Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Framework Híbrido com Aprendizado de Máquina Profundo para Desambiguação de Nomes de Autores

Natan de Souza Rodrigues

Tese apresentada como requisito parcial
para conclusão do Doutorado em Informática

Orientadora
Profª. Drª. Célia Ghedini Ralha

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Framework Híbrido com Aprendizado de Máquina Profundo para Desambiguação de Nomes de Autores

Natan de Souza Rodrigues

Tese apresentada como requisito parcial
para conclusão do Doutorado em Informática

Profª. Drª. Célia Ghedini Ralha (Orientadora)
IE/CIC/UnB

Prof. Dr. Marcos André Gonçalves
UFMG

Profª. Drª. Maristela Terto de Holanda
IE/CIC/UnB

Prof. Dr. Marlo Vieira dos Santos e Souza
UFBA

Profª . Drª. Cláudia Nalon
Coordenadora do Programa de Pós-Graduação em Informática

Brasília, 11 de agosto de 2025

Dedicatória

À minha pequena Laura, que transformou a minha vida com sua chegada, e à minha grande avó Aida, *in memoriam*, cuja força e sabedoria sempre me guiaram.

Agradecimentos

Agradeço, primeiramente, aos meus pais, Josimeire e José, à minha irmã, Naomí, e à minha companheira Taís, pelo apoio incondicional, incentivo e amor em cada etapa desta caminhada.

À minha orientadora, Profª. Drª. Célia Ghedini Ralha, agradeço profundamente pela confiança, paciência e orientação ao longo de todo o desenvolvimento desta pesquisa. Sua dedicação foi fundamental para a realização deste trabalho e sua presença constante, desde minha graduação, teve papel essencial na minha formação acadêmica.

Estendo meus agradecimentos aos professores do Departamento de Ciência da Computação (CIC) da Universidade de Brasília (UnB) pelas aulas, pelo conhecimento compartilhado e pela formação sólida ao longo da minha trajetória. Em especial, à Profª. Drª. Alba Cristina Magalhães de Melo, ao Prof. Dr. Li Weigang e ao Prof. Dr. José Carlos Ralha. Expresso também meus agradecimentos à Profª. Drª. Gecynalda Soares da Silva Gomes (UFBA) pelo apoio e orientação na condução das análises estatísticas relacionadas aos experimentos realizados nesta tese.

À UnB e ao CIC, pela estrutura e pelo ambiente acolhedor que possibilitaram o desenvolvimento desta pesquisa.

Ao colega Dr. Aurélio Costa pela parceria, pela colaboração e pelas valiosas trocas de conhecimento ao longo do desenvolvimento deste trabalho.

Aos colegas docentes da Universidade Estadual de Goiás, em especial ao Prof. Me. Aulo Plácio, pela parceria, apoio e valiosos ensinamentos.

Aos amigos, em especial ao Heitor Brito, pela amizade, incentivo e companheirismo.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a concretização deste trabalho.

"Computadores fazem arte."

Chico Science

Resumo

A desambiguação de nomes de autores (*Author Name Disambiguation* – AND) é uma tarefa desafiadora em repositórios bibliográficos digitais, marcada por ambiguidade nominal, variações linguísticas e metadados incompletos. Esta tese propõe o *framework* híbrido ADAN (*Automatic Disambiguation Author Name*), o qual combina aprendizado de máquina profundo com um algoritmo de agrupamento hierárquico aglomerativo aprimorado por grafos (*Graph-enhanced Hierarchical Agglomerative Clustering* - GHAC). Utiliza técnicas de Processamento de Linguagem Natural (PLN) com modelos baseados em *transformers* como SciBERT e MiniLM, e Redes Convolucionais de Grafos (RCG). O *framework* ADAN foi definido arquiteturalmente com quatro camadas: entrada e pré-processamento, extração de *embeddings* e construção da rede heterogênea, aprendizado com RCG e clusterização com GHAC. A camada de entrada conta com uma interface gráfica de usuário (*Graph User Interface* - GUI) que permite carregar os dados, configurar os parâmetros do modelo e visualizar os resultados da tarefa de AND. O *framework* ADAN é configurável, possibilitando adaptação a diferentes bases e níveis de complexidade estrutural e semântica. Os experimentos foram realizados utilizando três conjuntos de dados comuns na literatura: AMiner-12, DBLP e LAGOS-AND. Em cenários com metadados limitados, tal como o AMiner-12, o ADAN apresenta resultados competitivos atingindo média de pF1 de 0,6717 e K-Metric de 0,8981, superando trabalhos de referência em até 37,6% em *Average Cluster Purity (ACP)* e 20,21% em K-Metric. Com o conjunto de dados DBLP, o ADAN apresentou ganhos expressivos e valores estatisticamente significativos segundo as médias e intervalos de confiança obtidos, com 33,9% em pF1 e 29,8% em K-Metric e demais métricas permanecendo dentro dos intervalos de confiança de 95% inferior e superior. Utilizando o LAGOS-AND, os resultados apresentam B-cubed F1 de 90,8, superando em até 21,43% as abordagens anteriores com o mesmo conjunto de dados. Os resultados indicam que o *framework* ADAN oferece uma solução eficaz e adaptável para a tarefa de AND, apresentando desempenho consistente em cenários com alta ambiguidade e diversidade estrutural.

Palavras-chave: AND, PLN, RCG, Repositórios Bibliográficos Digitais

Abstract

Author Name Disambiguation (AND) is a challenging task in digital bibliographic repositories, marked by name ambiguity, linguistic variations, and incomplete metadata. This thesis proposes the hybrid framework ADAN (Automatic Disambiguation Author Name), which combines deep machine learning with a Graph-enhanced Hierarchical Agglomerative Clustering (GHAC) algorithm. It integrates Natural Language Processing (NLP) techniques using transformer-based models such as SciBERT and MiniLM, along with Graph Convolutional Networks (GCNs). The ADAN framework is architecturally defined with four layers: input and preprocessing, embedding extraction and heterogeneous network construction, learning with GCNs, and clustering with GHAC. The input layer includes a Graphical User Interface (GUI) that allows users to upload data, configure model parameters, and visualize the results of the AND task. The ADAN framework is configurable, allowing for adaptation to datasets with different levels of structural and semantic complexity. Experiments were conducted using three commonly used datasets in the literature: AMiner-12, DBLP, and LAGOS-AND. In scenarios with limited metadata, such as AMiner-12, ADAN presented competitive results with pF1 average of 0,6717 and K-Metric of 0,8981, outperforming the reference works in 37,6% of Average Cluster Purity (ACP) and 20,21% in K-Metric. On the DBLP dataset, ADAN presented significant gains and statistically significant values according to the means and confidence intervals, with 33,9% of pF1 and 29,8% of K-Metric, and other metrics remaining within the lower and upper 95% confidence intervals. Using LAGOS-AND, ADAN achieved a B-cubed F1 of 90.8, outperforming previous approaches by up to 21.43%. These results indicate that the ADAN framework offers an effective and adaptable solution for the AND task, showing consistent performance in scenarios with high ambiguity and structural diversity.

Keywords: AND, NLP, GCN, Digital Bibliographic Repositories

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

ADAN *Automatic Disambiguation Author Name*. xi, 5, 6, 8, 59, 66, 67, 76, 80–82, 85–88, 90, 93–100, 103–105, 107–109, 111, 112

AND Desambiguação de Nomes de Autores ou *Author Name Disambiguation*. 2–7, 9–11, 20, 21, 28, 30–34, 36, 38–41, 44, 46–59, 66, 67, 69, 70, 73, 76, 80, 81, 86–93, 99–102, 104, 105, 107, 108, 111, 112

BERT *Bidirectional Encoder Representations from Transformers*. 13–16, 18, 75, 82, 83, 90

CAPES Coordenação de Aperfeiçoamento de Pessoal de Nível Superior. 110

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico. 10

EUA Estados Unidos da América. 41, 43

GHAC *Graph-enhanced Hierarchical Agglomerative Clustering*. 5, 6, 19–21, 59, 66, 78–80, 86, 87, 90, 91, 100, 103, 105, 107–109

GUI *Graphical User Interface*. 69, 70, 80, 82, 84, 108, 112

HAC *Hierarchical Agglomerative Clustering*. 20, 57, 79, 91, 93

IA Inteligência Artificial. 2, 12, 32, 66

KL Kullback-Leibler. 18

LLM Grandes Modelos de Linguagem ou *Large Language Models*. 104, 105, 111, 112

MiniLM *Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-trained Transformers*. 16–18, 73, 83, 90, 91, 93–96, 98, 100–103, 105, 107, 108, 111

MLM Modelagem de Linguagem Mascarada. 14

MSE Erro Quadrático Médio ou *Mean Squared Error*. 28, 78, 86, 91

NSP Previsão da Próxima Frase ou *Next Sentence Prediction*. 14, 15

PLN Processamento de Linguagem Natural. ix, 2, 5–7, 9, 12, 13, 15, 16, 18, 59, 66, 70, 73, 75–77, 81, 86, 87, 90, 94, 96, 100, 101, 103, 108, 109

QP Questões de Pesquisa. 5, 66, 86, 88, 107

RCG Redes Convolucionais em Grafos. 5, 6, 9, 20–22, 25–28, 59, 66, 67, 72, 76–78, 80, 81, 83, 85–87, 90, 91, 93–96, 98–101, 103–105, 107–112

ReLU Unidade Linear Retificada ou *Rectified Linear Unit*. 26–28, 77, 91

RNC Redes Neurais Convolucionais. 12, 21, 26

RNR Redes Neurais Recorrentes. 13, 26

SciBERT *Pretrained Language Model for Scientific Text*. 15, 16, 73, 82, 83, 90, 91, 94–96, 100–102, 108

TEMAC Teoria do Enfoque Metaanalítico Consolidado. 32, 33

UnB Universidade de Brasília. 10

WoS *Web of Science*. 33, 38, 39, 41, 46, 60, 61, 63

Sumário

Capítulo 1

Introdução

Repositórios bibliográficos digitais como DBLP,[1] AMiner,[2] CiteSeerX,[3] e PubMed[4] disponibilizam informações de citações bibliográficas, oferecendo funcionalidades que permitem a identificação de trabalhos científicos, autores e suas respectivas redes sociais acadêmicas.

O DBLP, por exemplo, lista em julho de 2025, 8.007.895 trabalhos científicos na área de Ciência da Computação, incluindo artigos em periódicos, conferências, *workshops*, entre outros tipos de publicação (e.g., monografias, teses, livros), além de reunir informações de aproximadamente 3.8 milhões de autores. Entre janeiro de 2025 e julho de 2025, foram adicionadas 246.845 novas publicações. Verificou-se ainda que a quantidade de publicações aumentou 249,88% entre 2015 (3.204.638) e 2025 (8.007.895) (DBLP, 2025). O AMiner, em julho de 2025, armazena informações de aproximadamente 310 milhões de publicações, 57 milhões de autores e 2.5 bilhões de citações bibliográficas (AMiner, 2025).

Devido ao grande volume de registros bibliográficos armazenados, os repositórios digitais tornam-se uma importante fonte de informação para a comunidade acadêmica e científica mundial, permitindo a busca por publicações relevantes de forma centralizada (Ferreira et al., 2020). Além do recurso de pesquisa bibliográfica, essas bibliotecas digitais também fornecem análises úteis e outras funcionalidades, as quais são utilizadas para uma melhor tomada de decisão por agências de financiamento científico e instituições acadêmicas (Hussain and Asghar, 2017).

Comumente, nesses repositórios, diferentes nomes de autores podem compartilhar a mesma referência bibliográfica devido a abreviações de nomes e erros tipográficos, o que os torna indistinguíveis no conjunto de informações, representando desafios para a recuperação de informação (Xiong et al., 2021a). O problema de ambiguidade de nomes de autores em repositórios bibliográficos digitais ocorre quando autores distintos têm o

[1] `https://dblp.org/`
[2] `https://www.aminer.cn/`
[3] `https://citeseerx.ist.psu.edu/`
[4] `https://pubmed.ncbi.nlm.nih.gov/`

mesmo registro de nome (homônimos) e quando um autor tem variados registros de nomes no mesmo conjunto de dados (sinônimos) (Zhou et al., 2024b). Dessa forma, mesmo que o problema de ambiguidade de nomes de autores tenha sido estudado por décadas, ele ainda permanece sem uma solução canônica.

A ambiguidade de nomes de autores pode afetar significativamente o desempenho da recuperação de documentos e informações por meio de mecanismos de pesquisa na *Web*, além de obstruir a integridade de entidade para bancos de dados integrados. Os esforços para resolver esse problema trazem uma questão de pesquisa importante, especialmente em repositórios bibliográficos digitais que atualmente estão se tornando mais centrados na pessoa do que em documentos (Shin et al., 2014).

Diversas abordagens têm sido propostas na literatura para resolver o problema de ambiguidade de nomes de autores. Alguns métodos baseiam-se em técnicas heurísticas e estratégias de resolução progressiva, como o trabalho de Backes and Dietze (2022), que propõe uma abordagem em que as comparações entre nomes são realizadas gradualmente, reduzindo a complexidade computacional. Com o avanço da Inteligência Artificial (IA), surgiram novas soluções que aplicam técnicas de aprendizado de máquina para capturar padrões mais complexos. O trabalho de Sun et al. (2020), combina redes neurais com técnicas de similaridade de grafos para realizar a tarefa de Desambiguação de Nomes de Autores ou *Author Name Disambiguation* (AND) em nível par-a-par. Outro exemplo é o trabalho de Boukhers and Asundi (2022), que emprega uma rede neural para aprender representações vetoriais de coautores e títulos de publicações, promovendo maior precisão na identificação de autores.

Recentemente, técnicas baseadas em aprendizado de máquina profundo com grafos vêm se consolidando como alternativas promissoras. Trabalhos como os de Rastogi et al. (2023) e Zhou et al. (2024a) exploram o uso de modelos capazes de capturar simultaneamente dependências estruturais e textuais em redes acadêmicas. Abordagens híbridas também têm ganhado destaque. Choi et al. (2024) propõem um método que combina regras heurísticas, redes neurais e algoritmos de agrupamento. Complementarmente, Huang et al. (2024) apresentam uma abordagem de transferência entre domínios, que utiliza representações textuais geradas por modelos de Processamento de Linguagem Natural (PLN) integradas a estruturas de grafos.

Conforme apresentado por Ferreira et al. (2020), diversas abordagens têm sido aplicadas à tarefa de AND, explorando técnicas de aprendizado de máquina, modelagem semântica e análise de grafos. No entanto, observa-se um espaço para o desenvolvimento de soluções integradas, que combinem diferentes perspectivas metodológicas na resolução da ambiguidade de nomes de autores em repositórios bibliográficos digitais.

Considerando o cenário apresentado, esta pesquisa assume um caráter investigativo

e propositivo, com o objetivo de compreender os principais desafios da tarefa de AND. Neste sentido, é proposto uma abordagem de solução com um *framework* híbrido, que articule diferentes métodos de aprendizado de máquina profundo para resolver de forma computacionalmente eficaz a tarefa de AND.

1.1 Motivação

Com o crescimento contínuo de publicações científicas e o conjunto de informações disponíveis nos repositórios bibliográficos digitais, o problema de ambiguidade de nomes de autores se tornou cada vez mais complexo. Segundo Bollen et al. (2007), há um aumento substancial de artigos de pesquisa científica, resultando em frequentes propostas de métodos de AND.

Os repositórios bibliográficos digitais, por proverem informações relevantes de pesquisa científica, autoria, coautoria e outras informações pertinentes à comunidade acadêmica, podem retornar erroneamente informações quando há ambiguidade de nomes de autores em seu conjunto de dados, ou quando o método aplicado à AND não apresenta uma acurácia satisfatória.

A literatura apresenta diversas abordagens para a tarefa de AND, como os baseados em similaridade de *strings*, análise de redes de coautoria e técnicas de aprendizado supervisionado (Debarshi et al., 2019; Hung et al., 2014; Hussain and Asghar, 2017). No entanto, tais métodos podem enfrentar limitações. Estratégias baseadas apenas em *strings* podem ser sensíveis ao tamanho e à forma como os nomes dos autores são representados (da Silva, 2007). Métodos supervisionados podem exigir grandes volumes de dados rotulados, enquanto técnicas que dependam exclusivamente da rede de coautoria podem falhar quando os autores possuem poucas conexões ou colaborações incomuns (Ferreira et al., 2012).

Apesar de avanços relevantes, conforme discutido por Ferreira et al. (2020), esse cenário evidencia que a ambiguidade de nomes de autores continua sendo um problema de investigação crescente por falta de uma solução canônica. Especialmente considerando os ambientes de larga escala, com grande volume de registros bibliográficos, dados incompletos, desbalanceados ou heterogêneos. Ainda há desafios quanto à integração de diferentes tipos de informação, à adaptação a contextos variados e à escalabilidade das soluções existentes. Neste sentido, torna-se necessário o desenvolvimento de soluções que preencham as lacunas presentes nas abordagens atuais de AND.

1.2 Problema

Repositórios bibliográficos digitais necessitam de modelos eficientes de AND para garantir a validade e autenticidade da informação. Conforme apresentado em Ferreira et al. (2020), existem vários problemas motivacionais que devem ser considerados para o desenvolvimento de soluções confiáveis para a tarefa de AND, principalmente quando aplicadas a repositórios bibliográficos digitais de grande volume. Considerando os problemas elencados, neste trabalho será dado foco aos seguintes:

- Poucos dados nas citações bibliográficas – muitos repositórios bibliográficos digitais apresentam apenas informações básicas sobre as citações disponíveis (metadados), como os nomes dos autores (coautores), títulos das obras, local e ano de publicação. Além disso, em alguns casos, os nomes dos autores contêm apenas a inicial e o último sobrenome, e o título do local de publicação abreviado. Novas estratégias que buscam derivar informações implícitas (por exemplo, tópicos) ou coletar informações adicionais da *Web* são promissoras nesse cenário.
- Eficiência – com a enorme quantidade de trabalhos publicados atualmente em diversas áreas do conhecimento, os métodos atuais de AND precisam lidar com o problema de forma eficiente. No entanto, poucos dos métodos propostos na literatura apresentam essa preocupação explícita.
- Praticidade e custo – muitos métodos para AND são baseados em aprendizado supervisionado, requerendo uma grande quantidade de dados rotulados manualmente para indicar se dois nomes ambíguos correspondem ou não ao mesmo autor, indicando os autores corretos para as referências. Os dados rotulados servem como treinamento para os procedimentos de aprendizado de máquina. No entanto, a criação desses dados de treinamento é custosa, prejudicando a aplicação prática desses métodos, principalmente porque os repositórios bibliográficos evoluem e mais treinamento é necessário para aprender novos padrões.
- Padrões de publicações distintos – a maioria dos repositórios bibliográficos digitais utilizadas para avaliar métodos de AND está relacionada à Ciência da Computação. No entanto, outras áreas do conhecimento (por exemplo, Humanidades, Medicina, Geologia) podem ter padrões de publicação diferentes (por exemplo, publicações com um único autor ou com muitos coautores), causando dificuldades adicionais para a geração de métodos de solução.
- Eficácia – os métodos para AND devem ser eficazes, desambiguando corretamente os nomes dos autores em citações bibliográficas sem afetar o desempenho. Embora

muitos métodos tenham sido relatados recentemente, ainda há espaço para melhorias em relação à eficácia, conforme evidenciado em estudos comparativos da literatura.

1.3 Questões de Pesquisa

Diante dos problemas apresentados (Seção 1.2), evidencia-se a necessidade de investigações e propostas de solução que abordem a escassez de metadados, eficiência quanto a grande quantidade de trabalhos publicados em diferentes áreas de conhecimento com praticidade e redução de custo de soluções supervisionadas, além de padrões de publicações distintos e manutenção de eficácia.

Nesse sentido, o desenvolvimento de novas soluções baseadas em técnicas avançadas de aprendizado de máquina abre um novo cenário de pesquisa para métodos eficazes de AND. O problema central reside na falta de abordagens suficientemente robustas, baseadas em técnicas de aprendizado profundo, que sejam ajustadas à complexidade dos dados presentes em repositórios bibliográficos digitais, capazes de lidar com a ambiguidade de nomes de autores de forma eficaz.

Para direcionar a investigação conduzida neste trabalho, foram formuladas Questões de Pesquisa (QP), conforme descritas a seguir:

- QP1: De que forma um *framework* híbrido, que combine aprendizado de máquina profundo, com Redes Convolucionais em Grafos (RCG), técnicas de PLN baseadas em *transformers* e agrupamento hierárquico aprimorado, provê um método eficaz para a tarefa de AND em repositórios bibliográficos digitais?
- QP2: Em que medida o *framework* híbrido proposto para AND apresenta desempenho superior aos trabalhos de referência na literatura?

1.4 Hipótese e Objetivos

A hipótese desta tese é que um *framework* híbrido, que combine aprendizado de máquina profundo com RCG, técnicas de PLN baseadas em *transformers* e agrupamento hierárquico aprimorado, apresenta desempenho superior em termos de eficiência e eficácia na tarefa de AND, quando comparado a trabalhos de referência reportados na literatura.

O objetivo geral desta tese é propor e avaliar o *framework* ADAN para a tarefa de AND em repositórios bibliográficos digitais.

Os objetivos específicos são:

- Implementar uma arquitetura que combina técnicas de PLN baseadas em *transformers*, RCG e *Graph-enhanced Hierarchical Agglomerative Clustering* (GHAC);

- Realizar experimentos empíricos em bases de dados de referência na literatura de AND (AMiner-12, DBLP e LAGOS-AND);
- Comparar quantitativamente o desempenho do ADAN com métodos utilizados em trabalhos de referência e métricas padronizadas na literatura.

Como resultado deste trabalho de pesquisa, foram realizadas publicações científicas, abordando a revisão da literatura, os aspectos metodológicos e resultados experimentais da proposta. As contribuições resultantes desta pesquisa estão listadas na Seção 6.1.

1.5 Metodologia

Esta pesquisa possui natureza **aplicada**, uma vez que busca oferecer uma solução tecnológica para o problema de AND em repositórios bibliográficos digitais, um desafio recorrente em ambientes de alta ambiguidade nominal, variações linguísticas e metadados incompletos. Quanto à abordagem, a pesquisa é **quantitativa**, dado que a avaliação do *framework* proposto baseia-se em métricas formais de desempenho. Do ponto de vista metodológico, classifica-se como **experimental**, pois propõe, implementa e avalia um artefato computacional em condições controladas, seguindo princípios da experimentação em Engenharia de Software e Ciência da Computação (Hevner et al., 2004; Wieringa, 2014; Wohlin et al., 2012; Zelkowitz and Wallace, 1997).

Procedimentos Metodológicos

O procedimento adotado consistiu no desenvolvimento do framework híbrido ADAN, estruturado em quatro camadas: (i) entrada e pré-processamento, (ii) extração de *embeddings* e construção de rede heterogênea, (iii) aprendizado com RCG e (iv) agrupamento hierárquico aprimorado com GHAC. O *framework* foi implementado com suporte a técnicas de PLN, utilizando modelos baseados em *transformers* (SciBERT, MiniLM).

Etapas da Investigação

A investigação foi conduzida em três etapas principais:

1. Implementação do *framework* ADAN – definição da arquitetura, integração das técnicas de aprendizado profundo, RCG e agrupamento GHAC.
2. Execução dos experimentos – aplicação do *framework* em três conjuntos de dados amplamente utilizados na literatura (AMiner-12, DBLP e LAGOS-AND), selecionados por sua relevância e diversidade estrutural.

3. Avaliação e análise comparativa – mensuração do desempenho por meio de métricas amplamente adotadas na literatura de AND, como *Pairwise F1*, *K-Metric*, e *B-Cubed*, possibilitando a comparação sistemática com métodos de referência.

Estratégia de Validação

A validação dos resultados seguiu abordagem empírica e comparativa, baseada em experimentos reprodutíveis com conjuntos de dados padronizados. Para assegurar a robustez das conclusões, os resultados obtidos foram confrontados com trabalhos de referência da literatura de AND, permitindo verificar ganhos relativos em diferentes cenários de ambiguidade nominal e disponibilidade de metadados. Foram aplicadas métricas de estatística descritiva e inferencial, tais como média, desvio padrão e intervalo de confiança, para analisar e descrever conjuntos de dados, verificar a significância dos resultados e reforçar a validade das conclusões (Bolfarine and Sandoval, 2001).

Etapas do Processo de Desenvolvimento

Para alcançar os objetivos propostos nesta tese, foram especificadas algumas etapas do processo de desenvolvimento da pesquisa, a qual possui natureza metodológica aplicada, experimental e qualitativa, subdivididas em três etapas principais:

1. Estudo e avaliação de técnicas

    - Estudar os fundamentos teóricos relacionados à AND, PLN e às estratégias de aprendizado e de agrupamento em grafos, avaliando sua aplicabilidade na área;
    - Realizar uma revisão da literatura de AND, com ênfase nos avanços das últimas duas décadas.

2. Desenvolvimento e validação da solução

    - Projetar e implementar um *framework* híbrido para AND, combinando as abordagens estudadas na etapa anterior;
    - Coletar e preparar os conjuntos de dados reais a partir de bases públicas;
    - Validar a proposta por meio de experimentos com os conjuntos de dados reais coletados, comparando os resultados com métodos de referência descritos na literatura.

3. Análise e disseminação dos resultados

    - Analisar os resultados obtidos com base em diferentes repositórios bibliográficos, utilizando métricas consolidadas na área de AND;

- Publicar os resultados em conferências e periódicos especializados na área de Computação;
- Redigir e defender a tese, apresentando os principais resultados da pesquisa, contribuições e limitações observadas.

1.6 Estrutura do Documento

A estrutura deste documento de tese inclui no Capítulo 2 os fundamentos teóricos englobando os conceitos utilizados nesta pesquisa; no Capítulo 3 é apresentada uma revisão da literatura; no Capítulo 4 é apresentado o *framework* híbrido ADAN, incluindo o modelo arquitetural e tecnologias utilizadas; no Capítulo 5 são apresentados os experimentos, resultados obtidos e discussão; no Capítulo 6 são apresentadas as considerações finais, incluindo as contribuições, publicações, limitações do trabalho e os próximos passos relacionados a esta pesquisa.

Capítulo 2

Fundamentação Teórica

Neste capítulo, apresentam-se os principais conceitos que fundamentam esta pesquisa. A Seção 2.1 aborda a tarefa de AND e os desafios relacionados à ambiguidade de nomes de autores. A Seção 2.2 trata do PLN, abordando técnicas utilizadas para representar textos de forma vetorial. A Seção 2.3 introduz os fundamentos sobre grafos e técnicas de agrupamento em estruturas desse tipo. Na Seção 2.4, discutem-se as RCG como abordagens de aprendizado profundo voltadas à modelagem estrutural em grafos. Por fim, a Seção 2.5 apresenta as principais métricas utilizadas na avaliação da tarefa de AND.

2.1 Desambiguação de Nomes de Autores

De acordo com Lee et al. (2007), a dificuldade de se obter conteúdo bibliográfico válido e consistente vem de possíveis erros de entrada de dados, formatos de citação diferentes, nomes de autores ambíguos e abreviações de títulos de locais de publicação. Entre essas dificuldades, a ambiguidade de nomes em conteúdo bibliográfico exige muita investigação científica.

A ambiguidade de nomes de pessoas representa um grande desafio para diversas aplicações, como recuperação de informação e análise de dados bibliográficos. Ao pesquisar trabalhos e publicações acadêmicas pelo nome de um autor, os resultados apresentados podem conter uma longa lista de citações de múltiplos autores com o mesmo nome (homônimos). Informações sobre o impacto acadêmico de autores em determinadas áreas são fundamentais para mensurar a contribuição de um determinado autor para a comunidade científica. Dessa forma, há necessidade de manter a precisão e a consistência dos dados apresentados nesses repositórios bibliográficos digitais (Sun et al., 2020).

Um exemplo de ambiguidade de nomes em um repositório bibliográfico digital é ilustrado na Figura 2.1, em que, após uma rápida pesquisa no AMiner pelo nome da pes-

quisadora “Célia Ralha” da Universidade de Brasília (UnB), são retornados dois registros diferentes. No entanto, após verificação dos artigos atribuídos em cada registro retornado e comparando-os com os trabalhos citados no Currículo Lattes [1] do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), fica constatado que todos os registros referem-se à mesma autora.

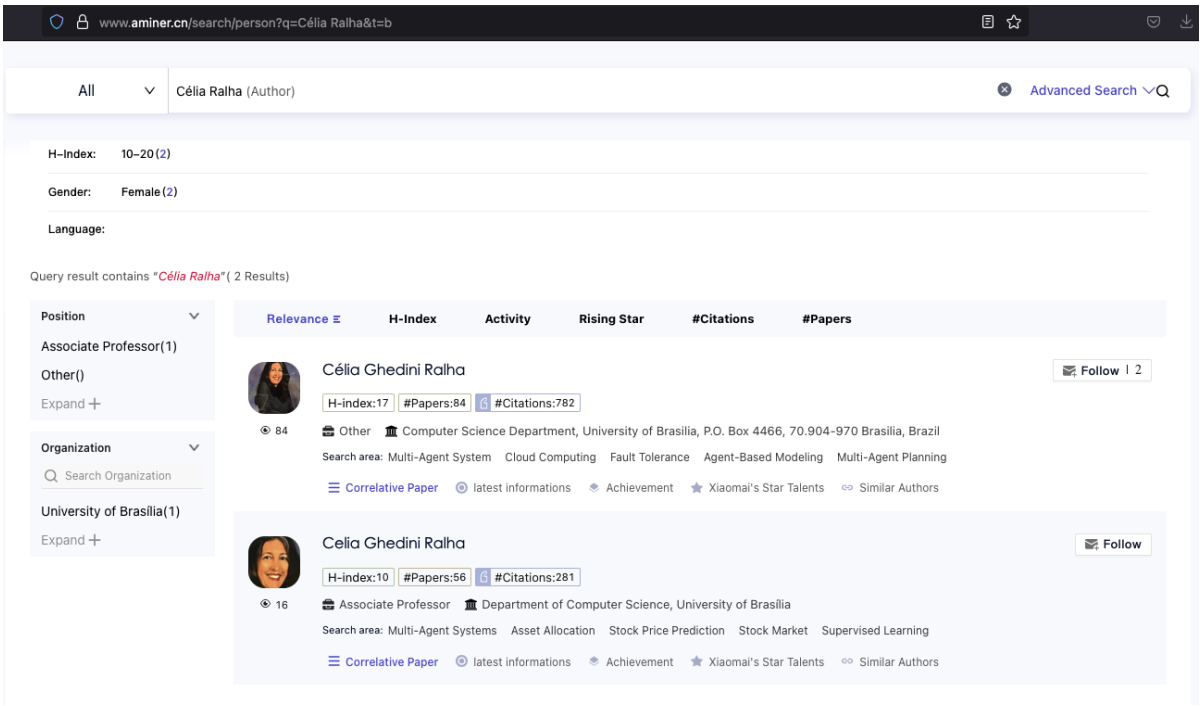


Figura 2.1: Resultado da busca pelo nome da autora “Célia Ralha” no AMiner retorna dois registros diferentes para a mesma pessoa. Fonte: AMiner, 2025. Disponível em: `https://www.aminer.cn/search/person?q=CeliaRalha`. Acesso em: 18 de Agosto de 2025.

Além disso, na página do pesquisador “Natan Rodrigues” no AMiner, observa-se a vinculação incorreta com o professor “Li Weigang”, da *School of Software* da *Northwestern Polytechnical University* (conforme indicado pela seta vermelha na Figura 2.2). Contudo, trata-se de um homônimo e não do docente da UnB com quem o pesquisador efetivamente colaborou. Esse é mais um exemplo claro dos efeitos da ambiguidade de nomes, onde a fusão de perfis distintos pode comprometer a integridade das informações em repositórios acadêmicos. A identificação incorreta também foi verificada por meio da análise dos trabalhos listados no perfil e confirmada com base nas informações disponíveis no Currículo Lattes do autor [2].

A resolução do problema da ambiguidade de nomes de autores em repositórios bibliográficos digitais é denominada, neste trabalho, de tarefa de AND. No geral, ao executar

[1] `http://lattes.cnpq.br/5632722847264046`
[2] `http://lattes.cnpq.br/1837696400623623`

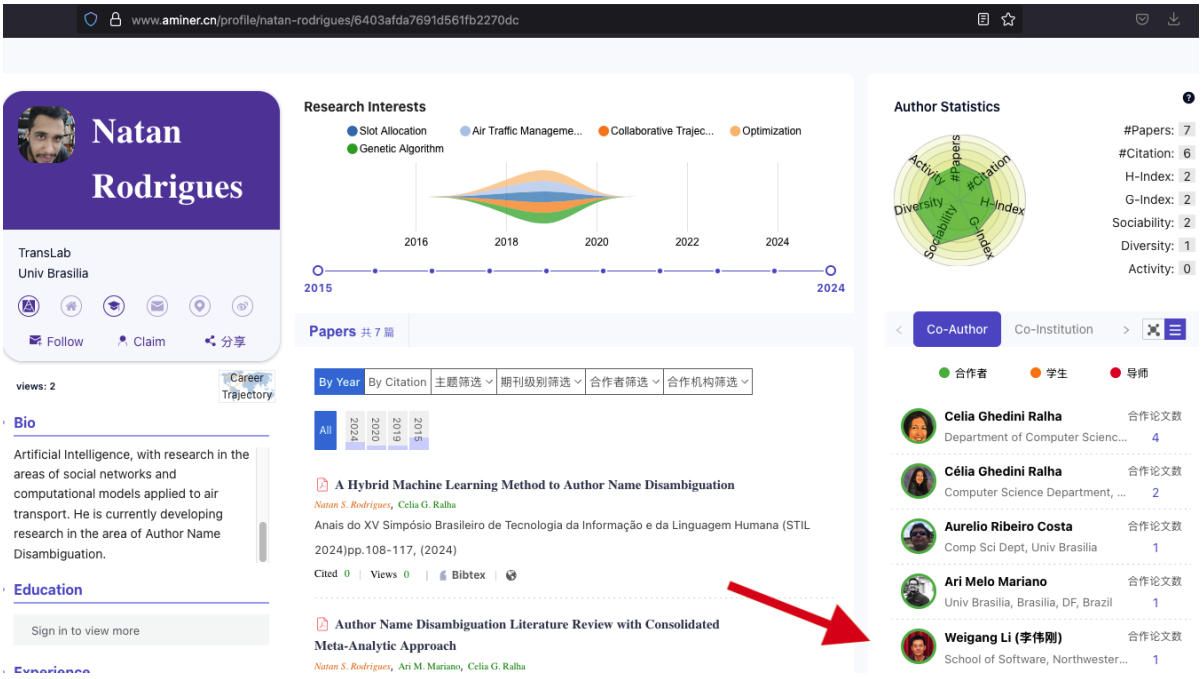


Figura 2.2: Exemplo de ambiguidade de nomes no AMiner. A seta vermelha destaca a vinculação incorreta entre o pesquisador "Natan Rodrigues" e o pesquisador "Li Weigang", que não corresponde ao verdadeiro coautor de seus trabalhos. Fonte: AMiner, 2025. Disponível em: `https://www.aminer.cn/profile/natan-rodrigues/6403afda7691d561fb2270dc`. Acesso em: 18 de Agosto de 2025.

uma tarefa de AND são utilizados atributos de publicações, como coautorias, títulos, resumos, local, afiliação e ano de publicação. Entretanto, nem todo repositório bibliográfico digital fornece todos esses atributos, disponibilizando informações limitadas que dificultam a resolução da ambiguidade em larga escala. Além disso, nos últimos anos, grandes quantidades de dados de publicações têm sido geradas e alocadas em repositórios bibliográficos, o que torna o problema da ambiguidade de nomes ainda mais desafiador do que no passado (Ferreira et al., 2012).

De acordo com Ferreira et al. (2020), a tarefa de AND é definida da seguinte forma: seja $C = \{c_1, c_2, \ldots, c_k\}$ um conjunto de registros de citações. Para uma dada instância $c_i \in C$, o valor de cada atributo *nomes de autores* refere-se a um autor distinto e está associado a um registro de autoria r_j.

O objetivo de um método de AND é produzir uma função de desambiguação que particione o conjunto de registros de autoria $\{r_1, r_2, \ldots, r_m\}$ em n subconjuntos $\{a_1, a_2, \ldots, a_n\}$, de modo que cada partição a_i contenha, idealmente, apenas os registros de autoria pertencentes ao mesmo autor.

A Figura 2.3, adaptada de Ferreira et al. (2020), ilustra o fluxo de trabalho da tarefa de AND, dividida em quatro etapas principais: Pré-processamento (1), Definição de Registros de Autoria Ambíguos (2), Agrupamento de Registros de Autoria Ambíguos (3) e

Desambiguação (4).

