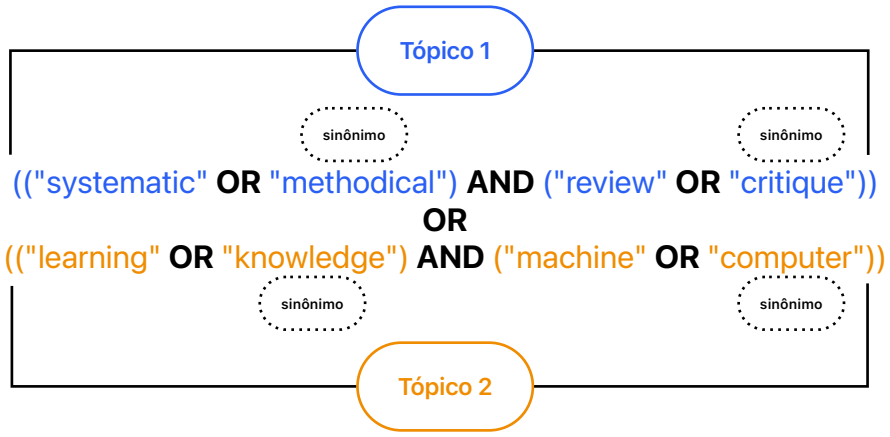

Figura 4.4: Ilustração de uma *string* gerada pela SeSGx-LLM. Baseado em Alves et al. (2022).

Ao final de cada *string*, se for necessário, ainda é possível adicionar o intervalo de ano de publicação. Isto é realizado pela ferramenta complementando a *string* com a restrição "PUBYEAR > ano mínimo de publicação" e "PUBYEAR < ano máximo de publicação".

Com este processo concluído, a *string* está pronta para uso na ferramenta

de busca Scopus, outras plataformas de buscas podem necessitar de ajustes de sintaxe para o correto funcionamento. A aplicação e obtenção de métricas de desempenho das *strings* é detalhada na sub-seção a seguir.

4.4 Aplicação das Strings de Busca

O último processo da SeSGx-LLM consiste em possibilitar a avaliação de qualidade das *strings* geradas pela ferramenta.

As *strings* criadas são aplicadas em plataformas de busca. Por padrão a SeSGx-LLM emprega as *strings* de busca na plataforma Scopus. Essa plataforma foi definida pela consistência em retornar elevados valores de revocação (Mourão et al. (2020)). A Scopus também dispõe de uma API que permite automatizar essa etapa.

A SeSGx-LLM é uma ferramenta pensada para estratégias de busca híbridas. Por esse motivo, através da lista de referência de cada texto dado como entrada, a SeSGx-LLM realiza a extração do relacionamento entre os artigos. Para essa finalidade, utiliza-se uma representação em grafos direcionados (Figura 4.5). No grafo de citações, os nós representam os artigos, e as arestas representam que um artigo X cita ou é citado pelo artigo Y.

A nível de código, os grafos são obtidos por meio de uma matriz de adjacência. Essa matriz é constituída a partir da busca dos títulos dos artigos do GS nos textos, caso um artigo referencie outro, o título estará presente no corpo do texto. Essa busca é feita a partir de técnicas de *string matching*, em específico a SeSGx-LLM utiliza a distância de Levenshtein (Levenshtein et al. (1966)), também conhecida como distância de edição, como forma de identificar os títulos que são referenciados em um texto. Desta forma, se o título de $M[0][0]$ está presente no texto de $M[0][1]$ tem-se que $M[0][1] = 1$ e $M[1][0] = 1$.

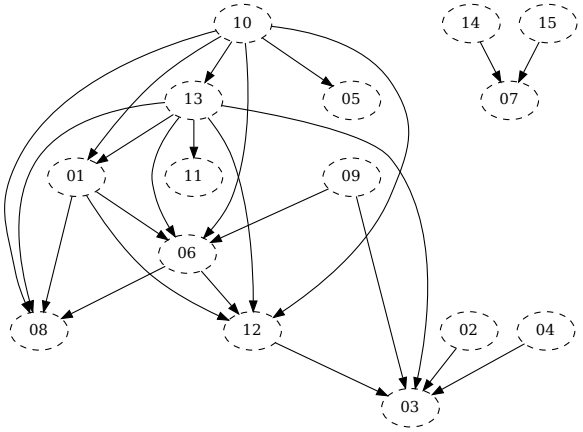

Figura 4.5: Exemplo de um grafo de citações.

Com os resultados retornados pela Scopus, e o grafo de citações criado, é possível obter métricas para avaliação do desempenho das *strings*. Essas métricas são:

- Revocação$_{sts}$: revocação alcançada em relação ao *start set*.
- Precisão$_{sts}$: precisão alcançada em relação ao *start set*.
- F1-score$_{sts}$: F1-score alcançada em relação ao *start set*.
- Revocação$_{final}$: revocação referente a etapa de *Snowballing*. Com base no grafo é possível determinar quais estudos seriam encontrados, a partir daqueles que foram diretamente atingidos pela *string* na busca automatizada, via análise de citações.

O *start set* representa o conjunto inicial que dará início ao processo de *Snowballing*. Como foi abordado anteriormente, a SeSGx-LLM foi pensada como uma ferramenta de apoio à busca híbrida, dessa maneira parte-se do pressuposto que os resultados da ferramenta serão complementados pelos resultados da aplicação da técnica de *Snowballing*. Por essa razão, utiliza-se métricas específicas em relação ao *start set*. Para a SeSGx-LLM entende-se por *start set* todos os artigos atingidos pela *string* de busca que foram identificados como pertencentes ao GS.

4.5 Considerações Finais

A SeSGx-LLM é composta por quatro grandes processos que irão resultar na criação e extração de métricas de *strings* de busca de forma automatizada. A partir da natureza modular da ferramenta, é possível ter diversos LLMs e estratégias aplicados. Por essa razão, no próximo capítulo será abordado detalhadamente todo o processo de experimentação realizado com o fim de gerar evidência de diversas combinações de modelos e algoritmos. Dessa maneira busca-se encontrar as melhores técnicas que irão de fato trazer benefícios para autores de Estudos Secundários.

CAPÍTULO

5

Resultados

Para obter evidências científicas do uso de LLMs no contexto de Estudos Secundários, neste trabalho foi desenvolvida uma ferramenta capaz de realizar experimentos que pudessem medir o desempenho desses modelos, a SeSGx-LLM. Nesse Capítulo será abordado o processo e design de experimentação realizados, assim como os resultados obtidos por meio da execução da SeSGx-LLM.

5.1 Design do Experimento

O experimento descrito a seguir foi projetado para avaliar a eficácia da ferramenta SeSGx-LLM. As *strings* de busca geradas por LLMs são colocadas em comparação àquelas que foram geradas pelas abordagens tradicionais (LDA e BERT). A partir das métricas alcançadas pelas *strings* de cada estratégia foi possível entender quais os benefício e limitações dos LLMs dentro do contexto de ESs.

Para avaliar o desempenho da SeSGx-LLM, foram selecionados 10 Estudos Secundários publicados como grupo experimental. Como entrada da ferramenta, foram utilizados os GSs definidos pelos autores de cada um dos 10 ESs. 3 dos estudos selecionados (Azeem, Hosseini e Vasconcellos) foram previamente utilizados como grupo experimental do estudo de Alves et al. (2022). Já os demais estudos foram selecionados visando expandir a diversidade e quantidade de estudos em prol da generalização dos resultados apresentados. Para a seleção desses estudos, utilizamos como critério de inclusão estudos secundários dentro domínio da computação e que possuíam publicados a lista

completa do GS utilizado. A lista completa do grupo experimental pode ser observada na Tabela 5.1.

Para cada execução da ferramenta, um terço do GS de cada estudo foi aleatoriamente selecionado, para representar o QGS. Ou seja, para de fato gerar as *strings* de busca, a SeSGx-LLM possui acesso apenas a um subconjunto aleatório do GS. Dessa forma é possível simular um cenário real no qual o autor do ES possui conhecimento de apenas alguns dos estudos relevantes inicialmente. O uso do GS é necessário apenas para que seja possível extrair as métricas de completude posteriormente.

Tabela 5.1: Grupo experimental.

Autor	**GS**	**QGS**	**Domínio**
Abayomi-Alli et al. (2022)	56	18	Ampliação de dados e métodos de aprendizado profundo para classificação de sonora.
Azeem et al. (2019)	15	5	Técnicas de aprendizado de máquina para detecção de *code smells*.
Bertolino et al. (2019)	147	49	Testes em nuvem.
Böhmer and Rinderle-Ma (2015)	152	51	Testes de *software*.
van Dinter et al. (2021)	41	13	Automação de revisões sistemáticas.
Dissanayake et al. (2022)	72	24	Segurança de *software*.
Cutigi Ferrari et al. (2018)	146	48	Técnicas de redução de custos para testes de mutação.
Hosseini et al. (2019)	46	15	Previsão de defeitos entre projetos.
Mohan et al. (2023)	76	25	Modelos preditivos para processos de tratamento de águas residuais.
Vasconcellos et al. (2017)	30	10	Abordagens para alinhamento estratégico de processos de *software*.