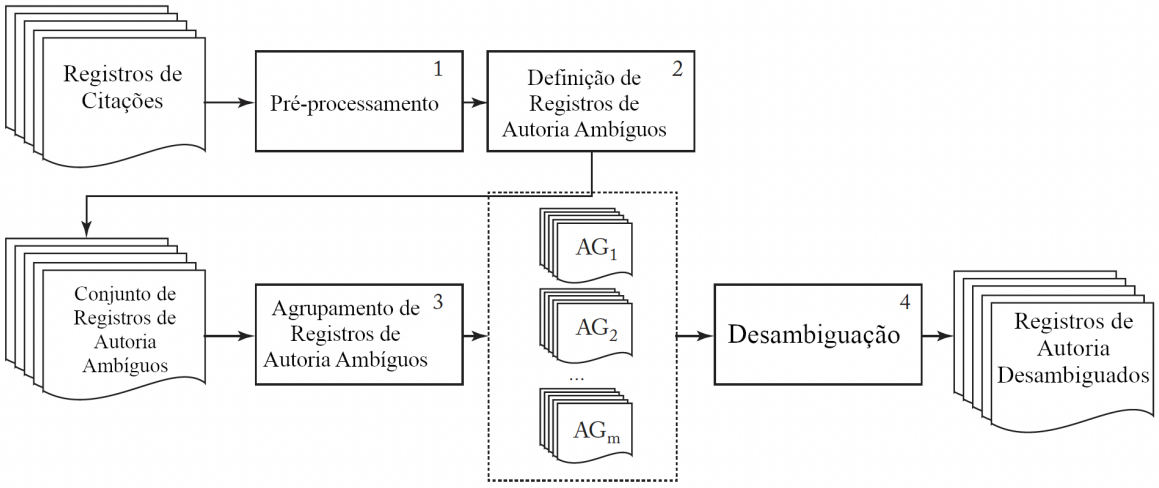


Figura 2.3: Fluxo de trabalho da tarefa de AND. Fonte: Traduzido de Ferreira et al. (2020).

2.2 Processamento de Linguagem Natural

O PLN surgiu na década de 1950 como um campo situado na interseção entre a IA e a linguística, voltado ao desenvolvimento de métodos e técnicas para o tratamento computacional de línguas naturais. Ao longo de sua evolução, o campo de PLN passou de abordagens baseadas em regras e gramáticas formais para técnicas mais robustas e flexíveis, incorporando métodos de aprendizado de máquina e processamento avançado de linguagem (Nadkarni et al., 2011).

Um exemplo de abordagem disseminada na área de tratamento de linguagem é o Word2Vec (Mikolov et al., 2013), que tem como base a utilização de vetores para a representação de palavras. Com um treinamento utilizando um grande volume de dados não rotulados, o Word2Vec consegue capturar relações semânticas e sintáticas entre as palavras. O modelo, por exemplo, identifica que a relação entre *rei* e *rainha* é semelhante à relação entre *homem* e *mulher*. Dessa forma, essa representação vetorial possibilita a captura de informações semânticas e sintáticas a partir de grandes volumes de dados não rotulados

Uma arquitetura utilizada em PLN são as Redes Neurais Convolucionais (RNC). Essas redes aplicam filtros convolucionais sobre janelas de palavras, permitindo a detecção de padrões locais. Em um problema de análise de sentimentos, por exemplo, as RNC podem aprender a identificar expressões de sentimentos específicas em pequenas janelas de palavras, permitindo uma classificação dos sentimentos em um determinado texto (Chen, 2015).

Por sua vez, as Redes Neurais Recorrentes (RNR) têm resultados satisfatórios em tarefas como a tradução automática e geração de texto. Sua arquitetura exclusiva foi projetada para lidar com dependências temporais, possibilitando que elas capturem e processem informações contextuais ao longo de uma sequência. Esse recurso também é vantajoso em tarefas como a análise de sentimentos em textos, em que a compreensão correta do contexto é essencial para a interpretação da polaridade emocional (Graves, 2013).

BERT: *Bidirectional Encoder Representations from Transformers*

Um marco relevante na área de PLN foi a apresentação do modelo *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2018). Esse modelo é caracterizado por ser uma rede neural, tendo como base a arquitetura *transformers*, podendo compreender o contexto das palavras em uma frase de forma bidirecional, ou seja, considera o contexto anterior e o posterior.

O BERT utiliza *transfer learning*, pré-treinando seus parâmetros em grandes conjuntos de textos não rotulados, com pequenas modificações para executar tarefas em domínios específicos. Essa abordagem baseada em *transfer learning* é poderosa para PLN, em que um modelo pré-treinado em grandes conjuntos de dados não rotulados é adaptado para domínios específicos, permitindo que o conhecimento geral da linguagem seja transferido e melhore o desempenho em variadas tarefas (Yosinski et al., 2014). Conforme Devlin et al. (2018), durante o pré-treinamento, o BERT aprende representações de palavras e frases que capturam uma compreensão geral da linguagem. Posteriormente, são realizados ajustes finos para tarefas específicas, adicionando uma camada de classificação e ajustando-a com base em conjuntos de dados rotulados.

Os dados de entrada do BERT seguem a abordagem *WordPiece* (Wu et al., 2016), em que as palavras são divididas em partes menores. Essa abordagem é eficiente com palavras desconhecidas e, além disso, bastante flexível. Por exemplo, a frase "Eu gosto de chocolate". Por meio da *token*ização do *WordPiece*, ela seria segmentada em subunidades como "Eu gosto de ch ##oco ##late". Consequentemente, uma frase de quatro palavras se torna uma entrada com seis subunidades de palavras a ser processada pelo BERT.

Além disso, dois *tokens* especiais são adicionados. O *token* "[CLS]" é inserido no início de cada entrada, sendo uma representação agregada de todo o texto utilizado para tarefas de classificação em nível de frase. O *token* "[SEP]" é usado para separar frases distintas dentro da mesma entrada. Isso permite que o BERT lide com tarefas que envolvem frases únicas, como análise de sentimentos, bem como tarefas que utilizam pares de frases, como

respostas a determinadas perguntas, em que os segmentos de pergunta e resposta são separados pelo *token* "[SEP]", repetido também no final da entrada. Dessa forma, o *Wordpiece* e a inclusão dos *tokens* especiais "[CLS]" e "[SEP]" são componentes integrais para formatar adequadamente os dados de entrada no BERT. A Figura 2.4 apresenta o exemplo do texto *"eu gosto de chocolate, me traz felicidade"* pré-processado.

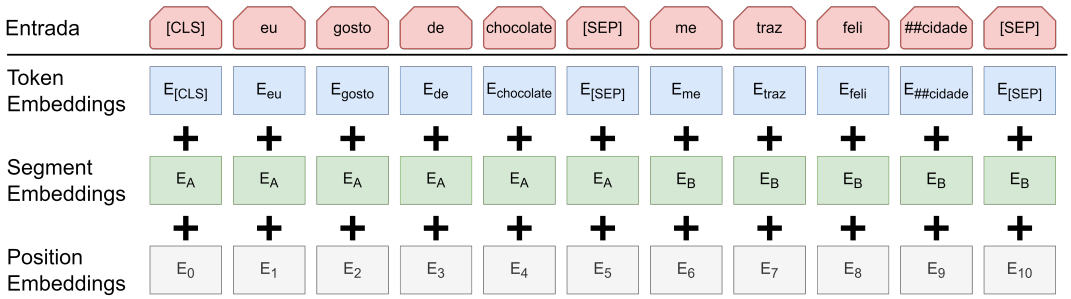


Figura 2.4: Pré-processamento textual do modelo BERT e criação de *embeddings* na camada inicial. Fonte: Adaptado de Devlin et al. (2018).

A formação de *embeddings* na camada inicial do BERT é ilustrada na Figura 2.4. Cada subunidade possui seu próprio *embedding*, denominado *Token Embedding*. Para identificar a qual parte da entrada cada *token* pertence, o modelo aprende o *Segment Embedding* A, atribuído até o primeiro *token* "[SEP]", e o *Segment Embedding* B, aplicado à sequência seguinte. Além disso, o BERT utiliza *Position Embeddings* para identificar a posição relativa entre os *tokens*. A *Position Embedding* no BERT atribui representações numéricas a *tokens* com base em sua posição na sequência de entrada, permitindo que o modelo capture informações estruturais e de ordem na linguagem. Isso ajuda o BERT a distinguir a ordem das palavras e a capturar as relações espaciais entre os *tokens*. Dessa forma, considerando a entrada do codificador inicial, cada *token* recebe a soma dos três tipos de *embeddings* (Devlin et al., 2018; Yang et al., 2019a).

De acordo com Eler (2022), e conforme apresentado na Figura 2.5, o pré-treinamento do BERT é composto por duas tarefas:

1. Modelagem de Linguagem Mascarada (MLM) - o modelo deve aprender e completar corretamente a frase. O BERT processa os *tokens* da frase e substitui alguns desses *tokens* por um especial, nomeado de "[MASK]". A arquitetura de *transformers* permite que se examine todos os *tokens* da frase de forma simultânea, característica de sua arquitetura bidirecional (considera o contexto anterior e posterior das palavras na frase).

2. Previsão da Próxima Frase ou *Next Sentence Prediction* (NSP) - verifica se uma frase posterior é uma sequência natural depois de uma determinada frase anterior. O NSP tem grande importância para as tarefas que envolvem frases, perguntas e respostas

e outras inferências de linguagem natural. Porém, segundo Aroca-Ouellette and Rudzicz (2020), a tarefa NSP pode ter um baixo desempenho em alguns modelos gerando variações do modelo BERT, que ou excluem a tarefa NSP ou propõem alternativas, como RoBERTA (Liu et al., 2019), DistilBERT (Sanh et al., 2019), XLNet (Yang et al., 2019b), ALBERT (Lan et al., 2019).

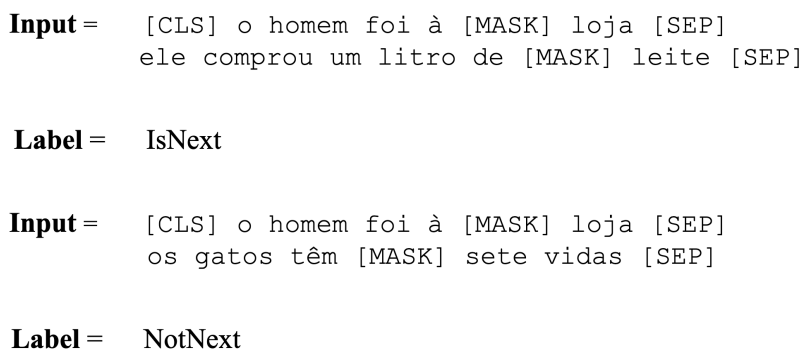


Figura 2.5: Tarefas de pré-treinamento do BERT. Fonte: Adaptado de Devlin et al. (2018).

Em conclusão, o modelo BERT incorporou ao PLN a combinação do *transfer learning* com a arquitetura *transformer*. O modelo apresenta um desempenho notável em várias tarefas de PLN, o que o tornou um modelo amplamente adotado na comunidade de pesquisa e desenvolvimento. Além disso, pesquisadores e desenvolvedores podem explorar e aplicar o BERT em suas próprias tarefas, devido à disponibilização de seu código-fonte e implementações abertas no GitHub.[3]

Existem variações do BERT desenvolvidas para domínios específicos, como o *Pretrained Language Model for Scientific Text* (SciBERT), criado para o processamento de textos científicos. Esse modelo adota a mesma arquitetura do BERT, mas foi pré-treinado com um *corpus* de 1,14 milhão de publicações da base Semantic Scholar [4], cobrindo áreas como Ciência da Computação e Biomedicina. Em experimentos realizados com tarefas típicas de PLN, como reconhecimento de entidades, classificação de relações e análise de citações, o SciBERT demonstrou desempenho superior ao BERT, obtendo ganhos médios de até +2,43 em pF1. Esses resultados indicam que o SciBERT é mais adequado para aplicações científicas, contribuindo para melhores representações semânticas em textos acadêmicos (Beltagy et al., 2019).

Para aprimorar a compreensão e a utilização do BERT, pode-se comparar suas diferentes variações e extensões. A Tabela 2.1 apresenta uma comparação de vários modelos BERT, incluindo as características e *links* para os repositórios do GitHub.

[3] `https://github.com/google-research/bert`
[4] `https://www.semanticscholar.org/`

Tabela 2.1: Comparação dos Modelos BERT.

Modelo	Repositório no GitHub	Características
BERT	google-research/bert	Modelo BERT original do Google
RoBERTa	pytorch/fairseq/roberta	Melhoria do BERT com treinamento mais longo e modelos maiores
XLNet	zihangdai/xlnet	Modelo auto-regressivo do BERT com treinamento baseado em permutação
DistilBERT	transformers/en/model_doc/distilbert	Versão compacta do BERT para inferência mais rápida
ALBERT	google-research/albert	Versão leve do BERT com tamanho de modelo e etapas de treinamento reduzidas
SciBERT	allenai/scibert	Modelo BERT pré-treinado para tarefas científicas

Fonte: Elaboração própria.

MiniLM: *Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-trained Transformers*

Conforme discutido anteriormente, o BERT pode ser utilizado em tarefas de PLN, mas sua arquitetura complexa e o número elevado de parâmetros podem dificultar sua adoção em cenários com restrições de tempo de inferência ou de uso de memória. Nesse contexto, o modelo *Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-trained Transformers* (MiniLM), proposto em Wang et al. (2020c), surge como uma solução leve e eficiente, oferecendo desempenho competitivo com uma quantidade menor de parâmetros.

O MiniLM é treinado por meio de uma técnica conhecida como *destilação de conhecimento*, em que um modelo menor (aluno) aprende a imitar o comportamento de um modelo maior e mais robusto (professor), como o BERT, por exemplo. No entanto, diferentemente de abordagens tradicionais, que focam apenas na saída final dos modelos, o MiniLM vai além e aprende também os padrões internos do modelo professor, especialmente aqueles ligados aos mecanismos de atenção.

O mecanismo de atenção é o componente que permite aos modelos do tipo *Transformer* atribuírem pesos diferentes às palavras de uma frase conforme o contexto. Em outras palavras, ele define o grau de relevância que cada palavra atribui às demais para compor uma representação contextualizada. Por exemplo, na frase "o aluno entregou a tarefa ao professor", o modelo aprende que "entregou" se relaciona mais com "aluno" e "tarefa" do que com "professor", ajustando sua representação interna com base nessa dinâmica.

Durante o processo de destilação de conhecimento, o MiniLM aprende dois aspectos centrais do mecanismo de atenção do professor:

1. Distribuição de Atenção: refere-se aos pesos de atenção gerados entre cada par de palavras. O MiniLM é treinado para imitar esses pesos, aproximando suas distribuições das geradas pelo modelo professor.

2. Relação entre Valores: além da atenção entre palavras, o MiniLM também aprende como as informações (valores) são combinadas e processadas pelo modelo maior. Isso é feito por meio de operações matemáticas entre vetores de valores, que são transformados em matrizes de relações. Esse aspecto aprofunda a capacidade do modelo aluno de replicar o comportamento interno do professor.

A Figura 2.6 ilustra esse processo. À esquerda, está representado o modelo professor, baseado em *Transformer*, com diversas camadas (blocos). Para o processo de destilação, utiliza-se apenas a última camada, da qual são extraídos os vetores de autoatenção: *queries*, *keys* e *values*, cada um com dimensão $\mathbb{R}^{A_h \times |x| \times d_k}$, em que A_h é o número de cabeças de atenção e $|x|$ o comprimento da sequência de entrada.

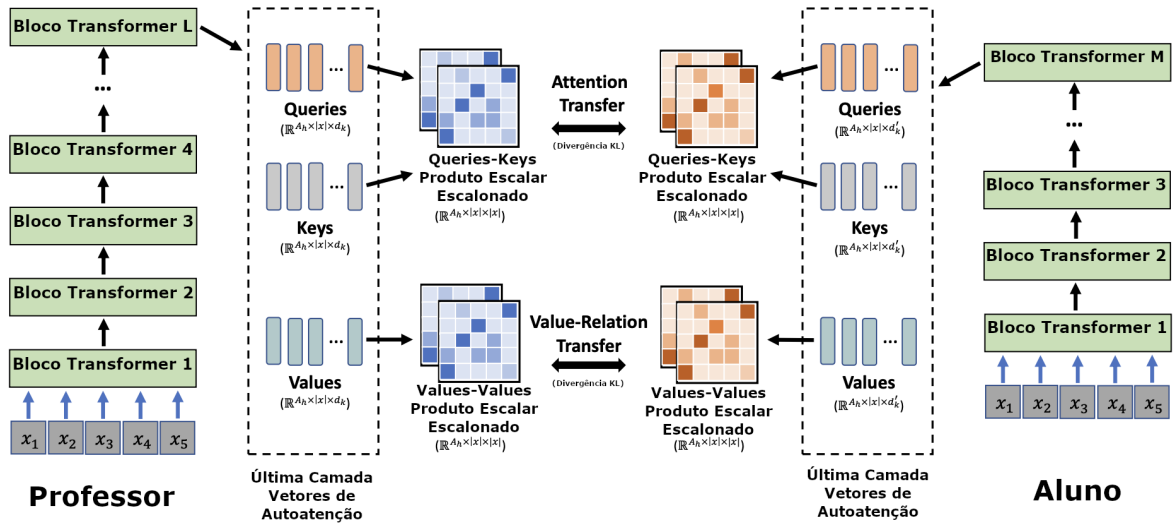


Figura 2.6: Processo de destilação no MiniLM: o modelo aluno aprende os mapas de atenção e as relações internas do modelo professor. Fonte: Adaptado de Wang et al. (2021).

A partir desses vetores, o modelo professor calcula duas representações principais:

- O Produto Escalar Escalonado (*Scaled Dot-Product*) entre *queries* e *keys*, que resulta nos mapas de atenção (*attention distributions*) com dimensão $\mathbb{R}^{A_h \times |x| \times |x|}$. Essas distribuições indicam o grau de importância que cada *token* atribui aos demais;

- O produto escalar escalonado entre *values*, também com dimensão $\mathbb{R}^{A_h \times |x| \times |x|}$, que forma uma matriz de relações semânticas entre os valores, denominada *value relation*.

À direita da Figura 2.6 está o modelo aluno, com estrutura mais compacta, ou seja, menos camadas e menor dimensionalidade (d'_k). Esse modelo também gera seus próprios vetores *queries*, *keys* e *values* a partir da última camada e é treinado para imitar as representações do professor por meio de duas etapas:

1. *Attention Transfer*, em que os mapas de atenção (*queries-keys*) são aproximados;
2. *Value-relation Transfer*, em que o modelo aluno aprende a reproduzir a matriz de relações internas entre os valores (*values-values*).

Ambas as representações são comparadas por meio da divergência de Kullback-Leibler (KL), uma medida estatística que quantifica o quão diferente é uma distribuição de probabilidade em relação a outra. Nesse contexto, ela é utilizada para aproximar as distribuições de atenção e de relações internas produzidas pelo modelo aluno em relação às do professor. Essa abordagem permite ao modelo aluno capturar os padrões internos do comportamento da autoatenção do modelo professor sem exigir correspondência camada a camada, nem igualdade de dimensionalidade. Como resultado, o MiniLM consegue aprender representações contextuais ricas de forma eficiente, com um custo computacional significativamente menor.

Com base nos resultados apresentados nos trabalhos de Wang et al. (2021, 2020c), o modelo se destaca por manter alto desempenho em tarefas de PLN mesmo com um número significativamente menor de parâmetros. Especificamente, o MiniLM com 66 milhões de parâmetros alcança mais de 99% da acurácia do BERT em *benchmarks* como SQuAD 2.0 (Rajpurkar et al., 2018) e GLUE (Wang et al., 2018), sendo ao mesmo tempo duas vezes mais rápido em tempo de inferência (Wang et al., 2020c). Além disso, em comparações diretas, o MiniLM supera modelos similares com o mesmo número de parâmetros, como DistilBERT (Sanh et al., 2019) e TinyBERT (Jiao et al., 2020), obtendo resultados superiores em várias tarefas, como análise de sentimentos, inferência textual e reconhecimento textual.

Posteriormente, foi proposto o MiniLMv2 (Wang et al., 2021), que incorpora novas estratégias de destilação multitarefa, incluindo o aprendizado de representações intermediárias das camadas, bem como das saídas de classificação. Essas melhorias resultaram em avanços consistentes de desempenho, reforçando o potencial dos modelos compactos para tarefas de PLN que exigem alta eficiência computacional.

2.3 Grafos

Nesta seção, serão apresentadas as propriedades dos grafos, incluindo uma definição formal, os principais conceitos (nós, arestas, grau dos nós e matriz de adjacência), bem como sua importância para a compreensão dos métodos utilizados neste trabalho. Também será apresentada a estrutura das redes heterogêneas, que permitem modelar diferentes tipos de entidades e suas relações em um mesmo grafo. Por fim, será discutido o método GHAC, que utiliza essas estruturas heterogêneas para realizar agrupamentos com base em similaridade.

A teoria dos grafos é um instrumento versátil e poderoso para a modelagem de sistemas complexos, abordando uma ampla gama de problemas. Esses problemas vão desde a modelagem de processos industriais e logística até sistemas de comunicação, redes de fluxo e seleção de rotas, entre outros. A teoria dos grafos possui aplicações em diversas áreas do conhecimento, como Engenharia, Ciência da Computação, Genética, Física, Química, Antropologia e Linguística (Bollobás, 1998; Newman, 2010).

Formalmente, um grafo pode ser definido como um par ordenado $G = (V, E)$, em que V é o conjunto de nós e E é o conjunto de arestas. Cada aresta $e \in E$ e conecta dois nós distintos $v_i, v_j \in V$, indicando que há uma relação ou interação entre esses nós. O grau de um nó em um grafo é o número de arestas incidentes a ele. Em um grafo não direcionado, o grau de um nó é igual ao número de vizinhos que ele possui. Denotamos o grau de um nó v_i como $d(v_i)$ (Newman, 2010).

A matriz de adjacência é uma representação matricial de um grafo, que descreve as conexões entre os nós. Essa matriz traz uma representação fundamental dos relacionamentos em um grafo, sendo amplamente utilizada em algoritmos de processamento de grafos (Bollobás, 1998; Newman, 2010).

Trata-se de uma matriz quadrada de ordem n, em que n é o número de nós no grafo. A entrada A_{ij} na matriz de adjacência indica se existe uma aresta entre os nós v_i e v_j. Em um grafo não direcionado, a matriz de adjacência é simétrica. A matriz de adjacência A pode ser expressa como:

$$A_{ij} = \begin{cases} 1, & \text{se existe uma aresta entre os nós } v_i \text{ e } v_j, \\ 0, & \text{caso contrário.} \end{cases}$$

A matriz de características serve para caracterizar um grafo. Essa estrutura contém informações sobre cada nó do grafo. Cada linha da matriz corresponde a um nó e cada coluna corresponde a uma característica específica. Essas características podem incluir informações textuais, numéricas ou categóricas associadas a cada nó. A matriz de carac-

terísticas X pode ser definida como:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1f} \\ x_{21} & x_{22} & \dots & x_{2f} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nf} \end{bmatrix}$$

em que n é o número de nós do grafo e f é o número de características.

Os conceitos básicos de grafos apresentados servem para compreender como as RCG podem ser aplicadas a essas estruturas.

Redes Heterogêneas

Conforme definido por Shi et al. (2017), redes heterogêneas são grafos compostos por múltiplos tipos de entidades (nós) e relações (arestas). Formalmente, uma rede heterogênea é um grafo $G = (V, E, \mathcal{T}_v, \mathcal{T}_e)$, em que V é o conjunto de nós, E é o conjunto de arestas, $\mathcal{T}_v$ denota o conjunto de tipos de nós e $\mathcal{T}_e$ denota o conjunto de tipos de arestas. Cada nó $v \in V$ está associado a um tipo $\phi(v) \in \mathcal{T}_v$, e cada aresta $e \in E$ possui um tipo $\psi(e) \in \mathcal{T}_e$.

Neste trabalho, no contexto de dados bibliográficos, os nós geralmente representam publicações, autores, palavras-chave ou veículos de publicação (e.g., conferências ou periódicos), e as arestas capturam relações como autoria, coautoria, compartilhamento de palavras-chave ou publicação em um mesmo veículo. Redes heterogêneas auxiliam na modelagem de estruturas relacionais e são amplamente utilizadas em tarefas como recomendação e predição de ligações.

GHAC: *Graph-enhanced Hierarchical Agglomerative Clustering*

O *Hierarchical Agglomerative Clustering* (HAC) é uma técnica clássica de agrupamento que inicia com cada ponto de dados formando seu próprio *cluster* unitário e, iterativamente, funde os pares mais próximos com base em medidas de dissimilaridade, até formar um agrupamento hierárquico (Müllner, 2011). No entanto, esse processo tradicional não considera relações topológicas explícitas entre os dados.

Para contornar essa limitação, e considerando sua aplicabilidade à tarefa de AND, o GHAC foi definido por Qiao et al. (2019) como um método que integra informações topológicas provenientes de grafos heterogêneos ao processo tradicional de HAC. Inicialmente, cada documento é considerado como um *cluster* individual. A cada iteração, o algoritmo funde o par de *clusters* com maior similaridade, calculada com base na média

da similaridade cosseno entre as representações vetoriais (*embeddings*) dos documentos, que codificam suas características semânticas e estruturais em um espaço numérico.

O GHAC opera sobre um grafo heterogêneo $G = (V, E)$, onde V representa o conjunto de nós que modelam entidades como, por exemplo, publicações, autores ou veículos de publicação, e $E \subseteq V \times V$ corresponde ao conjunto de arestas que indicam relações entre essas entidades. Cada nó de documento $v \in V$ está associado a um vetor de *embedding* $z_v \in \mathbb{R}^d$. A similaridade entre dois *clusters* C_i e C_j é calculada utilizando a média da similaridade cosseno entre todos os pares de documentos pertencentes a esses clusters:

$$\text{sim}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{u \in C_i} \sum_{v \in C_j} \frac{z_u^\top z_v}{\|z_u\| \cdot \|z_v\|}.$$

O algoritmo realiza iterativamente a fusão dos *clusters* mais similares até que um critério de parada seja atingido. Quando o número de *clusters* esperados K é conhecido, por exemplo, com base na quantidade de autores reais associados ao nome ambíguo, o algoritmo é interrompido assim que exatamente K *clusters* forem formados. Em situações nas quais o valor de K não é previamente conhecido, o número ideal de *clusters* pode ser determinado automaticamente por meio da avaliação de critérios de qualidade de agrupamento, como a modularidade.

2.4 Redes Convolucionais de Grafos

As RCG, propostas por Kipf and Welling (2016), são um poderoso modelo de aprendizado de máquina que estende as RNC para lidar com dados estruturados representados como grafos. As RCG têm recebido considerável atenção no campo de aprendizado de representações de grafos devido à sua capacidade de capturar dependências locais e globais dentro de um grafo.

As RCG têm sido amplamente aplicadas em diversas áreas, incluindo AND. Um exemplo de aplicação de grafos em AND é o trabalho de Pratama et al. (2019), no qual os autores utilizaram uma RNC de grafos profunda. Eles exploraram a estrutura de coautoria em redes de colaboração científica, capturando as informações de coautoria em um grafo e combinando-as com *embeddings* de palavras e características adicionais dos autores. A abordagem proposta obteve resultados promissores, superando outras técnicas de AND.

Portanto, a utilização de grafos em AND proporciona uma representação rica e abrangente, permitindo capturar as relações e interações entre os elementos. A abordagem baseada em grafos leva em consideração o contexto compartilhado entre os elementos e pode melhorar a precisão e a qualidade dos resultados na tarefa de AND.

O trabalho apresentado por Hamilton (2020) servirá como base na apresentação das definições e conceitos acerca de RCG, elencados nas subseções que seguem.

Representação e Entrada de Dados

Ao desenvolver um modelo de aprendizado profundo (e.g., RCG) para aplicação em grafos, é essencial considerar um método de entrada de dados que respeite a estrutura intrínseca dessas representações e leve em consideração as características específicas associadas a cada nó. Nesse contexto, é apropriado conceber um modelo capaz de receber, como entrada, tanto a matriz de adjacência A, quanto a matriz de características X, vinculadas a um determinado grafo G.

Entretanto, enfrentamos um desafio ao prosseguir com essa abordagem. A representação da matriz de adjacência A depende da disposição dos nós em suas linhas e colunas, resultando em diferentes matrizes que podem representar o mesmo grafo. Isso introduz uma sensibilidade à ordem dos nós na matriz de adjacência, algo que deve ser evitado.

Dessa forma, qualquer função f que aceite uma matriz de adjacência A associada a um grafo G como entrada deve atender a uma das seguintes propriedades: invariância por permutação $f(PAP) = f(A)$ ou equivariância por permutação $f(PAP) = Pf(A)$. A invariância por permutação indica que a função não é afetada pela ordem dos nós na matriz de adjacência, enquanto a equivariância por permutação assegura que a saída da função seja permutada juntamente com a matriz de adjacência de entrada (Gilmer et al., 2017).

As RCG são projetadas para satisfazer a equivariância por permutação por meio de operações de agregação local baseadas na matriz de adjacência, permitindo combinar as características de cada nó com as de seus vizinhos imediatos, sem necessariamente preservar toda a informação global das arestas. Conforme descrito por Kipf and Welling (2016), essa abordagem realiza a propagação camada a camada de representações de nós, em que a atualização de cada nó é obtida a partir de uma combinação ponderada de suas próprias características e das de seus vizinhos, normalizada pela estrutura do grafo.

Passagem de Informações

Como discutido anteriormente, o modelo de RCG utiliza duas fontes de dados: a matriz de adjacência A, que representa as conexões entre os nós, e a matriz de características X, que contém as informações de cada nó. Essas informações são combinadas em um vetor imerso h_u, associado a cada nó u, com dimensão d, correspondente ao número de características em X.

A passagem de informações ocorre nos vetores imersos por meio de iterações. Ao inserir as matrizes no modelo, antes do início do algoritmo de transferência de informações, estamos na iteração zero. Nessa fase, o vetor imerso h_u do nó u é o vetor de características desse mesmo nó u. Logo, podemos expressá-lo por $h_u^{(0)} = x_u, \quad \forall u \in V$. Essa abordagem facilita a visualização da dimensão do vetor imerso, pois, na iteração zero, esse vetor é basicamente o vetor de características, cuja dimensão é a quantidade total de características.

Posteriormente, para obtermos as informações do nó u e classificá-lo, o modelo precisa agregar as informações dos nós vizinhos de u. No entanto, para coletar as informações dos vizinhos de u, é necessário agregar as informações dos vizinhos dos vizinhos de u, em um processo recursivo.

Supondo que esse processo ocorra k vezes, então definimos o vetor imerso $h_u^{(k)}$, que é atualizado com as informações dos seus vizinhos. O objetivo dessa estrutura de modelo é, após K iterações desse método de transmissão de mensagens, criar um vetor imerso final $z_u^{(K)}$ para cada nó u em V, composto por todas as informações de cada nó. Podemos pensar nos vetores imersos $h_u^{(k)}$ como versões preliminares que são atualizadas ao longo das iterações, e nos vetores $z_u^{(K)}$, como os vetores imersos finais após todas as iterações.

Nessa estrutura, duas atividades distintas são realizadas:

1. Coleta – agrega as informações dos vizinhos $v \in N(u)$ do nó de interesse u.
2. Atualização – substitui o vetor anterior $h_u^{(k-1)}$ pelo vetor atualizado $h_u^{(k)}$.

Essas atividades podem ser vistas como duas funções. A função de coleta de informações recebe como entrada os vetores de informações $h_v^{(k)}$, onde $v \in N(u)$, sendo $N(u)$ a vizinhança do nó u. O resultado dessa função é a mensagem compilada por essas informações, usando apenas os nós da vizinhança de u. Essa mensagem serve como entrada para a função relacionada à atualização do vetor de informações, que, utilizando o vetor anterior $h_u^{(k-1)}$ como segunda entrada, realiza a atualização e gera o vetor $h_u^{(k)}$. A Figura 2.7 ilustra esse processo.

Logo, podemos representar matematicamente esse modelo de transmissão de mensagens. Vamos considerar a função f responsável pela coleta de informações e a função g encarregada pela atualização dessas informações, conforme Equação 2.1.

$$h_u^{(k+1)} = g^{(k)}\left(h_u^{(k)}, f^{(k)}\left(h_v^{(k)}, \forall v \in N(u)\right)\right) \tag{2.1}$$

Temos que o segundo argumento da função g refere-se aos vetores imersos coletados dos vizinhos v do nó de interesse u. Assim, podemos definir a mensagem $m_{N(u)}^{(k)}$ conforme a Equação 2.2.

$$m_{N(u)}^{(k)} = f^{(k)}\left(h_v^{(k)}, \forall v \in N(u)\right) \tag{2.2}$$

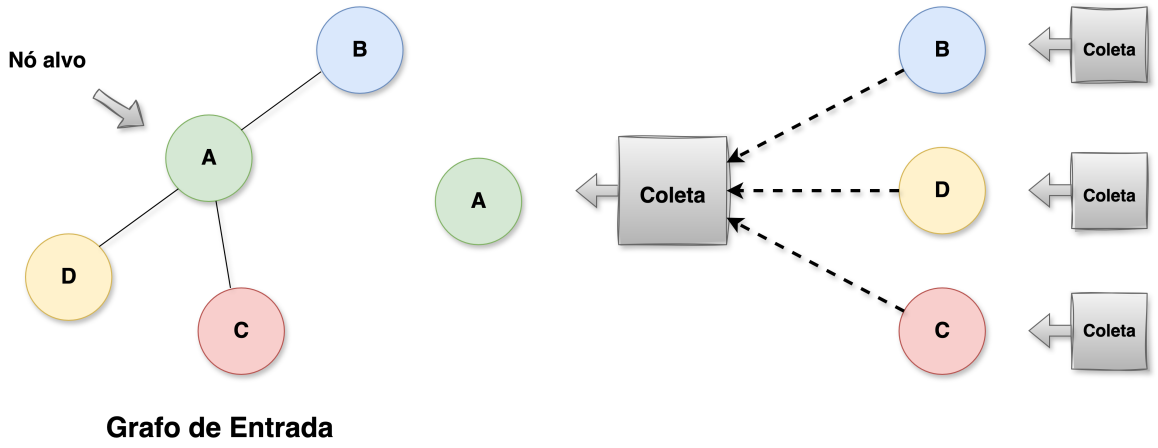


Figura 2.7: Transmissão de mensagens. Fonte: Elaboração própria.

Onde

$$h_u^{(k+1)} = g^{(k)}\left(h_u^{(k)}, m_{N(u)}^{(k)}\right).$$

No fim de K iterações, definimos o vetor imerso final do modelo conforme a Equação 2.3.

$$z_u = h_u^{(K)}, \quad \forall u \in V \tag{2.3}$$

Passagem de Informações aplicada aos Grafos

Após a definição dos conceitos de passagem de informações, exploramos agora sua aplicação em estruturas de grafos.

Como se trata de um modelo de aprendizado de máquina, é necessário incorporar parâmetros treináveis. Para isso, introduzimos uma matriz de pesos W, composta por dois termos: W_a, associado ao próprio nó u, e W_b, relacionado aos seus vizinhos.

Seguindo a formulação de Hamilton (2020), a coleta da mensagem $m_{N(u)}^{(k)}$ será obtida somando os vetores imersos $h_v^{(k)}$ de todos os nós vizinhos do nó de interesse u conforme Equação 2.4:

$$m_{N(u)}^{(k)} = \sum_{v \in N(u)} h_v^{(k)} \tag{2.4}$$

Dessa forma, podemos definir uma estrutura cuja aplicação em grafos torna-se viável (Equação 2.5).

$$h_u^{(k)} = W_a^{(k)} h_u^{(k-1)} + W_b^{(k)} \sum_{v \in N(u)} h_v^{(k-1)} \tag{2.5}$$

O primeiro termo da Equação 2.5 refere-se ao nó de interesse e sua matriz de parâmetros associada, enquanto o segundo corresponde aos seus vizinhos. A matriz de características X também está implícita nesse modelo, uma vez que $h_u^{(0)} = x_u, \forall u \in V$.

No entanto, a Equação 2.5 é linear e, portanto, poderia ser tratada como um modelo de regressão. Para torná-la não linear e justificar seu uso em aprendizado profundo, introduzimos uma função de ativação σ, resultando na Equação 2.6:

$$h_u^{(k)} = \sigma\left(W_a^{(k)} h_u^{(k-1)} + W_b^{(k)} \sum_{v \in N(u)} h_v^{(k-1)}\right) \tag{2.6}$$

Podemos ainda simplificar o modelo da Equação 2.6 adicionando o nó de interesse aos seus vizinhos na função de coleta de mensagem, resultando no modelo com auto-laço (Equação 2.7).

$$h_u^{(k)} = \sigma\left(\sum_{v \in N(u) \cup \{u\}} W^{(k)} h_v^{(k-1)}\right) \tag{2.7}$$

Essa abordagem proporciona a eliminação da necessidade de dividir a matriz de parâmetros W em duas matrizes distintas e a exclusão total de um termo na expressão. No entanto, é preciso cautela, pois aqui não é mais possível diferenciar quais informações provêm dos vizinhos do nó e quais são do próprio nó.

Normalização de Vizinhanças

Segundo Hamilton (2020), um dos desafios no modelo de RCG está na coleta de mensagens, especialmente quando os nós possuem diferentes quantidades de vizinhos, o que é comum em grafos reais.

Vamos considerar um cenário fictício onde um grafo representa uma rede de colaboração científica, em que cada nó representa um pesquisador e cada aresta, uma coautoria. Suponha que o pesquisador u tenha colaborado com cinco coautores, enquanto v colaborou com 50. Ao somar as informações dos vizinhos, a contribuição de v tende a ser numericamente maior, não necessariamente por ser mais relevante, mas por refletir a extensão de sua rede. Isso pode distorcer a interpretação da informação agregada.

Para mitigar esse efeito, aplica-se uma normalização na função de coleta, baseada nos graus dos nós. Utilizamos a normalização simétrica proposta por Kipf and Welling (2016), que pondera as contribuições dos vizinhos considerando o grau dos nós envolvidos.

Assim, a função de coleta de informações, representada por $mN(u)$, para o pesquisador u é calculada como a soma ponderada das contribuições dos pesquisadores vizinhos, levando em consideração a normalização simétrica, conforme apresentado na Equação 2.8,

onde $N(u)$ representa a vizinhança do pesquisador u, h_v é o vetor de informações do pesquisador v, e $|N(u)|$ e $|N(v)|$ são os graus dos nós u e v, respectivamente.

$$mN(u) = \sum_{v \in N(u)} \frac{h_v}{\sqrt{|N(u)| \cdot |N(v)|}} \tag{2.8}$$

Essa abordagem mitiga distorções causadas pela discrepância no número de colaborações, tornando a coleta de informações proporcional à relevância das contribuições e sensível à estrutura da rede.

Essa formulação, que integra normalização simétrica, auto-laço e função de ativação não linear, foi proposta originalmente por Kipf and Welling (2016) e é amplamente adotada como base para modelos modernos de RCGs.

Funções de Ativação

A escolha adequada da função de ativação é essencial para o desempenho de modelos de redes neurais, pois define como as informações fluem entre os neurônios das diferentes camadas da rede. Essas funções podem ser lineares ou não lineares, limitadas ou infinitas, e sua seleção depende tanto da tarefa (como predição ou classificação) quanto da arquitetura utilizada, como RNC, RNR ou RCG.

Neste trabalho, são avaliadas três funções de ativação:

- Função Linear;
- Função Sigmoide; e
- Função Unidade Linear Retificada ou *Rectified Linear Unit* (ReLU).

Cada uma apresenta características distintas e pode impactar de forma diferente o desempenho do modelo de RCG. Embora a ReLU seja amplamente utilizada, é importante considerar os contextos em que as demais podem ser vantajosas.

- Função Linear – Definida por $f(x) = cx$, onde c é constante, a função linear é a mais simples e não introduz não-linearidades. Por isso, é pouco eficaz para modelar padrões complexos, sendo mais adequada para camadas de saída em problemas de regressão.
- Função Sigmoide – A função sigmoide é expressa por $f(x) = \frac{1}{1+e^{-x}}$, mapeando a entrada para o intervalo (0, 1). É útil para classificação binária, mas sofre com o problema do desvanecimento do gradiente em redes profundas, dificultando o treinamento eficiente.

- Função ReLU – A função ReLU, dada por $f(x) = \max(0, x)$, ativa apenas entradas positivas e zera as negativas. Sua simplicidade e eficiência a tornam ideal para camadas ocultas, mitigando o desvanecimento do gradiente e favorecendo a aprendizagem de representações mais profundas e discriminativas.

Empilhamento de Camadas

O empilhamento de camadas é uma característica essencial das RCG, permitindo ao modelo capturar relações cada vez mais complexas e aprender representações hierárquicas dos dados em grafos. A cada camada, os vetores dos nós são atualizados por meio de convolução e normalização, incorporando informações de vizinhos cada vez mais distantes.

Esse processo amplia a capacidade de generalização do modelo e possibilita a representação de contextos tanto locais quanto globais. Com isso, as RCGs realizam aprendizado *end-to-end*, ajustando simultaneamente as representações dos nós e a tarefa final (Defferrard et al., 2016; Kipf and Welling, 2016).

Esse empilhamento tem demonstrado vantagens significativas em problemas de aprendizado de grafos e têm sido amplamente adotadas em diversas aplicações, como análise de redes sociais, sistemas de recomendação e outros.

Implementação

A implementação de modelos baseados em RCGs pode assumir diferentes configurações, dependendo da tarefa a ser resolvida, como classificação supervisionada, aprendizado semi-supervisionado ou estratégias auto-supervisionadas. Em conformidade com a proposta original de Kipf and Welling (2016) e os desdobramentos apresentados por Hamilton (2020), a estrutura típica de uma RCG envolve a aplicação sucessiva de camadas convolucionais sobre o grafo, intercaladas por funções de ativação não lineares e por mecanismos de agregação de vizinhos normalizados.

Em tarefas de classificação de nós, é comum empregar uma função de perda baseada na log-verossimilhança negativa, combinada com a função *softmax*, que transforma a saída da rede em uma distribuição de probabilidade entre as classes. A Equação 2.9 define a função *softmax* aplicada ao vetor imerso z_u de um nó u. Seja $z_{u,i}$ a i-ésima componente do vetor z_u, $y_u \in \{1, \ldots, c\}$ o índice da classe verdadeira do nó u, e c o número total de classes:

$$\text{softmax}(z_u)_i = \frac{e^{z_{u,i}}}{\sum_{j=1}^{c} e^{z_{u,j}}} \tag{2.9}$$

em que $z_u \in \mathbb{R}^c$ é o vetor de saída do nó u antes da normalização, e $\text{softmax}(z_u)_i$ é a probabilidade do nó pertencer à classe i.

Com base nessa probabilidade, define-se a função de perda L, que calcula a log-verossimilhança negativa para todos os nós rotulados no conjunto de treinamento V_{treino}, conforme a Equação 2.10:

$$L = \sum_{u \in V_{\text{treino}}} -\log\left(\text{softmax}(z_u)_{y_u}\right) \tag{2.10}$$

em que $y_u \in \{1, \ldots, c\}$ representa a classe verdadeira do nó u, e $\text{softmax}(z_u)_{y_u}$ corresponde à probabilidade prevista para a classe correta.

Entretanto, nem toda aplicação de RCG exige a predição direta de rótulos. Em cenários voltados à geração de representações vetoriais mais expressivas, como em tarefas de agrupamento ou análise de similaridade, pode-se adotar uma função de perda de reconstrução que minimize a diferença entre os vetores imersos finais z_u e os vetores de entrada x_u. Nesse contexto, uma alternativa viável é o uso do erro quadrático médio (Erro Quadrático Médio ou *Mean Squared Error* (MSE)) como função de perda, permitindo que o modelo aprenda representações enriquecidas a partir da estrutura do grafo, mesmo na ausência de rótulos explícitos.

Essa configuração corresponde a um aprendizado auto-supervisionado, no qual o sinal de supervisão é extraído da própria estrutura dos dados, sem necessidade de anotação externa (Zhu et al., 2020). Além disso, trata-se de um regime transdutivo, pois todos os nós do grafo estão disponíveis durante o treinamento, ainda que nem todos sejam utilizados no cálculo direto da função de perda (Hamilton, 2020).

Independentemente da função de perda adotada, a arquitetura da rede pode ser composta por múltiplas camadas convolucionais empilhadas, intercaladas por funções de ativação como a ReLU, e finalizadas por uma camada linear. A otimização dos parâmetros é realizada por métodos baseados em gradiente, com taxa de aprendizado ajustável conforme o problema.

Ao final do treinamento, os vetores z_u obtidos constituem representações aprendidas dos nós, que podem ser aplicadas em tarefas posteriores como classificação, agrupamento ou recomendação.

2.5 Métricas de Avaliação

Para avaliar o desempenho da solução proposto na tarefa de AND, são utilizadas três métricas amplamente adotadas na literatura, conforme destacado por Ferreira et al.

(2020): *Pairwise F1* ($pF1$), *K-Metric* e *B-Cubed*. Cada uma dessas métricas captura diferentes aspectos da qualidade da solução.

A métrica $pF1$ é voltada à avaliação par a par, verificando se pares de documentos foram corretamente atribuídos ao mesmo autor. Já a K-Metric e a B-Cubed são métricas de agrupamento, utilizadas para mensurar a qualidade global das partições geradas. A K-Metric considera simultaneamente a pureza e a completude dos agrupamentos. Por sua vez, a B-Cubed realiza uma análise local mais refinada, penalizando tanto o sobreagrupamento quanto a fragmentação indevida.

Pairwise F1 A *Pairwise Precision* (pP) mede a proporção de pares de documentos corretamente agrupados entre todos os pares atribuídos ao mesmo autor:

$$pP = \frac{\text{Pares Corretamente Classificados}}{\text{Total de Pares Agrupados}}. \tag{2.11}$$

A *Pairwise Recall* (pR) quantifica a capacidade do *framework* de recuperar todos os pares de documentos que realmente pertencem ao mesmo autor:

$$pR = \frac{\text{Pares Corretamente Recuperados}}{\text{Total de Pares Reais do Mesmo Autor}}. \tag{2.12}$$

A pontuação $pF1$ corresponde à média harmônica entre pP e pR:

$$pF1 = \frac{2 \cdot pP \cdot pR}{pP + pR}. \tag{2.13}$$

K-Metric A *K-Metric* é uma métrica composta baseada na média geométrica entre duas submétricas: a *Average Cluster Purity* (ACP) e a *Average Author Purity* (AAP) (Santana et al., 2017). Essa métrica avalia, de forma global, a qualidade do agrupamento, considerando tanto a pureza quanto a completude dos agrupamentos.

A ACP avalia a pureza de cada *cluster* previsto, ou seja, o quanto seus documentos pertencem a um mesmo autor real:

$$ACP = \frac{1}{N} \sum_{i=1}^{e} \sum_{j=1}^{t} \frac{n_{ij}^2}{n_i}, \tag{2.14}$$

em que:

- N é o número total de documentos;
- e é o número de *clusters* previstos;
- t é o número de *clusters* reais (autores reais);
- n_{ij} é o número de documentos compartilhados entre o *cluster* previsto i e o real j;

- n_i é o número total de documentos no *cluster* previsto i.

A AAP avalia a coesão dos documentos de um autor real dentro dos *clusters* previstos:

$$AAP = \frac{1}{N} \sum_{j=1}^{t} \sum_{i=1}^{e} \frac{n_{ij}^2}{n_j}, \tag{2.15}$$

em que n_j representa o número de documentos do autor real j.

A K-Metric é definida como a média geométrica entre ACP e AAP:

$$K = \sqrt{ACP \cdot AAP}. \tag{2.16}$$

B-Cubed A métrica *B-Cubed*, proposta por Bagga and Baldwin (1998), foi originalmente concebida para avaliação de resoluções de co-referência, mas tem sido amplamente empregada em tarefas de AND (Ferreira et al., 2020).

Esta métrica calcula a precisão e a revocação finais com base na precisão (P_r) e na revocação (R_r) de cada registro de autoria r, definidas da seguinte forma:

$$P_r = \frac{n_i^r}{n_i}, \qquad R_r = \frac{n_i^r}{n_j},$$

em que:

- n_i^r é o número total de registros de autoria que se referem ao mesmo autor associado a r e pertencem ao mesmo *cluster* empírico i (ou seja, o *cluster* previsto) que contém r;
- n_i é o número total de registros de autoria no *cluster* empírico i que contém r;
- n_j é o número total de registros de autoria no *cluster* teórico j (ou seja, o *cluster* de referência) que contém r.

A precisão (bP) e a revocação (bR) finais são calculadas por:

$$bP = \sum_{r=1}^{N} w_r P_r, \qquad bR = \sum_{r=1}^{N} w_r R_r,$$

em que N é o número total de registros de autoria na coleção e w_r é o peso atribuído ao registro r, comumente definido como $w_r = \frac{1}{N}$.

A pontuação F1 do B-Cubed (bF_α) é a média harmônica ponderada entre bP e bR, definida por:

$$bF_\alpha = \frac{1}{\alpha \cdot \frac{1}{bP} + (1-\alpha) \cdot \frac{1}{bR}}.$$

Quando $\alpha = 0,5$, tem-se a média harmônica padrão entre bP e bR. Essa métrica penaliza tanto o agrupamento excessivo quanto a fragmentação, sendo especialmente útil para avaliar a consistência local do agrupamento na tarefa de AND.

Capítulo 3

Revisão da Literatura

Em consequência da crescente quantidade de publicações e repositórios bibliográficos, houve a necessidade de novos métodos e técnicas para AND. Especialmente nos últimos anos, muitos métodos foram propostos para resolução desse problema, tais como similaridade de palavras, baseados em aprendizado de máquina e outras técnicas de IA.

Neste capítulo será apresentada uma revisão da literatura na área de AND utilizando o método da Teoria do Enfoque Metaanalítico Consolidado (TEMAC) (Mariano and Rocha, 2017). Ao final deste capítulo, será apresentada uma visão geral das pesquisas na área, incluindo uma tabela com artigos publicados entre 2020 e 2025.

3.1 Método

Tradicionalmente, revisões sistemáticas da literatura concentram-se em acessar vários repositórios bibliográficos digitais para enriquecer as descobertas da revisão (Kitchenham, 2004; Kitchenham et al., 2009). Neste trabalho, uma abordagem meta-analítica consolidada foi escolhida para realizar a revisão de literatura devido à vantagem metodológica, uma vez que seu processo reduz o acesso a repositórios bibliográficos digitais científicos privados, minimizando o viés e maximizando a possibilidade de cobertura.

O TEMAC surgiu como uma solução exploratória, apoiada em estratégias anteriores e fundamentada em princípios bibliométricos, concentrando-se na necessidade de unificar vários métodos sistemáticos com uma estrutura meta-analítica com publicações recentes (Mariano et al., 2019; Vera-Olivera et al., 2021). Além disso, estudos de revisão meta-analítica na área de AND ainda não foram conduzidos, marcando um ponto de partida para futuras pesquisas que contribuirão para o corpo de conhecimento dos estudos de revisões sistemáticas existentes.

O método do TEMAC inclui três etapas, preparação da pesquisa (Etapa 1), apresentação e inter-relação dos dados (Etapa 2), e detalhamento, modelo integrador e validação por evidências (Etapa 3), as quais serão descritas na sequência.

Etapa 1 - Preparação da Pesquisa

A etapa de preparação da revisão é vital, uma vez que escolhas erradas geram resultados insatisfatórios, por exemplo, palavras de pesquisa inadequadas. Mariano et al. (2019) afirma que uma das etapas essenciais em estudos de revisão é a leitura de artigos para definir critérios específicos para inclusão e exclusão de estudos. Desta forma, foram definidos como critérios específicos para inclusão (CI) e exclusão (CE) na seleção de trabalhos acadêmicos:

- CI-1: Aborda principalmente AND como componente integral do estudo;
- CI-2: Publicado em conferências ou periódicos revisados por pares, disponíveis em importantes repositórios bibliográficos;
- CE-1: Trabalhos que não estão disponíveis em repositórios online;
- CE-2: Associados principalmente a domínios diferentes de Sistemas de Informação, Ciência da Computação e Engenharia;
- CE-3: Publicados fora do período de 2003-2022.

Nesta revisão da literatura foram utilizados os repositórios bibliográficos *Web Of Science* (WoS) e *Scopus*, ambos reconhecidos em várias comunidades acadêmicas. O espaço-tempo da pesquisa é outro fator essencial, pois os bancos de dados têm restrições de tempo distintas. As áreas de conhecimento são extraídas por meio da leitura dos trabalhos relacionados ao tema de AND. Dessa forma, durante a etapa de preparação da pesquisa, foram respondidas as quatro perguntas que se seguem.

1. Qual é o descritor, *strings* de busca ou palavras-chave da pesquisa?
   O descritor *"author name disambiguation"* foi usado sem conectivos lógicos (e/ou), para incluir estudos em diferentes contextos nos repositórios bibliográficos.
2. Quais são as bases de dados?
   Os repositórios bibliográficos *Web of Science* (WoS) e Scopus, valorizados em várias comunidades acadêmicas, foram escolhidos devido ao fato de que as obras dentro destes emanam de conferências ou periódicos revisados por pares. Outro aspecto importante refere-se a ampla cobertura temporal do WoS e o abrangente escopo de periódicos de Ciência e Tecnologia na base Scopus. Esses repositórios são complementares e amplamente utilizados em revisões de literatura.

3. Qual é o campo espaço-temporal da pesquisa?
   A delimitação temporal é crucial, uma vez que os bancos de dados possuem coberturas temporais variadas. Esses documentos abrangeram o período de 2003, que marcou o primeiro trabalho sobre AND na *Web*, até 2022.

4. Quais são as áreas de conhecimento?
   Determinamos as áreas de conhecimento após examinar os documentos incluídos nas bases WoS e Scopus. Após essa análise, as áreas foram categorizados em Ciência da Computação, Ciências Sociais e da Informação, Medicina, Engenharia e Matemática. A Tabela 3.1 apresenta as áreas de conhecimento relacionadas a AND em ambos repositórios.

Acreditamos que a utilização de uma abordagem de revisão meta-analítica possa estabelecer uma base adequada para a condução de um estudo exploratório na área de pesquisa de AND, garantindo a inclusão de trabalhos relevantes para a construção de um conhecimento atualizado.

Etapa 2 - Apresentação e Inter-relação dos Dados

A apresentação dos trabalhos e os dados de inter-relação são baseados na abordagem meta-analítica consolidada. A abordagem inclui técnicas quantitativas e aspectos bibliométricos baseados em três leis.

- A "Lei de Bradford", proposta por Brookes (1969), permite encontrar periódicos que publicam mais sobre um determinado tópico. Os periódicos científicos de uma área devem ser ordenados de maneira decrescente de acordo com sua produtividade, gerando núcleos nos quais alguns periódicos geralmente respondem por uma parcela significativa das publicações totais. Enquanto um grande número de periódicos publica menos artigos na área (Heradio et al., 2020). Esta lei também mede a dispersão bibliográfica, ou seja, o quanto o conhecimento está disperso em diferentes periódicos. A "Lei de Bradford"é calculada a partir dos n periódicos que mais publicaram artigos sobre o assunto, formando o núcleo. A medida que nos afastamos do núcleo, observa-se uma proporção crescente de artigos nas zonas subsequentes $1{:}n{:}n^2{:}n^3$. No contexto deste estudo, a Lei de Bradford facilita citar um número limitado de periódicos científicos na área de AND, os quais coletivamente respondem por uma parte substancial do total de publicações.

- A "Lei do Elitismo"ou "Lei do Príncipe"deriva da "Lei de Lotka"(Lotka, 1926), um dos modelos mais discutidos em bibliometria, que afirma que o número de autores fazendo n contribuições é aproximadamente $1/n^2$ daqueles que fazem uma única

Tabela 3.1: Áreas de conhecimento nos repositórios da WoS e Scopus.

WoS	Scopus
Information Science & Library Science	*Computer Science*
Computer Science Information Systems	*Social Sciences*
Computer Science Interdisciplinary Applications	*Mathematics*
Computer Science Theory Methods	*Engineering*
Computer Science Artificial Intelligence	*Decision Sciences*
Engineering Electrical Eletronic	*Medicine*
Computer Science Software Engineering	*Multidisciplinary*
Multidisciplinary Sciences	*Bussiness, Management and Accounting*
Telecommunications	*Arts and Humanities*
Computer Science Hardware Architecture	*Materials Sciences*
Medical Informatics	*Agricultural and Biological Sciences*
Health Care Sciences Services	*Biochemistry, Genetics and Molecular Biology*
Mathematics Interdisciplinary Applications	*Energy*
Medicine General Internal	*Neuroscience*
Operations Research Management Science	*Physics and Astronomy*
Physics Multidisciplinary	
Bussiness	
Cardiac Cardiovascular Systems	
Computer Science Cybernetics	
Education Educational Research	
Education Scientific Disciplines	
Engineering Mechanical	
Management	
Mathematical Computational Biology	
Medicine Research Experimental	
Physics Fluid Plasmas	
Physics Mathematical	
Regional Urban Planning	
Social Sciences Mathematical Methods	
Statistics Probability	

Fonte: Elaboração própria.

publicação. As contribuições de autores que fazem uma única contribuição representam cerca de 60% de toda a publicação em um campo específico. A Lei do Elitismo busca revelar os autores e artigos mais importantes (mais citados), empregando a

raiz quadrada do número total de autores para destacar o que é considerado uma elite. No contexto deste estudo, se n representa o número total de autores, $\sqrt{n}$ representaria a elite da área estudada. Neste estudo, os autores mais citados revelam os autores e documentos mais importantes responsáveis por mais da metade das contribuições na área de AND.

- A "Lei 80/20"(regra de Pareto) é inspirada nos sistemas de informação usados no comércio e na indústria, em que 80% da demanda de informações é atendida por 20% do conjunto de fontes de informação (Trueswell, 1969). Essa lei busca periódicos, países, universidades e áreas mais relevantes que publicam mais e a escolha de palavras-chave mais representativas.

A apresentação e a inter-relação dos dados usando a abordagem meta-analítica consolidada permitem uma revisão dos autores e citações mais relevantes, periódicos, países, organizações ou universidades e áreas de conhecimento mais relacionadas ao campo de pesquisa. Para realizar a revisão da análise de dados, foi utilizada a ferramenta bibliométrica VOSViewer (VOSviewer, 2022) e o BiblioTools (Grauwin, 2022).

A ferramenta VOSviewer, utilizada para visualização e análise de redes bibliométricas, oferece informações sobre padrões, relacionamentos e tendências na literatura de pesquisa e também permite a criação de representações visuais de dados bibliométricos. A visualização de redes de coautoria e cocitação, juntamente com *clusters* de pesquisa, autores influentes e novas direções de pesquisa, ajuda a revelar relacionamentos entre autores e documentos em revisões bibliométricas. Em resumo, a ferramenta VOSviewer auxilia na exploração de grandes conjuntos de dados bibliométricos, contribuindo para a compreensão do panorama dos campos de pesquisa.

O BiblioTools é um conjunto de *scripts* na linguagem *Python* para análise bibliométrica integrável a diferentes repositórios digitais, com diversas funções, como mineração de dados, processamento de dados, análise de dados, visualização de palavras-chave e geração automatizada de relatórios. O BiblioTools possibilita refinar e limpar dados brutos com um *script* de pré-processamento, preparando o conjunto de dados para análise. Exploramos nossos dados, produzindo diferentes redes de coocorrência, como co-palavras, coautores e cocitações. Esse conjunto também permite a visualização de redes e *clusters* de acoplamento bibliográfico, fornecendo informações sobre publicações, autores e conexões de tópicos de pesquisa.

Dessa forma, utilizando o BiblioTools e o VOSviewer, extraímos informações conforme segue:

1. Uma análise dos periódicos com o maior número de documentos sobre o tema;
2. Evolução das publicações em periódicos e conferências por ano;

3. Autores que mais publicaram *versus* autores mais citados;
4. Os países que mais publicaram;
5. Organizações ou universidades que mais publicaram;
6. Áreas de conhecimento que mais publicam;
7. Frequência de palavras-chave.

Etapa 3 - Detalhamento, Modelo Integrador e Validação por Evidências

Na terceira etapa, análises mais profundas permitem uma melhor compreensão do tema abordado, selecionando os principais autores, abordagens, linhas de pesquisa e validação por evidências, com uma comparação dos resultados das diferentes bases de dados utilizadas.

Essa evidência é obtida com a análise de cocitações e mapas de acoplamento bibliográfico. O método de cocitação conecta diferentes autores e documentos com base em sua aparição conjunta nas listas de referências obtidas nos repositórios bibliográficos. Por outro lado, o método de acoplamento bibliográfico conecta autores e documentos com base no número de referências que compartilham entre si. Em outras palavras, enquanto a cocitação apresenta trabalhos constantemente citados em conjunto e pode mostrar semelhanças entre os estudos, o acoplamento bibliográfico usa a premissa de que os trabalhos que citam os mesmos artigos têm contextos semelhantes e indicam as frentes de pesquisa atuais usando um espaço-tempo atualizado.

As análises de cocitações e acoplamento bibliográfico são comumente utilizadas em revisões sistemáticas (Ankrah et al., 2022; Crispim et al., 2022; Garakhanova, 2023; Khider et al., 2023; Müller, 2023). Elas desempenham funções importantes, como revelar as principais abordagens de pesquisa, estabelecer frentes de pesquisa e apontar direções futuras (Mariano and Rocha, 2017). Ao estabelecer uma ligação entre referências (passado) e trabalhos mais proeminentes (futuro), elas desempenham a função de *snowballing* de maneira automatizada. Assim, por meio da cocitação, é possível compreender as principais abordagens do passado, enquanto o acoplamento bibliográfico identifica os principais estudos atuais. Além disso, a nuvem de palavras-chave é essencial para revelar linhas de pesquisa, demonstrando as diferentes aplicações em áreas específicas (Correa and Cruz, 2005). A nuvem de palavras-chave geralmente é construída usando a frequência das palavras-chave e pode ser aprimorada pela coocorrência dessas palavras.

Uma representação genérica da análise de cocitação e acoplamento bibliográfico é apresentada na Figura 3.1.

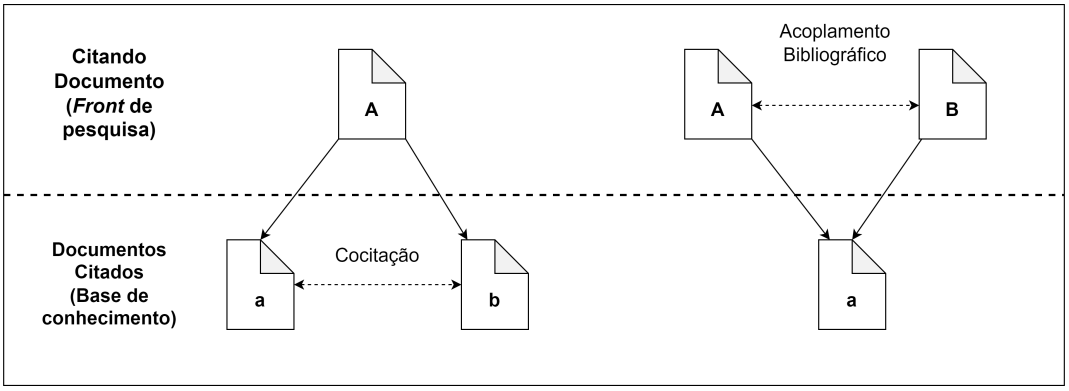


Figura 3.1: Cocitação e acoplamento bibliográfico. Fonte: Adaptado de Vogel and Güttel (2013).

3.2 Resultados

De acordo com a taxonomia proposta por Ferreira et al. (2012, 2020) e apresentada na Figura 3.2, podemos classificar os métodos de AND de acordo com o tipo de abordagem e a evidência explorada. Existem vários tipos de abordagens classificadas em categorias como Agrupamento de Autor e Atribuição de Autor. As evidências podem ser classificadas como: informação de citação, informação da *Web* e evidência implícita.

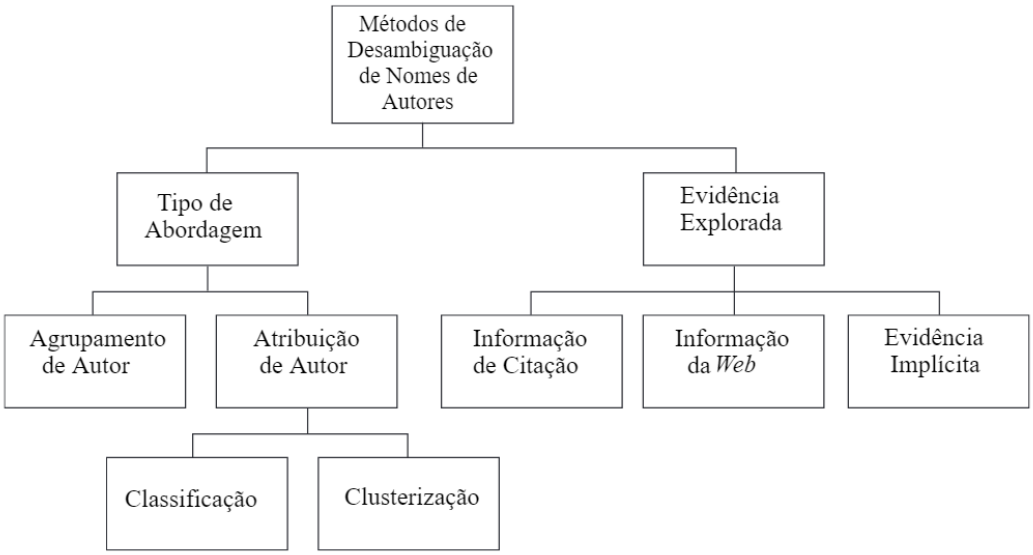


Figura 3.2: Taxonomia de AND. Fonte: Traduzido de Ferreira et al. (2012, 2020).

Os resultados da pesquisa nas bases de dados retornaram 197 documentos no repositório da Scopus, dos quais 137 também estavam no *Web of Science* (WoS). A exportação de dados manteve os registros completos do trabalho, incluindo os campos *Author, Title, Abstract, Keywords, Addresses* e *Cited References*. Para uma ampla investigação de revisão da literatura, fizemos uma fusão com os documentos obtidos nos repositórios

WoS e Scopus, quando 14 documentos exclusivos compunham o WoS. A Figura 3.3 apresenta um Diagrama de Venn com 211 documentos recuperados em ambos os repositórios bibliográficos.

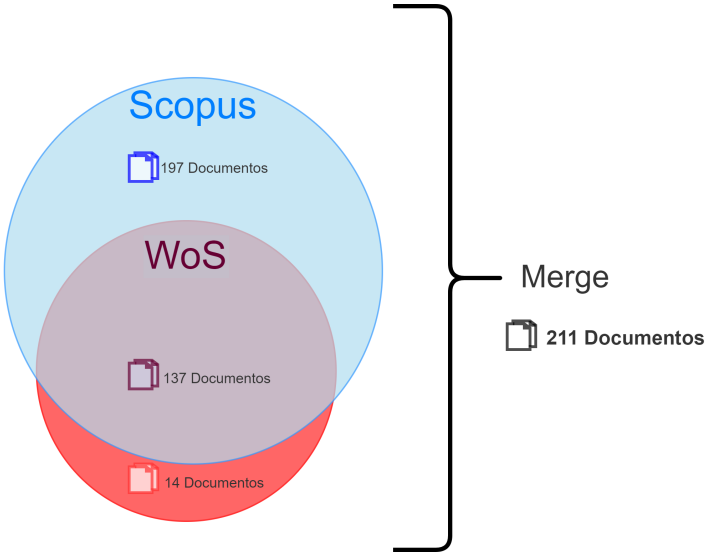


Figura 3.3: Merge dos documentos obtidos nos repositórios Scopus e WoS. Fonte: Elaboração própria.

Resultados da Etapa 2 - Apresentação e Inter-relação dos Dados

Nesta etapa, apresentamos a inter-relação de dados e informações quantitativas da revisão da literatura. A Tabela 3.2 apresenta os tipos de documentos nos repositórios WoS e Scopus. Na WoS, aproximadamente 53% (81) são artigos de periódico, 41% (62) artigos de conferência, 2% (3) revisão, 2,6% (4) *erratum* e acesso antecipado e 0,6% (1) documento de dados. De acordo com o repositório da Scopus, aproximadamente 45,6% (90) dos documentos são artigos de periódico, 43,1% (85) são artigos de conferência, o restante é dividido em revisões de conferências 7,1% (14), revisão 2,5% (5), capítulo de livro 0,5% (1), artigo de dados 0,5% (1) e *erratum* 0,5% (1). Considerando os documentos obtidos a partir dos repositórios mesclados, 46,4% (98) são artigos de periódico, 41,2% (87) artigos de conferência, 6,6% (14) revisões de conferências, 2,3% (5) revisões, 3,3% (7) são divididos em capítulos de livros, artigos de dados, *erratum* e acesso antecipado.

A Tabela 3.3 apresenta os periódicos com a maior quantidade de documentos publicados. O *Scientometrics*, *h-index* de 123 e *SJR* de 0,929 é o que tem mais publicações na área de AND, com 21 publicações no WoS e nos repositórios combinados, e 19 no respositório da Scopus. O restante dos periódicos tem 23 publicações no repositório mesclado. O *IEEE Access* tem o maior *h-index* com 158 e um *SJR* de 927, mas apenas quatro

Tabela 3.2: Tipos de documentos em cada repositório.

Tipo de Documento	WoS	Scopus	Mesclado
Artigo em Periódico	81	90	98
Artigo em Conferência	62	85	87
Revisão de Conferência	0	14	14
Revisão	3	5	5
Capítulo de Livro	0	1	1
Artigo de Dados	1	1	1
Erratum	2	1	3
Acesso Antecipado	2	0	2
Total	151	197	211

Fonte: Elaboração própria.

publicações no repositório mesclado. Também é possível verificar que, nesse conjunto de documentos, os periódicos relacionados à área de Ciência da Informação são a maioria.

Tabela 3.3: Periódicos com maior quantidade de documentos.

Nome do Periódico	*h-index*	SJR	WoS	Scopus	Mesclado
Scientometrics	123	0,929	21	19	21
Journal of the Association for Information Science and Technology	150	0,848	8	8	8
Journal of the American Society for Information Science and Technology	18	—	4	4	4
Journal of Information Science	69	0,761	3	3	3
Journal of Informetrics	77	1,437	4	3	4
IEEE Access	158	927	2	3	4

Fonte: Elaboração própria.

A distribuição de documentos entre as bases de dados é semelhante, nota-se que a maioria são artigos publicados em periódicos e conferências. A Figura 3.4 apresenta a evolução das publicações desses dois tipos de documentos por ano. Observe que, em todos os casos, os documentos de periódico e conferência se alternaram ao longo dos anos. Entretanto, considerando os anos de 2020 a 2022, a quantidade de artigos publicados em periódicos aumentou.

Conforme mostrado na Figura 3.5, as áreas de conhecimento da Ciência da Computação (34,1%), Ciências Sociais (13,8%) e Engenharia (10,8%) estão relacionadas a mais da metade do número total de documentos de AND na base de dados mesclada (58,7%).

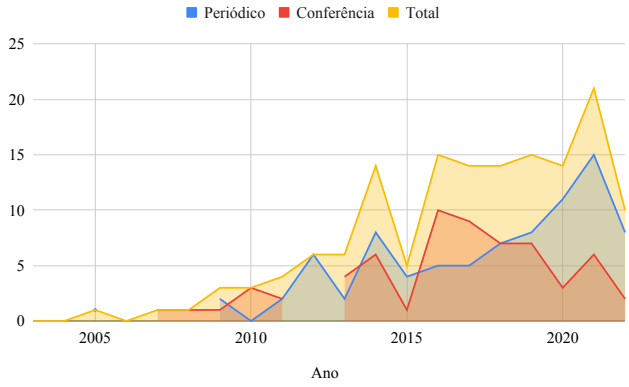
A liderança da Ciência da Computação pode estar relacionada ao fato de que AND é um problema em aberto na área, desencadeando métodos e abordagens para resolvê-lo. Embora alguns trabalhos usem bancos de dados relacionados ao domínio da Medicina (PubMed MEDLINE (2025)), essa área corresponde a 3,8% de todos os documentos.

Com base no método de revisão da literatura, é possível identificar os autores com o maior número de publicações e os mais citados. Conforme apresentado na Tabela 3.4, o número de documentos diminui à medida que o número de autores aumenta. Observando esse padrão, verificamos que 90% dos autores têm menos de cinco publicações, mas isso pode não se referir aos documentos mais importantes. Portanto, considerando os conjuntos de dados WoS, Scopus e base de dados mescladas, selecionamos os autores com pelo menos cinco trabalhos, representando a média entre o número de documentos e autores.

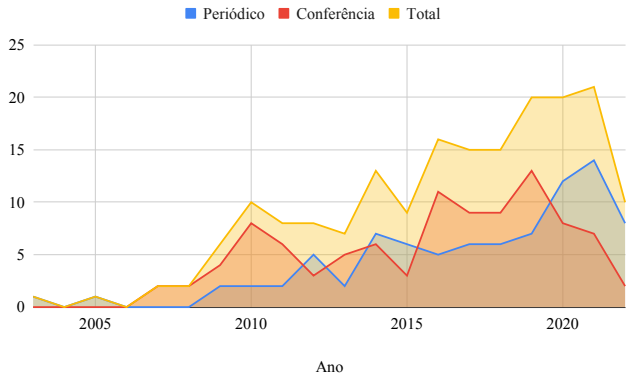
A seleção de um número menor, por exemplo, 4 ou 3, retorna um grande número de autores, o que dificulta uma análise mais detalhada dos artigos desta revisão. Com essa restrição, o repositório da WoS retornou oito autores, Scopus e os repositórios mesclados retornaram 13. Analisamos esses autores, comparando o número de documentos e citações.

Conforme mostrado na Tabela 3.5, o autor com o maior número de documentos no repositório da WoS foi Kim, J. (12). Entretanto, os resultados do Scopus e dos repositórios mesclados mostraram que Gonçalves, M. A. foi o autor com o maior número de documentos (13). Esse autor foi o mais citado na WoS (305). No entanto, com autores não identificados anteriormente no WoS, mas encontrados no Scopus e nos repositórios mesclados, foi verificado que Torvik, V. I. é o autor com mais citações (549).

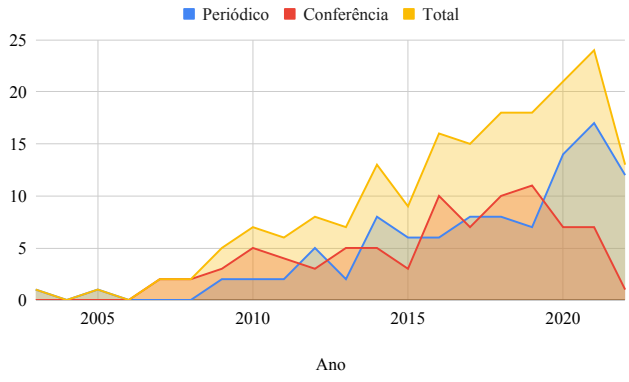
Conforme mostrado na Figura 3.6, foi computado o número de documentos e citações por país usando informações dos repositórios da WoS, Scopus e os repositórios mesclados. Seguindo a especificação de autores citados e a quantidade de documentos por autor (Tabela 3.5), filtramos os países da base de dados combinada com cinco ou mais publicações. Com relação aos documentos por região, os Estados Unidos da América (EUA), a China e a Alemanha lideram a lista de países que mais publicam, considerando os dois repositórios e o mesclado. No repositório mesclado, por exemplo, os Estados Unidos da América (EUA) têm 45 (21,3%) documentos, a China 41 (19,4%), a Alemanha 28 (13,2%) e o Brasil 18 (8,5%).



(a) WoS



(b) Scopus



(c) Repositórios mesclados

Figura 3.4: Evolução das publicações em periódicos e conferências por ano. Fonte: Elaboração própria.