Para cada ES foram feitas 10 execuções da ferramenta utilizando todos os LLMs em estudo, cada execução possuindo um QGS aleatoriamente selecionado distinto. Ao final do processo de experimentação foram realizadas 100 execuções. Como o abordado na Seção 4.3, a SeSGx-LLM conta com quatro diferentes hiperparâmetros para realizar a construção das *strings* de busca: frequência dos documentos (min_fd), número de tópicos ($n\_topicos$), número de palavras por tópico ($n\_palavras\_por\_topico$) e quantidade de palavras enriquecidas por palavra do tópico ($n\_palavras\_enriquecidas\_por\_palavra$). Para este trabalho foi definido a utilização do seguinte conjunto dos hiperparâmetros:

- min_fd = [0.1, 0.2, 0.3, 0.4]

- $n\_topicos$ = [1, 2, 3, 4, 5]
- $n\_palavras\_por\_topico$ = [5, 6, 7, 8, 9, 10]
- $n\_enriquecimento$ = [0, 1, 2, 3, 4, 5]
- $umap\_vizinhos$ = [3, 5, 7]
- $kmeans\_clusters$ = [1, 2, 3, 4, 5]

Cada combinação de hiperparâmetros resulta em uma *string* de busca. Portanto, a partir desse conjunto de hiperparâmetros são gerados ao máximo: 720 *strings* utilizando o LDA como estratégia para a extração de tópicos. E 540 *strings* quando as LLMs extraem os tópicos. A quantidade máxima de *strings* geradas é igual a quantidade de combinações possíveis entre os parâmetros de cada estratégia (Figura 5.1). Entretanto, é possível que conjuntos de parâmetros distintos resultem na mesma *string* de busca, em vista disso, para o cálculo das métricas, são utilizadas apenas as *strings* únicas. Ao final do processo de experimentação foram geradas 409243 *strings* de busca únicas.

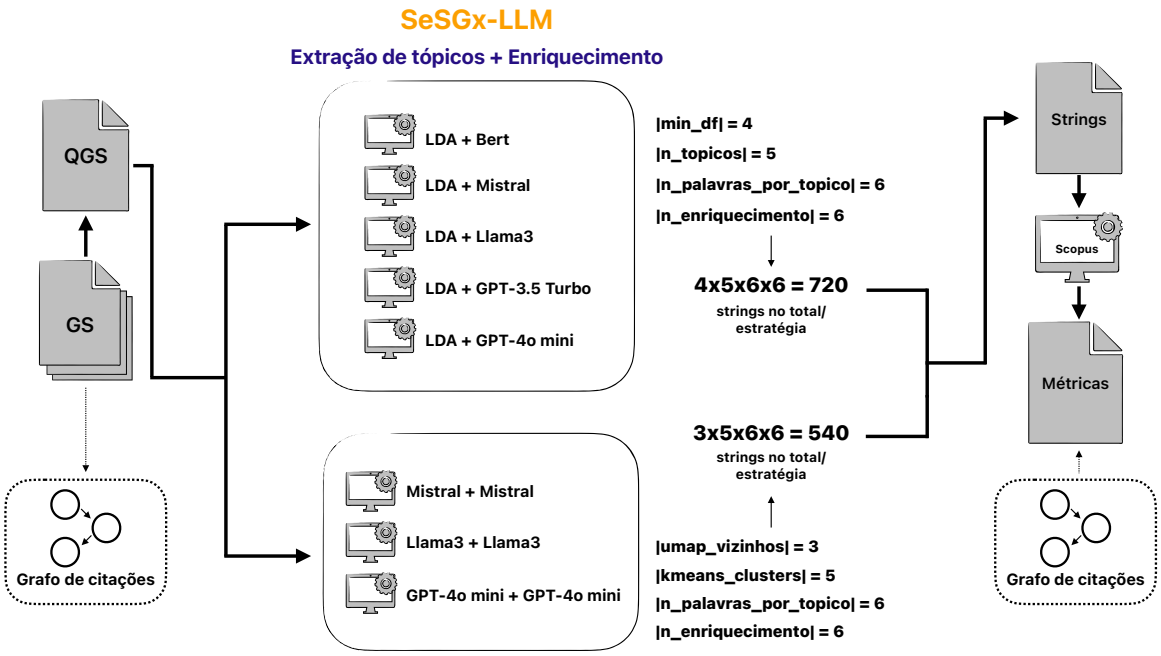

Figura 5.1: Ilustração do processo experimental realizado com a SeSGx-LLM.

Para as execuções realizadas com o GPT-3.5 Turbo, temos que o custo total da API da OpenAI foi de $1.62 dólares americanos. Posterior as execuções realizadas com o GPT-3.5 Turbo, um novo modelo foi anunciado pela OpenAI, o GPT-4o Mini, apresentado como o modelo de melhor custo benefício da empresa. O GPT-4o Mini foi utilizado com os mesmo parâmetros que foram

utilizado no GPT-3.5 Turbo. Entretanto, ao contrário do GPT-3.5 Turbo, o GPT-4o Mini ainda foi aplicado na tarefa de extração de tópicos.

O uso desses modelos para extração de tópicos possui um custo mais elevado por requisição, pois o *input* esse processo será composto por um número maior de *tokens* em relação ao *input* utilizado no processo de enriquecimento. Isso ocorre devido a necessidade de inserir ao *prompt* os metadados dos textos (títulos e resumos) que deseja-se extrair os termos latentes. Por essa razão, foi optado apenas pela utilização do GPT-4o Mini como alternativa paga, para a experimentação na etapa de extração de tópicos. Visto que, o GPT-4o Mini possui o melhor desempenho, é mais atualizado e pode ser 60% mais barato em relação ao GPT-3.5 Turbo. Ao total, para realizar todas as requisições necessárias para a experimentação, o custo do modelo GPT-4o Mini foi de $0.70 dólares americanos. Mesmo se tratando de um serviço pago, o uso do modelos proprietários da OpenAI mostrou-se uma alternativa de baixo custo.

O Mistral 7B foi escolhido primordialmente para ser a alternativa sem custos para a SeSGx-LLM. Por essa razão, foi optada pela não utilização da API proprietária do Mistral, sendo assim o modelo foi utilizado localmente. Mesmo se tratando de um LLM de tamanho relativamente compacto, a utilização do Mistral 7B e outros LLMs localmente pode ser complexa pela grande necessidade de *hardware* que esse modelos exigem. Para o modelo Llama3 8B é observado a mesma problemática.

Como forma de contornar esse problema, foi optado pela utilização da ferramenta Ollama [1]. Ollama é um *software* livre que possibilita o uso de LLMs *open-source* localmente, otimizando o uso dos modelos. Para rodar os modelos Mistral 7B e Llama3 8B, o Ollama utiliza uma quantização de quatro bits. Entretanto, o custo estimado para utilizar a API do Mistral 7B seria de menos de $1 dólar americano.

Cada execução da ferramenta utiliza um QGS aleatório como insumo. Os títulos e resumos dos estudos pertencentes ao QGS são diretamente utilizados na etapa de extração de tópicos feita tanto pelo LDA quanto pelos LLMs. Os mesmos tópicos servem então de entrada para o enriquecimento que será realizados pelos modelos GPT-4o Mini, GPT-3.5 Turbo, Mistral 7B e Llama3 8B. Dessa forma é possível certificar a validade do experimento.

Para garantir resultados reprodutíveis em meio as múltiplas execuções do LDA, o algoritmo não foi inicializado aleatoriamente. Na SeSGx-LLM, é utilizada a versão do LDA implementada pela biblioteca scikit-learn [2] e pode-se garantir a inicialização não aleatória ao definir o hiperparâmetro $random\_state$ para zero.

Como foi abordado anteriormente, para possibilitar o uso dos LLMs, é ne-

[1] https://ollama.com
[2] https://scikit-learn.org/

cessária a definição de um *prompt* que descreve o que deseja-se gerar. Cada etapa da SeSGx-LLM que utiliza LLMs possui um *prompt próprio*. São eles:

> Extração de tópicos:
> "You are a helpful topic extractor. From now on consider that a topic is a set of descriptors words that describe a document or a set of documents. You have to return only a JSON object and nothing more, follow this example: "keywords": [set, of , words]. The key value for the JSON object is "keywords". I will provide you with some documents as context so you can extract a set of keywords. Given the following documents: {context}. Generate a topic with 15 keywords. Please do not generate any more or any less than I've asked and remember to structure your response as a JSON object and return nothing else than a JSON. Please return only a JSON with only on pair of key-value that being "keywords": [set, of, words]. Remember to be generic and try to return simple words, avoid compound words."

> Enriquecimento:
> "You are a helpful synonym generator. Answer with a JSON object and nothing more. Follow this example 'synonyms': ['house', 'home']. Given the following context: {context}. Generate this amount of synonyms: {number_similar_words} for this topic:
> {word_to_be_enriched}."

Relativo ao *prompt* de extração de tópicos, tem-se o campo reservado "*context*", que deve ser substituído. Esse campo corresponde aos metadados (título e resumo) dos textos que devem ter seus tópicos extraídos. Entretanto, é possível que os artigos de interesse de um Estudo Secundário, mesmo que relacionados a um mesmo domínio, tenham sub-domínios entre si. Partindo desse pressuposto, a SeSGx-LLM foi projetada para, a partir do algoritmo *K-means*, agrupar os artigos do QGS baseado nas *embeddings* de seus metadados. Dessa forma é possível encontrar os tópicos latentes dentro de cada sub-grupo temático existente, expandindo ao máximo o vocabulário que será empregado as *strings* de busca. Entretanto, para cobrir o caso em que é valido extrair os tópicos latentes dado todos os textos pertencentes ao QGS, foi utilizado no processo de experimentação o parâmetro $kmeans\_clusters = 1$, ou seja, tem-se apenas um *cluster* que engloba todos os textos do QGS.

No *prompt* de Enriquecimento há três valores que serão substituídos. O primeiro valor é o "*context*", esse campo representa uma frase existente nos metadados, em que um dos termos de um determinado tópico extraído é utilizado. Dessa forma o LLM tem um contexto em que pode basear a geração de

um termo sinônimo. Originalmente, esse mesmo processo é feito pelo BERT, entretanto a pedição de um sinônimo se da pela técnica de *Masked Language Model*, onde mascara-se o termo em questão e busca-se pela *embedding* que melhor se encaixa na frase dado seu contexto bidirecional. O segundo valor a ser substituído é o "*{number_similar_words}*" que representa a quantidade de palavras similares que o modelo deve gerar. E por fim, o campo "*{word_to_be_enriched}*" refere-se ao termo que deve ser enriquecido.

Com todas as *strings* geradas por cada estratégia, pode-se aplicá-las a Scopus, através da API disponibilizada pela plataforma de busca. A decisão de se aplicar as *strings* de busca na Scopus foi baseada na disponibilidade de uma API que permitisse a automatização desse processo e na sua consistência em retornar altos valores de precisão, como levantado por Mourão et al. (2020). É importante enfatizar que, foi respeitada a mesma restrição de ano de publicação que o autor original utilizou. Ou seja, se para determinado ES, o autor buscou por artigos dentro do anos 2000 a 2014, essa característica foi replicada nas *strings* geradas pela SeSGx-LLM.

Para avaliar o desempenho das *strings* de busca geradas pela ferramenta foram utilizadas as métricas de completude. Os estudos relevantes que foram retornados na busca automatizada a partir da *string*, foram utilizados como *start set*. Com base nesse *start set*, são calculadas as métricas de precisão, revocação e F1-score. Note que o cálculo dessas métricas foi possibilitado pelo uso de Estudos Secundários já publicados, ou seja, existe o conhecimento prévio de um GS estabelecido pelo autores. É realizado também uma simulação do processo de *Snowballing* utilizando o *start set* alcançado e feito o cálculo da revocação do *Snowballing*. Portanto, temos as seguintes métricas sendo coletadas:

- Precisão$_{sts}$: precisão baseada no *start set* encontrado via busca automatizada, resultado direto do emprego das *strings* geradas.

- Revocação$_{sts}$: revocação baseada no *start set* encontrado via busca automatizada, resultado direto do emprego das *strings* geradas.

- F1-score$_{sts}$: F1-score baseado no *start set* encontrado via busca automatizada, resultado direto do emprego das *strings* geradas.

- Revocação$_{final}$: revocação baseada no *start set* encontrado via busca automatizada em conjunto dos estudos relevantes encontrados via *Snowballing*, tanto *Backward* quanto *Forward*.

5.2 Resultados

Nessa seção serão discutidos os resultados alcançados pela SeSGx-LLM utilizando os LLMs GPT-4o Mini, GPT-3.5 Turbo, Mistral 7B, Llama3 8B na etapa de enriquecimento, e os modelos GPT-4o Mini, Mistral 7B, Llama3 8B para extração de tópicos. Para critério de comparação do uso de LLMs, foi aplicado também algoritmo LDA para a extração de tópicos e o PLM BERT para enriquecimento. Dessa forma temos as combinações de estratégias apresentadas na Tabela 5.2.

Tabela 5.2: Combinações de estratégias experimentadas.

Extração de Tópicos	**Enriquecimento**	**Nome da combinação**
LDA	BERT	LDA-BERT
LDA	Mistral 7B	LDA-Mistral
LDA	Llama3 8B	LDA-Llama3
LDA	GPT-3.5 Turbo	LDA-GPT3.5T
LDA	GPT-4o Mini	LDA-GPT4oM
Mistral 7B	Mistral 7B	Mistral
Llama3 8B	Llama3 8B	Llama3
GPT-4o Mini	GPT-4o Mini	GPT4oM

Primeiramente, serão analisados as médias e desvios dos resultados gerais atingidos considerando todas as *strings* geradas por todas as combinações de estratégias. E por fim, serão discutidos os resultados das melhores cinco *strings* de busca geradas, considerando a Revocação$_{sts}$. A análise das cinco melhores *strings* será dividida entre a análise das estratégias de enriquecimento, ou seja, considerando *strings* com $n\_enriquecimento > 0$, e análise das estratégias de extração de tópicos, *strings* com $n\_enriquecimento = 0$.

Como abordado anteriormente, o LDA foi configurado para não possuir inicialização aleatória, portanto todas as *strings* geradas, a partir de tópicos extraídos com LDA e que não possuam palavras enriquecidas, serão iguais. Logo, todas as *strings* das combinações LDA-BERT, LDA-Mistral, LDA-Llama3, LDA-GPT3.5T, LDA-GPT4oM, com $n\_enriquecimento = 0$ serão iguais. Dessa forma, iremos comparar os resultados dessas *strings* de busca com os resultados das *strings* sem enriquecimento que foram geradas utilizando tópicos extraídos pelo Mistral 7B, Llama3 8B e GPT-4o Mini.

5.2.1 Resultados Gerais

Para analisar as estratégias utilizadas no experimento realizado nesta pesquisa, será discutida as médias e desvios padrões dos resultados alcançados. Para isso, os resultados de todas as *strings* de busca únicas que foram geradas por cada estratégia foram contabilizados e dispostos nas Tabelas 5.3 a

5.6.

Pelas médias gerais é possível observar que o uso do LDA como extrator de tópicos e LLMs como enriquecedor de termos foram predominantes em relação as métricas Revocação$_{sts}$, F1-score$_{sts}$ e Revocação$_{final}$. Sendo o Mistral 7B o modelo que obteve a melhor médias dessas três métricas, seguido pelo Llama3 8B.

Em relação a Precisão$_{sts}$, as estratégia que utilizam apenas LLMs, tanto para extrair tópicos quanto para gerador de sinônimos, obtiveram a melhor média em sete dos dez ESs do grupo experimental. Os estudos de Ferrari e Vasconcellos obtiveram as melhores médias de Precisão$_{sts}$ com a combinação entre LDA e Mistral 7B. E por fim, a melhor média de Precisão$_{sts}$ para Hosseini foi atingida com o LDA e Llama3 8B. A médias dos resultados da combinação entre LDA e BERT não se destacou em nenhum dos ESs do grupo experimental.

A partir dos resultados de média geral foi possível notar que, os LLMs tendem a identificar termos específicos dos textos que lhe foram apresentados, gerando termos precisos referente ao domínio em questão. Contudo, isso implica na diminuição da Revocação$_{sts}$, ou seja, os termos extraídos não têm abrangência suficiente para atingir os textos que pertencem ao domínio de interesse, mas não empregam aquele vocabulário em específico.

Como a diferença da Revocação$_{sts}$ foi superior a diferença da Precisão$_{sts}$ entre as estratégias com "LDA-LLM", "LDA-PLM" em relação a combinação "LLM-LLM", o F1-score$_{sts}$ foi superior para as abordagens com o LDA como extrator de tópicos. Em geral, os LLMs que se destacaram foram o Mistral 7B e o Llama3 8B, sendo os modelos com maior consistência nas métricas. Foram poucas os destaques dos modelos da OpenAI. O GPT-4o Mini obteve a melhor média de Precisão$_{sts}$ apenas quando utilizado tanto para extração quanto para enriquecimento dos tópicos. O GPT-3.5 Turbo, quando utilizado em conjunto ao LDA, obteve o melhor F1-score$_{sts}$, ambos os casos referentes ao ES de Azeem.

Em relação a métrica de Revocação$_{final}$, em todos os grupos experimentais as melhores estratégias foram capazes de atingir em média mais de 50% dos estudos. Dois das três maiores ESs, em termos de tamanho do GS, Bohmer (152) e Ferrari (146), inclusive atingiram a marca superior a 80% dos estudos relevantes atingidos após a aplicação do *Snowballing*. Ferrari foi o estudo que obteve a maior média de Revocação$_{final}$ com a estratégia LDA-Mistral, com média de 99% do GS atingido.

A combinação LDA-Mistral foi responsável também, pela maior média de Revocação$_{sts}$ dentre todos os ESs do grupo experimental, alcançando a marca de 38% de estudos retornados diretamente pelo uso da *string*, nos ESs de

Hosseini e Ferrari. Vale ressaltar que, quanto maior o valor da Revocação$_{sts}$ mais abrangente e completo o *start set* será, consequentemente aumentando as chances de atingir novos estudos relevantes a partir do *Snowballing*.

No âmbito de Estudos Secundários, a escolha entre revocação e precisão é significativa para o resultado da pesquisa. Uma *string* com alta revocação significa maior completude do ES, e como contraponto, é observado que esse cenário é usualmente acompanhado por maior ruído, ou seja, baixa precisão. Como consequência há maior esforço durante a tarefa de identificação de estudos relevantes. Entretanto, uma alta precisão muitas vezes está atrelada a baixa revocação, comprometendo a completude do estudo. Dessa forma, vale enfatizar a importância de uma alta revocação. Mesmo que a *string* de busca não seja rigorosamente precisa, a alta revocação é indicativo da pertinência dos termos usados e sua disposição dentro da *string* gerada, e o refino dessa mesma *string* é capaz de mitigar os ruídos encontrados.

5.2.2 Melhores Strings com Enriquecimento

Neste trabalho, levanta-se a hipótese de que LLMs podem contribuir positivamente nas etapas de extração e enriquecimento de tópicos relevantes que resultarão na geração de *strings* de busca. Entretanto, se faz necessário avaliar o impacto desses modelos de forma individual em cada uma dessas etapas, a extração dos tópicos e o enriquecimento. Por esse motivo, a primeira discussão a ser realizada será baseada nas *strings* de busca que utilizaram termos enriquecidos em sua composição, ou seja, $n\_enriquecimento > 0$.

Para esta análise foram agrupadas, por estratégia, as cinco melhores *strings* de busca geradas pela SeSGx-LLM com base primeiro na Revocação$_{sts}$ e posteriormente no F1-score$_{sts}$. A Revocação$_{sts}$ é uma das métricas mais importantes a serem consideradas na análise de completude de uma *string* visto que, por definição, o ES pretende reunir todo e qualquer EP disponível publicamente. Ou seja, é importante que a *string* de busca seja capaz de retornar o máximo de estudos relevantes possível. Além disso, como abordado na Seção 2.2.3, é indispensável que o *start set* seja diverso e representativo, assim um alto valor de Revocação$_{sts}$ pode implicar diretamente na taxa de sucesso do *Snowballing*. Por esse motivos, essa métrica foi escolhida para ordenar os resultados.