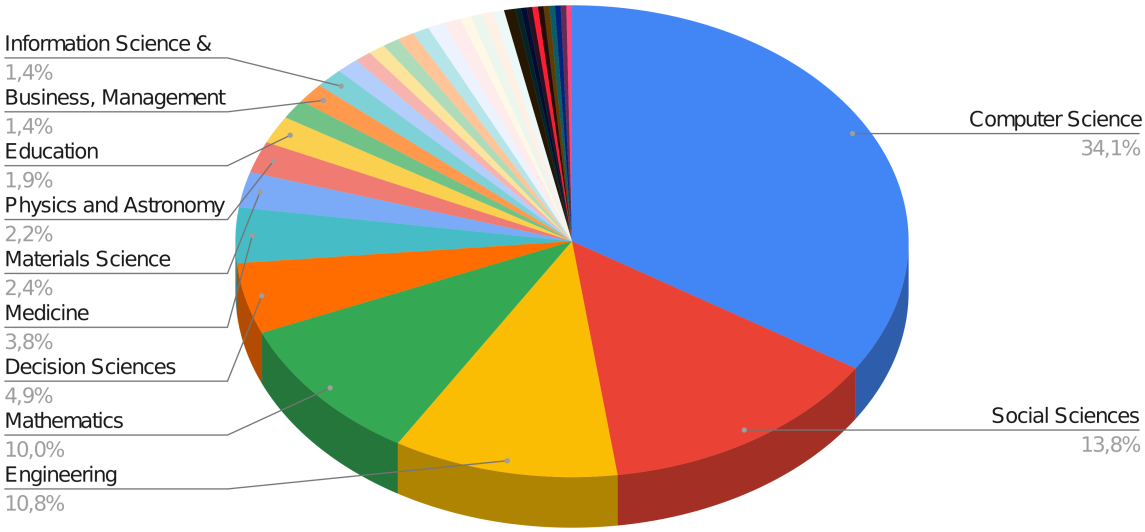


Figura 3.5: Distribuição de documentos por área de conhecimento nos repositórios mesclados (WoS e Scopus). Fonte: Elaboração própria.

Tabela 3.4: Quantidade de autores vs. quantidade de publicações nos repositórios mesclados (WoS e Scopus).

Quantidade de Documentos	Quantidade de Autores
13	2
12	1
11	1
7	1
6	2
5	6
4	8
3	13
2	53
1	351

Fonte: Elaboração própria.

De acordo com o gráfico na Figura 3.7, as informações sobre o número de citações mudam em comparação com o número de documentos. Os EUA, o Brasil e a China apresentam o maior número de citações. Os EUA lideram confortavelmente com 1.337

Tabela 3.5: Autores com mais documentos (Docs) e citações.

Autor	Docs (WoS)	Citações (WoS)	Docs (Scopus)	Citações (Scopus)	Docs (mesclado)	Citações (mesclado)
Gonçalves, M. A. Orcid: 0000-0002-2075-3363	9	305	13	507	13	507
Kim, J. Orcid: 0000-0001-6481-2065	12	85	13	166	13	166
Ferreira, A. A Orcid: 0000-0002-2487-6600	8	302	12	500	12	500
Laender, A. H. F. Orcid: 0000-0001-5032-2233	7	297	11	499	11	499
Asghar, S. Orcid: 0000-0001-6883-3584	6	43	6	72	7	72
Hussain, I. Orcid: 0000-0002-1586-1503	6	43	6	72	6	72
Smalheiser, N. R. Orcid: 0000-0003-1079-3406	3	306	6	524	6	524
Chandra, J. Orcid: 0000-0001-5994-9024	5	13	5	18	5	18
Giles, C. L. Orcid: 0000-0002-1931-585X	4	35	5	168	5	168
Mondal, S. Orcid: 0000-0002-2159-3410	5	13	5	18	5	18
Torvik, V. I. Orcid: 0000-0002-0035-1850	3	340	5	549	5	549
Veloso, A. Orcid: 0000-0002-9177-4954	3	69	5	166	5	166
Zhang, L. Orcid: 0000-0003-2104-0194	2	5	5	10	5	10

Fonte: Elaboração própria.

citações. Além disso, é possível verificar que o Brasil (510) tem mais citações de documentos do que a China (311), apesar de um número menor de documentos. O número de citações da China é muito próximo do número da Alemanha (51). No entanto, a Alemanha é o terceiro país que mais publica.

Foi filtrado o número de publicações por ano nos dois repositórios e no mesclado. Conforme proposto no método de revisão da literatura, os documentos selecionados foram de 2003 a 2022. Analisando a Figura 3.4 que utiliza o repositório da Wos, Scopus e mesclado, é possível observar um aumento de publicações na área de AND desde 2003, mas com uma diminuição de publicações em 2015, considerando que em 2014 houve crescimento. Também é importante observar que, mesmo com um cenário de pandemia mundial em 2020 e 2021, houve um crescimento de publicações em relação aos anos anteriores. Em

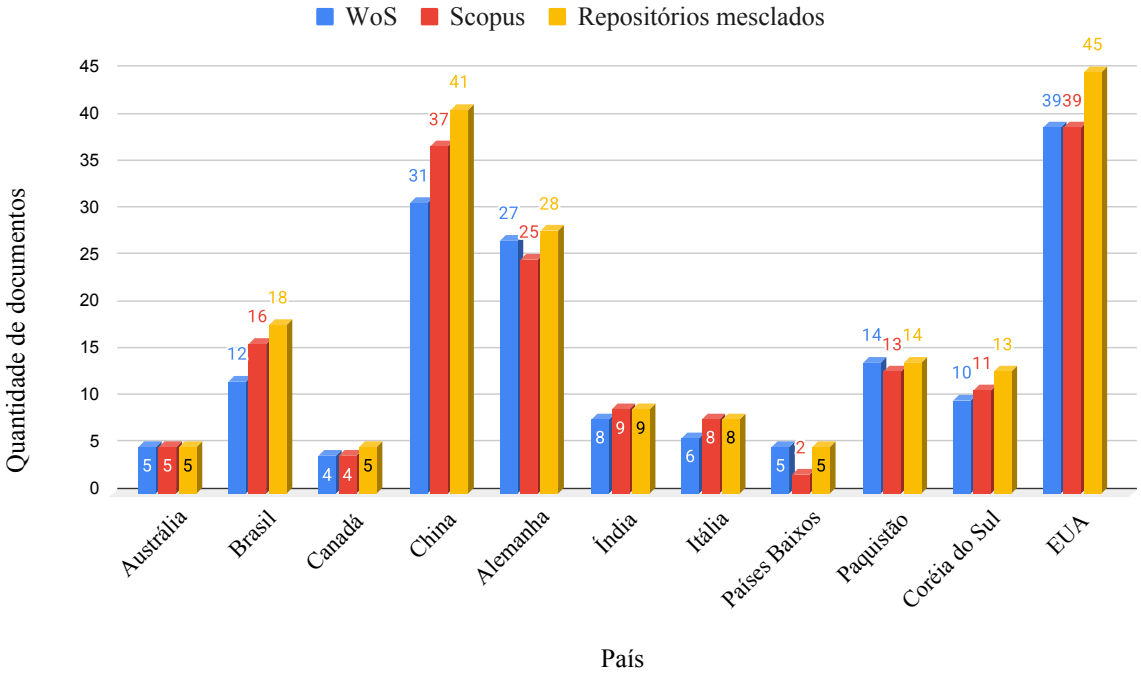


Figura 3.6: Documentos por país nos repositórios mesclados (WoS e Scopus). Fonte: Elaboração própria.

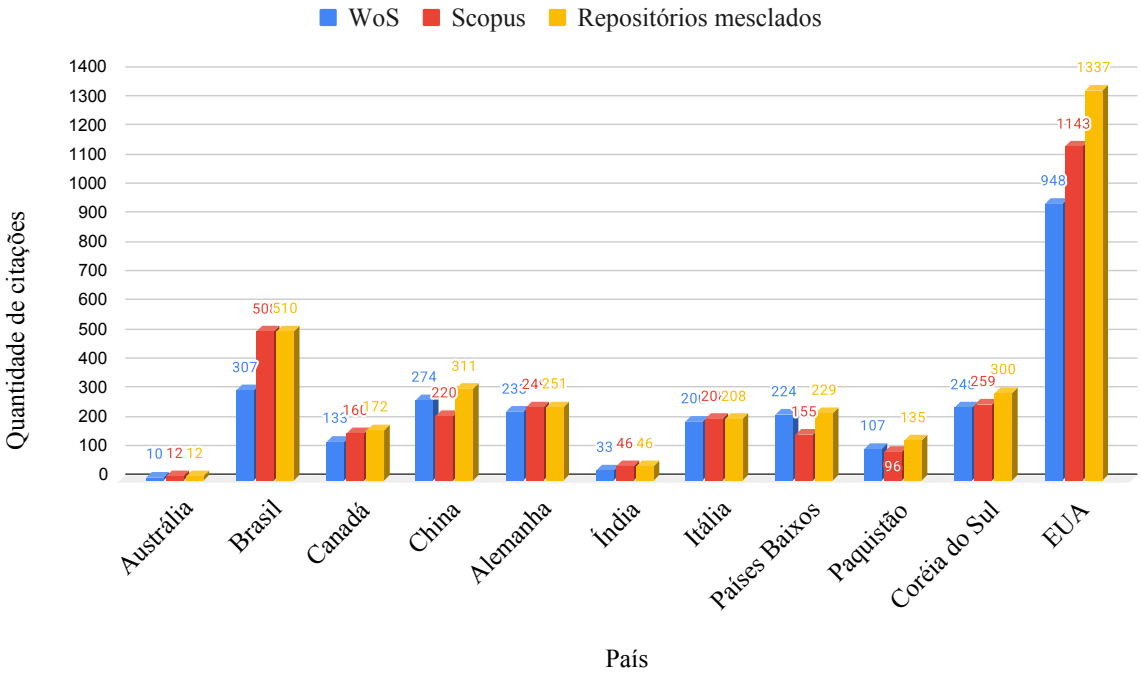


Figura 3.7: Citações por país nos repositórios mesclados (WoS e Scopus). Fonte: Elaboração própria.

2022, não obtivemos o número total de publicações, sendo necessária uma nova pesquisa em 2023 para validar o crescimento anual.

A mesma análise de publicações e citações de organizações que publicam estudos na área de AND foi realizada. Foram incluídas as organizações com mais de 20 citações em cada repositório (WoS e Scopus), conforme apresentado na Tabela 3.6. Considerando os repositórios mesclados, descobrimos que as organizações norte-americanas e brasileiras publicam regularmente com grande quantidade de citação de documentos. Verificamos que a maioria das publicações é feita em conjunto no Brasil, como, por exemplo, o Departamento de Ciência da Computação da Universidade Federal de Minas Gerais e o Departamento de Computação da Universidade Federal de Ouro Preto. Essas duas organizações têm 18 publicações e 682 citações. Diferentemente das organizações brasileiras, as organizações norte-americanas geralmente não publicam juntas. Entretanto, cada organização tem muitas publicações. A *Information Sciences School da Illinois University at Urbana-Champaign* tem 11 publicações e 498 citações. O *Institute for Research on Innovation & Science da Michigan University* tem 12 publicações com 192 citações.

Tabela 3.6: Documentos e citações por organização.

Organização	País	Documentos (WoS)	Citações (WoS)	Documentos (Scopus)	Citações (Scopus)	Documentos (mesclado)	Citações (mesclado)
School of Information Sciences University of Illinois at Urbana-Champaign	USA	11	482	2	58	11	498
Departamento de Ciência da Computação Universidade Federal de Minas Gerais	Brasil	9	305	7	359	10	392
Departamento de Computação Universidade Federal de Ouro Preto	Brasil	7	228	4	231	8	290
Institute for Research on Innovation Science, University of Michigan	USA	12	176	6	76	12	192
School of Information Management Wuhan University	China	4	65	4	28	7	115
Heidelberg Institute for Theoretical Studies GGMBH	Alemanha	4	58	4	65	4	65
Microsoft Research	USA	3	25	2	32	4	44
Mathematics Department Fiz Karlsruhe, Berlin	Alemanha	2	36	2	41	2	41
Computer Science and Engineering Pennsylvania State University, Univ. Park	USA	4	42	2	27	4	61

Fonte: Elaboração própria.

Usando os títulos e resumos dos documentos no repositório mesclado, geramos uma nuvem de palavras, conforme mostrado na Figura 3.8. A nuvem incluiu palavras relacionadas ao problema AND e às abordagens de solução, como *methods*, *data*, *clustering*, *information*, *network*, *learning*, *similarity*, *publications*, *model*, *libraries*, e *graph*. Na análise de cocitação e acoplamento bibliográfico, essas abordagens validam o uso recorrente dos métodos de AND.

Figura 3.8: Nuvem de palavras considerando os títulos dos documentos e resumos nos respositórios mesclados (WoS e Scopus). Fonte: Elaboração própria.

Resultados da Etapa 3 - Detalhamento, Modelo Integrador e Validação por Evidências

Os resultados da Etapa 3 da metodologia de revisão incluem a análise de cocitação, a análise de acoplamento bibliográfico e a visão geral das publicações na área de AND.

A Tabela 3.7 apresenta uma relação direta entre as referências mostradas nas Figuras 3.9 e 3.11, e suas representações correspondentes no texto, o que contribui para melhorar a legibilidade e clareza de nossos resultados.

Análise de cocitação

Na análise de cocitação da Figura 3.9, foi identificado quais autores são cocitados, indicando uma similaridade entre suas linhas de pesquisa e trabalhos. Há quatro pontos vermelhos escuros que representam núcleos de cocitação. A seguir, detalharemos os principais estudos de cada *cluster* e os classificaremos de acordo com a taxonomia usada nesta revisão da literatura, apresentada na Figura 3.2.

- *Cluster* 1: Este *cluster* é composto por dois trabalhos, Shin et al. (2014) e Ferreira et al. (2014), que utilizam informações de coautoria para a AND. Essa similaridade indica a proximidade no mapa de calor e justifica a alta cocitação do *cluster*. No entanto, a abordagem computacional para a AND é diferente. O trabalho de Shin et al. (2014) adota uma abordagem baseada em grafos construídos com relações

Tabela 3.7: Correspondência entre as referências citadas nas Figuras 3.9 e 3.11.

Análise de Cocitação (Figura 3.9)	Análise de Acoplamento (Figura 3.11)
Shin D., 2014 Shin et al. (2014) - *Cluster* 1	Zhang W., 2019a Zhang et al. (2019) - *Cluster* 1
Ferreira A. A., 2014 Ferreira et al. (2014) - *Cluster* 1	Kim K., 2019a Kim et al. (2019b) - *Cluster* 1
Kim J., 2018 Kim et al. (2019a) - *Cluster* 2	Xu J., 2020 Xu et al. (2020) - *Cluster* 2
Kim J., 2019 Kim (2019) - *Cluster* 2	Colavizza G., 2020 Colavizza et al. (2020) - *Cluster* 3
Levin M., 2012 Levin et al. (2012) - *Cluster* 3	
Ferreira A. A., 2012 Ferreira et al. (2012) - *Cluster* 3	
Cota R. G., 2010 Cota et al. (2010) - *Cluster* 3	
Smalheiser N. R., 2009 Smalheiser and Torvik (2009) - *Cluster* 3	
Torvik V. I., 2009 Torvik and Smalheiser (2009) - *Cluster* 4	
Torvik V. I., 2005 Torvik et al. (2005) - *Cluster* 4	

Fonte: Elaboração própria.

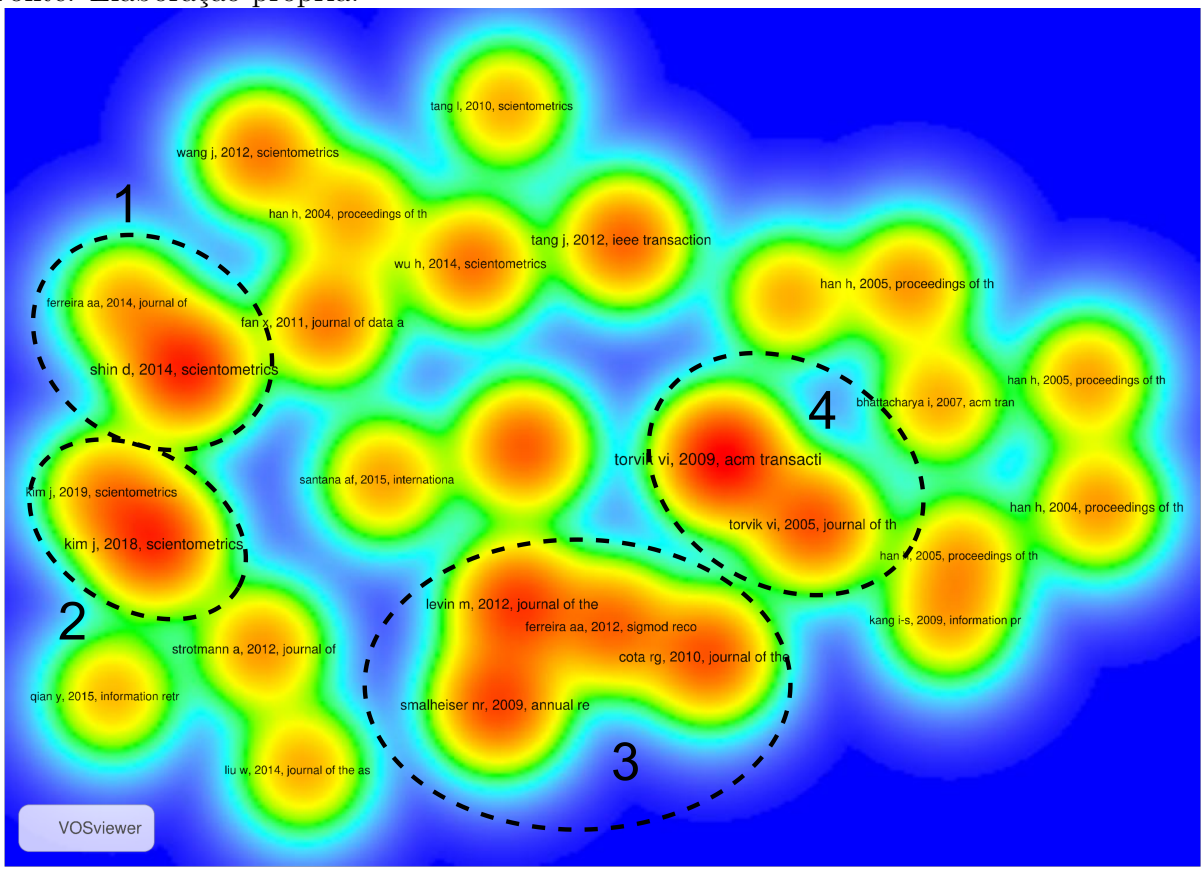


Figura 3.9: Mapa de calor apresentando os *clusters* na análise de cocitação. As linhas pontilhadas circulares com rótulos numerados indicam cada cluster. Fonte: Elaboração própria.

de coautoria para a tarefa de AND usando os bancos de dados DBLP e ArnetMiner. De acordo com a taxonomia, podemos classificar o tipo de abordagem como

Agrupamento de Autores e a evidência explorada como Informação de Citação e Informação da *Web*.

O trabalho apresentado por Ferreira et al. (2014) utiliza uma abordagem de três etapas para o AND. Primeiramente, usando uma heurística baseada em coautoria, as citações são agrupadas. Usando similaridades, alguns desses *clusters* serão selecionados para se tornarem dados de treinamento na segunda etapa. Na terceira etapa, os *clusters* selecionados são adicionados a um desambiguador de nomes associativos com capacidade de auto-treinamento. Esse trabalho é classificado, de acordo com a taxonomia, como Atribuição de Autor e a evidência explorada como Informação de Citação e Informação da *Web* (ou seja, dados extraídos do DBLP e BDBComp).

- *Cluster* 2: Embora nenhum dos dois trabalhos neste *cluster* sugira uma solução direta para o problema de AND, eles oferecem estratégias que auxiliam abordagens de resolução. O surgimento do *cluster* é justificado por sua alta cocitação na literatura, servindo de base para outros estudos. Kim et al. (2019a) apresenta um método para gerar dados rotulados para compor abordagens de aprendizado de máquina. Com execuções de teste, a proposta alcançou alto desempenho em comparação com trabalhos na literatura. Kim (2019) implementa uma estrutura que integra cinco medidas de validação para abordagens de AND usando agrupamento. Essa integração pode ajudar os acadêmicos da área de AND a comparar as semelhanças e diferenças das várias medidas de validação antes de selecionar as que melhor caracterizam o desempenho de agrupamento de seus métodos AND.

- *Cluster* 3: Smalheiser and Torvik (2009) apresenta uma breve revisão da literatura com foco na definição e nos desafios do problema AND. Ferreira et al. (2012) realizou uma revisão da literatura com abordagens para a resolução de AND, sugerindo uma taxonomia para classificar essas abordagens. Foi identificado que as duas revisões estão próximas em comparação com todo o mapa de calor, evidenciando uma grande cocitação desses artigos nos bancos de dados estudados.

  Dois outros trabalhos propõem abordagens para AND. Primeiro, Levin et al. (2012) apresenta um algoritmo auto-supervisionado que usa técnicas de *bootstrap* para agrupamento e um algoritmo de treinamento supervisionado. O trabalho usa informações das citações dos autores e outros atributos, como *e-mail*, nomes dos autores e idioma. Classificamos esse trabalho no tipo de abordagem como agrupamento de autores e a evidência explorada como informações de citação.

  O trabalho apresentado por Cota et al. (2010) usa uma abordagem baseada em heurística para AND com funções de similaridade de registros de evidência de autoria

extraídos de DBLP e BDBComp. De acordo com a taxonomia, o tipo de abordagem é Agrupamento de autores e Informações de citação de evidências exploradas.

- *Cluster* 4: Esse *cluster* contém dois artigos dos mesmos autores. O primeiro Torvik and Smalheiser (2009), uma abordagem probabilística, denominada Authority, para resolver o problema AND no banco de dados MEDLINE (2025) usando informações como título, nome do periódico, coautoria, idioma e outros recursos. O Authority calcula a similaridade entre dois artigos analisando os nomes e os *e-mails* dos autores. O modelo também apresenta formas de gerar automaticamente conjuntos de treinamento, métodos para estimar a probabilidade entre os nomes dos autores e um algoritmo de agrupamento aglomerativo baseado na máxima verossimilhança para calcular grupos de artigos que representam os autores estudados.

  O segundo trabalho, Torvik et al. (2005), também usa um modelo probabilístico para AND, mas usa apenas os nomes dos autores, descartando outras informações, como endereços de e-mail e afiliações. Assim, fica evidente que o trabalho de 2009 é uma evolução do trabalho de 2005. De acordo com a taxonomia, ambos os trabalhos utilizam o tipo de abordagem como *Author Grouping* e como evidência explorada *Citation Information*. Os outros *clusters* de cocitação apresentados pelo mapa de calor na Figura 3.9 não apresentaram padrões detectados por este estudo.

  A Figura 3.10 apresenta outra análise de cocitação usando a densidade de *clusters* com agrupamento entre todos os trabalhos de cocitação e permite uma visão de outras semelhanças entre os documentos em cada grupo. Assim, podemos analisar os outros trabalhos não tão evidentes na Figura 3.9. Observe que há três grupos gerais: verde, azul e vermelho. Uma característica comum é o espaço-tempo entre os documentos. O vermelho tem documentos de 2005 a 2010 e o azul de 2009 a 2015. Essa característica de espaço-tempo não aparece no *cluster* verde, pois ele é mais diversificado, com documentos de 2004 a 2019. Observe que há trabalhos nas bordas do cluster, unindo grupos com base na característica de data, como Ferreira et al. (2012) que liga os *clusters* azul e vermelho.

  O *cluster* verde é bastante diversificado, pois há vários tipos de abordagens, como mapas cognitivos e análise de rede (Tang and Walsh, 2010), modelos probabilísticos (Tang et al., 2012), modelos baseados em heurística (Santana et al., 2015), agrupamento hierárquico aglomerativo (Wu et al., 2014) e aprendizado supervisionado (Han et al., 2004; Wang et al., 2012).

  A cobertura do *cluster* vermelho funciona usando abordagens de agrupamento (Bhattacharya and Getoor, 2005; Han et al., 2005a,b). O estudo conduzido por Kang et al. (2009) investiga a influência dos atributos de coautoria para resolver o pro-

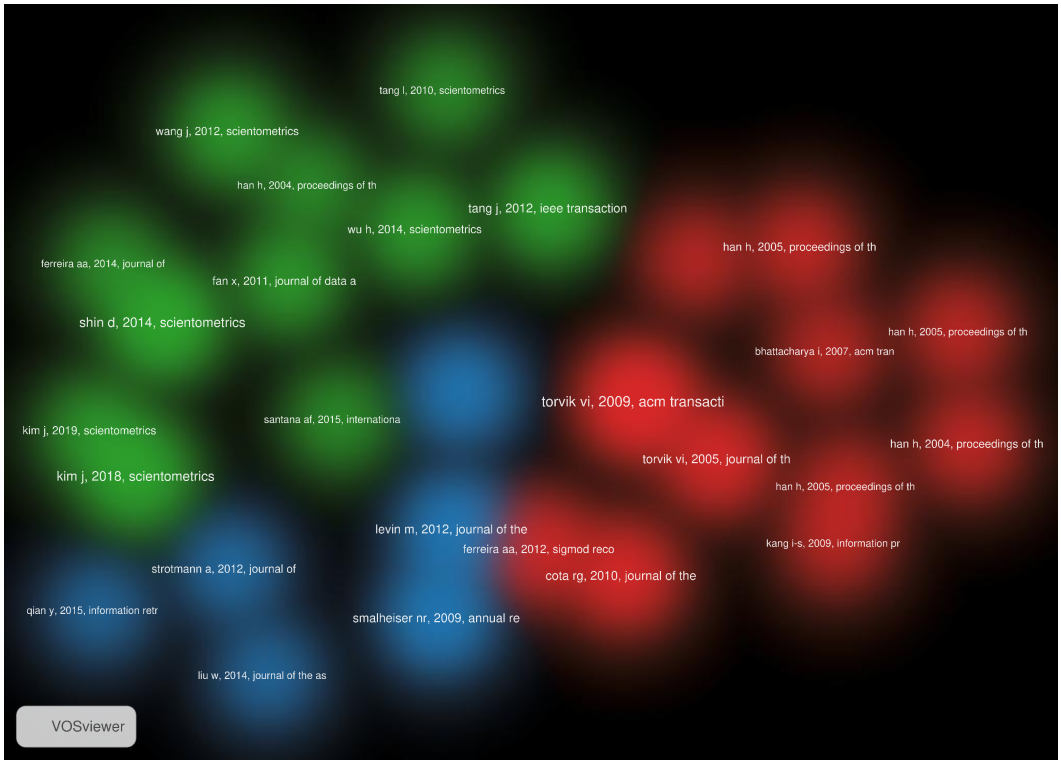


Figura 3.10: Cocitações com análise de densidade. Fonte: Elaboração própria.

blema AND. O *cluster* azul apresenta uma semelhança com a pesquisa que usa a semelhança de *cluster* e o agrupamento aglomerativo para AND (Liu et al., 2014; Qian et al., 2015). Em contraste, Strotmann and Zhao (2012) apresentam uma pesquisa que indica a influência da tarefa de AND em estudos de análise de base bibliográfica e de citação.

Análise de acoplamento bibliográfico

A análise de acoplamento bibliográfico proporciona uma compreensão do estado atual da área de pesquisa de AND. Apresentamos nesta seção as frentes de pesquisa de AND, incluindo trabalhos de 2019 a 2022. Os trabalhos são classificados considerando a abordagem de ANDusando a taxonomia apresentada na Figura 3.2.

A Figura 3.11 apresenta um acoplamento bibliográfico de trabalhos usando um mapa de calor para os repositórios mesclados da WoS e Scopus. Observe que existem três *clusters* explicitamente numerados, destacando as atuais frentes de pesquisa em AND. Em seguida, apresentamos um resumo dos trabalhos incluídos nos três *clusters*.

- *Cluster* 1: Zhang et al. (2019) usaram uma abordagem de incorporação de nó de grafo para resolver o problema AND. Esse tipo de solução é inspirado no modelo de *word embedding*, mas adaptado para uma solução estrutural de grafo. Um grafo é construído com relações de coautoria, usando o método de passeio aleatório e então

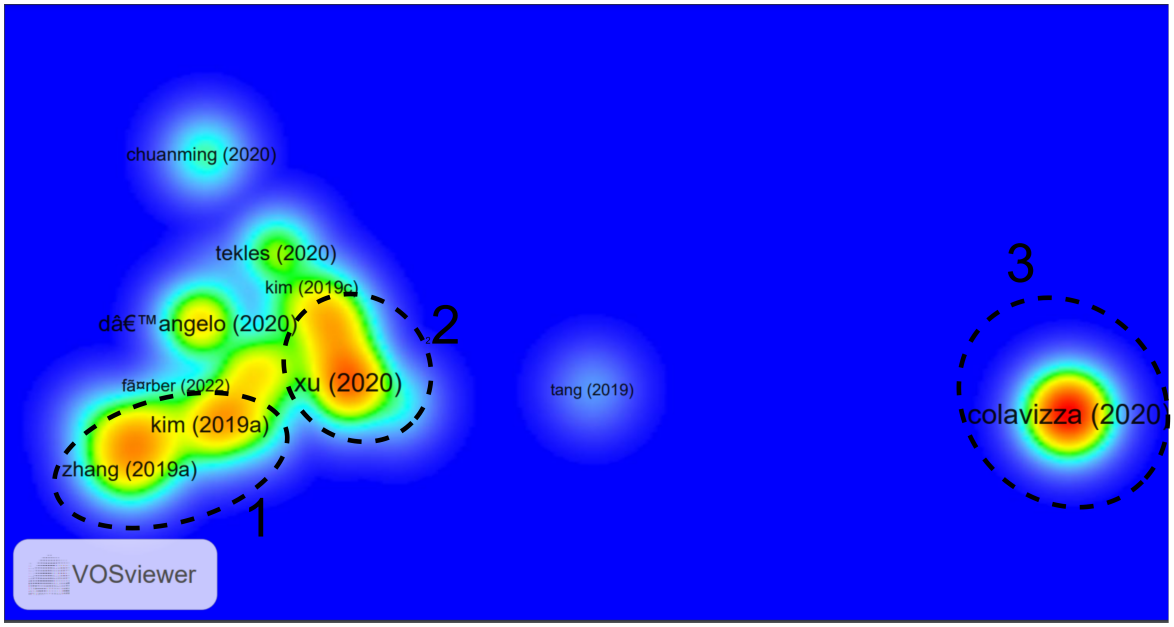


Figura 3.11: Mapa de calor apresentando os *clusters* na análise de acoplamento bibliográfico. As linhas pontilhadas circulares com rótulos numerados indicam cada cluster. Fonte: Elaboração própria.

atribuir um *cluster* a autores reais. A proposta usou os dados do CiteSeerX com resultados aprimorados em comparação com abordagens semelhantes.

Kim et al. (2019b) propõem um método híbrido de classificação por pares para estimar a probabilidade de um registro de autor estar correto em um repositório bibliográfico. Essa solução usa recursos globais extraídos do texto por meio de treinamento supervisionado em um conjunto de dados de citações de um autor. Essa classificação de texto e o treinamento supervisionado usam métodos de *word embedding*, como *bag of words* e TF-IDF, com dados do PubMed e do ArnetMiner. De acordo com a taxonomia, os trabalhos de Zhang et al. (2019) e Kim et al. (2019b) podem ser classificados como abordagens de Agrupamento de Autores porque usam cálculo de similaridade com treinamento e aprendizado de máquina. Os autores usam *word embedding* como base para o método AND, justificando a proximidade dos estudos observados no mapa de calor apresentado na Figura 3.11.

- *Cluster* 2: Em Xu et al. (2020), os autores criam um grafo de conhecimento com informações do repositório PubMed, extraindo bioentidades dos resumos. Neste trabalho, os autores não propõem uma nova abordagem para AND. No entanto, foram utilizadas abordagens já conhecidas na literatura, como Authority (que utiliza uma abordagem baseada em grafos) e Semantic Scholar (que utiliza um classificador binário de treinamento para associar pares de nomes de autores e criar *clusters* de autores). O grafo de conhecimento construído permitiu a criação de ligações entre

entidades, como, artigos, autores e afiliações. No passo de AND, os resultados alcançaram escores F1 de 98.09%. Podemos classificar a abordagem deste trabalho como Agrupamento de Autores e a evidência explorada como Informação de Citação e Informação da *Web*.

- *Cluster* 3: O trabalho de Colavizza et al. (2020) criou um sistema automático para declarações de disponibilidade de dados em repositórios bibliográficos usando o PubMed. Os autores comparam nomes e sobrenomes com técnicas de similaridade de *strings*. Ele aparece no mapa de calor da literatura porque cita vários trabalhos influentes de AND.

Com a análise de acoplamento da literatura, observou-se o uso recorrente de técnicas consolidadas para AND, especialmente métodos de agrupamento e agrupamento de autores. O primeiro artigo identificado no conjunto analisado data de 2003. Como o objetivo do acoplamento bibliográfico é evidenciar frentes de pesquisa emergentes, foram inicialmente selecionados trabalhos publicados entre 2020 e 2022. Contudo, para complementar a revisão realizada no início de 2023, optou-se por estender a análise, incorporando também artigos publicados até julho de 2025. Os trabalhos mais recentes (2020–2025) foram organizados de acordo com a taxonomia apresentada na Tabela 3.8 e representados graficamente na Figura 3.12.

A análise de cocitação, conduzida ao longo do intervalo 2003–2022, confirmou a predominância de abordagens baseadas em agrupamento de autores, corroborando os resultados apresentados em Ferreira et al. (2020). A análise de acoplamento bibliográfico também revelou caminhos alternativos e promissores para a resolução do problema de ambiguidade de nomes de autores, destacando a diversidade de estratégias investigadas na área.

3.3 Visão Geral de Pesquisas Recentes

Nesta seção, apresentamos uma visão geral dos trabalhos sobre AND publicados entre 2020 e julho de 2025, organizados conforme a taxonomia da Figura 3.2. Este intervalo de tempo foi escolhido, pois revisões anteriores cobriram trabalhos até 2019 (Debarshi et al., 2019; Ferreira et al., 2012). A análise de acoplamento inclui alguns trabalhos apresentados nesta seção.

A visão geral dos trabalhos de pesquisa atuais é essencial para completar a análise meta-analítica, citando as técnicas específicas, estratégias e temas emergentes dentro da área de pesquisa de AND. Para organizar a análise da visão geral dos trabalhos de maneira concisa, a orientamos pelo livro dedicado ao estudo de AND em repositórios bibliográficos (Ferreira et al., 2020). Apresentamos uma sinopse de cada trabalho incluído na

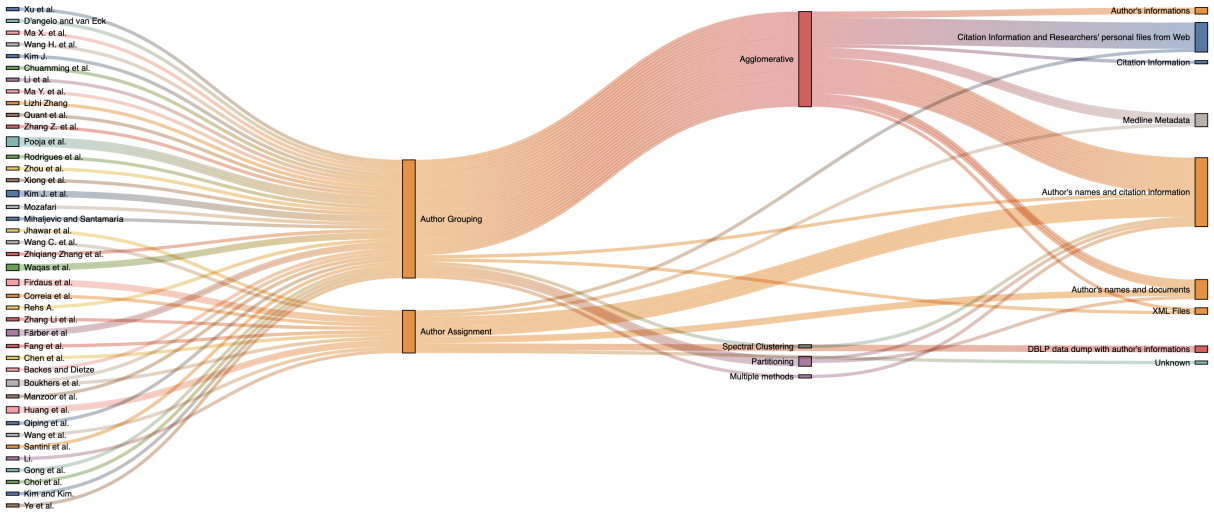


Figura 3.12: Diagrama Sankey para as abordagens de AND utilizadas entre 2020 e 2025. Fonte: Elaboração própria.

Tabela 3.8. A sinopse do trabalho apresenta as abordagens de AND, incluindo o agrupamento de autores, principalmente por meio do agrupamento aglomerativo, destacando-se como métodos prevalentes nos estudos recentes.

A abordagem de agrupamento de autores é especialmente adequada para conjuntos de dados com muitos dados de coautoria. Grandes conjuntos de dados bibliográficos podem se beneficiar dessa abordagem, ao contrário da abordagem de atribuição de autor, pois não depende de anotações manuais demoradas. O agrupamento aglomerativo no agrupamento de autores é uma abordagem comum, pois é simples de usar e pode produzir *clusters* hierárquicos a serem analisados em diferentes granularidades. Este método de agrupamento proporciona flexibilidade ao criar grupos de autores. A Figura 3.12 mostra como os trabalhos se relacionam entre si na taxonomia utilizada.