Porém, é importante que a busca seja precisa, evitando retornar ruído que aumentará o esforço despendido pelos pesquisadores. Dessa forma, considera-se também o F1-score$_{sts}$ como forma de determinar as cinco melhores *strings* de busca geradas. Esses resultados estão dispostos na Tabela 5.7.

A partir da Tabela 5.7 é possível observar que, as cinco melhores *strings* de busca de todos os grupos experimentais foram geradas a partir de tópicos extraídos pelo LDA. Isto é, não houve presença de nenhuma *string* gerada por

Tabela 5.3: Média e desvio padrão das métricas obtidas com cada estratégia (Alli, Azeem e Bertolino).

Métrica	Estratégia	Alli	Azeem	Bertolino
Precisão_{sts}	LDA-BERT	0.02 (± 0.05)	0.12 (± 0.2)	0.06 (± 0.15)
	LDA-Mistral	0.06 (± 0.07)	0.17 (± 0.24)	0.08 (± 0.15)
	LDA-Llama3	0.05 (± 0.06)	0.18 (± 0.24)	0.08 (± 0.15)
	LDA-Gpt3.5T	0.05 (± 0.07)	0.21 (± 0.25)	0.07 (± 0.14)
	LDA-Gpt4oM	0.05 (± 0.07)	0.19 (± 0.24)	0.07 (± 0.14)
	Mistral	**0.13** (± 0.25)	0.25 (± 0.36)	**0.18** (± 0.31)
	Llama3	**0.13** (± 0.25)	0.26 (± 0.36)	0.17 (± 0.3)
	Gpt4oM	0.11 (± 0.23)	**0.27** (± 0.37)	0.17 (± 0.3)
Revocação_{sts}	LDA-BERT	0.25 (± 0.2)	0.25 (± 0.2)	0.08 (± 0.09)
	LDA-Mistral	0.28 (± 0.2)	0.26 (± 0.16)	**0.15** (± 0.13)
	LDA-Llama3	**0.29** (± 0.21)	**0.27** (± 0.17)	0.14 (± 0.12)
	LDA-Gpt3.5T	0.28 (± 0.2)	**0.27** (± 0.17)	0.14 (± 0.11)
	LDA-Gpt4oM	0.28 (± 0.2)	0.26 (± 0.17)	0.14 (± 0.11)
	Mistral	0.07 (± 0.09)	0.18 (± 0.16)	0.03 (± 0.06)
	Llama3	0.07 (± 0.1)	0.18 (± 0.16)	0.04 (± 0.06)
	Gpt4oM	0.07 (± 0.09)	0.19 (± 0.16)	0.03 (± 0.05)
F1-score_{sts}	LDA-BERT	0.03 (± 0.05)	0.1 (± 0.13)	0.03 (± 0.04)
	LDA-Mistral	**0.07** (± 0.07)	0.13 (± 0.15)	**0.05** (± 0.04)
	LDA-Llama3	0.06 (± 0.06)	0.14 (± 0.15)	**0.05** (± 0.04)
	LDA-Gpt3.5T	0.06 (± 0.06)	**0.16** (± 0.15)	0.04 (± 0.04)
	LDA-Gpt4oM	0.06 (± 0.06)	0.15 (± 0.15)	0.04 (± 0.04)
	Mistral	0.04 (± 0.05)	0.1 (± 0.12)	0.02 (± 0.03)
	Llama3	0.03 (± 0.04)	0.11 (± 0.12)	0.02 (± 0.03)
	Gpt4oM	0.03 (± 0.04)	0.11 (± 0.12)	0.02 (± 0.02)
Revocação_{final}	LDA-BERT	0.53 (± 0.28)	0.68 (± 0.36)	0.61 (± 0.31)
	LDA-Mistral	0.58 (± 0.22)	**0.76** (± 0.29)	**0.75** (± 0.16)
	LDA-Llama3	**0.59** (± 0.23)	0.75 (± 0.3)	**0.75** (± 0.15)
	LDA-Gpt3.5T	**0.59** (± 0.23)	0.74 (± 0.3)	0.73 (± 0.18)
	LDA-Gpt4oM	**0.59** (± 0.23)	0.75 (± 0.3)	0.74 (± 0.17)
	Mistral	0.32 (± 0.26)	0.64 (± 0.37)	0.48 (± 0.36)
	Llama3	0.33 (± 0.26)	0.64 (± 0.36)	0.5 (± 0.36)
	Gpt4oM	0.33 (± 0.26)	0.64 (± 0.36)	0.5 (± 0.36)

estratégias puramente baseadas nos LLMs para extração e enriquecimento de tópicos.

Em quatro de dez ESs, as cinco melhores *strings* de busca foram geradas pela combinação LDA-Mistral, i.e., Bertolino, Dinter, Dissanayake e Vasconcellos. O LDA-BERT também foi unânime em um dos ESs, o Alli. Azeem, Bohmer, Ferrari, Mohan e Hosseini obtiveram resultados diversos, exceto por Azeem e Mohan, o LDA-Mistral esteve presente em pelo menos uma das cinco melhores *strings* geradas em todos os outros ESs do grupo experimental.

O LDA-Llama3 também se destacou no ranqueamento das cinco melhores *strings* de busca geradas. As duas melhores *strings* de Ferrari e as quatro

Tabela 5.4: Média e desvio padrão das métricas obtidas com cada estratégia (Bohmer, Dinter e Dissanayake).

Métrica	Estratégia	Bohmer	Dinter	Dissanayake
Precisão$_{sts}$	LDA-BERT	0.06 (± 0.12)	0.02 (± 0.06)	0.01 (± 0.04)
	LDA-Mistral	0.12 (± 0.17)	0.04 (± 0.08)	0.02 (± 0.05)
	LDA-Llama3	0.09 (± 0.14)	0.02 (± 0.07)	0.02 (± 0.04)
	LDA-Gpt3.5T	0.08 (± 0.13)	0.02 (± 0.07)	0.02 (± 0.04)
	LDA-Gpt4oM	0.08 (± 0.13)	0.02 (± 0.07)	0.02 (± 0.04)
	Mistral	0.12 (± 0.26)	**0.15** (± 0.26)	0.08 (± 0.21)
	Llama3	**0.13** (± 0.27)	0.13 (± 0.24)	**0.11** (± 0.25)
	Gpt4oM	0.11 (± 0.24)	0.12 (± 0.23)	0.09 (± 0.22)
Revocação$_{sts}$	LDA-BERT	0.22 (± 0.13)	0.11 (± 0.16)	0.05 (± 0.08)
	LDA-Mistral	**0.26** (± 0.11)	**0.22** (± 0.17)	**0.1** (± 0.11)
	LDA-Llama3	0.24 (± 0.12)	0.19 (± 0.16)	0.09 (± 0.09)
	LDA-Gpt3.5T	0.24 (± 0.12)	0.16 (± 0.16)	0.09 (± 0.09)
	LDA-Gpt4oM	0.24 (± 0.12)	0.16 (± 0.16)	0.08 (± 0.09)
	Mistral	0.03 (± 0.05)	0.11 (± 0.11)	0.05 (± 0.07)
	Llama3	0.03 (± 0.06)	0.11 (± 0.1)	0.05 (± 0.08)
	Gpt4oM	0.03 (± 0.05)	0.11 (± 0.1)	0.04 (± 0.07)
F1-score$_{sts}$	LDA-BERT	0.06 (± 0.08)	0.02 (± 0.05)	0.01 (± 0.03)
	LDA-Mistral	**0.11** (± 0.1)	0.04 (± 0.07)	**0.02** (± 0.03)
	LDA-Llama3	0.09 (± 0.09)	0.03 (± 0.06)	**0.02** (± 0.03)
	LDA-Gpt3.5T	0.08 (± 0.08)	0.03 (± 0.06)	**0.02** (± 0.03)
	LDA-Gpt4oM	0.08 (± 0.09)	0.03 (± 0.06)	**0.02** (± 0.03)
	Mistral	0.02 (± 0.03)	**0.05** (± 0.06)	**0.02** (± 0.02)
	Llama3	0.02 (± 0.03)	**0.05** (± 0.05)	**0.02** (± 0.02)
	Gpt4oM	0.02 (± 0.03)	**0.05** (± 0.05)	**0.02** (± 0.02)
Revocação$_{final}$	LDA-BERT	0.77 (± 0.25)	0.42 (± 0.43)	0.37 (± 0.35)
	LDA-Mistral	**0.84** (± 0.05)	**0.73** (± 0.29)	**0.55** (± 0.3)
	LDA-Llama3	**0.84** (± 0.06)	0.68 (± 0.34)	0.54 (± 0.31)
	LDA-Gpt3.5T	0.83 (± 0.11)	0.62 (± 0.38)	0.51 (± 0.32)
	LDA-Gpt4oM	0.83 (± 0.13)	0.61 (± 0.38)	0.5 (± 0.32)
	Mistral	0.45 (± 0.41)	0.71 (± 0.31)	0.41 (± 0.34)
	Llama3	0.45 (± 0.41)	0.69 (± 0.32)	0.41 (± 0.35)
	Gpt4oM	0.45 (± 0.41)	0.69 (± 0.33)	0.4 (± 0.34)

melhores *strings* de Mohan foram produzidas pelo LDA-Llama3. Em complemento, a estratégia LDA-Llama3 também esteve presente nos ESs Bohmer, sendo responsável pelo segundo e terceiro melhor resultado, e em Hosseini, como sendo o gerador do quarto melhor resultado.

Já os modelos GPT foram responsáveis pela melhores*strings* de Azeem e Bohmer, que foram gerados pelo LDA-GPT3.5T e LDA-GPT4oM respectivamente. O LDA-GPT3.5T ainda foi o gerador da quinta melhor *string* de busca em Mohan e Hosseini. Enquanto o LDA-GPT4oM não obteve mais aparições nos *rankings* de melhores cinco *strings* geradas.

Perceptivelmente, o modelo Mistral 7B foi o modelo mais consistente em gerar termos sinônimos aos tópicos extraídos. Também se faz evidente a pre-

Tabela 5.5: Média e desvio padrão das métricas obtidas com cada estratégia (Ferrari, Hosseini e Mohan).

Métrica	Estratégia	Ferrari	Hosseini	Mohan
Precisão$_{sts}$	LDA-BERT	0.14 (± 0.13)	0.11 (± 0.17)	0.01 (± 0.03)
	LDA-Mistral	0.2 (± 0.13)	0.15 (± 0.17)	0.03 (± 0.03)
	LDA-Llama3	**0.18** (± 0.13)	**0.16** (± 0.18)	0.03 (± 0.03)
	LDA-Gpt3.5T	**0.16** (± 0.14)	0.16 (± 0.18)	0.02 (± 0.03)
	LDA-Gpt4oM	0.16 (± 0.14)	0.15 (± 0.17)	0.02 (± 0.03)
	Mistral	**0.18** (± 0.34)	0.12 (± 0.25)	**0.18** (± 0.3)
	Llama3	0.17 (± 0.35)	0.14 (± 0.26)	**0.18** (± 0.3)
	Gpt4oM	0.13 (± 0.3)	0.12 (± 0.25)	0.16 (± 0.28)
Revocação$_{sts}$	LDA-BERT	0.27 (± 0.21)	0.34 (± 0.18)	0.16 (± 0.13)
	LDA-Mistral	**0.38** (± 0.23)	**0.38** (± 0.14)	0.22 (± 0.13)
	LDA-Llama3	0.35 (± 0.22)	0.37 (± 0.14)	**0.23** (± 0.13)
	LDA-Gpt3.5T	0.35 (± 0.23)	0.36 (± 0.14)	0.21 (± 0.13)
	LDA-Gpt4oM	0.31 (± 0.22)	0.36 (± 0.13)	0.2 (± 0.13)
	Mistral	0.05 (± 0.09)	0.13 (± 0.14)	0.06 (± 0.07)
	Llama3	0.06 (± 0.09)	0.12 (± 0.14)	0.06 (± 0.08)
	Gpt4oM	0.04 (± 0.06)	0.12 (± 0.13)	0.06 (± 0.08)
F1-score$_{sts}$	LDA-BERT	0.13 (± 0.1)	0.1 (± 0.12)	0.02 (± 0.03)
	LDA-Mistral	**0.19** (± 0.09)	**0.15** (± 0.12)	**0.04** (± 0.03)
	LDA-Llama3	0.17 (± 0.1)	**0.15** (± 0.12)	**0.04** (± 0.03)
	LDA-Gpt3.5T	0.16 (± 0.1)	**0.15** (± 0.12)	0.03 (± 0.03)
	LDA-Gpt4oM	0.14 (± 0.1)	0.14 (± 0.12)	0.03 (± 0.03)
	Mistral	0.03 (± 0.04)	0.04 (± 0.06)	0.03 (± 0.03)
	Llama3	0.03 (± 0.04)	0.05 (± 0.05)	0.03 (± 0.03)
	Gpt4oM	0.02 (± 0.03)	0.04 (± 0.05)	0.03 (± 0.03)
Revocação$_{final}$	LDA-BERT	0.95 (± 0.2)	0.86 (± 0.26)	0.58 (± 0.28)
	LDA-Mistral	**0.99** (± 0.02)	**0.94** (± 0.04)	0.69 (± 0.1)
	LDA-Llama3	**0.99** (± 0.05)	**0.94** (± 0.06)	**0.7** (± 0.09)
	LDA-Gpt3.5T	0.98 (± 0.08)	**0.94** (± 0.05)	0.69 (± 0.12)
	LDA-Gpt4oM	0.98 (± 0.11)	**0.94** (± 0.05)	0.68 (± 0.13)
	Mistral	0.77 (± 0.41)	0.73 (± 0.39)	0.54 (± 0.26)
	Llama3	0.78 (± 0.4)	0.72 (± 0.4)	0.54 (± 0.25)
	Gpt4oM	0.76 (± 0.41)	0.71 (± 0.4)	0.55 (± 0.25)

dominância do LDA enquanto extrator de tópicos, já que nenhum dos melhores resultados foram obtidos utilizando LLMs para identificação de termos descritores.

> *String*-01 de busca gerada por LDA-Mistral, ES de Bertolino et al. (2019):
> TITLE-ABS-KEY(((“testing” OR “testing and verification” OR “quality assurance” OR “automated quality control”) AND (“service” OR “application” OR “platform” OR “system”) AND (“test” OR “examination” OR “evaluuation” OR “inspection”) AND (“cloud” OR “cloud computing” OR “network of servers” OR “distributed computing”) AND

Tabela 5.6: Média e desvio padrão das métricas obtidas com cada estratégia (Vasconcellos).

Métrica	Estratégia	Vasconcellos
Precisão$_{sts}$	LDA-BERT	0.07 (± 0.16)
	LDA-Mistral	**0.11** (± 0.18)
	LDA-Llama3	0.1 (± 0.18)
	LDA-Gpt3.5T	0.1 (± 0.17)
	LDA-Gpt4oM	**0.11** (± 0.18)
	Mistral	0.1 (± 0.21)
	Llama3	0.09 (± 0.2)
	Gpt4oM	0.09 (± 0.2)
Revocação$_{sts}$	LDA-BERT	0.27 (± 0.21)
	LDA-Mistral	**0.3** (± 0.2)
	LDA-Llama3	0.27 (± 0.19)
	LDA-Gpt3.5T	0.28 (± 0.2)
	LDA-Gpt4oM	0.29 (± 0.2)
	Mistral	0.15 (± 0.13)
	Llama3	0.15 (± 0.13)
	Gpt4oM	0.13 (± 0.12)
F1-score$_{sts}$	LDA-BERT	0.05 (± 0.07)
	LDA-Mistral	**0.08** (± 0.08)
	LDA-Llama3	0.07 (± 0.08)
	LDA-Gpt3.5T	0.07 (± 0.08)
	LDA-Gpt4oM	**0.08** (± 0.08)
	Mistral	0.05 (± 0.06)
	Llama3	0.05 (± 0.06)
	Gpt4oM	0.04 (± 0.06)
Revocação$_{final}$	LDA-BERT	0.68 (± 0.35)
	LDA-Mistral	**0.83** (± 0.16)
	LDA-Llama3	0.79 (± 0.22)
	LDA-Gpt3.5T	0.8 (± 0.22)
	LDA-Gpt4oM	0.82 (± 0.19)
	Mistral	0.66 (± 0.28)
	Llama3	0.66 (± 0.28)
	Gpt4oM	0.65 (± 0.29)

(“based” OR “founded upon” OR “established on” OR “supported by”)) OR ((“cloud” OR “cloud computing” OR “network of servers” OR “distributed computing”) AND (“testing” OR “testing and verification” OR “quality assurance” OR “automated quality control”) AND (“computing” OR “processing” OR “information technology” OR “data centers”) AND (“cloud computing” OR “cloud-based computing” OR “distributed computing” OR “networked computing”) AND (“software” OR “application” OR “program” OR “script”)) OR ((“test” OR “examination” OR “evaluuation” OR “inspection”) AND (“cloud” OR “cloud computing” OR “network of servers” OR “distributed com-

Tabela 5.7: Cinco melhores resultados de cada ES, **com** enriquecimento.

ES	Estratégia	Precisão$_{sts}$	Revocação$_{sts}$	F1-score$_{sts}$	Revocação$_{Final}$
Dissanayake	LDA-Mistral (1)	0.01	**0.611**	0.02	**0.931**
	LDA-Mistral (2)	0.014	0.583	0.027	0.903
	LDA-Mistral (3)	0.008	0.583	0.017	0.917
	LDA-Mistral (4)	**0.015**	0.569	**0.029**	0.903
	LDA-Mistral (5)	0.009	0.569	0.018	0.917
Ferrari	LDA-Llama3 (1)	0.063	**0.897**	0.118	**1.0**
	LDA-Llama3 (2)	0.03	**0.897**	0.058	**1.0**
	LDA-Mistral (3)	0.027	**0.897**	0.052	**1.0**
	LDA-Mistral (4)	**0.067**	0.89	**0.125**	**1.0**
	LDA-Mistral (5)	0.047	0.89	0.089	**1.0**
Hosseini	LDA-BERT (1)	0.007	**0.717**	0.013	**0.957**
	LDA-Mistral (2)	**0.034**	0.696	**0.065**	**0.957**
	LDA-Llama3 (3)	0.033	0.696	0.063	**0.957**
	LDA-Gpt3.5T (4)	0.019	0.696	0.037	**0.957**
	LDA-Mistral (5)	0.012	0.696	0.024	**0.957**
Mohan	LDA-Llama3 (1)	**0.014**	**0.645**	0.027	**0.803**
	LDA-Llama3 (2)	0.013	**0.645**	0.025	**0.803**
	LDA-Llama3 (3)	0.02	0.632	**0.038**	0.789
	LDA-Llama3 (4)	**0.014**	0.632	0.028	0.789
	LDA-Gpt3.5T (5)	0.01	0.632	0.019	**0.803**
Vasconcellos	LDA-Mistral (1)	**0.013**	**0.733**	**0.025**	**0.967**
	LDA-Mistral (2)	0.012	**0.733**	0.024	**0.967**
	LDA-Mistral (3)	0.012	**0.733**	0.023	**0.967**
	LDA-Mistral (4, 5)	0.01	**0.733**	0.02	**0.967**

puting") AND ("environment" OR "cloud infrastructure" OR "cloud ecosystem" OR "cloud platform") AND ("paper" OR "study" OR "research" OR "report") AND ("software" OR "application" OR "program" OR "script")) OR (("performance" OR "performance level" OR "efficiency" OR "productivity") AND ("cloud" OR "cloud computing" OR "network of servers" OR "distributed computing") AND ("computing" OR "processing" OR "information technology" OR "data centers") AND ("cloud computing" OR "cloud-based computing" OR "distributed computing" OR "networked computing") AND ("test" OR "examination" OR "evaluuation" OR "inspection")) OR (("applications" OR "software" OR "programs" OR "systems") AND ("testing" OR "testing and verification" OR "quality assurance" OR "automated quality control") AND ("software" OR "application" OR "program" OR "script") AND ("performance" OR "performance level" OR "efficiency" OR "productivity") AND ("cloud" OR "cloud computing" OR "network of servers" OR "distributed computing"))) AND PUBYEAR > 2011 AND PUBYEAR < 2018