Identificamos um conjunto de cinco trabalhos que usam modelos de aprendizado baseados em árvores, como *Gradient Boosting, Random Forest* e *Decision Tree* (Jhawar et al., 2020; Kim et al., 2021; Mihaljević and Santamaría, 2021; Rehs, 2021; Zhang and Ban, 2020). Os trabalhos podem usar outras técnicas de aprendizado de máquina em conjunto com as mencionadas, como *Naive Bayes* (Kim et al., 2021), *Logistic Regression* (Jhawar et al., 2020; Rehs, 2021) e *Network Graphs* (Mihaljević and Santamaría, 2021).

Alguns trabalhos abordam técnicas supervisionadas distintas, como Huang et al. (2024); Kim and Owen-Smith (2020) que usam o aprendizado por transferência. Kim and Owen-Smith (2021) mostram que o ORCID pode validar o desempenho de métodos supervisionados de AND que usam dados bibliográficos em grande escala. Boukhers and Asundi (2022) e Boukhers and Asundi (2024) usaram o banco de dados DBLP com uma rede neural para aprender as representações de coautores e títulos para que o AND pudesse

considerar a similaridade entre esses atributos. Fang et al. (2023) propõem um algoritmo de seleção de atributos diferenciável via *Gumbel-Softmax*. A técnica escolhe automaticamente os melhores atributos para desambiguação, superando métodos baseados em engenharia manual de atributos. O modelo proposto por Gong et al. (2023) é baseado em *Capsule Networks*, que funde características semânticas e estruturais dos artigos para representação e agrupamento mais robustos. Chen et al. (2023) propõem um *framework* supervisionado e distribuído para AND em cenários com pouca ou nenhuma informação disponível sobre os autores. Em vez de agrupar publicações definindo antecipadamente quantos autores existem, o modelo reformula o problema como uma tarefa de predição de ligação entre pares de publicações. Kim and Kim (2024) introduzem o ANDez, um sistema de código aberto que unifica várias técnicas de aprendizado de máquina para avaliação e comparação de modelos de AND.

Wang et al. (2023) propõem um modelo de aprendizado par-a-par que integra atributos textuais, discretos e de coautoria usando mecanismos de atenção. A abordagem permite identificar relações entre pares de publicações apresentando resultados eficazes em grandes repositórios bibliográficos. Gong et al. (2024) apresentam uma arquitetura que combina representações semânticas de artigos, cálculo de similaridade supervisionado, agrupamento e redes neurais com atenção para refinar os grupos com base em coautorias. Zhou et al. (2024b) apresentam um modelo que integra diferentes tipos de informações, como atributos dos documentos e relações locais e globais em grafos, por meio de um mecanismo híbrido de atenção. A abordagem combina múltiplos grafos de similaridade e extrai representações mais informativas.

Li et al. (2020) apresentam um algoritmo com várias estratégias de similaridade para a implementação do AND usando cálculos de rede de colaboração, afiliação e atributos de publicações dos autores. Outra abordagem com várias estratégias é apresentada por Rodrigues et al. (2021) usando a comparação de *strings* com a similaridade Jaccard, a distância Levenshtein e a comparação da rede de coautoria. Waqas and Qadir (2021) propõem uma heurística de várias camadas com uma abordagem de agrupamento. O agrupamento usa atributos inerentes ao autor e à publicação, como título, resumo, palavras-chave, e-mail e afiliação. A representação de palavras baseada em Word2Vec é usada para extrair os atributos.

D'Angelo and van Eck (2020) usam uma abordagem de pontuação baseada em regras com atributos de autor, publicação, citação e instituição. Os *clusters* com metadados permitem a indexação de um determinado autor para AND. Zhang et al. (2020) usam regras heurísticas combinadas com redes neurais para analisar atributos de publicação, como título e afiliação. Uma vantagem desse método é a possibilidade de estender a aplicação do método a outros conjuntos de dados. Choi et al. (2024) propõem uma

abordagem de AND que integra dados de diferentes sítios acadêmicos e usa um algoritmo baseado em regras com suporte a classificadores para escolher a melhor estratégia conforme os metadados disponíveis.

Mozafari (2021) propõe um algoritmo genético para determinar o coeficiente de similaridade entre dois autores para AND. O algoritmo determina a importância dos atributos nas publicações, elegendo um coeficiente ideal para comparação entre os autores.

Jinqi et al. (2020) propõem um algoritmo para colocar entidades e recursos em um grafo de rede para definir o grau de compartilhamento baseado na capacidade do nó do recurso. O grafo de rede usa relacionamentos entre o autor e os nós de publicação para calcular a capacidade de fluxo entre os nós, permitindo assim o agrupamento.

Zhang and Ban (2020) usam relações de publicação para construir grafos com as publicações fortemente relacionadas e agrupadas, formando *clusters* atômicos e reduzindo o tamanho dos grafos. Em outro estágio, um algoritmo de similaridade baseado em regras analisa e combina as informações de recursos dos grafos de publicações para executar a AND.

Zhou et al. (2021) apresentam uma abordagem com cinco grafos formados por atributos de publicação, coautoria, local, título, palavras-chave e afiliação. Cada atributo cria o nó em que as bordas são os pesos de similaridade entre os pares de publicações. Um grafo de fusão dos atributos é criado. Um algoritmo de passeio aleatório é aplicado ao grafo para determinar caminhos que representam as informações estruturais do nó local. Em seguida, um algoritmo *Perceptron* de múltiplas camadas é aplicado à estrutura do grafo.

Liu et al. (2024a) propõem um *framework* para tarefa de AND refinando iterativamente os grafos que conectam documentos, reduzindo incertezas nas associações. Em seguida, aplica uma técnica de aprendizado contrastivo que combina diferentes representações para gerar vetores mais precisos.

Santini et al. (2022) propõem um modelo de Grafo de Conhecimento (*Knowledge Graph Embedding* - KGE) utilizando informações do banco de dados ArnetMiner. O KGE possui três partes: extração de informações multimodais do KGE, um procedimento de bloqueio e agrupamento aglomerativo hierárquico. Qiping et al. (2022) utilizam informações de citação para construir uma rede de informação heterogênea. Aprendizado de representação para agrupamento de autores e desambiguação é aplicado, e uma análise de *cluster* com correspondência de regras é realizada.

Ma et al. (2020b) propõem incorporar um modelo Word2Vec em uma abordagem baseada em grafo. O algoritmo extrai atributos e as relações entre publicações, autores e coautores. O Word2Vec é utilizado para obter essas características, permitindo a inserção de outras características que possam aparecer no conjunto de dados. Posteriormente, um grafo com relações entre publicações e autores é construído. Em seguida, um algoritmo

de agrupamento e análise de similaridade entre nós e arestas é aplicado.

Trabalhos que apresentam soluções usando uma abordagem baseada em grafos como base associada a outras técnicas computacionais, por exemplo, agrupamento, incluem:

- Pooja et al. (2020) utilizam uma abordagem de agrupamento baseada em grafo para AND. Similaridade de Jaccard e cosseno caracterizam as relações entre autores e publicações nos grafos, e informações da *Web* refinam os resultados.
- Pooja et al. (2021) utilizam grafos com atributos de publicação e o modelo de representação de palavras Word2Vec para criar vetores que servirão como entrada para um agrupamento utilizando HAC, amplamente utilizado em estudos de AND.
- Pooja et al. (2022a) utilizam uma abordagem baseada em grafo configurada com vizinhanças multi-hop e aplicam HAC para AND na etapa final do algoritmo.
- Pooja et al. (2022b) utilizam grafos para construir a rede de autores e publicações e utiliza o agrupamento para AND. No entanto, o diferencial desta nova abordagem é a capacidade de trabalhar com informações online de repositórios bibliográficos digitais.
- Liu et al. (2024b) propõem um *framework* que integra aprendizado textual e relacional com aprendizado de similaridade, otimizando todas as etapas do processo de AND de forma conjunta. O modelo combina representações de texto e de grafos, ajustadas aos objetivos da tarefa, e aplica um módulo final de agrupamento com refinamento.
- Ye et al. (2025) apresentam um *framework* que combina aprendizado contrastivo multivisual com um módulo orientado por *cluster*. A abordagem torna o modelo mais robusto ao ruído presente nos grafos iniciais e permite refinar dinamicamente os rótulos de *cluster* ao integrar o aprendizado de representação com a etapa de agrupamento.

Os autores em Chuanming et al. (2020); Wang et al. (2020b); Xiong et al. (2021b); Zhang et al. (2021b) utilizam *word embedding*, grafos e agrupamento, respectivamente. Ma et al. (2020a) usam a mesma abordagem aplicada a consultores de literatura robótica.

Wang et al. (2020a) propõe uma técnica com classificação parcial em três etapas para resolver a tarefa de AND. A primeira usa uma restrição de propagação de probabilidade para inferir a distribuição de um nome de autor dado. Na segunda etapa, uma parte do nome do autor nos documentos é vinculada aos seus respectivos autores se o modelo apresentar alta confiança. Na última etapa, os parâmetros do algoritmo de classificação inicial são atualizados.

Firdaus et al. (2021b) propõem dois métodos para AND. No primeiro trabalho, a técnica usa várias etapas para a desambiguação: dados rotulados, atributos de publicação extraídos, rede neural profunda, classificação por *Random Walk*, *Naive Bayes* e SVM, e a validação do resultado é feita comparando as técnicas de classificação. No segundo trabalho, Firdaus et al. (2021a), a classificação com a técnica de rede neural profunda é melhorada com aprendizado sensível ao custo, considerando a variação de custo a partir de dados não classificados.

Manzoor et al. (2022) utilizam redes neurais convolucionais para classificação de conjuntos de dados desbalanceados e balanceados. De acordo com os autores, a solução é flexível ao aprender os atributos sem concatenar medidas de similaridade. O mesmo método também é Single Citation Based, o que pré-processa eficientemente o conjunto de dados, diminuindo os custos computacionais.

Färber and Ao (2022) não propõem uma nova abordagem, mas utilizam um método de classificação baseado em regras não supervisionado. O método não requer treinamento de dados, sendo adaptado para a proposta dos autores.

Correia et al. (2021) propõem um protótipo baseado em sistemas de crowd, permitindo interação e contribuição da *Web* para o público em geral corrigir ambiguidades de nomes, dados ausentes e referências incorretas em um repositório bibliográfico digital. Backes and Dietze (2022) apresentam uma técnica para AND progressivo com estruturas de lattice para inclusão de nomes. Waqas and Qadir (2022) não propõem uma resolução de AND, mas apresentam um conjunto de dados para auxiliar desenvolvedores. O conjunto de dados rotulado *CustAND* com 7886 registros de publicações é apresentado utilizando dados do DBLP e Google Scholar.

Huang et al. (2025) propõem um modelo de AND baseado em papéis atribuídos globalmente aos autores, considerando diferentes funções que um autor pode exercer ao longo de sua trajetória acadêmica. A abordagem organiza essas funções em um *framework* de reconhecimento de padrões, buscando melhorar a atribuição correta das publicações.

Li (2024) propõe um método baseado em impressões digitais semânticas para AND, combinando características de coautores, instituições e textos. A abordagem visa melhorar a precisão sem depender de dados adicionais ou alto custo computacional, mostrando bons resultados em um conjunto de dados anotados manualmente.

3.4 Discussão

A análise de cocitação e acoplamento bibliográfico apresentou a ocorrência de quatro grupos distintos. Na análise de cocitação, esses grupos evidenciaram o uso de abordagens baseadas em grafos, aprendizado supervisionado e heurísticas, com aplicação de técnicas

probabilísticas para resolver o problema AND. Por outro lado, a análise de acoplamento bibliográfico indica uma pesquisa atual voltada para o problema AND, com ênfase em *word embedding* e aprendizado supervisionado. É importante mencionar que a maioria das abordagens utiliza as bases bibliográficas ArnetMiner e DBLP para a extração de informações.

As análises de cocitação e acoplamento bibliográfico realizadas identificaram a existência de quatro grupos distintos de abordagens para a tarefa de AND. Essas análises forneceram informações sobre as tendências atuais de pesquisa e as estratégias mais comuns adotadas na área.

No contexto da análise de cocitação, os grupos identificados destacaram o uso de abordagens baseadas em grafos, aprendizado supervisionado e heurísticas, com aplicação de técnicas probabilísticas para resolver o problema AND. Isso indica que essas estratégias têm sido amplamente exploradas pela comunidade de pesquisa e podem fornecer um ponto de partida sólido para o desenvolvimento de uma abordagem eficaz.

Por outro lado, a análise de acoplamento bibliográfico revelou uma pesquisa atual centrada na tarefa de AND, com foco em *word embedding* e no uso de aprendizado supervisionado. Isso sugere que essas áreas específicas estão recebendo atenção significativa e podem permitir o aprimoramento da eficácia da AND.

Além disso, a menção de que a maioria das abordagens identificadas utiliza as bases bibliográficas ArnetMiner e DBLP para a extração de informações destaca a relevância dessas fontes de dados e sugere que uma abordagem eficaz deve considerar essas fontes como parte integrante do processo de AND.

A revisão da literatura permitiu identificar tendências recentes no uso de técnicas baseadas em aprendizado profundo e modelagem em grafos. Com base nessas evidências, delineou-se uma proposta de solução que integra PLN para representação semântica, estratégias de aprendizado em grafos por meio de RCG e algoritmos de agrupamento hierárquico orientados à estrutura de grafos, como o GHAC. Essa abordagem híbrida está alinhada com os avanços do estado da arte e se mostra promissora para lidar com os desafios da tarefa de AND em repositórios bibliográficos digitais. Os resultados e informações detalhadas desta revisão da literatura, incluindo gráficos interativos, mapas de calor e filtros, estão disponíveis em: `https://natansr.github.io/AND-TEMAC/`

A revisão apresentada, demonstra um esforço sistemático e abrangente para compreender as diferentes abordagens no contexto de AND em repositórios bibliográficos digitais, proporcionando uma base sólida e contextualizada para o desenvolvimento do modelo proposto no âmbito do *framework* ADAN apresentado no Capítulo 4.

Tabela 3.8: Classificação dos artigos de 2020 a 2025, conforme a taxonomia da Figura 3.2 (Ferreira et al., 2012, 2020).

Referência	Tipo de Abordagem			Evidência Explorada	Conjunto de Dados	Repositório da Citação
	Agrupamento de Autor Função de Similaridade agrupamento		Atribuição de Autor Método Classificação			
Kim and Owen-Smith (2020)	Transfer Learning	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	DBLP, ArnetMiner, KISTI, MEDLINE American Physical Society	Scopus Wos
Xu et al. (2020)	Autoridade com uma métrica probabilística aprendida; Semântica Acadêmica com aprendizado baseado em erro e baseado em hank.	Aglomerativo		Metadados do Medline	PubMed	Scopus Wos
D'Angelo and van Eck (2020)	Escore Baseado em Regras	Aglomerativo		Informações do Autor (nome, posição acadêmica, áreas de pesquisa e afiliação institucional)	Fonte de Dados do Ministério Italiano de Educação, Universidades e Pesquisa	Scopus Wos
Chuanming et al. (2020)	Aprendizado Não Supervisionado	Aglomerativo		Informações de Citação	CiteSeerX, ArnetMiner, DBLP	Scopus
Jhawar et al. (2020)			Classificação Baseada em Conjunto com Random Forest e Gradient Boosted Tree	Metadados do Medline	PubMed	Scopus
Li et al. (2020)	Heurístico	Particionamento		Arquivos XML com atributos de autor e publicações	WoS	Scopus
Ma et al. (2020b)	Autoencoder de Grafo e Incorporação de Grafo com Word2Vec	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	ArnetMiner	Scopus
Jinqi et al. (2020)	Fluxo Máximo em Grafo de Rede	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	ArnetMiner, Microsoft Academic	Scopus

Continua na próxima página

Tabela 3.8: Classificação dos artigos de 2020 a 2025, conforme a taxonomia da Figura 3.2 (Ferreira et al., 2012, 2020). (Continuação)

Referência	Tipo de Abordagem			Evidência Explorada	Conjunto de Dados	Repositório da Citação
	Agrupamento de Autor Função de Similaridade agrupamento		Atribuição de Autor Método Classificação			
Ma et al. (2020a)	Algoritmo Baseado em Meta-caminho com incorporação de nós em uma rede homogênea	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	ArnetMiner	Scopus
Wang et al. (2020a)			Técnica de Classificação Supervisionada com modelo baseado em passeio aleatório	Despejo de dados do DBLP com informações do autor	DBLP	Scopus Wos
Wang et al. (2020b)	Modelo de aprendizado de representação adversarial com rede de informação heterogênea	Aglomerativo		Nome do autor e informações de citação	ArnetMiner	Scopus Wos
Zhang and Ban (2020)	Desambiguação Baseada em Regras em um modelo de grafo	Aglomerativo		Nome do autor e informações de citação	ArnetMiner	Scopus
Zhang et al. (2020)	Rede Neural Convolucional para comparar *clusters* de publicações	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	ArnetMiner	Scopus
Pooja et al. (2020)	Abordagem Baseada em Podas de Arestas em Grafo	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	ArnetMiner, WoS	Scopus
Rodrigues et al. (2021)	Abordagem Multiestratégica com comparação de *strings* e redes de autores	Aglomerativo		Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	DBLP	Scopus
Zhou et al. (2021)	Similaridade de Grafo com Frequência Inversa do Documento	Particionamento		Nome do autor e informações de citação	ArnetMiner	Scopus Wos
Firdaus et al. (2021b)			Naïve Bayes, Random Forest, Support Vector Machine e Deep Neural Network	Despejo de dados do DBLP com informações do autor	DBLP	Scopus

Continua na próxima página

Tabela 3.8: Classificação dos artigos de 2020 a 2025, conforme a taxonomia da Figura 3.2 (Ferreira et al., 2012, 2020). (Continuação)

Referência	Tipo de Abordagem			Evidência Explorada	Conjunto de Dados	Repositório da Citação
	Agrupamento de Autor Função de Similaridade agrupamento		Atribuição de Autor Método Classificação			
Xiong et al. (2021a)	Aprendizado Não Supervisionado com Autoencoder Variacional	Aglomerativo		Nome do autor e informações de citação	ArnetMiner, DBLP, CiteSeerX	Scopus
Kim and Owen-Smith (2021)	Similaridades de Autoridade	Aglomerativo		Metadados do Medline	PubMed	Scopus Wos
Mozafari (2021)	Algoritmo Genético para aprendizado a partir das amostras disponíveis	Aglomerativo		Informações do autor (nome, posição acadêmica, áreas de pesquisa e afiliação institucional)	Ministério Iraniano da Ciência, Ministério da Saúde	Scopus
Mihaljević and Santamaría (2021)	Aprendizado Supervisionado com Decision Tree, Random Forest e Histogram-based Gradient Boosting	Aglomerativo		Nome do autor e documentos	NASA/ADS	Scopus
Correia et al. (2021)			Página da *Web* com formulário para campanha de crowdsourcing			Scopus Wos
Zhang et al. (2021b)	Redes de Atenção a Grafos	Agrupamento Espectral		Nome do autor e informações de citação	ArnetMiner	Scopus Wos
Pooja et al. (2022a)	Aprendizado de Representação Multidimensional com metadados e grafos de similaridade de autor	Aglomerativo		Nome do autor e informações de citação	ArnetMiner, DBLP, CiteSeer, Zbmath	Scopus
Zhang et al. (2021a)			Aprendizado Supervisionado com Random Forest	Informações de Citação e Arquivos Pessoais de Pesquisadores da *Web*	PubMed, Microsoft Academic, Semantic Scholar	Scopus Wos
Kim et al. (2021)	Gradient Boosting, Logistic Regression, Naïve Bayes e Random Forest			Nome do autor e informações de citação	KISTI, ArnetMiner, GESIS, UM-IRIS	Scopus Wos

Continua na próxima página

Tabela 3.8: Classificação dos artigos de 2020 a 2025, conforme a taxonomia da Figura 3.2 (Ferreira et al., 2012, 2020). (Continuação)

Referência	Tipo de Abordagem			Evidência Explorada	Conjunto de Dados	Repositório da Citação
	Agrupamento de Autor Função de Similaridade agrupamento		Atribuição de Autor Método Classificação			
Waqas and Qadir (2021)	Agrupamento heurístico em camadas múltiplas com Research2vec e similaridade cosseno	Aglomerativo		Nome do autor e informações de citação	ArnetMiner, BDBComp	Scopus Wos
Firdaus et al. (2021a)			Deep Neural Network com Sensibilidade a Custos	Nome do autor e informações de citação	DBLP	Scopus
Rehs (2021)	Random Forest e Logistic Regression	Particionamento		Nome do autor e documentos	WoS	Scopus Wos
Färber and Lamprecht (2021)	Baseado em Regras com similaridade Jaro-Winkler	Aglomerativo		Arquivos XML com atributos de autor e publicação	OpenAire, WikiData	Scopus Wos
Pooja et al. (2022a)	Graph Convolution Baseado em Atenção com vizinhança multihop	Aglomerativo		Nome do autor e informações de citação	ArnetMiner	Scopus
Backes and Dietze (2022)	Fusão progressiva de blocos	Aglomerativo		Nome do autor e documentos	WoS	Scopus Wos
Manzoor et al. (2022)	Convolutional Neural Network para classificação	Aglomerativo		Metadados do Medline	PubMed	Scopus Wos
Boukhers and Asundi (2022)	Rede Neural para aprender autor e coautores	Aglomerativo		Nome do autor e informações de citação	DBLP	Scopus Wos
Färber and Ao (2022)	Abordagem Não Supervisionada com classificador baseado em regras	Aglomerativo		Nome do autor e documentos	MAKG	Scopus Wos
Qiping et al. (2022)	Aprendizado de Representação em Rede	Aglomerativo		Nome do autor e informações de citação	ArnetMiner, DBLP, CiteSeerX	Scopus Wos
Santini et al. (2022)	Incorporação Multimodal de Grafos de Conhecimento	Aglomerativo		Nome do autor e informações de citação	ArnetMiner, ORCID	Scopus Wos

Continua na próxima página

Tabela 3.8: Classificação dos artigos de 2020 a 2025, conforme a taxonomia da Figura 3.2 (Ferreira et al., 2012, 2020). (Continuação)

Referência	Tipo de Abordagem			Evidência Explorada	Conjunto de Dados	Repositório da Citação
	Agrupamento de Autor Função de Similaridade agrupamento		Atribuição de Autor Método Classificação			
Waqas and Qadir (2022)	Verificação cruzada manual e similaridade cosseno para detectar ambiguidades	Aglomerativo		Informações de Citação e arquivos pessoais de pesquisadores na *Web*	Google Scholar, DBLP	Scopus Wos
Fang et al. (2023)	*Gumbel-Softmax*		Similaridade supervisionada	Nome do autor e informações de citação	S2AND	Scopus Wos
Gong et al. (2023)	Características Semânticas e Estruturais	Aglomerativo		Nome do autor e informações de citação	AMiner	Scopus
Choi et al. (2024)	Matriz de similaridade com *sim2diss*	Aglomerativo		Nome do autor e informações de citação	NTIS, SCIENCEON, DBPIA, KCI	Scopus Wos
Chen et al. (2023)	Similaridade cosseno e *Jaccard*		Classificação Supervisionada (*linkage*)	Nome do autor e informações de citação	AMiner	Scopus Wos
Kim and Kim (2024)	Múltiplos métodos de aprendizado de máquina integrados)	K-Means, Spectral, DBSCAN e Aglomerativo	Classificadores Integrados com *benchmark* de aprendizado de máquina	Similaridade entre pares e atributos acessíveis	Próprio conjunto de dados	Scopus Wos
Huang et al. (2024)	*Embeddings* textuais e de grafos com aprendizado por transferência		Classificação Supervisionada em Grafos	Nome do autor e documentos	Kejso	Scopus
Wang et al. (2023)	Co-atenção supervisionada		Classificação par-a-par	Nome do autor, documentos e informações de citação	Desconhecido	Scopus Wos
Huang et al. (2025)	Fatores pessoais, culturais e institucionais		Classificação com Reconhecimento de Padrões	Nome do autor e informações de citação	ORCID	Scopus
Liu et al. (2024a)	Aprendizado constrativo		Classificação supervisionada	Nome do autor e informações de citação	DBLP	Scopus

Continua na próxima página

Tabela 3.8: Classificação dos artigos de 2020 a 2025, conforme a taxonomia da Figura 3.2 (Ferreira et al., 2012, 2020). (Continuação)

Referência	Tipo de Abordagem			Evidência Explorada	Conjunto de Dados	Repositório da Citação
	Agrupamento de Autor Função de Similaridade agrupamento		Atribuição de Autor Método Classificação			
Zhou et al. (2024b)	Similaridade de *Jaccard*, *Dice*, TF-IDF e IDF		Classificação Supervisionada com atenção	Nome do autor e documentoss	OAG-*WhoisWho* e AD-AND	Scopus Wos
Li (2024)	Impressão digital semântica	Aglomerativo	Classificação supervisionada	Nome do autor e informações de citação	Próprio conjunto de dados	Scopus
Boukhers and Asundi (2024)	Coautores e domínios de pesquisa		Classificação Supervisionada (RNN)	Nome do autor e informações de citação	DBLP	Scopus Wos
Ye et al. (2025)	Aprendizado Constrativo Multivisual	Particionamento		Nome do autor e informações de citação	AMiner, WhoisWho, LAGOS-AND	Scopus Wos

Fonte: Elaboração própria.

Capítulo 4

Proposta

O foco deste capítulo é apresentar a resposta à QP1, conforme proposta na Seção 1.3:

> QP1: De que forma um *framework* híbrido, que combine aprendizado de máquina profundo, com RCG, técnicas de PLN baseadas em *transformers* e agrupamento hierárquico aprimorado, provê um método eficaz para a tarefa de AND em repositórios bibliográficos digitais?

O problema de ambiguidade de nomes de autores em repositórios bibliográficos digitais representa um desafio devido à presença de homonímia, paronímia e à diversidade nas formas de citação de nomes de autores nas diversas línguas. Com base nos fundamentos teóricos discutidos no Capítulo 2, e considerando as características dos métodos analisados, este trabalho propõe um *framework* híbrido denominado ADAN.

O ADAN busca aprimorar os resultados da tarefa de AND em repositórios bibliográficos digitais por meio da combinação de técnicas de PLN, RCG e GHAC. A importância do uso destas técnicas de IA para a tarefa de AND é apresentada nas respectivas seções: PLN (Seção 2.2), GHAC (Seção 2.3) e RCG (Seção 2.4). A abordagem híbrida apresentada no ADAN está consoante com os avanços do estado da arte da literatura de AND, conforme discutido na Seção 3.4.

Os resultados obtidos com o ADAN foram comparados aos resultados de trabalhos de referência da literatura, utilizando uma análise quantitativa. Para a análise do desempenho do *framework* foram utilizadas métricas de avaliação comumente adotadas para avaliação da tarefa de AND, como pF1, K-Metric e B-Cubed F1, conforme destacado por (Ferreira et al., 2020) e apresentadas na Seção 2.5.

4.1 Modelo Arquitetural

O objetivo principal desta seção é apresentar a estrutura arquitetural do ADAN, detalhando as operações realizadas em cada uma de suas camadas, desde a entrada dos dados

até a geração dos resultados finais. Busca-se, com isso, oferecer uma visão sistematizada das funcionalidades do modelo, evidenciando suas interdependências e o fluxo de processamento ao longo das etapas do *framework*.

A arquitetura do ADAN está alinhada ao fluxo tradicional da tarefa de AND, conforme proposto por Ferreira et al. (2020) e apresentado na Seção 2.1. A tarefa é composta por quatro etapas principais: (i) Pré-processamento; (ii) Definição de Registros de Autoria Ambíguos; (iii) Agrupamento de Registros de Autoria Ambíguos; e (iv) Desambiguação.

Conforme a Figura 4.1, os módulos de Entrada e Pré-processamento no ADAN estão diretamente relacionados à etapa de (i) Pré-processamento, sendo responsáveis por carregar, padronizar e estruturar os dados bibliográficos. A etapa de (ii) Definição dos Registros de Autoria Ambíguos é realizada pelos módulos de Extração de Embeddings com PLN e Construção da Rede Heterogênea, que transformam as informações textuais e relacionais em representações vetoriais e estruturadas. A etapa de (iii) Agrupamento de Registros de Autoria Ambíguos é conduzida pelos módulos de Fusão de Embeddings e RCG, que integram as representações e propagam as informações na topologia do grafo. Por fim, a (iv) Desambiguação é realizada pelos módulos de agrupamento com GHAC e Saída, responsáveis por agrupar os registros ambíguos e apresentar os resultados finais.

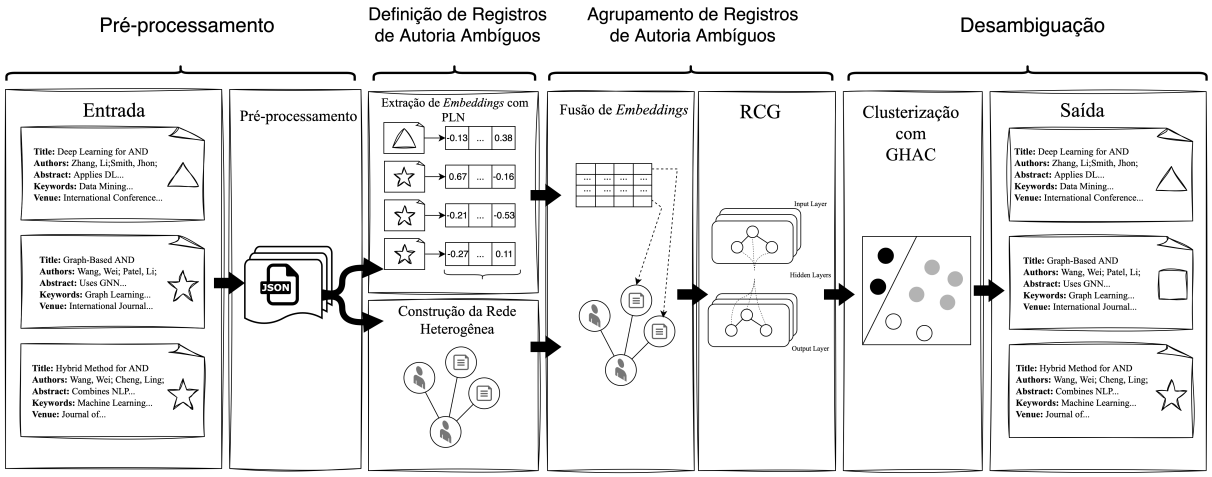


Figura 4.1: Fluxo das etapas do ADAN alinhado à tarefa de AND proposta por Ferreira et al. (2020). Fonte: Elaboração própria.

Além do fluxo geral apresentado, a Figura 4.2 apresenta uma versão mais detalhada da arquitetura do ADAN, organizada em quatro camadas principais (*layers*): L_1, L_2, L_3 e L_4. Essa representação em camadas tem como objetivo modularizar as operações realizadas em cada etapa da execução, desde a entrada dos dados até a produção dos resultados da desambiguação. As funcionalidades de cada camada serão descritas nas seções seguintes.

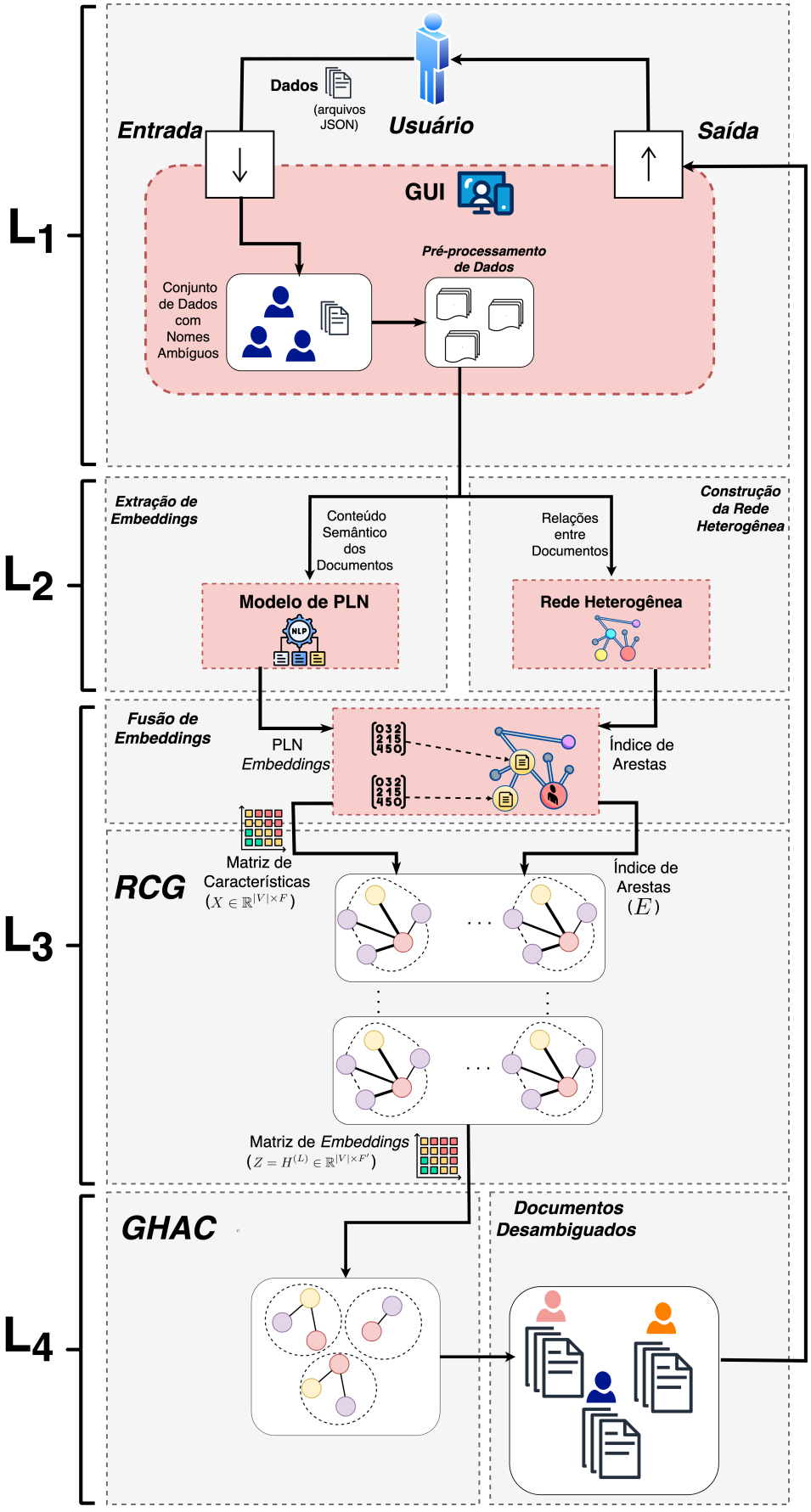


Figura 4.2: Arquitetura detalhada do ADAN. Fonte: Elaboração própria.

L_1 - Entrada e Pré-processamento

Esta camada é responsável pela entrada e pelo pré-processamento dos dados. Em L_1, a *Graphical User Interface* (GUI) permite a interação do usuário com os elementos do *framework* e a inclusão de novos dados bibliográficos relevantes para a tarefa de AND.

Após o envio completo do conjunto de dados para a tarefa de AND, é realizado um pré-processamento para garantir a consistência e a padronização das informações. Esse pré-processamento inclui a remoção de duplicatas de publicações, a normalização de campos textuais (e.g., remoção de caracteres especiais, símbolos e espaços em excesso) e a padronização de metadados (como a verificação da presença de campos obrigatórios), assegurando que os dados estejam prontos para serem utilizados nas etapas subsequentes da tarefa de AND.

Os dados de entrada consistem em arquivos no formato JSON,[1] contendo os metadados das publicações associadas a cada autor. A Listagem 4.1 apresenta um exemplo de entrada, com campos obrigatórios, tais como "Identificador" (`id`), "Título" (`title`) e "Autores" (`authors`). Além desses, o JSON pode conter campos opcionais como "Resumo" (`abstract`), "Palavras-chave" (`keywords`), "Veículo" (`venue`), "*E-mail*" (`email`) e "Afiliação" (`organization`) cuja disponibilidade varia de acordo com o conjunto de dados utilizado. Outro campo presente é o "Rótulo" (*label*), que identifica o autor real ao qual cada publicação pertence. No entanto, esse campo não é utilizado durante o pré-processamento. Ele é reservado exclusivamente para a fase de validação do processo da tarefa de AND, realizada na camada L_4.

Listagem 4.1: Exemplo de entrada JSON.

```
{
  "id": 2001,
  "label": 42,
  "title": "A Study on Smart Cities",
  "organization": "University of Example",
  "abstract": "This paper discusses key technologies in the development of smart
      cities.",
  "venue": "International Conference on Smart Systems",
  "email": null,
  "authors": ["Elbakyan A.", "John Titor", "Ronnie Dio"],
  "keywords": "smart cities, IoT, sustainability"
}
```

Concluído o pré-processamento, as informações extraídas são reorganizadas e salvas em arquivos intermediários no formato `.txt`, devidamente preparados para utilização na camada L_2, sendo que cada arquivo representa um aspecto específico dos dados. A

[1] `https://www.ecma-international.org/publications-and-standards/standards/ecma-404/`

Tabela 4.1 descreve os arquivos gerados, com seus respectivos conteúdos e formatos. O processo completo é descrito no Algoritmo 1, que apresenta o pseudocódigo da rotina de pré-processamento aplicada aos arquivos JSON utilizados como entrada.

Tabela 4.1: Exemplo de dados após o pré-processamento do JSON da Listagem 4.1

Arquivo	Conteúdo	Formato
`Paper_author.txt`	i2001 0, i2001 1, i2001 2	`paper_id id_author`
`Paper_author_names.txt`	i2001 elbakyan a., i1001 john titor, i2001 ronnie dio	`paper_id author_name`
`Paper_title.txt`	i2001 a study on smart cities	`paper_id title`
`Paper_abstract.txt`	i2001 this paper discusses key technologies in the development of smart cities.	`paper_id abstract`
`Paper_venue_name.txt`	i2001 international conference on smart systems	`paper_id venue_name`
`Paper_venue.txt`	i2001 7	`paper_id id_venue`
`Paper_keywords.txt`	i2001 smart cities, i2001 iot, i2001 sustainability	`paper_id keyword`

Fonte: Elaboração própria.

A GUI também permite a visualização dos resultados finais da desambiguação gerados pelo *framework*. Por viabilizar a entrada de dados e a visualização dos resultados, a GUI desempenha um papel central na melhoria da usabilidade do *ADAN*. Desta forma, beneficia usuários não especializados, reduzindo a barreira de entrada para a aplicação de técnicas de AND em ambientes reais de gestão bibliográfica e repositórios bibliográficos digitais.

L_2 — Extração de *Embeddings* e Construção da Rede Heterogênea

Nesta camada são realizadas duas operações centrais e complementares: a criação da rede heterogênea, a partir dos metadados das publicações, e a extração de representações semânticas vetoriais (*embeddings*), com base em modelos de PLN. Ambas as operações têm como objetivo fornecer informações contextuais e estruturais que serão exploradas nas etapas subsequentes da tarefa AND.

A rede é modelada como um grafo heterogêneo $G = (V, E)$, conforme a definição apresentada na Seção 2.3. O conjunto V representa o conjunto de nós tipados e $E \subseteq V \times V$ representa o conjunto de arestas tipadas que expressam diferentes tipos de relações semânticas entre as entidades. O conjunto de nós V é composto por subconjuntos disjuntos:

$$V = A \cup T \cup R \cup W \cup V \cup F \cup P,$$

sendo definidos como:

- A: autores — indivíduos responsáveis por uma ou mais publicações;

Algoritmo 1 Pré-processamento de publicações em JSON

Entrada: Diretório de entrada *input_dir*, diretório de saída *output_dir*, atributos selecionados $features$

Saída: Arquivos `.txt` com campos estruturados

Inicializar *authors_map*, *venues_map*, *keyid* $\leftarrow 0$
Carregar todos os arquivos JSON de *input_dir* em *all_data*
para cada publicação $entry \in all\_data$ **faça**
    Obter *id*, *title*, *abstract*, *venue*, *label*, *keywords*
    Extrair autores em *all_authors*
    se $abstract \neq \emptyset$ **e** "abstract" $\in features$ **então**
        Escrever (*i_id*, *abstract*) em `paper_abstract.txt`
    fim se
    para cada $author \in all\_authors$ **faça**
        se *author* não está em *authors_map* **então**
            Atribuir *authors_map*[*author*] $\leftarrow$ *keyid*; incrementar *keyid*
        fim se
        Escrever (*i_id*, *author_id*) em `paper_author.txt`
    fim para
    Escrever todos os nomes em `paper_author_names.txt`
    se $venue \neq \emptyset$ **e** "venue_name" $\in features$ **então**
        se *venue* não está em *venues_map* **então**
            Atribuir *venues_map*[*venue*] $\leftarrow$ *keyid*; incrementar *keyid*
        fim se
        Escrever (*i_id*, *venue_id*) em `paper_venue.txt`
        Escrever (*i_id*, *venue*) em `paper_venue_name.txt`
    fim se
    Limpar e normalizar *title*
    Escrever (*i_id*, *title*) em `paper_title.txt`
    se $keywords \neq \emptyset$ **e** "keywords" $\in features$ **então**
        para cada $kw \in keywords$ **faça**
            Escrever (*i_id*, *kw*) em `paper_keywords.txt`
        fim para
    fim se
fim para

- T: títulos — conteúdo textual dos títulos das publicações;
- R: resumos — descrições textuais dos conteúdos das publicações;
- W: palavras-chave — termos temáticos extraídos das publicações;
- C: veículos — conferências ou periódicos nos quais as publicações foram apresentadas;
- F: instituições — afiliações institucionais associadas aos autores, quando disponíveis;
- P: publicações — documentos a serem desambiguados.

As arestas E descrevem as interações entre os nós e são categorizadas da seguinte forma:

- *written_by*: (p, a), indica que a publicação $p \in P$ é de autoria do autor $a \in A$;
- *has_title*: (p, t), indica que a publicação possui o título $t \in T$;
- *has_abstract*: (p, r), indica que a publicação possui o resumo $r \in R$;
- *contains*: (p, w), indica que a publicação contém a palavra-chave $w \in W$;
- *published_in*: (p, c), indica que a publicação foi veiculada no periódico ou conferência $c \in C$;
- *affiliated_with*: (a, f), indica que o autor $a \in A$ está afiliado à instituição $f \in F$;
- *co_authored*: (a_i, a_j), indica que os autores $a_i, a_j \in A$ colaboraram na mesma publicação;
- *shared_word*: (p_i, p_j), indica que as publicações $p_i, p_j \in P$ compartilham ao menos uma palavra-chave.

Essa modelagem permite identificar conexões explícitas entre autores e publicações, padrões de colaboração e similaridades semânticas implícitas entre documentos, as quais não estão diretamente presentes nos metadados, mas podem ser inferidas a partir da estrutura do grafo. Esses relacionamentos enriquecem a representação dos dados e podem ser explorados por algoritmos de aprendizado, como a RCG, por exemplo.

A topologia do grafo é armazenada por um índice de arestas (*edge index*), que consiste em uma matriz esparsa contendo apenas os pares de nós conectados. Essa estrutura, adotada por bibliotecas como a PyTorch Geometric (Fey and Lenssen, 2019), evita o uso de matrizes de adjacência densas e pode melhorar significativamente a eficiência computacional durante o treinamento de modelos baseados em grafos.

O Algoritmo 2 descreve o processo de construção da rede heterogênea a partir dos dados pré-processados na L_1, enquanto a Figura 4.3 ilustra um exemplo visual da rede heterogênea gerada.

Figura 4.3: Exemplo de rede heterogênea. Fonte: Elaboração própria.

Simultaneamente à construção do grafo, realiza-se a extração de *embeddings* para cada documento, utilizando modelos de PLN como SciBERT e MiniLM, previamente treinados. Conforme descrito na Seção 2.2, esses modelos aplicam os princípios de *transfer learning* para gerar representações vetoriais que capturam o conteúdo semântico das publicações, sendo adequadas à tarefa de AND.

Considerando um conjunto de N publicações, cada documento i é descrito por três campos principais: título T_i, resumo R_i e palavras-chave K_i. A extração dos *embeddings* segue três etapas principais:

1. Tokenização: os campos T_i, R_i e K_i são segmentados em sequências de *tokens*, representadas como $\{t_{i,1}, \ldots, t_{i,L_i}\}$, $\{r_{i,1}, \ldots, r_{i,M_i}\}$ e $\{k_{i,1}, \ldots, k_{i,P_i}\}$, onde L_i, M_i e P_i indicam, respectivamente, o número de *tokens* dos campos título, resumo e palavras-chave.

Algoritmo 2 Geração da rede heterogênea a partir de arquivos `.txt` gerados na L_1.

Entrada: Diretório contendo os arquivos `.txt` gerados no pré-processamento
Saída: Objeto de grafo heterogêneo $G = (V, E)$ salvo como arquivo `.pkl`

se arquivo `paper_author.txt` não existir **então**
    retorne Erro: "Arquivo obrigatório `paper_author.txt` não encontrado"
fim se
Inicialize grafo $G \leftarrow (V, E) \leftarrow$ grafo vazio
Inicialize dicionário auxiliar `autores_por_paper` $\leftarrow \{\}$
▷ Leitura de autores e criação de arestas `written_by`
para cada linhas (p, a) em `paper_author.txt` **faça**
    Adicione nó $p \in P$ do tipo `paper`
    Adicione nó $a \in A$ do tipo `author`
    Adicione aresta (p, a) com rótulo `written_by`
    Adicione a à lista `autores_por_paper[p]`
fim para
▷ Criação de arestas de coautoria entre autores de um mesmo paper
para cada $p \in$ `autores_por_paper` **faça**
    `autores` $\leftarrow$ `autores_por_paper[p]`
    para cada pares (a_1, a_2) em `autores` com $a_1 \neq a_2$ **faça**
        se aresta (a_1, a_2) não existe **então**
            Adicione aresta (a_1, a_2) com rótulo `co_authored`
        fim se
    fim para
fim para
▷ Leitura de títulos
para cada linhas (p, t) em `paper_title.txt` **faça**
    Crie nó $t \in T$ com ID `title_p`
    Adicione aresta (p, t) com rótulo `has_title`
fim para
se `paper_abstract.txt` existe **então**
    para cada linhas (p, r) **faça**
        Crie nó $r \in R$ com ID `abstract_p`
        Adicione aresta (p, r) com rótulo `has_abstract`
    fim para
fim se
se `paper_keywords.txt` existe **então**
    para cada linhas (p, w) **faça**
        Adicione nó $w \in W$
        Adicione aresta (p, w) com rótulo `contains`
    fim para
fim se
se `paper_venue.txt` existe **então**
    para cada linhas (p, c) **faça**
        Adicione nó $c \in C$
        Adicione aresta (p, c) com rótulo `published_in`
    fim para
fim se
se `paper_author_names.txt` existe **então**
    para cada linhas $(p, names)$ **faça**
        Crie nó com ID `author_names_p` com conteúdo *names*
        Adicione aresta $(p, names)$ com rótulo `has_author_names`
    fim para
fim se
se `paper_organization.txt` existe **então**
    para cada linhas (p, f) **faça**
        Adicione nó $f \in F$ com ID `org_f`
        Adicione aresta (p, f) com rótulo `affiliated_with`
    fim para
fim se
Construa o índice de arestas (*edge index*) a partir das conexões E
Salve o grafo G (incluindo o *edge index*) como arquivo `.pkl`

2. Geração dos *embeddings*: os *tokens* são processados por modelos do tipo *Transformer*, que produzem vetores de dimensão fixa capazes de representar semanticamente os textos de entrada.

A operacionalização completa do processo de geração de representações vetoriais está formalizada no Algoritmo 3, o qual descreve as etapas de carregamento dos dados textuais, concatenação dos campos selecionados, *tokenização* e extração dos vetores semânticos por meio de um modelo de PLN. O resultado é uma matriz de *embeddings* $E \in \mathbb{R}^{N \times d}$, na qual cada vetor $v_i \in \mathbb{R}^d$ representa semanticamente uma publicação i. Essa matriz é armazenada em um arquivo `.pkl`, indexada pelos respectivos identificadores dos documentos.

Conforme apresentado anteriormente, a extração de *embeddings* utiliza o modelo de PLN baseado em *transformers*. No BERT, a saída da função `last_hidden_state` fornece um vetor de representação para cada *token* da sequência. Para obter um único vetor fixo que represente o texto como um todo, aplicou-se a técnica de *mean pooling*, isto é, a média dos vetores de todos os *tokens*. Essa estratégia permite resumir as informações semânticas distribuídas na sequência em uma única representação, adequada para tarefas de similaridade e agrupamento. Desta forma, na linha 9 do Algoritmo 3, tem-se que a geração do *embedding* é $v_p \leftarrow \texttt{M}(d_p).\texttt{last\_hidden\_state.mean}$.

Algoritmo 3 Extração de *embeddings* com modelo de PLN

Entrada: Arquivo do modelo pré-treinado M, diretório D contendo arquivos `.txt` (ex: `paper_title.txt`, `paper_abstract.txt`, `paper_venue.txt`, ...), lista de campos selecionados F
Saída: Arquivo `.pkl` contendo a matriz de embeddings $E \in \mathbb{R}^{N \times d}$, indexada por ID de publicação
1: Carregue o tokenizador e o modelo M
2: Inicialize dicionário vazio `paper_vec`
3: **para cada** arquivos de entrada $f \in F$ **faça**
4: Carregue os pares (p, f_p) do arquivo `paper_`f`.txt`, onde p é o ID da publicação e f_p o conteúdo textual
5: **fim para**
6: **para cada** IDs de publicações p **faça**
7: Concatene os campos disponíveis para p: $d_p \leftarrow \texttt{title}_p + \texttt{abstract}_p + \texttt{venue}_p$
8: Tokenize d_p com truncamento e padding
9: Gere o embedding $v_p \leftarrow \texttt{M}(d_p).\texttt{last\_hidden\_state.mean}$
10: `paper_vec[p]` $\leftarrow v_p$
11: **fim para**
12: Construa a matriz $E \in \mathbb{R}^{N \times d}$ a partir dos vetores em `paper_vec`, indexada por ID
13: Salve a matriz E no arquivo `.pkl`

A Figura 4.4 ilustra uma matriz de *embeddings* gerada para cinco publicações fictícias. O eixo vertical indica os identificadores das publicações, enquanto o eixo horizontal representa as dimensões dos vetores semânticos. O valor de d, ou seja, o número de dimensões, varia conforme o modelo de PLN adotado. Cada célula da matriz contém um valor numérico que expressa a intensidade da contribuição daquela dimensão específica na representação da publicação. Neste exemplo, os valores simulados seguem uma distribuição normal, variando aproximadamente entre $-3{,}0$ e $2{,}0$. Tons mais escuros indicam valores mais baixos, e tons mais claros, valores mais altos.

Valores mais altos em determinadas dimensões indicam maior intensidade numérica atribuída pelo modelo, mas isso não implica necessariamente maior relevância semântica. A interpretação depende de fatores como topologia da rede, ordem de apresentação dos dados e parâmetros de treinamento. Ainda assim, ativações elevadas podem, em alguns contextos, refletir padrões capturados pelo modelo, embora seu significado exato não seja diretamente interpretável.

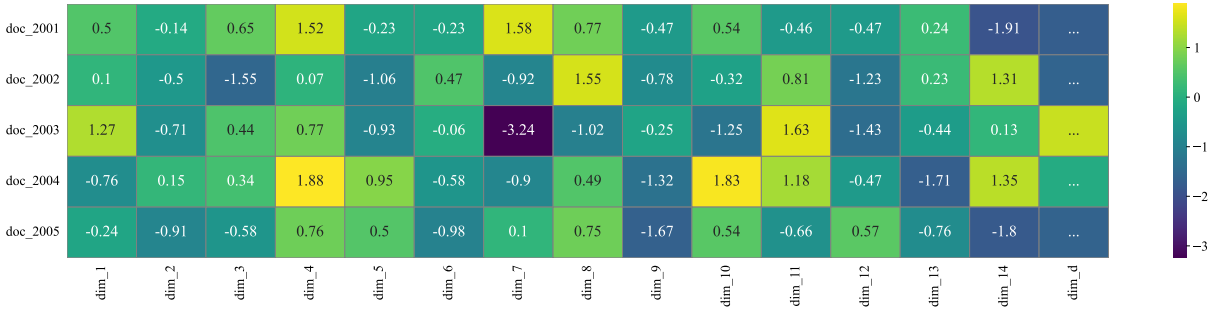


Figura 4.4: Matriz de *embeddings* semânticos das publicações gerada por modelo de PLN. Fonte: Elaboração própria.

Ao final dos procedimentos realizados na camada L_2, são produzidos dois insumos essenciais que serão encaminhados à camada L_3: (i) a rede heterogênea representada por um índice de arestas (*edge index*) e (ii) a matriz de *embeddings* das publicações, contendo as representações semânticas extraídas com PLN. Esses elementos servem como entrada para os mecanismos de fusão e processamento estruturados na próxima camada do ADAN.

L_3 — Aprendizado com RCG

A função desta camada é integrar as representações semânticas extraídas por modelos de PLN com a estrutura do grafo heterogêneo construída na etapa anterior. Essa integração é realizada pelo módulo de fusão de *embeddings*, que combina informações textuais e relacionais, gerando as entradas necessárias para o modelo de RCG. O objetivo é enriquecer a representação dos nós com informações semânticas e topológicas, ampliando a capacidade de generalização da rede.

O módulo de fusão recebe como entrada a matriz de *embeddings* das publicações, gerada na camada L_2, e a estrutura do grafo heterogêneo $G = (V, E)$, representada por um índice de arestas. O conjunto V representa todos os nós do grafo, incluindo publicações, autores, palavras-chave, veículos e instituições, enquanto $E \subset V \times V$ define suas conexões. Denotamos por $P \subset V$ o subconjunto de nós que representam publicações, ou seja, os documentos a serem desambiguados na tarefa de AND.

Como resultado, são produzidos dois componentes principais:

1. A matriz de atributos dos nós $X \in \mathbb{R}^{|V| \times F}$, em que cada linha representa um vetor de características de um nó $v \in V$. Os nós do subconjunto $P \subset V$, que correspondem às publicações, recebem os *embeddings* semânticos extraídos anteriormente. Os demais nós, que não possuem representação textual, são inicializados com vetores nulos. A variável F corresponde à dimensão dos *embeddings* semânticos extraídos pelo PLN.

2. O índice de arestas E, que preserva a topologia da rede construída na etapa anterior e será utilizado durante a propagação.

A operacionalização completa desse processo está formalizada no Algoritmo 4, que descreve as etapas de carregamento do grafo heterogêneo, identificação dos nós de publicação com *embeddings* disponíveis, construção da matriz de atributos dos nós X com vetores semânticos ou vetores nulos, e extração do índice de arestas E. Esses dois componentes compõem a entrada principal do modelo de RCG na próxima etapa.

Com X e E definidos, a RCG realiza um processo de propagação de mensagens entre os nós do grafo. A cada camada ℓ, os vetores de atributos são atualizados de acordo com a seguinte regra:

$$H^{(\ell)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(\ell-1)}W^{(\ell)}\right),$$

em que $H^{(0)} = X$, $\tilde{A}$ é a matriz de adjacência com laços próprios, $\tilde{D}$ é a matriz diagonal dos graus dos nós, $W^{(\ell)}$ são os pesos treináveis da camada ℓ, e $\sigma(\cdot)$ representa uma função de ativação. Neste trabalho, adota-se a função ReLU, definida por:

$$\sigma(x) = \max(0, x).$$

Algoritmo 4 Fusão dos *embeddings* com o grafo heterogêneo salvo na camada L_2

Entrada: Arquivo `.pkl` com grafo heterogêneo $G = (V, E)$; matriz de embeddings $E \in \mathbb{R}^{N \times d}$ contendo *embeddings* dos nós de publicação $P \subset V$ (`embeddings`)
Saída: Matriz de atributos dos nós $X \in \mathbb{R}^{|V| \times F}$; índice de arestas $E \subset V \times V$

Carregue o grafo heterogêneo $G = (V, E)$ a partir do arquivo `.pkl`
Recupere a lista de nós $V = \{v_1, v_2, \ldots, v_{|V|}\}$
Recupere o índice de arestas $E \subset V \times V$ salvo em G
Inicialize lista vazia `features`
▷ Determinação da dimensão dos embeddings
$F \leftarrow$ dimensão do primeiro vetor em `embeddings`
▷ Construção da matriz de atributos dos nós
para cada $v \in V$ **faça**
    se $v \in P$ e $v \in$ `embeddings` **então**
        $x_v \leftarrow$ `embeddings`$[v]$
    senão
        $x_v \leftarrow \vec{0}_F$ ▷ Vetor nulo para nós sem embedding
    fim se
    Adicione x_v à lista `features`
fim para
Construa matriz $X \in \mathbb{R}^{|V| \times F}$ a partir de `features`
retorne matriz de atributos X, índice de arestas E

Ao final de L camadas, a rede produz uma matriz $H^{(L)} \in \mathbb{R}^{|V| \times F'}$, em que cada linha contém a representação refinada de um nó do grafo. O valor F' indica a nova dimensionalidade dos *embeddings* aprendida pela RCG, resultante da combinação entre conteúdo semântico e estrutura relacional.

O treinamento é orientado por uma função de perda do tipo MSE, definida por:

$$\mathcal{L} = \frac{1}{|V|} \sum_{i=1}^{|V|} \|Z_i - X_i\|^2 ,$$

em que Z_i é o vetor final gerado para o nó i, e X_i é o vetor de entrada original. A otimização dos pesos é realizada por meio do algoritmo Adam (Kingma and Ba, 2015), com taxa de aprendizado ajustável.

A escolha do otimizador Adam se justifica por sua eficiência na convergência e boa capacidade de generalização em diferentes arquiteturas de redes neurais. Estudos como o de Makinde (2024) demonstram que o Adam supera outros otimizadores em termos de acurácia e velocidade de convergência, mesmo em tarefas complexas como a previsão de séries temporais. Essas vantagens também podem justificá-lo como uma escolha adequada no contexto de aprendizado sobre grafos.

O Algoritmo 5 apresenta o procedimento completo de treinamento da RCG, utilizando os dados produzidos pela etapa de fusão. A complexidade computacional do treinamento cresce linearmente com o número de épocas T, o número de camadas L, o número de arestas $|E|$ e a dimensão dos vetores de entrada F. A convolução em cada camada tem custo $\mathcal{O}(|E| \cdot F)$, e o cálculo da perda introduz um custo adicional de $\mathcal{O}(|V| \cdot F)$ por época. Assim, o custo total do treinamento pode ser estimado por:

$$\mathcal{O}(T \cdot L \cdot |E| \cdot F).$$

Essa estimativa está alinhada com análises anteriores da complexidade computacional de RCG, que apontam a convolução por camada como tendo custo $\mathcal{O}(|E| \cdot F)$, e o treinamento completo como proporcional a $\mathcal{O}(L \cdot |E| \cdot F)$ por época (Blakely et al., 2021).

Embora a RCG gere representações para todos os nós do grafo, apenas os *embeddings* referentes às publicações são utilizados na etapa seguinte. A submatriz $Z_{\text{doc}} \in \mathbb{R}^{|P| \times F'}$, que contém os vetores refinados dos nós de publicação, é encaminhada à camada L_4, responsável pelo agrupamento dos documentos com base em suas representações vetoriais.

L_4 — Agrupamento com GHAC

Essa camada recebe como entrada as representações finais dos nós geradas pela RCG, conforme citado anteriormente. A matriz $H^{(L)} \in \mathbb{R}^{|V| \times F'}$ contém os *embeddings* refinados

Algoritmo 5 Treinamento da RCG sobre grafo heterogêneo

Entrada: Matriz de atributos dos nós $X \in \mathbb{R}^{|V| \times F}$, arestas E, número de camadas L, número de épocas T
Saída: Embeddings refinados dos nós $Z \in \mathbb{R}^{|V| \times F'}$

Inicializar pesos $W^{(1)}, \ldots, W^{(L)}$ da RCG
para época $t = 1$ até T **faça**
    $H^{(0)} \leftarrow X$
    para camada $\ell = 1$ até L **faça**
        $H^{(\ell)} \leftarrow \text{ReLU}\left(\text{GCNConv}(H^{(\ell-1)}, E;\ W^{(\ell)})\right)$
    fim para
    $Z \leftarrow H^{(L)}$
    Calcular a perda: $\mathcal{L} \leftarrow \frac{1}{|V|} \sum_{v \in V} \|Z_v - X_v\|^2$
    Atualizar pesos $W^{(\ell)}$ com otimizador Adam
fim para
retorne Z

de todos os nós do grafo heterogêneo. A partir dessa matriz, são extraídos apenas os vetores dos nós correspondentes às publicações, denotados por $\{z_i\}_{i=1}^{n} \subset \mathbb{R}^{F'}$, que servirão como entrada para o processo de agrupamento.

Essa etapa aplica o algoritmo GHAC, conforme a definição apresentada na Seção 2.3, iniciando com cada publicação em um *cluster* individual $C_i = \{i\}$. A cada iteração, o GHAC calcula a similaridade entre todos os pares de *clusters* com base na média da similaridade cosseno entre os vetores de *embeddings* das publicações pertencentes aos *clusters*. Em seguida, os dois *clusters* mais similares são fundidos, e as similaridades são atualizadas. O processo de fusão continua até que o número de *clusters* atinja K, valor previamente definido com base no número de autores distintos associados ao nome ambíguo, conforme o conjunto de referência. Embora os rótulos não sejam usados na formação dos grupos, K é conhecido e define o critério de parada. A Listagem 4.2 exemplifica esse cenário com cinco publicações atribuídas ao nome ambíguo "J. Zhang", que apresentam três rótulos distintos (`label`), 0,1 e 2. Assim, define-se $K = 3$ como o número de agrupamentos esperados. O Algoritmo 6 descreve o funcionamento do GHAC.

O procedimento descrito no Algoritmo 6 segue os princípios gerais do HAC proposto por Qiao et al. (2019), mas difere no cálculo das similaridades. Em vez de reconstruir um grafo esparso e considerar apenas nós diretamente conectados, esta abordagem calcula a similaridade cosseno entre todos os pares de *embeddings* das publicações. Para n documentos em um bloco de desambiguação, com vetores $z_i \in \mathbb{R}^{F'}$ representando as publicações, essa operação resulta em uma matriz de similaridade densa construída com $\binom{n}{2}$ comparações em pares, cada uma com custo $\mathcal{O}(F')$. O número de combinações de n elementos tomados 2 a 2 é dado por $\binom{n}{2} = \frac{n(n-1)}{2}$. Assim, o custo total da etapa inicial é $\mathcal{O}(n^2 F')$.

Apesar de exigir maior custo computacional do que variantes baseadas em grafos esparsos, esta versão do GHAC simplifica o fluxo de execução, eliminando a reconstrução de grafos e integrando-se diretamente às representações geradas pela RCG (Qiao et al., 2019).

Ao final, cada grupo de publicações é associado a um autor distinto, e os rótulos preditos são comparados com os rótulos reais para avaliação do desempenho, conforme descrito na Seção 2.5. Os resultados da desambiguação são, então, enviados à camada L_1, responsável por exibi-los ao usuário via GUI.

Listagem 4.2: Exemplo de publicações com nome de autor ambíguo (“J. Zhang”).

```
[{
  "id": 201,
  "title": "Neural Approaches to NLP",
  "abstract": "...",
  "authors": ["J. Zhang", "Alice Smith"],
  "label": 0
 },
 {
  "id": 202,
  "title": "Computer Vision and Transformers",
  "abstract": "...",
  "authors": ["J. Zhang", "Wei Liu"],
  "label": 1
 },
 {
  "id": 203,
  "title": "Graph Neural Networks for Science",
  "abstract": "...",
  "authors": ["J. Zhang", "Alice Smith"],
  "label": 0
 },
 {
  "id": 204,
  "title": "Federated Learning in Healthcare",
  "abstract": "...",
  "authors": ["J. Zhang", "Jane Kim"],
  "label": 2
 },
 {
  "id": 205,
  "title": "Contrastive Learning Methods",
  "abstract": "...",
  "authors": ["J. Zhang", "Wei Liu"],
  "label": 1
 }]
```

As Figuras 4.5, 4.6 e 4.7 ilustram a GUI do ADAN, projetada para oferecer um ambiente interativo na execução da tarefa de AND. A primeira tela (Figura 4.5) permite selecionar as características a serem utilizadas, como título, resumo e nomes de autores, e acionar cada etapa do *framework*.

Em seguida, a tela de extração de *embeddings* (Figura 4.6) permite ao usuário escolher o modelo de PLN, definir o diretório de dados e marcar os campos textuais desejados.

A tela de treinamento da RCG (Figura 4.7) possibilita configurar o número de camadas e épocas, além de carregar os arquivos de grafo e *embeddings* necessários para o aprendizado. Como os hiperparâmetros podem impactar o desempenho da tarefa de AND de forma variável, o ADAN oferecerá flexibilidade para ajustes conforme as características do conjunto de dados utilizado. O repositório final da proposta inclui exemplos de configurações recomendadas para os conjuntos avaliados.

Algoritmo 6 GHAC

Entrada: Conjunto de embeddings $\{z_i\}_{i=1}^{n}$, número de clusters desejado K

Saída: Rótulos de cluster $\{y_i\}_{i=1}^{n}$

1: Inicializar os clusters $\mathcal{C} \leftarrow \{C_1, C_2, \ldots, C_n\}$, onde $C_i = \{i\}$

2: **para cada** par (C_i, C_j) com $i \neq j$ **faça**

3: Calcular a similaridade cosseno média:

$$\text{sim}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{u \in C_i} \sum_{v \in C_j} \frac{z_u^\top z_v}{\|z_u\| \cdot \|z_v\|}$$

4: **fim para**

5: **enquanto** $|\mathcal{C}| > K$ **faça**

6: Selecionar o par mais similar:

$$(C_p, C_q) = \arg\max_{(C_i, C_j)} \text{sim}(C_i, C_j)$$

7: Fundir: $C_{\text{novo}} \leftarrow C_p \cup C_q$

8: Atualizar $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_p, C_q\} \cup \{C_{\text{novo}}\}$

9: Recalcular as similaridades entre C_{novo} e os demais clusters

10: **fim enquanto**

11: Atribuir a cada publicação i o rótulo y_i conforme seu cluster final

12: **retorne** $\{y_i\}_{i=1}^{n}$

4.2 Tecnologias Utilizadas

As tecnologias e linguagens utilizadas para o desenvolvimento do ADAN incluem:

Figura 4.5: Tela inicial do ADAN, com seleção de atributos e acesso às etapas do *framework*. Fonte: Elaboração própria.

- Linguagem *Python*[2]: usada como a principal linguagem de programação na implementação em todas as etapas do ADAN. A escolha do *Python* deve-se à sua popularidade e riqueza de bibliotecas para processamento de texto, aprendizado de máquina, grafos, GUI e *PLN*.
- *Transformers*[3]: biblioteca da HuggingFace empregada na etapa de extração de *embeddings* semânticos de publicações, por meio de modelos como BERT (variação Sci-

[2] `https://www.python.org/`
[3] `https://huggingface.co/docs/transformers`

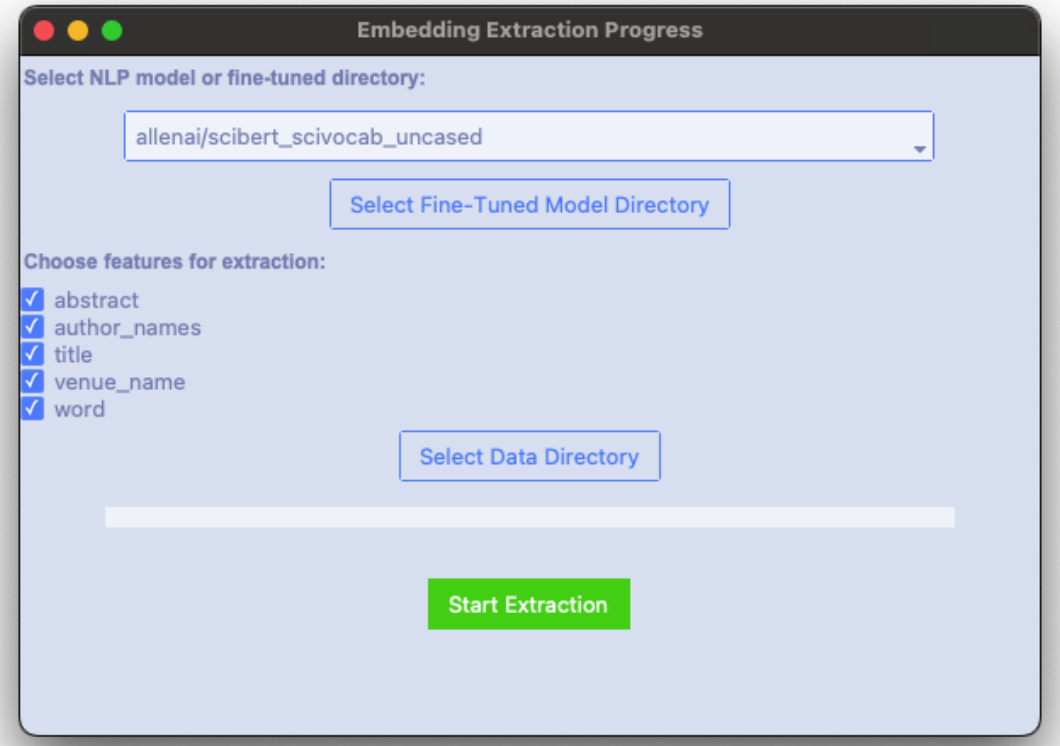


Figura 4.6: Tela para extração de *embeddings*, com escolha do modelo de PLN e dos campos textuais. Fonte: Elaboração própria.

BERT [4]) e MiniLM [5]. Esses modelos foram escolhidos por suas propriedades complementares. O SciBERT, pré-treinado com textos científicos, demonstrou maior eficácia que o BERT em tarefas acadêmicas (Beltagy et al., 2019). Já o MiniLM, além de compacto, mantém desempenho competitivo com custo computacional reduzido em relação a modelos legados como o BERT (Wang et al., 2020c).

- *PyTorch*[6]: biblioteca de aprendizado profundo utilizada na implementação do modelo de RCG, oferecendo ferramentas para operações matriciais, retropropagação e otimização.
- *PyTorch Geometric*[7]: extensão do PyTorch para aprendizado em grafos, utilizada na propagação de informações e no aprendizado de representações em grafos heterogêneos

[4] `https://huggingface.co/allenai/scibert_scivocab_uncased`
[5] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`
[6] `https://pytorch.org/`
[7] `https://pytorch-geometric.readthedocs.io/`

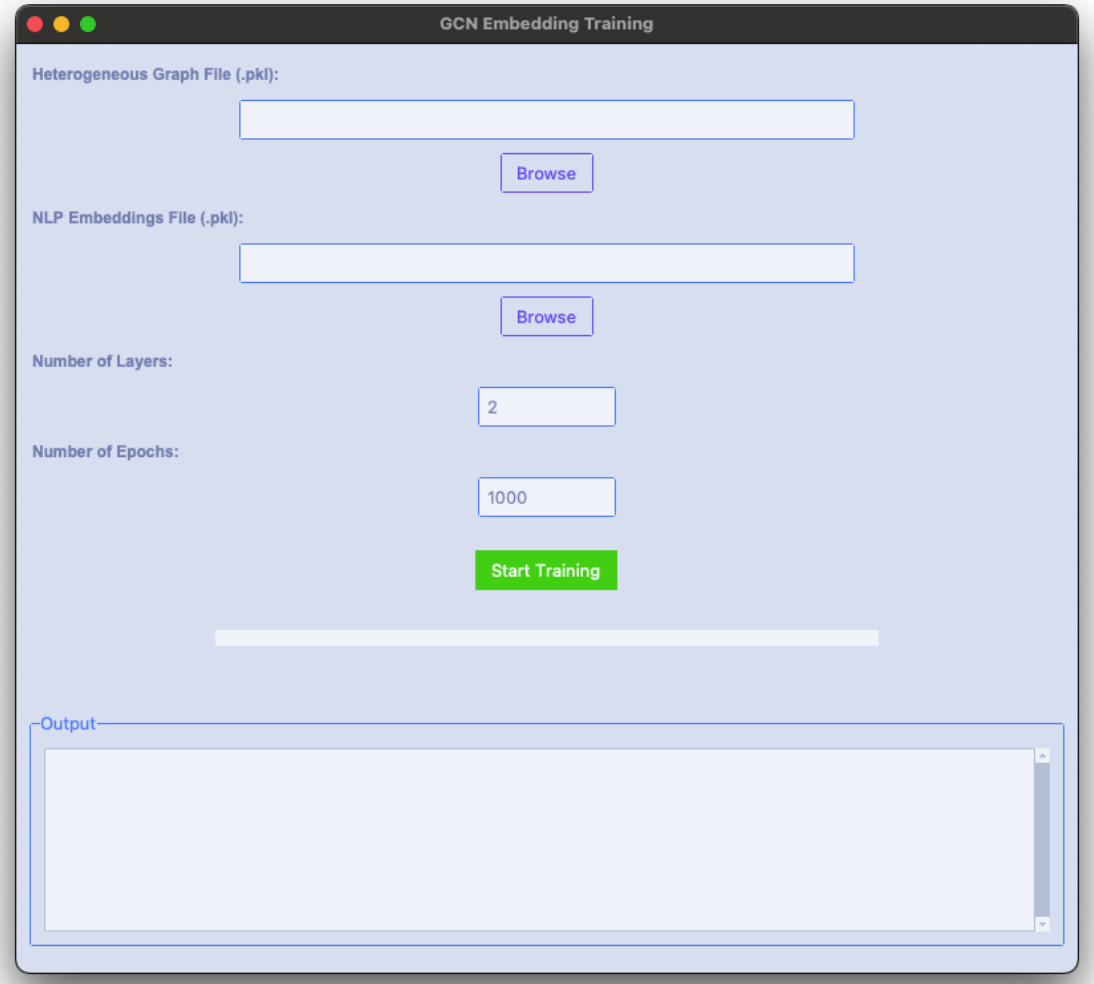


Figura 4.7: Tela de configuração da RCG, permitindo ajuste de camadas e número de épocas. Fonte: Elaboração própria.

- *Tkinter*[8]: biblioteca padrão do Python para construção de interfaces gráficas, permitindo a interação visual com o *framework*.
- *ttkbootstrap*[9]: pacote que estende o *Tkinter* com temas modernos e responsivos, utilizado para estilização da GUI.
- *NumPy*[10]: biblioteca para computação numérica vetorial, utilizada para manipulação de matrizes e operações lineares ao longo do pipeline.