String-02 de busca gerada por Llama3-Llama3, ES de Bertolino et al. (2019):
TITLE-ABS-KEY((("Cloud" OR "Cyberinfrastructure" OR "Virtual Infrastructure" OR "Remote Computing Environment" OR "Distribu-

ted System") AND ("Testing" OR "quality assurance" OR "software verification" OR "validation" OR "examining") AND ("Software" OR "Program" OR "Application" OR "Software program" OR "Computer system") AND ("Applications" OR "software" OR "programs" OR "executables") AND ("Services" OR "Applications" OR "Frameworks" OR "Platforms" OR "Infrastructures")) OR (("cloud" OR "sky" OR "atmosphere" OR "cyberinfrastructure" OR "networkspace") AND ("mobile" OR "handheld" OR "cellular" OR "wireless" OR "on-the-go") AND ("testing" OR "evaluation" OR "examination" OR "investigation") AND ("service" OR "infrastructure" OR "platform" OR "network" OR "resource") AND ("infrastructure" OR "platform" OR "system" OR "environment" OR "foundation"))) AND PUBYEAR > 2011 AND PUBYEAR < 2018

Tabela 5.8: Resultados das *strings* *String*-01 e *String*-02, referantes ao estudos de Bertolino et al. (2019), utilizadas como exemplos gerados pela SeSGx-LLM.

String	**Revocação**$_{sts}$	**Precisão**$_{sts}$	**F1-score**$_{sts}$
String-01 (LDA-Mistral)	**0.7142**	**0.032**	**0.0614**
String-02 (Llama3-Llama3)	0.653	0.0203	0.0395

Acima são apresentadas duas *strings* de busca que foram de fato geradas pela SeSGx-LLM, com base no ES de Bertolino et al. (2019), e seus resultados na Tabela 5.8. A partir dessas *strings*, é possível visualizar a especialização do vocabulário ocorrendo quando utilizado puramente LLMs, nesse caso em específico, o Llama3 8B.

Nesse exemplo ainda é possível observar na prática alguns pontos que resultaram no melhor desempenho do Mistral 7B. Bertolino et al. (2019), conduziu um ES referente a testes realizados em ambientes na nuvem (do inglês, *cloud*). O LDA realizou a extração do termo *"cloud"* como uma das palavras latentes do ES, e por meio de seu contexto, o Mistral 7B atribuiu como sinônimos os seguintes termos: *"cloud computing"*, *"network of servers"* e *"distributed computing"*.

Em contraponto, o Llama3 8B, de maneira análoga ao LDA, também extraiu "*cloud*" como palavra-chave. Porém, foi atribuído a este termo sinônimos relacionados ao seu valor semântico literal (*"sky"*, *"atmosphere"*), trazendo uma desconexão do tema do ES na *string* gerada. Isso resultou em um ganho percentual de 6.32% em revocação no resultado da *string* gerada pelo Mistral 7B, em conjunto com uma maior precisão, em comparação com os resultados alcançados pela *string* de busca gerada pelo modelo da Meta.

5.2.3 Melhores Strings sem Enriquecimento

Como abordado anteriormente, para efetuar do desempenho de LLMs enquanto extrator de tópicos, uma nova analise é proposta. Na Tabela 5.9 estão dispostos os cinco melhores resultados atingidos por *strings* que contêm $n\_enriquecimento = 0$. Dessa forma é possível visualizar o desempenho de *strings* de busca que são compostas apenas pelos termos descritores que foram identificados pelo LDA ou pelos LLMs, Mistral 7B, Llama3 8B, GPT-3.5 Turbo e GPT-4o Mini.

Ao comparar apenas as estratégias que foram aplicadas na etapa de extração de tópicos, o LDA se demonstrou notavelmente mais consistente que os LLMs. Em oito de dez ESs as cinco melhores *strings* de busca foram geradas a partir de termos identificados pelo LDA. Nos dois ESs remanescentes, Azeem e Dinter, o LDA foi responsável pelas duas melhores *strings* de Dinter, e apenas em Azeem o Mistral, GPT-4o Mini e Llama3 8B foram capazes de superar o LDA.

É possível ainda observar que, em relação aos melhores resultados das melhores *strings* **com** enriquecimento, as *strings* de busca que não obtiveram os tópicos enriquecidos, em sua grande maioria, atingiram menores valores de Revocação$_{sts}$. Isto é indicador do efeito prático que a utilização de termos sinônimos tornam a *string* mais abrangente.

A partir desses resultados, é possível concluir que os LLMs não demonstraram ser a melhor abordagem para identificação de termos descritores. Ao analisar os tópicos que foram extraídos por LLMs em comparação com os tópicos oriundos do LDA, é possível observar uma maior especialização do vocabulário quando os LLMs são aplicados, ou seja, são identificados termos muitos específicos limitando o vocabulário da busca. Uma forma de contornar essa problemática foi adicionar ao *prompt* aplicado, a preferência por termos simples e genéricos que pudessem descrever o texto dado como contexto. Entretanto, a abordagem estatística do LDA ainda sim se demonstrou superior.

5.3 Respostas das Questões de Pesquisa

De maneira a sumarizar as conclusões com base nos dados levantados, foram propostas seis questões de pesquisa. Com base no experimento projetado e executado, responde-se as perguntas propostas da seguinte maneira:

QP-01. Em quais cenários os LLMs, apresentam maior benefício na geração de *strings* de busca, em termos de precisão e revocação, ao compará-los a métodos tradicionais LDA e BERT?
Os LLMs demonstraram uma ótima alternativa para geração de termos similares. Em conjunto com a abordagem mais tradicional, o LDA como extrator

Tabela 5.9: Cinco melhores resultados de cada ES, **sem** enriquecimento.

ES	**Estratégia**	**Precisão**$_{sts}$	**Revocação**$_{sts}$	**F1-score**$_{sts}$	**Revocação**$_{Final}$
Alli	LDA (1)	0.012	**0.804**	0.025	0.929
	LDA (2)	0.015	0.786	0.029	0.929
	LDA (3)	0.054	0.768	**0.1**	**0.946**
	LDA (4, 5)	**0.063**	0.75	0.116	**0.946**
Azeem	Mistral (1)	0.018	**0.8**	0.035	**1.0**
	Gpt4oM (2)	0.018	**0.8**	0.035	**1.0**
	Llama3 (3)	0.018	**0.8**	0.035	**1.0**
	LDA (4)	**0.073**	0.733	**0.133**	**1.0**
	Mistral (5)	0.028	0.733	0.055	**1.0**
Bertolino	LDA (1)	0.037	**0.592**	0.07	0.918
	LDA (2)	0.051	0.565	0.094	**0.932**
	LDA (3)	**0.079**	0.558	**0.138**	0.905
	LDA (4)	0.018	0.558	0.035	0.905
	LDA (5)	0.055	0.551	0.1	0.891
Bohmer	LDA (1)	0.018	**0.586**	0.035	**0.901**
	LDA (2)	0.026	0.566	0.05	0.888
	LDA (3, 4)	**0.027**	0.559	**0.052**	0.888
	LDA (5)	0.021	0.559	0.04	0.882
Dinter	LDA (1)	0.008	**0.732**	0.016	**0.902**
	LDA (2)	**0.009**	0.683	**0.018**	**0.902**
	Mistral (3, 4, 5)	**0.009**	0.683	0.017	0.878
Dissanayake	LDA (1)	0.007	**0.514**	0.015	**0.861**
	LDA (2)	0.023	0.5	0.044	0.819
	LDA (3)	**0.037**	0.486	**0.069**	0.833
	LDA (4)	**0.037**	0.486	**0.069**	0.819
	LDA (5)	0.023	0.486	0.044	0.819
Ferrari	LDA (1)	0.045	**0.863**	0.085	**1.0**
	LDA (2)	0.056	0.856	0.105	0.993
	LDA (3)	0.044	0.849	0.084	**1.0**
	LDA (4)	0.086	0.842	0.156	0.993
	LDA (5)	**0.135**	0.822	**0.232**	**1.0**
Hosseini	LDA (1)	**0.049**	**0.652**	**0.09**	0.935
	LDA (2)	0.026	**0.652**	0.051	**0.957**
	LDA (3)	0.02	**0.652**	0.038	**0.957**
	LDA (4)	0.047	0.63	0.087	0.935
	LDA (5)	0.041	0.63	0.076	**0.957**
Mohan	LDA (1)	0.011	**0.645**	0.022	**0.816**
	LDA (2)	0.011	0.618	0.022	0.803
	LDA (3)	0.009	0.618	0.019	**0.816**
	LDA (4)	**0.019**	0.605	**0.037**	0.803
	LDA (5)	0.016	0.605	0.03	0.803
Vasconcellos	LDA (1)	0.007	**0.7**	0.015	0.933
	LDA (2)	0.006	**0.7**	0.012	**0.967**
	LDA (3)	**0.031**	0.667	**0.059**	0.933
	LDA (4)	0.012	0.667	0.024	0.933
	LDA (5)	0.012	0.667	0.023	0.933

de tópicos, a SeSGx-LLM obteve, em média, seus melhores resultados. Dentre os modelos utilizados, destaca-se o desempenho obtido por uma alternativa *open-source*, o Mistral 7B. Mesmo diante de modelos mais robustos e considerados, através de *benchmarks*, superiores, como o GPT-4o Mini, o Mistral 7B mostrou predominância dentre os melhores resultados. Para estudos de

temas amplos, recomenda-se o uso do Mistral 7B para maior revocação, enquanto para temas específicos, o LDA pode ser mais eficiente.