[8] `https://docs.python.org/3/library/tkinter.html`
[9] `https://ttkbootstrap.readthedocs.io/`
[10] `https://numpy.org/`

- *SciPy*[11]: biblioteca complementar ao NumPy, utilizada em operações matemáticas avançadas e no cálculo de distâncias e similaridades.
- *Pandas*[12]: biblioteca para análise e organização de dados tabulares, especialmente útil na manipulação de conjuntos de publicações.
- *Scikit-learn*[13]: biblioteca amplamente empregada em tarefas de pré-processamento, cálculo de métricas, e como alternativa para métodos de agrupamento.
- *NetworkX*[14]: biblioteca para criação e análise de redes complexas, empregada na modelagem das redes de coautoria e estruturação do grafo heterogêneo.
- *tqdm*[15]: ferramenta de barra de progresso utilizada para monitoramento de etapas demoradas, como a extração de *embeddings* e o agrupamento.
- *pickle*,[16] *json*,[17] *os*,[18] *threading*,[19] *subprocess*,[20] *re*:[21] módulos da biblioteca padrão do Python utilizados para serialização, leitura de arquivos, manipulação do sistema operacional, controle de concorrência e processamento textual.

4.3 Discussão

Segundo Kipf and Welling (2016), a formulação original da RCG foi concebida para classificação de dados rotulados para o treinamento das redes. Além disso, uma matriz de adjacência é necessária para definir a relação entre os dados de entrada, resultando que todos os dados, incluindo os de treinamento, validação e teste, normalmente formem apenas uma estrutura de grafo para treinamento. No entanto, para melhorar a capacidade de aprendizado e o desempenho do modelo sob dados de treinamento limitados, faz-se necessário utilizar estratégias de aprendizado auto-supervisionado, capazes de explorar as informações disponíveis a partir dos próprios dados de estrutura de grafos de entrada.

trabalho, o *framework* ADAN adota um modelo de aprendizado auto-supervisionado, onde a RCG não realiza a classificação de documentos, mas atua no refinamento das representações no grafo gerado a partir da entrada de dados. Cada publicação é inicialmente

[11] `https://scipy.org/`
[12] `https://pandas.pydata.org/`
[13] `https://scikit-learn.org/`
[14] `https://networkx.org/`
[15] `https://tqdm.github.io/`
[16] `https://docs.python.org/3/library/pickle.html`
[17] `https://docs.python.org/3/library/json.html`
[18] `https://docs.python.org/3/library/os.html`
[19] `https://docs.python.org/3/library/threading.html`
[20] `https://docs.python.org/pt-br/3/library/subprocess.html`
[21] `https://docs.python.org/3/library/re.html`

associada a um *embedding* gerado utilizando técnicas de PLN baseadas em *transformers*, que compõe o vetor de atributos do nó correspondente na rede heterogênea, formada por diferentes tipos de nós (publicação, autor, co-autor, titulo, resumo, palavra-chave, veículo, afiliação). A partir da rede heterogênea, a RCG propaga informações pela estrutura do grafo, de modo que cada nó atualize sua representação ao combinar seu próprio vetor com os de seus vizinhos, resultando em *embeddings* enriquecidos.

Conforme exposto, o treinamento da RCG ocorre sem rótulos manuais, utilizando a função de perda MSE entre os vetores de entrada e os de saída, no qual o próprio dado fornece o sinal de supervisão. Nesta etapa da arquitetura do ADAN, a RCG é responsável apenas por ajustar as representações. A validação do treinamento ocorre com o algoritmo de agrupamento hierárquico aprimorado GHAC, responsável pela desambiguação entre os autores, uma vez que recebe os *embeddings* refinados pela saída da RCG.

Assim, o método distingue-se das abordagens supervisionadas tradicionais, que requerem a divisão em conjuntos de treino, validação e teste e dependem de funções de perda definidas em relação a rótulos manuais. No presente trabalho, não há divisão de dados de treinamento e teste durante o aprendizado da RCG. Além disso, a validação é realizada em um momento posterior, quando os agrupamentos gerados pelo GHAC são comparados com os rótulos de referência (*ground truth*) dos conjuntos de dados.

Estudos anteriores, como Zhu et al. (2020), propõe estratégias de aprendizado auto-supervisionado para explorar as informações disponíveis a partir dos próprios dados de estrutura de grafos de entrada, sem depender de grandes volumes de dados anotados. Os resultados experimentais demonstraram a capacidade de generalização, bem como a portabilidade das estratégias propostas, podendo melhorar o desempenho das RCG no aprendizado de recursos. Desta forma, os resultados do estudo reforçam a viabilidade do aprendizado auto-supervisionado em RCG, tal como utilizada no *framework* ADAN.

Considerando a resposta à QP1, a forma que o *framework* híbrido ADAN provê um método eficaz para a tarefa de AND, abrange a estratégia de aprendizado auto-supervisionado com a combinação de técnicas de aprendizado de máquina profundo. As técnicas incluem um modelo de PLN baseado em *transformers* para extração de *embeddings* a partir de conteúdos semânticos dos textos das publicações. Esses *embeddings* são incorporados a uma rede heterogênea formada por publicações, autores e outras informações bibliográficas. A RCG propaga essas representações no grafo, permitindo que cada nó refine seu vetor ao combinar sua própria informação com a dos vizinhos. O treinamento é auto-supervisionado, pois a rede ajusta os parâmetros minimizando o MSE entre os vetores de entrada e os produzidos, sem necessidade de rótulos manuais. Na etapa seguinte, o algoritmo GHAC, recebe os *embeddings* refinados e realiza o agrupamento hierárquico das publicações, separando autores distintos em *clusters* de autores

desambiguados.

Dessa forma, a combinação das técnicas utilizadas no ADAN cumpre um papel específico e complementar: modelos de PLN baseados em *transformers* fornecem informação semântica, a RCG enriquece as representações explorando a estrutura do grafo e o GHAC efetiva a desambiguação com agrupamentos coerentes. Assim, o *framework* híbrido provê um método eficaz para a tarefa de AND em repositórios bibliográficos digitais.

O Capítulo 5 apresenta os experimentos conduzidos para validar a proposta do *framework* ADAN.

Capítulo 5

Experimentos

Este capítulo apresenta os experimentos realizados, incluindo os conjuntos de dados utilizados nas tarefas de AND, a configuração experimental para utilização dos trabalhos de referência (*baselines*), adotados para a avaliação comparativa do *framework*. Também foi conduzido um estudo de ablação com o objetivo de analisar o impacto de diferentes componentes no desempenho geral do ADAN. Por fim, são apresentados os resultados obtidos, seguidos de discussões e das limitações observadas ao longo da avaliação.

Como consequência, é apresentada a resposta à QP2, formulada na Seção 1.3:

> QP2: Em que medida o *framework* híbrido proposto para AND apresenta desempenho superior aos trabalhos de referência na literatura?

5.1 Conjuntos de Dados

O *framework* proposto foi avaliado utilizando três conjuntos de dados comumente utilizados como referência em tarefas de AND: AMiner-12,[1] DBLP[2] e LAGOS-AND.[3] Esses conjuntos diferem entre si em relação à complexidade da ambiguidade, à disponibilidade dos metadados e à escala, permitindo uma avaliação abrangente do desempenho do *framework* em diferentes cenários.

O AMiner-12 (Tang et al., 2012) inclui 109 nomes ambíguos, abrangendo 7.447 publicações atribuídas a 1.546 indivíduos distintos. Disponibiliza metadados estruturados, como título, coautores, afiliação institucional e veículo de publicação, o que possibilita a extração de características contextuais e relacionais relevantes.

A versão utilizada do DBLP (Qian et al., 2015) é composta por 6.478 artigos vinculados a 679 nomes ambíguos e 1.463 autores distintos. Seus metadados incluem título,

[1] `https://www.aminer.cn/disambiguation`
[2] `https://github.com/yaya213/DBLP-Name-Disambiguation-Dataset`
[3] `https://zenodo.org/records/7313380`

resumo, veículo de publicação e vínculos de coautoria, o que permite uma modelagem tanto semântica quanto estrutural.

Para a avaliação da eficiência e padrões de publicações distintos, foi utilizado o LAGOS-AND-BLOCK-TRIMMED, uma versão do conjunto LAGOS-AND (Zhang et al., 2023) (neste documento referenciado somente como LAGOS-AND). O LAGOS-AND foi desenvolvido como um conjunto de dados em larga escala para *benchmarking* de sistemas de AND sob condições realistas e heterogêneas. O LAGOS-AND usa diferentes repositórios bibliográficos digitais, como o PubMed, Microsoft Academic Graph, and Semantic Scholar, contemplando diferentes áreas de conhecimento como a Medicina, Matemática, Ciência da Computação e Biomedicina.

A versão utilizada neste trabalho contém milhares de blocos de desambiguação extraídos de bibliotecas digitais reais, apresentando ampla variedade de metadados, incluindo título, resumo, coautores, palavras-chave, afiliações e veículo de publicação. O LAGOS-AND contém aproximadamente 500.000 publicações, 290.000 autores distintos e mais de 10.000 blocos de nomes ambíguos.

A Tabela 5.1 resume os principais atributos e estatísticas dos conjuntos utilizados neste estudo, destacando a diversidade em termos de escala e cobertura de metadados. Na tabela, utilizou-se o símbolo "+" para indicar a presença do atributo, "-" para sua ausência e "+/-" quando a presença do atributo varia entre os registros.

Tabela 5.1: Conjuntos de dados utilizados nos experimentos.

Atributo	AMiner-12	DBLP	LAGOS-AND
Publicações	7.447	6.478	∼500.000
Autores distintos	1.546	1.463	∼290.000
Nomes ambíguos	109	679	10.000+ blocos
Título	+	+	+
Resumo	-	+	+
Coautores	+	+	+
Palavras-chave	-	+	+/-
Afiliações	+	-	+/-
Veículo	+	+	+/-

Fonte: Elaboração própria.

Em resumo, o AMiner-12 e DBLP constituem *benchmarks* compactos e bem estruturados, adequados para avaliações controladas. Por outro lado, o LAGOS-AND representa um cenário realista de grande escala e alta heterogeneidade, sendo essencial para validar a escalabilidade e a robustez de sistemas de AND.

5.2 Configuração Experimental

Esta seção descreve a configuração experimental utilizada nesta pesquisa, incluindo objetivos dos experimentos, envolvendo os três conjuntos de dados (AMiner-12, DBLP, LAGOS-AND). Na sequência, detalham-se as configurações, incluindo a extração de *embeddings*, o aprendizado com RCG, a estratégia de agrupamento com GHAC, as métricas de validação adotadas (*Pairwise F1, K-metric, B-Cubed*) e os ambientes computacionais utilizados na execução dos experimentos. Para fins de padronização, os experimentos foram conduzidos com configurações de parâmetros iguais, os quais estão disponíveis no repositório do *framework* ADAN.[4]

Objetivos

Os experimentos foram conduzidos com o propósito de avaliar a aplicabilidade e a validade do *framework* ADAN na tarefa de AND, considerando os problemas motivacionais discutidos na Seção 1.2. Para isso, foi adotada uma abordagem empírica, variando os modelos de *transformers* de PLN empregados na extração de *embeddings*, o número de camadas da RCG e as épocas de treinamento, a fim de observar como essas escolhas impactam o desempenho. Para cada conjunto de dados, os experimentos foram realizados sob diferentes configurações com o objetivo de avaliar a flexibilidade do *framework* proposto e seu comportamento em cenários distintos.

Configuração da Extração de *Embeddings*

O módulo de extração de *embeddings* foi testado com dois modelos de PLN: SciBERT (Beltagy et al., 2019) e MiniLM (Wang et al., 2020c), permitindo analisar o impacto da representação semântica na tarefa de AND.

Conforme apresentado na Seção 2.2, neste trabalho, utilizam-se modelos de linguagem baseados em *transformers* para gerar representações vetoriais dos documentos científicos, o SciBERT e o MiniLM. O SciBERT é uma variação do BERT especializada em literatura científica e captura padrões linguísticos e vocabulários específicos do domínio acadêmico, proporcionando representações semânticas mais precisas para textos técnicos (Beltagy et al., 2019). Já o MiniLM, por sua arquitetura compacta, oferece um bom equilíbrio entre eficiência computacional e riqueza semântica (Wang et al., 2020c).

Esses modelos de PLN são empregados para processar os campos de título, resumo e palavras-chave de cada publicação. Os textos desses campos são concatenados em uma única sequência textual, a partir da qual se obtêm representações semânticas contextuais.

[4] `https://github.com/natansr/adan.git`

A dimensão do vetor final $\mathbf{v}_i \in \mathbb{R}^d$ varia conforme o modelo utilizado, por exemplo, $d = 768$ para o SciBERT e $d = 384$ para o MiniLM.

Formalmente, uma publicação p_i, composta por título, resumo e palavras-chave, é representada como uma única sequência textual concatenada. Por exemplo, um documento com título *"Author Name Disambiguation"*, resumo *"A method to solve AND in bibliographic datasets"* e palavras-chave associadas pode ser organizado da seguinte forma:

p_i = "Author Name Disambiguation. A method to solve AND in bibliographic datasets."

Essa sequência é tokenizada e truncada automaticamente para até 512 *tokens*, respeitando os limites da arquitetura do modelo. Caso contenha menos *tokens*, o preenchimento (*padding*) é aplicado de forma automática.

O modelo processa os *tokens* $t_{i,j}$ e gera *embeddings* $\mathbf{e}_{i,j} \in \mathbb{R}^d$ para cada um. A representação final da publicação é obtida pela média dos *embeddings* dos *tokens* válidos:

$$\mathbf{v}_i = \frac{1}{n} \sum_{j=1}^{n} \mathbf{e}_{i,j}, \quad \mathbf{v}_i \in \mathbb{R}^d,$$

em que $n \leq 512$ corresponde ao número de *tokens* considerados após o truncamento, e j representa a posição de cada *token* válido na sequência. Esses vetores $\mathbf{v}_i$ capturam informações semânticas e contextuais relevantes para a tarefa de AND, sendo utilizados como entrada para as etapas subsequentes baseadas em grafos.

Configuração do Aprendizado com RCG

A RCG foi configurada com três profundidades distintas: 1, 2 ou 3 camadas, com o objetivo de avaliar o impacto da profundidade da rede no desempenho. Cada configuração foi submetida a treinos com 1000, 2000 e 3000 épocas, analisando-se como a duração do treinamento influencia a convergência do modelo e a qualidade na tarefa de AND. A escolha desses valores baseou-se em evidências empíricas obtidas em estudos anteriores Rodrigues and Ralha (2024, 2025), nos quais essas faixas mostraram-se eficazes em diferentes cenários de AND. Foram utilizados tamanho de lote 128, função de ativação ReLU, função de perda MSE e otimizador Adam com taxa de aprendizado de 0,0005.

Estratégia de Agrupamento

O processo de agrupamento é realizado exclusivamente pelo método GHAC, apresentado na Seção 2.3. Enquanto o HAC tradicional apenas avalia distâncias entre vetores, o GHAC parte de *embeddings* extraídos da RCG, que já integram informações semânticas dos textos e relações estruturais do grafo heterogêneo, como vínculos entre autores e

publicações. Dessa forma, o agrupamento não considera apenas similaridade num espaço vetorial, mas também a conectividade implícita do grafo. A implementação utiliza a classe *AgglomerativeClustering* da biblioteca *Python scikit-learn*, configurada com distância cosseno pré-computada e *average linkage*, interrompendo o processo no número esperado de autores K.

Métricas de Validação

Nos experimentos, adotaram-se métricas comumente utilizadas na literatura de AND, tais como Pairwise F1, K-Metric e B-Cubed, conforme descrito na Seção 2.5.

Para assegurar rigor experimental, a configuração que apresentou maior incidência de resultados superiores nas métricas de validação, entre diferentes combinações de parâmetros testadas, foi repetida 30 vezes, de modo a avaliar a estabilidade frente às variações aleatórias do treinamento. Nessas repetições, foram calculadas medidas de estatística descritiva (média e desvio padrão) e também de estatística inferencial (intervalo de confiança de 95%). Quando a variância populacional é desconhecida, o intervalo de confiança deve ser construído com a distribuição t de Student (Bolfarine and Sandoval, 2001), definido por:

$$IC_{1-\alpha} = \left(\bar{X} \pm t_{\alpha/2,\, n-1} \frac{s}{\sqrt{n}} \right),$$

em que:

- $\bar{X}$ representa a média amostral dos valores da métrica em 30 execuções;
- s corresponde ao desvio padrão amostral, que quantifica a variação entre as execuções;
- n é o número de repetições independentes realizadas, aqui igual a 30;
- $t_{\alpha/2,\, n-1}$ é o valor crítico da distribuição t de Student com $n-1$ graus de liberdade, associado ao nível de confiança de 95%.

Ambiente de Execução

Os experimentos foram conduzidos em duas configurações de *hardware* distintas, dependendo da escala de cada conjunto de dados. Para os conjuntos DBLP e AMiner-12, utilizou-se um *MacBook Air* com processador M1, 16 GB de memória RAM e 256 GB de armazenamento SSD, executando o *macOS Sequoia*. Para os experimentos com o LAGOS-AND, empregou-se um ambiente de alto desempenho fornecido pelo *Google Colab Pro*, equipado com uma *GPU NVIDIA A100* (40 GB de VRAM), 80 GB de RAM e 220 GB de armazenamento SSD.

5.3 Trabalhos de Referência

Para validar o ADAN na tarefa de AND, foi comparado o seu desempenho com trabalhos de referência identificados na literatura. Foram selecionados estudos com técnicas e conjuntos de dados semelhantes, garantindo uma avaliação objetiva com base nas métricas apresentadas na Seção 2.5. Os trabalhos considerados para comparação são:

- Pooja et al. (2020) (ATGEP): Utiliza grafos de similaridade entre autores e tópicos para tarefa de AND, empregando sobreposição de coautores e similaridade temática para identificar agrupamentos, sem depender de aprendizado de representações.
- Pooja et al. (2021): Propõe um método de aprendizado não supervisionado que integra características relacionais (coautoria) e não relacionais (título, resumo, veículo de publicação) utilizando HAC para a tarefa de AND em conjuntos com rotulagem esparsa.
- Pooja et al. (2022a): Introduz um método que combina RCG com mecanismos de atenção para aprimorar a representação semântica em grafos heterogêneos de documentos, com foco em informações contextuais.
- Zhang et al. (2023) (LAGOS-AND): Propõe um *benchmark* abrangente para AND, com blocos de autores ambíguos em larga escala. Os métodos de referência incluem abordagens tradicionais baseadas em engenharia de atributos (e.g., similaridade de nomes, Jaccard, TF-IDF, Doc2Vec) e um modelo neural com rede *feedforward* treinada com similaridades de título e resumo.

5.4 Resultados

Esta seção apresenta os resultados obtidos utilizando os conjuntos de dados descritos na Seção 5.1, com base na configuração experimental descrita na Seção 5.2 e nos trabalhos de referência listados na Seção 5.3. As métricas de avaliação apresentadas na Seção 2.5 serão utilizadas para avaliar o desempenho do *framework* ADAN sob diferentes configurações e propriedades dos dados, aplicando-se métricas de validação estatística nos resultados utilizando os conjuntos de dados AMiner-12 e DBLP. Na sequência, são analisados e discutidos os resultados experimentais obtidos.

Conjunto de Dados AMiner-12 Conforme apresentado na Tabela 5.2, o MiniLM apresentou o melhor desempenho em todas as configurações. A configuração identificada com maior incidência de resultados superiores nas métricas de validação é o MiniLM, com duas camadas na RCG e 3000 épocas de treinamento, obtendo 0,627 em pR, 0,908 em

AAP e 0,898 em K-Metric. A Tabela 5.3 apresenta os resultados estatísticos obtidos em 30 execuções independentes do *framework* ADAN com essa configuração. Essas execuções foram realizadas para reduzir a influência de erros aleatórios, garantir a reprodutibilidade e a confiabilidade dos resultados, conforme discutido na Seção 5.2 (Métricas de Validação).

Tabela 5.2: Comparação de desempenho entre SciBERT e MiniLM com diferentes configurações de camadas na RCG e épocas de treinamento no conjunto de dados AMiner-12.

Modelo PLN	Camadas RCG	Épocas	pP	pR	pF1	ACP	AAP	K-Metric
SciBERT	3	3000	0,684	0,589	0,616	0,888	0,892	0,890
SciBERT	3	2000	0,684	0,592	0,618	0,885	0,890	0,887
SciBERT	3	1000	0,676	0,555	0,592	0,881	0,879	0,879
SciBERT	2	3000	0,688	0,596	0,623	0,887	0,891	0,889
SciBERT	2	2000	0,681	0,589	0,615	0,886	0,888	0,886
SciBERT	2	1000	0,694	0,577	0,614	0,884	0,884	0,884
SciBERT	1	3000	0,689	0,600	0,625	0,889	0,895	0,891
SciBERT	1	2000	0,684	0,599	0,623	0,891	0,891	0,891
SciBERT	1	1000	0,687	0,581	0,612	0,884	0,886	0,885
MiniLM	3	3000	0,748	0,622	0,664	0,879	0,901	0,889
MiniLM	3	2000	0,738	0,622	0,658	0,880	0,896	0,887
MiniLM	3	1000	0,708	0,586	0,625	0,861	0,885	0,872
MiniLM	**2**	**3000**	0,753	**0,627**	0,668	0,890	**0,908**	**0,898**
MiniLM	2	2000	0,748	0,621	0,663	0,880	0,899	0,889
MiniLM	2	1000	0,732	0,604	0,646	0,872	0,891	0,881
MiniLM	1	3000	0,743	0,625	0,663	**0,891**	0,902	0,896
MiniLM	1	2000	**0,757**	0,622	**0,669**	0,884	0,902	0,892
MiniLM	1	1000	0,728	0,603	0,645	0,878	0,894	0,886

Fonte: Elaboração própria.

Com base nas métricas de validação estatística apresentadas na Tabela 5.3, considerando o resultado médio de 30 execuções, o desempenho do ADAN foi comparado com diferentes trabalhos de referência identificados na literatura apresentando a variação percentual do ADAN na Tabela 5.4. Note que os melhores resultados são apresentados por Pooja et al. (2020) para pP (0,836) e AAP (0,909), Pooja et al. (2021) para pR (0,844) e pF1 (0,784), e ADAN para ACP (0,890) e K-Metric (0,898). Os maiores ganhos do ADAN foram de 37,65% na métrica de ACP (de 0,647 para 0,890) e 20,21% no K-Metric (de 0,747 para 0,898), apresentando qualidade superior de agrupamento, mesmo em um cenário desafiador com metadados escassos como o AMiner-12 (não inclui resumo, depende majoritariamente de título e coautoria).

Conforme apresentado na Tabela 5.5, os melhores valores dos trabalhos de referência estão fora dos intervalos de confiança de 95% inferior e superior do ADAN, indicando que não são estatisticamente significantes. Desta forma, apenas os melhores resultados do ADAN (ACP e K-Metric) são estatisticamente significantes (marcados com ✓), pois permanecem dentro dos respectivos intervalos de confiança, confirmando a consistência do desempenho.

Tabela 5.3: Resultados estatísticos das 30 execuções independentes no conjunto de dados AMiner-12 (configuração: MiniLM com duas camadas na RCG e 3000 épocas).

Métrica	Média	Desvio Padrão	IC 95% Inferior	IC 95% Superior
pP	0,7507	0,0052	0,7488	0,7527
pR	0,6328	0,0053	0,6308	0,6348
pF1	0,6717	0,0044	0,6701	0,6734
ACP	0,8908	0,0032	0,8896	0,8920
AAP	0,9063	0,0017	0,9057	0,9069
K-Metric	0,8981	0,0021	0,8973	0,8989

Fonte: Elaboração própria.

Tabela 5.4: Comparação entre diferentes trabalhos de referência com o conjunto de dados AMiner-12. As colunas de ganho mostram a variação percentual do ADAN, considerando a média de 30 execuções.

Métrica	Pooja et al. (2020)	Ganho ADAN(%)	Pooja et al. (2021)	Ganho ADAN(%)	Pooja et al. (2022a)	Ganho ADAN(%)	ADAN
pP	**0,836**	–10,15	0,756	–0,69	0,724	3,69	0,750
pR	0,578	9,47	**0,844**	–25,04	0,751	–15,72	0,632
pF1	0,621	8,17	**0,784**	–14,31	0,715	–6,06	0,671
ACP	0,647	37,65	0,782	13,91	0,810	9,98	**0,890**
AAP	**0,909**	–0,74	0,856	5,88	0,798	13,56	0,906
K-Metric	0,747	20,21	0,814	10,32	0,800	12,26	**0,898**

Fonte: Elaboração própria.

Conjunto de Dados DBLP Como apresentado na Tabela 5.6, o MiniLM registra a maior incidência de resultados superiores nas métricas de validação, com duas camadas na RCG e 1000 épocas de treinamento, obtendo 0,880 em pP, 0,878 em pF1, 0,975 em ACP, 0,978 em AAP e 0,976 no K-Metric. As configurações com 3 camadas e 2000 épocas também apresentaram desempenho semelhante, indicando certa estabilidade do MiniLM com diferentes camadas na RCG. O SciBERT manteve estabilidade entre as configurações, mas ficou abaixo do MiniLM em todas as métricas.

A Tabela 5.7 apresenta os resultados estatísticos do MiniLM com duas camadas na RCG e 1000 épocas de treinamento, repetida em 30 execuções independentes. As médias

Tabela 5.5: Comparação dos melhores resultados obtidos com o conjunto de dados AMiner-12 com os intervalos de confiança (IC) do ADAN. Símbolos: ✓dentro do IC 95% Inferior (I)/Superior (S) e – fora do IC 95%.

Métrica	Melhor Resultado	Trabalhos	IC 95% [I/S]	
pP	0,836	Pooja et al. (2020)	[0,7488/0,7527]	–
pR	0,844	Pooja et al. (2021)	[0,6308/0,6348]	–
pF1	0,784	Pooja et al. (2021)	[0,6701/0,6734]	–
ACP	0,890	ADAN	[0,8896/0,8920]	✓
AAP	0,909	Pooja et al. (2020)	[0,9057/0,9069]	–
K-Metric	0,898	ADAN	[0,8973/0,8989]	✓

Fonte: Elaboração própria.

Tabela 5.6: Comparação de desempenho entre SciBERT e MiniLM com diferentes configurações de camadas na RCG e épocas de treinamento no conjunto de dados DBLP.

Modelo PLN	Camadas RCG	Épocas	pP	pR	pF1	ACP	AAP	K-Metric
SciBERT	3	3000	0,819	0,816	0,813	0,952	0,950	0,951
SciBERT	3	2000	0,818	0,816	0,813	0,951	0,950	0,950
SciBERT	3	1000	0,812	0,811	0,806	0,949	0,948	0,948
SciBERT	2	3000	0,818	0,815	0,812	0,952	0,950	0,950
SciBERT	2	2000	0,817	0,815	0,811	0,951	0,950	0,950
SciBERT	2	1000	0,817	0,815	0,811	0,951	0,949	0,950
SciBERT	1	3000	0,816	0,814	0,810	0,951	0,950	0,950
SciBERT	1	2000	0,819	0,815	0,813	0,952	0,950	0,951
SciBERT	1	1000	0,819	0,816	0,813	0,951	0,950	0,950
MiniLM	3	3000	0,879	0,878	0,876	0,973	**0,978**	0,975
MiniLM	3	2000	0,875	**0,882**	0,876	**0,975**	**0,978**	**0,976**
MiniLM	3	1000	0,876	0,879	0,875	0,973	0,976	0,974
MiniLM	2	3000	0,877	0,880	0,876	0,974	0,977	0,975
MiniLM	2	2000	0,879	0,880	0,877	0,973	0,977	0,975
MiniLM	**2**	**1000**	**0,880**	0,881	**0,878**	**0,975**	**0,978**	**0,976**
MiniLM	1	3000	0,876	0,878	0,875	0,973	0,976	0,974
MiniLM	1	2000	0,877	0,880	0,876	0,974	0,977	0,975
MiniLM	1	1000	0,879	0,879	0,877	**0,975**	0,977	**0,976**

Fonte: Elaboração própria.

apresentam valores próximos com valores de desvio padrão pequenos e intervalos de confiança estreitos, o que evidencia a consistência do desempenho do ADAN no conjunto DBLP.

De acordo com as métricas estatísticas apresentadas na Tabela 5.7, considerando o

Tabela 5.7: Resultados estatísticos das 30 execuções independentes no conjunto de dados DBLP (configuração: MiniLM com duas camadas na RCG e 1000 épocas).

Métrica	Média	Desvio Padrão	IC 95% Inferior	IC 95% Superior
pP	0,8814	0,0013	0,8809	0,8819
pR	0,8889	0,0008	0,8886	0,8892
pF1	0,8827	0,0010	0,8823	0,8831
ACP	0,9786	0,0005	0,9784	0,9788
AAP	0,9808	0,0004	0,9807	0,9810
K-Metric	0,9796	0,0004	0,9795	0,9798

Fonte: Elaboração própria.

resultado médio de 30 execuções, o desempenho do ADAN foi comparado com os trabalhos de referência da literatura. A Tabela 5.8 apresenta a comparação com os trabalhos de Pooja et al. (2020, 2021), uma vez que Pooja et al. (2022a) não apresentou resultados disponíveis e comparáveis para esse conjunto de dados.

Tabela 5.8: Comparação entre diferentes trabalhos de referência com o conjunto de dados DBLP. As colunas de ganho mostram a variação percentual do ADAN, considerando a média de 30 execuções.

Métrica	Pooja et al. (2020)	Ganho ADAN(%)	Pooja et al. (2021)	Ganho ADAN(%)	ADAN
pP	0,789	11,70	0,853	3,33	**0,881**
pR	0,690	28,81	0,706	25,89	**0,888**
pF1	0,659	33,90	0,761	15,97	**0,882**
ACP	0,841	16,35	0,853	14,71	**0,978**
AAP	0,726	35,07	0,768	27,67	**0,980**
K-Metric	0,754	29,88	0,807	21,43	**0,979**

Fonte: Elaboração própria.

Em relação a Pooja et al. (2021), o ADAN melhora a métrica pF1 em 15,9%, passando de 0,761 para 0,882. Comparado a Pooja et al. (2020), o ganho é ainda mais expressivo, com aumento de 33,9% (de 0,659 para 0,882). Na qualidade do agrupamento, observam-se incrementos de 29,8% no K-Metric em relação a 0,754 e de 21,4% em relação a 0,807, ambos superados pelo valor de 0,979 alcançado pelo ADAN. Além disso, nas métricas ACP (0,978), AAP (0,980), e pR (0,888), o ADAN também supera de forma consistente os métodos da literatura, confirmando sua superioridade no conjunto DBLP. É importante destacar que, conforme apresentado na Tabela 5.9, em todas as métricas de validação os resultados do ADAN permanecem dentro dos respectivos intervalos de confiança de 95% inferior e superior, sendo estatisticamente significantes (marcados com ✓).

Tabela 5.9: Comparação dos melhores resultados obtidos no conjunto de dados DBLP com os intervalos de confiança (IC) do ADAN. Símbolos: ✓dentro do IC 95% Inferior (I)/Superior (S) e – fora do IC 95%.

Métrica	Melhor Resultado	Trabalho	IC 95% [I/S]	
pP	0,881	ADAN	[0,8809/0,8819]	✓
pR	0,888	ADAN	[0,8886/0,8892]	✓
pF1	0,882	ADAN	[0,8823/0,8831]	✓
ACP	0,978	ADAN	[0,9784/0,9788]	✓
AAP	0,980	ADAN	[0,9807/0,9810]	✓
K-Metric	0,979	ADAN	[0,9795/0,9798]	✓

Fonte: Elaboração própria.

A riqueza dos metadados, incluindo título, resumo, veículo de publicação e coautoria, permite a construção de grafos heterogêneos que potencializam tanto os *embeddings* semânticos quanto os estruturais. Essa estrutura possibilita que a RCG capture padrões contextuais e topológicos relevantes, resultando em elevada eficácia no processo de agrupamento.

Conjunto de Dados LAGOS-AND O desempenho do ADAN no conjunto LAGOS-AND utilizando *embeddings* do MiniLM combinados com o modelo de RCG é apresentado na Tabela 5.10. São reportadas as métricas B-Cubed em diferentes números de épocas de treinamento. Os melhores resultados foram obtidos com 800 épocas, alcançando $bP = 0{,}907$, $bR = 0{,}914$ e $bF = 0{,}908$.

Tabela 5.10: Desempenho do ADAN no conjunto de dados LAGOS-AND.

Épocas	bP	bR	bF
1	0,769	0,781	0,769
50	0,804	0,813	0,803
100	0,832	0,831	0,828
200	0,877	0,883	0,877
400	0,897	0,904	0,898
800	**0,907**	**0,914**	**0,908**

Fonte: Elaboração própria.

A Figura 5.1 apresenta visualmente a evolução das métricas B-Cubed ao longo do treinamento. As linhas mostram tendência crescente consistente para todas as métricas, com os maiores ganhos concentrados entre 1 e 200 épocas. A partir desse ponto, as melhorias tornam-se mais sutis, sugerindo saturação do desempenho próximo às 800 épocas.

Antes da análise comparativa, esclarecemos os métodos de referência propostos por Zhang et al. (2023). A abordagem *Base Features* (BF) utiliza funções de similaridade manuais com base no nome do autor, veículo, ano e afiliação, empregando métricas como similaridade de Jaccard e distância numérica. O grupo *Content Features* (CF) incorpora similaridades textuais com base em título e resumo. O método $\mathrm{CF}_{\mathrm{jaccard}}$ calcula a similaridade de Jaccard sobre *tokens*, enquanto $\mathrm{CF}_{\mathrm{tfidf}}$ e $\mathrm{CF}_{\mathrm{doc2vec}}$ empregam *embeddings* TF-IDF e Doc2Vec, respectivamente. A variante $\mathrm{CF}_{\mathrm{nn}}$ utiliza uma rede neural totalmente conectada treinada com pares de título e resumo. O melhor método de referência reportado combina BF com $\mathrm{CF}_{\mathrm{nn}}$.

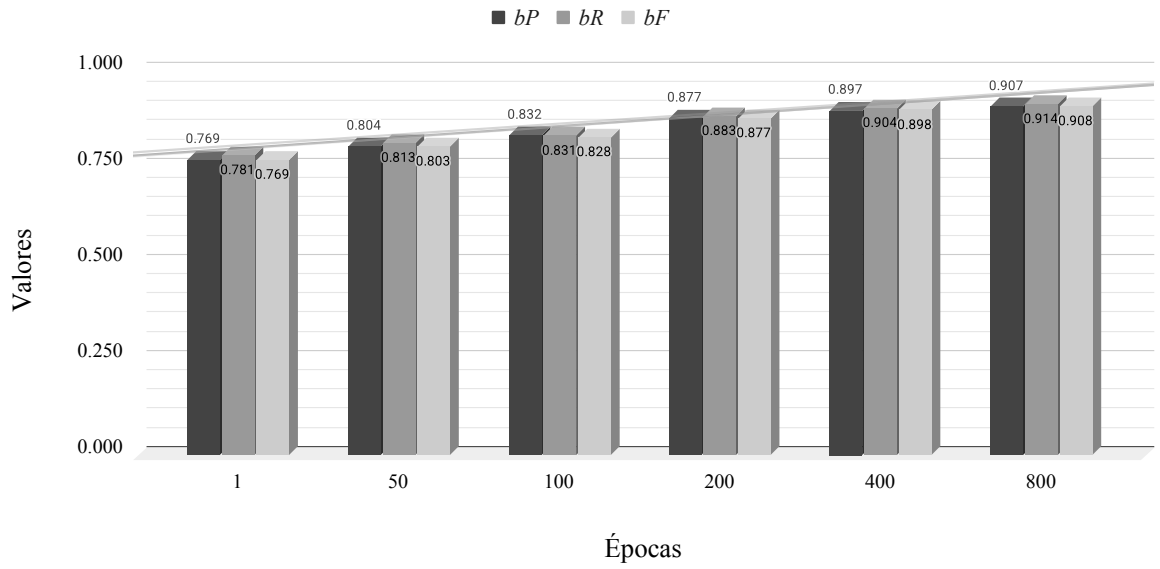


Figura 5.1: O conjunto de dados LAGOS-AND ao longo das épocas de treinamento da RCG com bP, bR e bF. Fonte: Elaboração própria.

A Tabela 5.11 resume os resultados comparativos considerando uma execução do ADAN. O ADAN supera, na maioria dos casos, os métodos de referência nas três métricas B-Cubed. Em comparação com o BF + $\mathrm{CF}_{\mathrm{nn}}$, que alcança $bF = 81{,}16$, o modelo proposto melhora esse valor em 11,8%. A métrica bP aumenta de 79,68 para 90,7 (ganho de 13,83%) e bR de 89,59 para 91,4 (ganho de 2,02%). Esses avanços confirmam a eficácia da combinação de *embeddings* contextuais com aprendizado em grafos para tarefas de AND em larga escala.

O ADAN demonstra robustez ao ser aplicado a blocos ambíguos de autores. Diferentemente dos métodos de referência do LAGOS-AND, que dependem de métricas de similaridade estáticas ou modelos neurais simples, o *framework* ADAN captura a semântica e a estrutura relacional por meio de RCG, permitindo agrupamentos mais precisos em contextos heterogêneos e esparsos.

Tabela 5.11: Resultados comparativos no LAGOS-AND: ADAN *versus* métodos de referência de Zhang et al. (2023). Cada coluna de ganho mostra a variação percentual do ADAN em relação ao método correspondente.

Método	*bP*	Ganho ADAN(%)	*bR*	Ganho ADAN(%)	*bF*	Ganho ADAN(%)
MAG-Author-ID	**97,68**	–7,14	71,11	28,53	77,00	17,92
Name Similarity	70,37	28,89	87,63	4,30	74,78	21,43
BF	75,85	19,58	86,62	5,52	77,40	17,31
BF + $CF_{jaccard}$	77,60	16,88	89,07	2,61	79,61	14,06
BF + CF_{tfidf}	77,27	17,38	90,09	1,45	79,93	13,60
BF + $CF_{doc2vec}$	74,14	22,34	**91,62**	–0,24	78,69	15,39
BF + CF_{nn}	79,68	13,83	89,59	2,02	81,16	11,88
ADAN (800 épocas)	**90,70**	–	91,40	–	**90,80**	–

Fonte: Elaboração própria.