QP-02. Por meio da capacidade semântica existente em LLMs, esses modelos superam a abordagem estatística do LDA em termos de extração de tópicos latentes?
Vale ressaltar que nenhum dos modelos testados demonstrou ser substituto como alternativa direta ao LDA enquanto extrator de tópicos. O algoritmo de modelagem de tópicos foi superior na maioria dos ESs utilizados como grupo experimental.

QP-03. LLMs superam o PLM BERT como enriquecedor de termos para automatizar a geração de *strings* de busca?
O domínio semântico atrelado aos LLMs se comprovou válido enquanto enriquecedor dos tópicos extraídos pelo LDA. A partir de um breve contexto, os LLMs superaram a maioria dos resultados atingidos pelo PLM BERT.

QP-04: Dentre as possibilidades de uso de modelos de linguagem (e.g., PLMs e LLMS), e modelos estatísticos (e.g., LDA), quais as melhores combinações de abordagens que minimizam o esforço do pesquisador, medido por meio da precisão, e mantém a completude da pesquisa, medido por meio da revocação?
A combinação LDA-Mistral exprimiu a maior consistência tanto dos resultados de precisão do *start set* quanto de revocação do *start set* e revocação final.

QP-05: Dentre os LLMs experimentados, qual obteve melhor desempenho na geração de *strings* de busca e qual obteve melhor custo-benefício?
O Mistral se manteve predominante dentre os melhores resultados obtidos com LLMs. O Llama 8B também obteve bons resultados, porém com menos consistência, e em menor destaque os modelos GPT, sendo o pior o modelo GPT3.5 Turbo. Por se tratar de um modelo de código aberto, o Mistral definitivamente atingiu o melhor custo-benefício. Entretanto, mesmo ao utilizar a versão paga, necessária quando aplicado a fins comerciais, o custo do uso do modelo seria baixo devido ao tamanho reduzido tanto do *input* quanto *output* gerados pelo Mistral 7B, considerando apenas a atividade de enriquecimento, que foi o ponto de destaque dos LLMs.

QP-06: LLMs podem auxiliar na geração de *strings* de busca quando o corpus de entrada utilizado é limitado ou especializado?
Um dos pontos focais do processo geração de *strings* de busca no contexto de ESs é a quantidade reduzida de textos disponíveis para extração de termos. Por essa razão, a questão de pesquisa QP-06 visa justamente analisar o comportamento dos LLMs diante de uma quantidade limitada de contexto a disposição.

Ao lidar com extração de tópicos dos textos de entrada (QGS) foi observado

uma especialização do vocabulário. Isto é, os termos extraídos apresentavam um alto grau de especificidade gerando dificuldade posterior na geração de termos sinônimos. A tratativa desse problema foi o refino do *prompt* utilizado, aumentando a atenção do modelo para a extração de termos mais genéricos que pudessem descrever aquele corpus. O objetivo foi evitar ao máximo a extração de palavras compostas e nomes próprios específicos ao texto apresentado como contexto.

Em suma, a contribuição deste trabalho foi apresentar o uso experimental de LLMs como facilitador na etapa de construção de *strings* de busca. A SeSGx-LLM foi disponibilizada como uma ferramenta de código aberto passível de novas implementações e melhorias. Foi concluído que LLMs possuem claro desempenho em enriquecer termos de modo a aumentar a abrangência da busca, permitindo aos pesquisadores se apoiarem em Inteligência Artificial como forma de diminuir o esforço em uma das várias etapas necessárias para a conclusão de um Estudo Secundário.

CAPÍTULO

6

Conclusão

O desenvolvimento de um ES é um trabalho complexo e moroso, e como levantado ao longo desse texto, carece de ferramentas que apoiem os pesquisadores. A partir dessa problemática, esse estudo propõe contribuir, com o apoio de LLMs, no processo de geração automática de *strings* de busca. Além da SeSGx-LLM estar disponível de maneira *open-source*, nesse trabalho também foram coletados dados como forma de publicar evidências da eficácia de diferentes LLMs dentro do contexto de geração de *strings* de busca. A SeSGx-LLM viabilizou a coleta de evidências ao gerar *strings* de busca, utilizando os modelos Mistral 7B, Llama3 8B, GPT-3.5 Turbo e GPT-4o Mini, e testá-las empiricamente na plataforma de indexação Scopus.

A SeSGx-LLM visa diminuir o esforço gasto por autores, permitindo que os mesmos possam focar na extração e documentação do conhecimento. Destaca-se ainda que a SeSGx-LLM independe de domínio de busca, ou seja, não há nicho específico para a utilização da ferramenta, podendo ser aplicada a diversas áreas como o âmbito da saúde, conhecido por possuir altas demandas por ESs.

As contribuições desse trabalho são: refatoração e publicação da ferramenta SeSGx-LLM, desenvolvimento de um *framework* para utilizar LLMs no contexto de extração e enriquecimento de tópicos, coleta de dados a partir da aplicação das *strings* de busca geradas pela ferramenta, com diversas abordagens e modelos (LLMs, LDA e BERT).

A partir do processo experimental foi possível concluir que, LLMs não demonstraram ser superiores ao LDA enquanto extrator de tópicos. Porém, os grandes modelos de linguagem possuem desempenho superior ao BERT como enriquecedor de termos. A utilização do LDA combinado aos LLMs beneficiam

positivamente a geração e refinamento de *strings* de busca.

6.1 Limitações

Este trabalho possuí limitações que devem ser consideradas. Dentre elas, é fundamental ressaltar que a SeSGx-LLM está prevista nesta pesquisa como uma ferramenta com fim experimental para análise de desempenho dos algoritmos aplicados. A ferramenta ainda não tem o preparo completo necessário para atender o usuário final.

Esse trabalho visa obter evidências científicas de diferentes abordagens que possam apoiar um usuário final. Com isso em mente, foi determinada a aplicação de LLMs de tamanhos menores mas com desempenhos confiáveis. É reconhecida a existência de modelos maiores com desempenhos melhores em diversos *benchmarks*, que poderiam igualmente ser aplicados a SeSGx-LLM. Entretanto, o objetivo desta pesquisa foi focar em soluções viáveis para um usuário que talvez não possa fornecer o *hardware* para modelos tão grandes. Em complemento a isto, a escolha dos LLMs também foi pensada de maneira a não depender totalmente de API pagas de terceiros. LLMs são altamente dependentes de um hardware que possa processá-los, dessa forma a SeSGx-LLM é diretamente dependente dos recursos computacionais que o usuário tem disponível.

Conforme dito neste Capítulo, as *strings* de busca geradas pela SeSGx-LLM foram aplicadas apenas na plataforma Scopus. Isto implica que os resultados são baseados em uma única plataforma, sendo assim, as métricas extraídas dependem totalmente da disponibilidades dos artigos relevantes na Scopus. Entretanto, a escolha dessa plataforma foi baseada na literatura por obter consistência em retornar altos valores de precisão (Mourão et al. (2020)). Para garantir que os resultados não sofressem com as atualizações (remoção ou adição) dos textos que são disponibilizados na plataforma, os experimentos foram executados de maneira a minimizar ao máximo a janela de tempo entre si. Contudo, caso esse experimento venha a ser replicado, é possível que os resultados das *strings* de busca sejam diferentes.

O QGS é utilizado para possibilitar tanto a geração das *strings* de busca como avaliá-las. Isto implica em uma grande dependência da ferramenta em relação ao QGS, de maneira que um QGS muito especializado, não representativo e limitado podem ditar a eficiência da ferramenta.

Foi identificado um equívoco no Estudo Secundário de Böhmer and Rinderle-Ma (2015). Apesar dos autores documentarem que foram inseridos ao GS 153 estudos, um dos artigos tratava-se de uma duplicata. Para evitar inconsistências nos resultados, o artigo duplicado foi retirado, dessa forma foi conside-

rado um GS de 152 estudos para esse ES.

6.2 Trabalhos Futuros

A SeSGx-LLM possui diversas oportunidades de melhorias e extensões a serem realizadas. Como abordado anteriormente, a SeSGx-LLM ainda deve ser preparada para o usuário final de fato. O uso de LLMs ainda pode ser aprimorado aplicando estratégias de refino de *prompts* para alcançar o máximo de desempenho dos modelos. Adicionalmente, levanta-se a problemática de domínios com vocabulário altamente especializado, que foi identificado como um fator problemático ao desempenho dos LLMs. Como forma de contornar esse cenário surge a necessidade de otimizar os modelos por meio de métodos de *fine-tuning* para domínios específicos.

Como trabalho futuro também destaca-se a hipótese de criar uma ferramenta que entregue os resultados da SeSGx-LLM sem intervenções externas aos *prompts* dados aos LLMs. Isto é, por meio de conceitos de engenharia de *prompt*, atestar a possibilidade de adicionar o contexto do estudo (*Retrieval-Augumented Generation*), descrição do raciocínio de composição de *strings* de busca (*Chain-of-Thought*) e por fim gerar *strings* baseando-se no contexto inserido, tudo apenas utilizando *prompts* de comando.

Com a vasta quantidade de modelos que são publicados frequentemente, surge a necessidade de experimentar modelos mais modernos e atualizados para a tarefa de geração de *strings*. Em conjunto, para um melhor resultado da ferramenta, há a necessidade de avaliação dos resultados das *strings* de busca em tempo real, promovendo uma maior dinamicidade de decisão de quais *strings* de fato podem ser valiosas ao pesquisador.

A adição de novas métricas como a eficiência do processo pode ser agregada a partir do desenvolvimento experimental de um ES, no qual dois autores realizam o processo de identificação de textos relevantes dentro de um mesmo domínio, com o fator diferencial sendo o uso da SeSGx-LLM para um dos autores. A partir desse processo, seria possível identificar na prática o ganho real em utilizar a ferramenta.

6.3 Disponibilidade de Artefatos

O código referente ao pacote SeSGx e a CLI que permite o uso da ferramenta pode ser encontrado dentro dos repositórios da organização SeSGx-LLM [1].

[1] https://github.com/sesgx

Referências Bibliográficas

Abacha, A. B., Yim, W.-w., Fu, Y., Sun, Z., Yetisgen, M., Xia, F., e Lin, T. (2024). Medec: A benchmark for medical error detection and correction in clinical notes. *arXiv preprint arXiv:2412.19260*.

Abayomi-Alli, O. O., Damaševičius, R., Qazi, A., Adedoyin-Olowe, M., e Misra, S. (2022). Data augmentation and deep learning methods in sound classification: A systematic review. *Electronics*, 11(22).

Aggarwal, C. C. (2015). *Mining Text Data*, páginas 429–455. Springer International Publishing, Cham.

Alves, L. F., Vasconcellos, F. J., e Nogueira, B. M. (2022). Sesg: a search string generator for secondary studies with hybrid search strategies using text mining. *Empirical Software Engineering*, 27(5):105.

Azeem, M. I., Palomba, F., Shi, L., e Wang, Q. (2019). Machine learning techniques for code smell detection: A systematic literature review and meta-analysis. *Information and Software Technology*, 108:115–138.

Benchekroun, Y., Dervishi, M., Ibrahim, M., Gaya, J.-B., Martinet, X., Mialon, G., Scialom, T., Dupoux, E., Hupkes, D., e Vincent, P. (2023). Worldsense: A synthetic benchmark for grounded reasoning in large language models.

Bertolino, A., Angelis, G. D., Gallego, M., García, B., Gortázar, F., Lonetti, F., e Marchetti, E. (2019). A systematic review on cloud testing. *ACM Comput. Surv.*, 52(5).

Biolchini, J., Mian, P. G., Natali, A. C. C., e Travassos, G. H. (2005). Systematic review in software engineering. *System engineering and computer science department COPPE/UFRJ, Technical Report ES*, 679(05):45.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., e Choi, Y. (2019). Piqa: Reasoning about physical commonsense in natural language.

Blei, D. M. e Lafferty, J. D. (2009). Topic models. In *Text mining*, páginas 101–124. Chapman and Hall/CRC.

Blei, D. M., Ng, A. Y., e Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., e Amodei, D. (2020). Language models are few-shot learners.

Böhmer, K. e Rinderle-Ma, S. (2015). A systematic literature review on process model testing: Approaches, challenges, and research directions.

Church, K. W. (2017). Word2vec. *Natural Language Engineering*, 23(1):155–162.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., e Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge.

Cutigi Ferrari, F., Viola Pizzoleto, A., e Offutt, J. (2018). A systematic review of cost reduction techniques for mutation testing: Preliminary results. In *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, páginas 1–10.

Devlin, J., Chang, M.-W., Lee, K., e Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, páginas 4171–4186.

Dissanayake, N., Jayatilaka, A., Zahedi, M., e Babar, M. A. (2022). Software security patch management - a systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology*, 144:106771.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., e Gardner, M. (2019). Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dybå, T. e Dingsøyr, T. (2008). Strength of evidence in systematic reviews in software engineering. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, páginas 178–187.

Fayyad, U., Piatetsky-Shapiro, G., e Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34.

Gehring, J., Auli, M., Grangier, D., Yarats, D., e Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *International conference on machine learning*, páginas 1243–1252. PMLR.

Grames, E. M., Stillman, A. N., Tingley, M. W., e Elphick, C. S. (2019). An automated approach to identifying search terms for systematic reviews using keyword co-occurrence networks. *Methods in Ecology and Evolution*, 10(10):1645–1654.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., e Steinhardt, J. (2021a). Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., e Steinhardt, J. (2021b). Measuring mathematical problem solving with the math dataset.

Hosseini, S., Turhan, B., e Gunarathna, D. (2019). A systematic literature review and meta-analysis on cross project defect prediction. *IEEE Transactions on Software Engineering*, 45(2):111–147.

Imtiaz, S., Bano, M., Ikram, N., e Niazi, M. (2013). A tertiary study: experiences of conducting systematic literature reviews in software engineering. In *Proceedings of the 17th international conference on evaluation and assessment in software engineering*, páginas 177–182.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. v. d., Graves, A., e Kavukcuoglu, K. (2016). Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.

Kitchenham, B. A., Budgen, D., e Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*, volume 4. CRC press.

Kitchenham, B. A., Dyba, T., e Jorgensen, M. (2004). Evidence-based software engineering. In *Proceedings. 26th International Conference on Software Engineering*, páginas 273–281. IEEE.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, páginas 707–710. Soviet Union.

Lewis, M. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., e Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. e Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, páginas 281–297. University of California Press.

Manning, C. D. (2009). *An introduction to information retrieval*. Cambridge university press.

Marcos-Pablos, S. e García-Peñalvo, F. J. (2020). Information retrieval methodology for aiding scientific database search. *Soft Computing*, 24(8):5551–5560.

Marshall, C., Brereton, P., e Kitchenham, B. (2014). Tools to support systematic reviews in software engineering: a feature analysis. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, páginas 1–10.

McInnes, L., Healy, J., e Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Mergel, G. D., Silveira, M. S., e da Silva, T. S. (2015). A method to support search string building in systematic literature reviews through visual text

mining. In *Proceedings of the 30th annual ACM symposium on applied computing*, páginas 1594–1601.

Mihaylov, T., Clark, P., Khot, T., e Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering.

Mohan, V., Ali, A.-F. M., e Ameedeen, M. A. (2023). A systematic survey on the research of ai-predictive models for wastewater treatment processes. *Iraqi Journal For Computer Science and Mathematics*, pagina 102–113.

Mourão, E., Pimentel, J. F., Murta, L., Kalinowski, M., Mendes, E., e Wohlin, C. (2020). On the performance of hybrid search strategies for systematic literature reviews in software engineering. *Information and software technology*, 123:106294.

Paaß, G. e Giesselbach, S. (2023). *Pre-trained Language Models*, páginas 19–78. Springer International Publishing, Cham.

Petitti, D. B. (2000). *Meta-analysis, decision analysis, and cost-effectiveness analysis: methods for quantitative synthesis in medicine*. OUP USA.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rajaraman, A. e Ullman, J. D. (2011). *Data Mining*, pagina 1–17. Cambridge University Press.

Reimers, N. e Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Rezende, S. O. (2003). *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda.

Riaz, M., Sulayman, M., Salleh, N., e Mendes, E. (2010). Experiences conducting systematic reviews from novices' perspective. In *14th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, páginas 1–10.

Ros, R., Bjarnason, E., e Runeson, P. (2017). A machine learning approach for semi-automated search and selection in literature studies. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, páginas 118–127.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., e Choi, Y. (2019). Winogrande: An adversarial winograd schema challenge at scale.

Sap, M., Rashkin, H., Chen, D., LeBras, R., e Choi, Y. (2019). Socialiqa: Commonsense reasoning about social interactions.

Souza, F. C., Santos, A., Andrade, S., Durelli, R., Durelli, V., e Oliveira, R. (2018). Automating search strings for secondary studies. In *Information Technology-New Generations: 14th International Conference on Information Technology*, páginas 839–848. Springer.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., e Wei, J. (2022). Challenging big-bench tasks and whether chain-of-thought can solve them.

Talmor, A., Herzig, J., Lourie, N., e Berant, J. (2019). Commonsenseqa: A question answering challenge targeting commonsense knowledge.

van Dinter, R., Tekinerdogan, B., e Catal, C. (2021). Automation of systematic literature reviews: A systematic literature review. *Information and Software Technology*, 136:106589.

Van Rijsbergen, C. (1979). Information retrieval: theory and practice. In *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, volume 79.

Vasconcellos, F. J., Landre, G. B., Cunha, J. A. O., Oliveira, J. L., Ferreira, R. A., e Vincenzi, A. M. (2017). Approaches to strategic alignment of software process improvement: A systematic literature review. *Journal of Systems and Software*, 123:45–63.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., e Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022a). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Weiss, S. M. e Indurkhya, N. (1998). *Predictive data mining: a practical guide*. Morgan Kaufmann.

Weiss, S. M., Indurkhya, N., Zhang, T., e Damerau, F. (2010). *Text mining: predictive methods for analyzing unstructured information*. Springer Science & Business Media.

Wohlin, C. (2014a). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, páginas 1–10.

Wohlin, C. (2014b). Writing for synthesis of evidence in empirical software engineering. In *Proceedings of the 8th acm/ieee international symposium on empirical software engineering and measurement*, páginas 1–4.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., e Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Zhang, H., Babar, M. A., e Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhong, W., Cui, R., Guo, Y., Liang, Y., Lu, S., Wang, Y., Saied, A., Chen, W., e Duan, N. (2023). Agieval: A human-centric benchmark for evaluating foundation models.

Zwakman, M., Verberne, L. M., Kars, M. C., Hooft, L., van Delden, J. J., e Spijker, R. (2018). Introducing palette: an iterative method for conducting a literature search for a review in palliative care. *BMC palliative care*, 17(1):1–9.