Estudo de Ablação

Como o ADAN integra múltiplos componentes com diferentes funções, foi conduzido um estudo de ablação com o objetivo de isolar e avaliar a contribuição específica de cada módulo para o desempenho geral do *framework*. As variações do modelo foram obtidas por meio da remoção ou substituição de componentes individuais, como o método de extração de *embeddings* com PLN e a presença da RCG. Para a avaliação, foram utilizadas as métricas pF1 e K-Metric. As variantes analisadas foram:

- ADAN completo — inclui todos os componentes da proposta apresentada na Seção 4: construção da rede heterogênea, extração de *embeddings* com SciBERT e MiniLM, RCG e agrupamento com GHAC.
- Ablação 1 — ADAN sem RCG — remove a etapa de aprendizado com RCG, realizando o agrupamento diretamente sobre os *embeddings* dos documentos utilizando o GHAC. Essa variante avalia o impacto da RCG na qualidade da tarefa de AND.
- Ablação 2 — ADAN com *embeddings* alternativos — substitui os modelos utilizados neste trabalho (SciBERT e MiniLM) por métodos de extração de *embeddings* baseados em Word2Vec e TF-IDF. O objetivo é analisar a efetividade dessas representações na modelagem semântica dos documentos.

Resultados - Ablação 1 A Tabela 5.12 apresenta os resultados comparativos da versão completa do ADAN com a versão sem RCG, nos conjuntos AMiner-12 e DBLP, utilizando os modelos SciBERT e MiniLM. Observa-se melhora significativa nas métricas quando a RCG é incorporada, especialmente no conjunto AMiner-12.

Tabela 5.12: Comparação dos resultados com e sem RCG nos conjuntos AMiner-12 e DBLP utilizando SciBERT e MiniLM.

Modelo de PLN	Métrica	AMiner-12			DBLP		
		Sem RCG	Com RCG	Ganho %	Sem RCG	Com RCG	Ganho %
SciBERT	pP	0,482	**0,694**	+43,97%	0,764	**0,817**	+6,94%
	pR	0,405	**0,577**	+42,47%	0,792	**0,815**	+2,90%
	pF1	0,424	**0,614**	+44,81%	0,770	**0,811**	+5,32%
	ACP	0,843	**0,884**	+4,86%	0,943	**0,951**	+0,85%
	AAP	0,839	**0,884**	+5,36%	0,943	**0,950**	+0,74%
	K Metric	0,840	**0,884**	+5,24%	0,942	**0,951**	+0,96%
MiniLM	pP	0,466	**0,732**	+57,08%	0,771	**0,880**	+14,14%
	pR	0,468	**0,604**	+29,06%	0,791	**0,881**	+11,37%
	pF1	0,454	**0,646**	+42,29%	0,775	**0,878**	+13,29%
	ACP	0,865	**0,872**	+0,81%	0,943	**0,975**	+3,39%
	AAP	0,843	**0,891**	+5,69%	0,938	**0,978**	+4,26%
	K Metric	0,853	**0,881**	+3,28%	0,940	**0,976**	+3,83%

Fonte: Elaboração própria.

No AMiner-12, o uso da RCG resultou em um ganho de 44,81% em pF1 e 5,24% em K-Metric com o SciBERT. Com MiniLM, os ganhos foram de 42,29% em pF1 e 3,28% em K-Metric.

No DBLP, os ganhos foram mais discretos. Com SciBERT, o pF1 aumentou 5,32% e o K-Metric, 0,96%. Com MiniLM, os incrementos foram de 13,29% na pF1 e 3,83% na K-Metric. Apesar da maior qualidade dos metadados nesse conjunto, a RCG ainda contribui para uma melhora consistente nos resultados de agrupamento.

Esses resultados indicam que a RCG tem impacto direto na qualidade da tarefa de AND, especialmente em contextos com maior ambiguidade ou metadados menos informativos. Sua capacidade de explorar conexões estruturais na rede heterogênea complementa as representações semânticas extraídas dos textos, promovendo agrupamentos mais coerentes.

Resultados - Ablação 2 A Tabela 5.13 mostra os resultados da substituição dos modelos SciBERT e MiniLM por TF-IDF e Word2Vec. O objetivo é avaliar a capacidade dessas representações mais simples em capturar a semântica dos documentos na tarefa de AND, considerando as métricas pF1 e K-Metric. Os modelos utilizados e apresentados na proposta inicial deste trabalho superaram as alternativas em todas as bases avaliadas.

Tabela 5.13: Comparação entre métodos de extração de *embeddings* (TF-IDF, Word2Vec, SciBERT e MiniLM) nos conjuntos AMiner-12 e DBLP.

Métrica	AMiner-12				DBLP			
	TF-IDF	Word2Vec	SciBERT	MiniLM	TF-IDF	Word2Vec	SciBERT	MiniLM
pP	0,574	0,509	**0,694**	0,732	0,798	0,785	0,817	**0,880**
pR	0,497	0,420	**0,577**	0,604	0,820	0,782	0,815	**0,881**
pF1	0,514	0,442	**0,614**	0,646	0,803	0,779	0,811	**0,878**
ACP	0,839	0,828	0,**884**	0,872	0,945	0,936	0,951	**0,975**
AAP	0,861	0,837	**0,884**	0,891	0,947	0,930	0,950	**0,978**
K-Metric	0,849	0,832	**0,884**	0,881	0,946	0,932	0,951	**0,976**

Fonte: Elaboração própria.

No conjunto AMiner-12, a substituição resultou em reduções consideráveis nos valores de pF1 e K-Metric. Para o pF1, o uso de TF-IDF gerou uma queda de 16,30% em relação ao SciBERT (de 0,614 para 0,514) e de 20,43% em relação ao MiniLM (de 0,646 para 0,514). Já o Word2Vec apresentou quedas de 28,01% em relação ao SciBERT (de 0,614 para 0,442) e de 31,60% em relação ao MiniLM (de 0,646 para 0,442).

No K-Metric, o TF-IDF apresentou uma redução de 3,96% em relação ao SciBERT (de 0,884 para 0,849) e de 3,63% em relação ao MiniLM (de 0,881 para 0,849). O Word2Vec obteve quedas de 5,88% em relação ao SciBERT (de 0,884 para 0,832) e de 5,56% em relação ao MiniLM (de 0,881 para 0,832).

No DBLP, embora os resultados gerais tenham sido mais elevados, observa-se a mesma tendência. Para o pF1, o uso de TF-IDF resultou em uma redução de 0,99% em relação ao SciBERT (de 0,811 para 0,803) e de 8,55% em relação ao MiniLM (de 0,878 para 0,803). Com o Word2Vec, as quedas foram de 3,94% em relação ao SciBERT (de 0,811 para 0,779) e de 11,28% em relação ao MiniLM (de 0,878 para 0,779).

No K-Metric, o TF-IDF apresentou redução de 0,53% em relação ao SciBERT (de 0,951 para 0,946) e de 3,07% em relação ao MiniLM (de 0,976 para 0,946). O Word2Vec teve quedas de 2,00% em relação ao SciBERT (de 0,951 para 0,932) e de 4,51% em relação ao MiniLM (de 0,976 para 0,932). Assim como no AMiner-12, os resultados reforçam a superioridade dos modelos contextuais na tarefa de AND, mesmo em conjuntos com maior riqueza textual como o DBLP.

No geral, os resultados desta ablação mostram que TF-IDF e Word2Vec são menos eficazes na tarefa de AND, especialmente em conjuntos com textos curtos ou pouco informativos, como o AMiner-12. Nesse cenário, *embeddings* contextuais do SciBERT e MiniLM oferecem melhor desempenho, refletindo diretamente na acurácia da desambiguação.

Aspectos Computacionais

Foram obtidas métricas de tempo de execução e uso de memória RAM para cada módulo do ADAN, seguindo a estrutura descrita na Seção 4.1, com as configurações experimentais detalhadas na Seção 5.2. A Tabela 5.14 apresenta os resultados para os conjuntos de dados DBLP, AMiner-12 e LAGOS-AND, utilizando diferentes configurações de *hardware* conforme o tamanho de cada conjunto. Os tempos estão expressos em segundos (s) e o uso de memória em *gigabytes* (GB). O modelo RCG utilizado foi configurado com duas camadas em todos os experimentos, enquanto o modelo de PLN adotado para a extração de *embeddings* foi o MiniLM. Para os conjuntos DBLP e AMiner-12, os valores refletem 1.000 épocas de treinamento. No caso do LAGOS-AND, o uso de memória corresponde à média de seis configurações (1, 50, 100, 200, 400 e 800 épocas).

Tabela 5.14: Tempo de execução e uso de memória para cada módulo nos conjuntos de dados AMiner-12, DBLP e LAGOS-AND.

Módulo	DBLP		AMiner-12		LAGOS-AND	
	Tempo (s)	RAM (GB)	Tempo (s)	RAM (GB)	Tempo (s)	RAM (GB)
Entrada	0,2	0,2	0,1	0,1	80,9	1,9
Extração de *Embeddings*	80,5	0,3	69,3	0,3	5734,2	7,1
Construção da Rede Heterogênea	0,2	0,4	0,3	0,3	122,4	8,9
Aprendizado com RCG	594	0,7	541	0,6	42869,2	29,1
Agrupamento com GHAC	1,2	0,3	0,5	0,3	7773,3	8,1
Total	676,1	1,9	611,2	1,6	56580,1	55,1

Fonte: Elaboração própria.

Nos conjuntos DBLP e AMiner-12, todos os módulos apresentaram baixo tempo de processamento e uso reduzido de memória. Os módulos "Aprendizado com RCG e "Extração de *Embeddings*" foram os mais custosos, mas executaram de forma satisfatória dentro da capacidade do *hardware* descrito na Seção 5.2. No conjunto LAGOS-AND, todos os módulos demandaram mais recursos. A maior parte do esforço computacional concentrou-se nos módulos "Aprendizado com RCG" e "Extração de *Embeddings*", enquanto a "Construção da Rede Heterogênea" apresentou um aumento notável no uso de memória. O módulo de "agrupamento com GHAC" também apresentou maior tempo de execução, embora seu uso de memória tenha-se mantido moderado. O módulo de "Entrada" permaneceu com baixo custo em todos os conjuntos de dados. A Tabela 5.14 também apresenta o tempo total de execução para cada conjunto de dados: LAGOS-AND foi o mais custoso (56.580 s), seguido por DBLP (676 s) e AMiner-12 (611 s).

A Figura 5.2 apresenta, em escala logarítmica, o tempo de execução referente ao módulo de "Aprendizado com RCG" responsável pela maior parte do custo computacional do ADAN. Os resultados confirmam que esse módulo constitui o principal gargalo de execução, com o tempo crescente conforme a complexidade do grafo construído para cada

conjunto de dados. Esse comportamento é compatível com o custo teórico $\mathcal{O}(T \cdot L \cdot |E| \cdot F)$, conforme discutido na Seção 4.1.

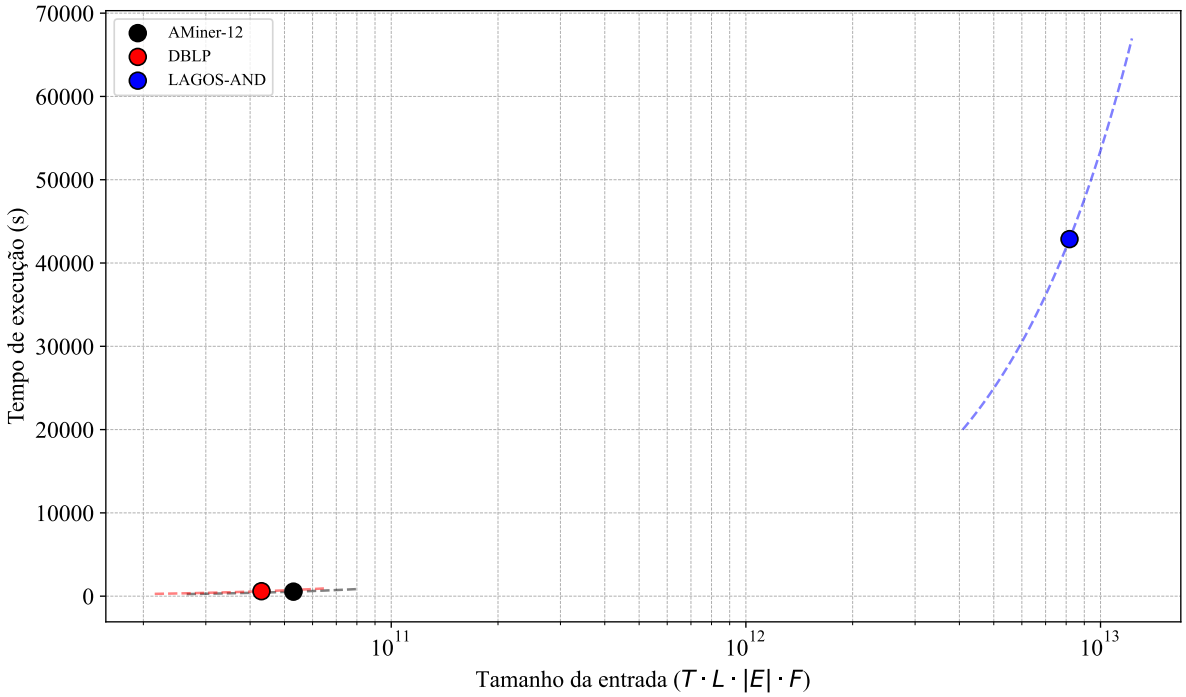


Figura 5.2: Ilustração do módulo "Aprendizado com RCG" avaliando o tempo de execução (em segundos) em função do tamanho da entrada ($T \cdot L \cdot |E| \cdot F$), em escala logarítmica. Fonte: Elaboração própria.

O conjunto LAGOS-AND apresentou o maior tempo de execução nesse módulo (42.869 s), refletindo sua estrutura de grafo significativamente mais ampla. O AMiner-12 obteve desempenho mais rápido (541 s), enquanto o DBLP exigiu um tempo de execução ligeiramente maior (594 s), mesmo contendo maior riqueza de atributos semânticos. Esses resultados indicam que o fator mais impactante no desempenho é o tamanho e a conectividade do grafo, e não a complexidade das informações associadas aos nós.

Os resultados apresentados nesta seção evidenciam a importância de estudos futuros voltados à otimização da eficiência computacional do ADAN, especialmente na camada L_3 do modelo arquitetural, que contém a etapa de "Aprendizado com RCG".

5.5 Discussão

Considerando os avanços atuais da Computação à época da realização desta pesquisa e da redação desta tese, é a utilização de Grandes Modelos de Linguagem ou *Large Language Models* (LLM) para a tarefa de AND. Recentemente, LLMs têm sido considerados promissores nessa tarefa, devido à sua capacidade de capturar relações semânticas e contextuais

complexas (van Lieshout, 2024; Yan and AsirAsir, 2024; Zhao and Chen, 2025). Trabalhos recentes indicam que abordagens híbridas, que combinam LLMs com modelos tradicionais, tendem a obter melhores resultados do que o uso isolado desses modelos (Yan and AsirAsir, 2024). Além disso, estratégias de *prompting* e ajuste fino de parâmetros podem contribuir para melhorias marginais, embora com elevado custo computacional (Naveed et al., 2024; van Lieshout, 2024).

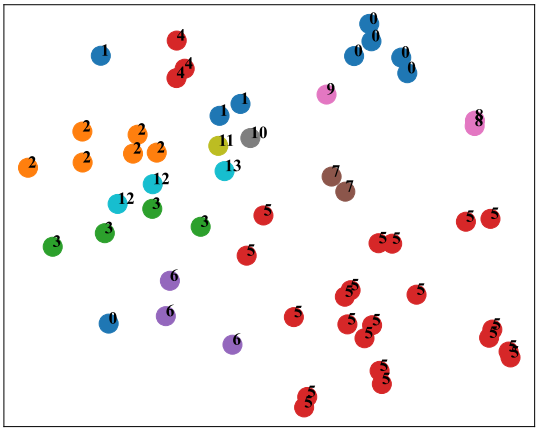
Nesse contexto, realizamos um teste preliminar com duas LLM, DeepSeek-R1-Distill-Llama-8B[5] e Gemma 3-12B-it,[6] aplicadas a um autor ambíguo do conjunto de dados AMiner12. Os modelos foram instruídos a agrupar as publicações do nome ambíguo "Koichi Furukawa" com base em título, coautores, veículo e ano. Embora ambos tenham identificado corretamente áreas de pesquisa relevantes, a desambiguação foi inconsistente em relação à rotulação verdadeira das publicações, indicando limitações na precisão quando não há integração com informações estruturais. O relatório completo com os *prompts*, análises e respostas está disponível em um repositório público no GitHub.[7]

Os experimentos realizados com conjunto de dados AMiner-12 apresentam alguns resultados competitivos. Considerando os resultados com significância estatística apresentados nas Tabela 5.3, o MiniLM atinge média de pF1 de 0,6717 e K-Metric de 0,8981, demonstrando robustez em cenários com metadados escassos.

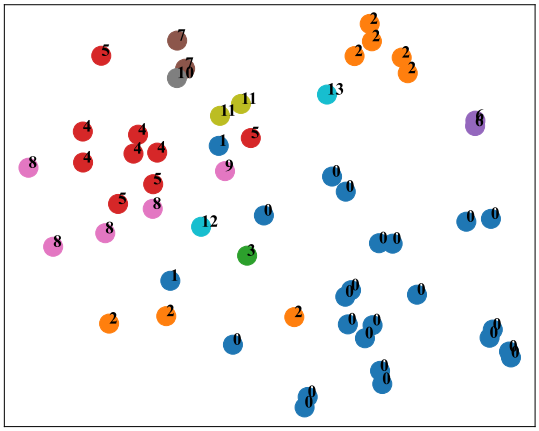
A Figura 5.3 apresenta os resultados de agrupamento para as publicações de três autores ambíguos do conjunto de dados AMiner-12, utilizando representações geradas com o modelo MiniLM, seguido de uma RCG com duas camadas e 1000 épocas de treinamento, e agrupamento com GHAC. Os autores ilustrados são: "Fan Wang" (com 14 rótulos reais), "Yue Zhao" (com 9 rótulos reais) e "Yoshio Tanaka" (com 2 rótulos reais). Os valores de pF1 correspondentes foram de 0,868, 0,945 e 1,00, respectivamente, utilizando apenas uma execução do ADAN.

Observa-se que a tarefa de AND torna-se progressivamente mais desafiadora à medida que o número de classes associadas ao autor aumenta. Por exemplo, "Fan Wang", que possui o maior número de instâncias ambíguas, apresentou predições de classes das publicações mais fragmentadas e um pF1 inferior em comparação aos demais autores. Cada linha da Figura 5.3 representa um autor ambíguo, composta por duas subfiguras: à esquerda, visualizamos os rótulos previstos pelo modelo; à direita, os rótulos extraídos dos dados reais. As publicações são agrupadas e coloridas de acordo com os rótulos atribuídos, permitindo uma comparação visual clara entre os agrupamentos obtidos e os corretos.

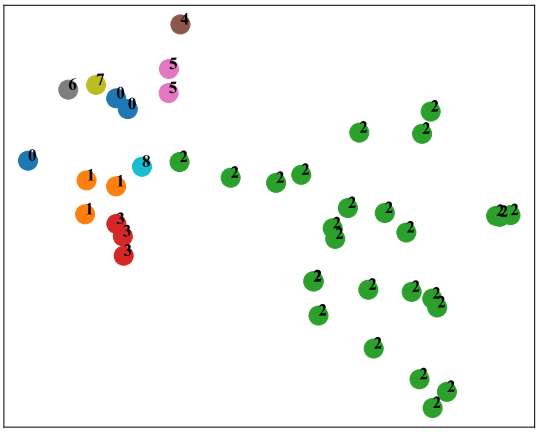
[5] `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B`
[6] `https://huggingface.co/google/gemma-3-12b-it`
[7] `https://github.com/natansr/and_llm_test.git`



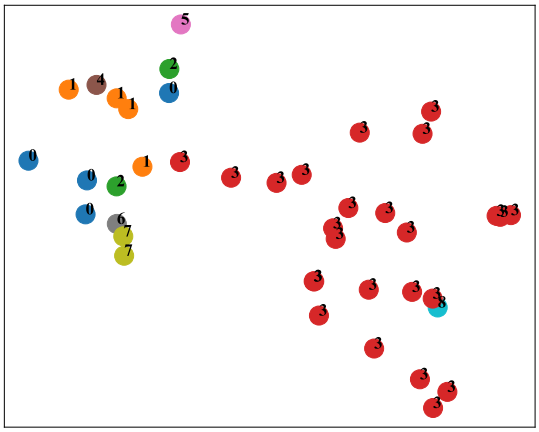
(a) Rótulos Previstos - “Fan Wang”



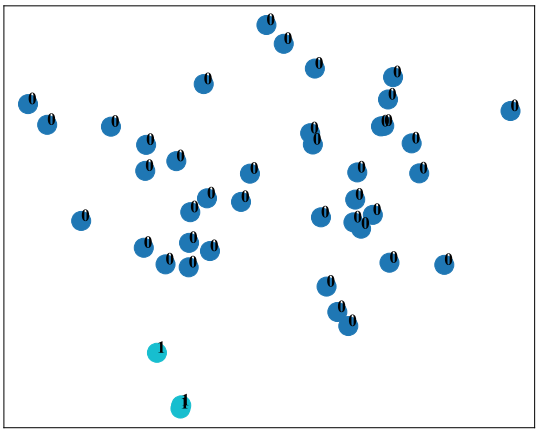
(b) Rótulos Reais - “Fan Wang”



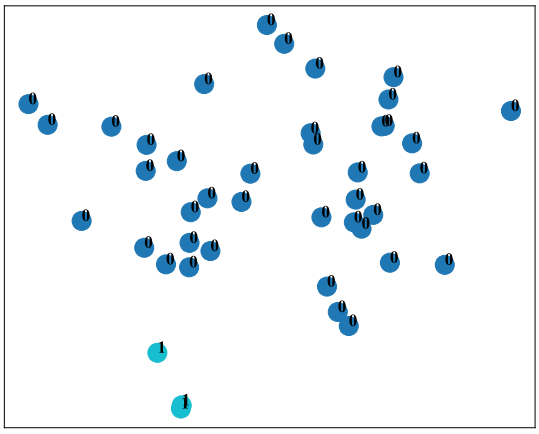
(c) Rótulos Previstos - “Yue Zhao”



(d) Rótulos Reais - “Yue Zhao”



(e) Rótulos Previstos - “Yoshio Tanaka”



(f) Rótulos Reais - “Yoshio Tanaka”

Figura 5.3: Visualização comparativa entre os rótulos reais e os rótulos previstos pelo ADAN para três autores ambíguos do conjunto AMiner-12. Fonte: Elaboração própria.

Essa visualização evidencia a capacidade do modelo de recuperar identidades distintas de autores, ao mesmo tempo em que revela o impacto da ambiguidade na performance.

Autores associados a um número reduzido de classes, como “Yoshio Tanaka”, tendem a gerar agrupamentos mais compactos e coerentes, enquanto autores com maior grau de ambiguidade, como “Fan Wang”, resultam em saídas mais dispersas e complexas.

No conjunto de dados DBLP, considerando os resultados com significância estatística nas Tabelas 5.7 e 5.8, a combinação do MiniLM com duas camadas na RCG e 1000 épocas apresentou, em média pF1 de 0,882, superando os trabalhos anteriores em 33,9% em pF1 e 29,8% em K-Metric.

No LAGOS-AND, um *benchmark* em larga escala e heterogêneo, considerando uma execução, o ADAN atinge uma pontuação bF de 90,8, superando em 21,43% o método de *Name Similarity* reportado em Zhang et al. (2023), conforme apresentado na Tabela 5.11.

O *framework* ADAN apresenta desempenho competitivo em relação aos trabalhos de referência nos conjuntos de dados avaliados, obtendo ganhos tanto em consistência de agrupamento quanto na acurácia da tarefa de AND.

Os resultados obtidos nos conjuntos DBLP, AMiner-12 e LAGOS-AND confirmam a efetividade da integração entre *embeddings* gerados por modelos baseados em *transformers*, o aprendizado relacional por meio da RCG e o agrupamento com o algoritmo GHAC para a tarefa de AND.

Conforme apresentado neste capítulo e considerando a QP2, apresentada na Seção 1.3, os experimentos realizados com o *framework* híbrido ADAN demonstraram desempenho superior nas métricas de validação de acordo com o trabalho de referência de Pooja et al. (2020), com ganhos expressivos e valores estatisticamente significativos segundo as médias e intervalos de confiança obtidos (Tabela 5.7), com 33,9% em pF1 e 29,8% em K-Metric, utilizando o conjunto de dados DBLP (Tabela 5.8). Verificou-se ainda com o DBLP, que o ADAN apresenta melhores resultados comparados ao trabalho de Pooja et al. (2021).

O Capítulo 6 apresenta as conclusões, contribuições e limitações desta tese.

Capítulo 6

Conclusões

Esta tese apresenta o desenvolvimento do *framework* híbrido ADAN para a tarefa de AND em repositórios bibliográficos digitais. O ADAN combina aprendizado de máquina profundo com técnicas de PLN, baseadas em modelos de *Transformer* (i.e., SciBERT e MiniLM), RCG e o algoritmo de agrupamento GHAC.

A arquitetura modular do *framework* ADAN é composta por quatro camadas principais, possibilitando a parametrização de diversos componentes, como a profundidade da RCG, o número de épocas de treinamento e o modelo de PLN utilizado. Como o *framework* ADAN integra múltiplos componentes, foi realizado um estudo de ablação para avaliar o impacto de diferentes configurações e a presença ou ausência de determinados módulos. Como resultado, foi possível identificar as combinações mais eficazes e adaptar o *framework* a distintos níveis de complexidade semântica e estrutural.

Os experimentos conduzidos com os conjuntos AMiner-12, DBLP e LAGOS-AND evidenciaram a eficácia do *framework* ADAN frente à complexidade da ambiguidade de nomes de autores. Mesmo em cenários com metadados limitados, como no AMiner-12, o modelo obteve ACP de 0,891 e K-Metric de 0,898, superando os métodos de referência em até 37,6% e 20,2%, respectivamente. No DBLP, em comparação com as abordagens de Pooja et al. (2020, 2021), o ADAN atingiu pF1 de 0,882 e K-Metric de 0,979, com ganhos de até 33,9% e 29,8%. Já no LAGOS-AND, alcançou $bF = 0{,}908$, com melhoria de 21,4% em relação às abordagens anteriores.

Além dos resultados quantitativos, o ADAN contribui metodologicamente ao oferecer uma arquitetura extensível, de código aberto e com GUI, facilitando seu uso e reprodutibilidade por outros pesquisadores da área.

6.1 Contribuições

Durante a realização dessa pesquisa, foram considerados os problemas motivacionais elencados na Seção 1.2. A solução proposta pelo *framework* híbrido denominado ADAN atendeu os referidos problemas conforme segue:

- Poucos dados nas citações bibliográficas: o ADAN emprega modelos de PLN baseados em *transformers*, capazes de extrair informações semânticas a partir de metadados limitados. Essa representação semântica é complementada pela RCG, que explora padrões relacionais de coautoria, suprindo a limitação de metadados em repositórios bibliográficos digitais.
- Praticidade e custo: a RCG foi treinada com estratégia auto-supervisionadas, aprendendo diretamente das relações de coautoria e vizinhança sem necessidade de rótulos manuais, enquanto o GHAC realiza agrupamento de forma não supervisionada. Essa abordagem reduz a dependência de dados rotulados, minimizando custos.
- Eficiência e Padrões de publicações distintos: o ADAN demonstrou capacidade de adaptação a padrões de publicações heterogêneos. No DBLP, com publicação em Ciência da Computação e no AMiner-12, com metadados reduzidos, o ADAN manteve desempenho consistente. Essa característica foi reforçada nos experimentos com o conjunto de dados , que integra diferentes repositórios bibliográficos e áreas de conhecimento distintas.
- Eficácia: os resultados experimentais, considerando a significância estatística, mostraram ganhos expressivos em métricas como pF1, ACP e K-Metric. Em especial, no DBLP, o ADAN superou Pooja et al. (2020) em 33,9% no pF1 e 29,88% no K-Metric. Considerando os trabalhos de Pooja et al. (2020, 2021), os resultados de pF1 e K-Metric foram superiores em 15,97% e 21,43%, respectivamente, indicando eficácia da proposta.

Publicações

Durante a execução desta pesquisa, foram obtidas publicações em periódicos e conferências da área de Computação, conforme segue.

Periódicos

1. Rodrigues, N. S. and Ralha, C. G. A Novel Framework with ComMAND: A Combined Method for Author Name Disambiguation. *Information Processing & Management*, vol. 63, Issue 1, January 2026, 104304. `https://doi.org/10.1016/j.ipm.`

2025.104304. Estrato A1 Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) (2017-2020). Fator de Impacto (JCR 2023): 7.4.

2. Rodrigues, N. S. and Ralha, C. G. A Flexible and Configurable System to Author Name Disambiguation. *IEEE Access*, vol. 13, pp. 125606–125617 (2025). Open access. https://doi.org/10.1109/ACCESS.2025.3589957. Estrato A3 CAPES (2017-2020). Fator de Impacto (JCR 2023): 3.4.

3. Rodrigues, N.S., Mariano, A.M. and Ralha, C.G. Author Name Disambiguation Literature Review with Consolidated Meta-analytic Approach. *International Journal on Digital Libraries*, vol. 25, pp. 765—785 (2024). Open access. https://doi.org/10.1007/s00799-024-00398-1. Estrato A1 CAPES (2017-2020). Fator de Impacto (JCR 2023): 1.6.

Conferências

1. Rodrigues, N., & Ralha, C. (2024). A Hybrid Machine Learning Method to Author Name Disambiguation. In *XV Brazilian Symposium in Information and Human Language Technology (STIL)*, pp. 108–117. Belém, PA, Brazil: SBC. doi: 10.5753/stil.2024.245440. Estrato B1 CAPES (2017-2020).

2. Rodrigues, N. S., Ralha, C. G., Costa, A. R., Lemos, L. C. (2020). Multi-strategic Approach for Author Name Disambiguation in Bibliography Repositories. In *Annual International Conference on Information Management and Big Data (SIMBig20)*, vol. 1, pp. 1—14. Lima, Peru. doi: 10.1007/978-3-030-76228-5_5. Estrato B3 CAPES (2017-2020).

6.2 Limitações

Apesar dos avanços observados nos resultados apresentados na Seção 5.4, algumas limitações necessitam ser citadas:

- Ajuste de hiperparâmetros: O desempenho do modelo varia com relação à profundidade da RCG e ao número de épocas de treinamento. Os experimentos mostram que o uso de mais de duas camadas na RCG frequentemente leva à suavização excessiva (*over-smoothing*), que pode prejudicar a acurácia da desambiguação.

- Sensibilidade aos metadados: A ausência de resumos e palavras-chave em alguns conjuntos (como o AMiner-12) limita a capacidade do modelo de explorar conteúdo semântico. A RCG se beneficia de atributos relacionais. Desta forma, seu desempenho é reduzido na ausência de informações.

- Distribuição da ambiguidade: Em conjuntos como o LAGOS-AND, o desbalanceamento entre classes é um desafio real, pois alguns autores aparecem com poucos registros, dificultando a modelagem de grupos minoritários. O uso de funções de perda reponderadas ou amostragem sensível à classe pode mitigar esse problema.
- Avaliação por bloco: Os blocos de nomes ambíguos variam em estrutura, e o desempenho pode depender do tamanho e densidade de cada bloco. Blocos maiores, com muitos coautores e afiliações sobrepostas, podem exigir estratégias diferentes daquelas aplicadas a blocos menores e isolados.
- Escalabilidade: Embora o MiniLM ofereça um bom equilíbrio entre qualidade e eficiência, sua combinação com RCG acarreta custos adicionais, especialmente em grafos de maior porte. No LAGOS-AND, o tempo de treinamento aumenta significativamente com o tamanho do grafo e o número de épocas.
- Comparação com LLM: Embora LLM representem uma direção promissora, seu uso isolado ainda apresenta fragilidades para tarefas de AND. Estudos apontam limitações relacionadas à dependência de metadados, à necessidade de estratégias sofisticadas de comparação, à sensibilidade a atributos pouco informativos e ao alto custo computacional (Naveed et al., 2024; van Lieshout, 2024; Zhao and Chen, 2025). Nossos testes preliminares com dois modelos de LLM confirmaram essas dificuldades para a tarefa de AND, resultando em desambiguação inconsistentes em relação à rotulação verdadeira dos documentos. Assim, optamos por não incluir comparações formais nesta pesquisa, deixando essa investigação para trabalhos futuros.

6.3 Trabalhos Futuros

Considerando as limitações discutidas na Seção 6.2, as direções futuras desta pesquisa envolvem não apenas a superação dos desafios identificados, mas também o aprimoramento e a expansão do *framework* híbrido proposto. Os principais objetivos incluem a redução de custos computacionais em grafos extensos, a escalabilidade em cenários ruidosos ou desbalanceados, a mitigação de efeitos semânticos indesejados nos metadados textuais, como sentimento e toxicidade, e a ampliação da aplicabilidade do ADAN em contextos reais de curadoria bibliográfica, com a exploração do potencial de LLM em combinação com modelos estruturais e semânticos especializados.

Nesse sentido, as ações previstas para trabalhos futuros incluem:

- Ampliar os experimentos com novos conjuntos de dados oriundos de diferentes domínios e instituições, a fim de fortalecer a validação externa e a robustez do sistema proposto;

- Avaliar arquiteturas alternativas de RCG, como GraphSAGE e GAT, com vistas a melhorar o desempenho em diferentes estruturas de grafo e reduzir os custos computacionais associados ao treinamento em larga escala;
- Estender a GUI do ADAN, com foco em responsividade e versão adaptada para ambientes *Web*, visando facilitar sua aplicação prática em tarefas de curadoria e análise bibliográfica;
- Investigar variantes de modelos de linguagem baseados em *transformers*, com foco em estratégias de fusão híbrida entre *embeddings* textuais e estruturais, além de explorar o uso de LLMs integrados a abordagens tradicionais de AND.

Referências

AMiner (2005-2025). Search and mining of academic social networks. `https://www.aminer.cn/`. Tsinghua University, Beijing, 100084. China. 1

Ankrah, J., Monteiro, A., and Madureira, H. (2022). Bibliometric analysis of data sources and tools for shoreline change analysis and detection. *Sustainability*, 14(9):4895. 37

Aroca-Ouellette, S. and Rudzicz, F. (2020). On losses for modern language models. *arXiv preprint arXiv:2010.01694*. 15

Backes, T. and Dietze, S. (2022). Lattice-based progressive author disambiguation. *Information Systems*, 109:102056. 2, 58, 63

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer. 30

Beltagy, I., Cohan, A., and Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text. *CoRR*, abs/1903.10676. 15, 83, 90

Bhattacharya, I. and Getoor, L. (2005). Relational clustering for multi-type entity resolution. In *Proceedings of the 4th International Workshop on Multi-Relational Mining*, MRDM '05, page 3–12, New York, NY, USA. Association for Computing Machinery. 50

Blakely, D., Lanchantin, J., and Qi, Y. (2021). Time and space complexity of graph convolutional networks. `https://qdata.github.io/deep2Read/talks-mb2019/Derrick_201906_GCN_complexityAnalysis-writeup.pdf`. Acessado em: Julho de 2025. 78

Bolfarine, H. and Sandoval, M. C. (2001). *Introdução à inferência estatística*, volume 2. Sociedade Brasileira de Matemática (SBM). 7, 92

Bollen, J., Rodriguez, M. A., Van de Sompel, H., Balakireva, L. L., and Hagberg, A. (2007). The largest scholarly semantic network...ever. In *Proceedings of the 16th Int. Conf. on World Wide Web*, page 1247–1248. ACM. 3

Bollobás, B. (1998). *Modern Graph Theory*. Springer. 19

Boukhers, Z. and Asundi, N. B. (2022). Whois? deep author name disambiguation using bibliographic data. In Silvello, G., Corcho, O., Manghi, P., Di Nunzio, G. M., Golub, K., Ferro, N., and Poggi, A., editors, *Linking Theory and Practice of Digital Libraries*, pages 201–215, Cham. Springer International Publishing. 2, 54, 63

Boukhers, Z. and Asundi, N. B. (2024). Deep author name disambiguation using dblp data. *International Journal on Digital Libraries*, 25(3):431–441. 54, 65

Brookes, B. C. (1969). Bradford's law and the bibliography of science. *Nature*, 224:953–956. 34

Chen, Y. (2015). Convolutional neural network for sentence classification. Master's thesis, University of Waterloo. 12

Chen, Y., Jiang, Z., Gao, J., Du, H., Gao, L., and Li, Z. (2023). A supervised and distributed framework for cold-start author disambiguation in large-scale publications. *Neural Computing and Applications*, 35(18):13093–13108. 55, 64

Choi, D., Jang, J., Song, S., Lee, H., Lim, J., Bok, K., and Yoo, J. (2024). Name disambiguation scheme based on heterogeneous academic sites. *Applied Sciences*, 14(1). 2, 55, 64

Chuanming, Y., Yunci, Z., Aochen, L., and Lu, A. (2020). Author name disambiguation with network embedding. *Data Analysis and Knowledge Discovery*, 4(2/3):48–59. 57, 60

Colavizza, G., Hrynaszkiewicz, I., Staden, I., Whitaker, K., and McGillivray, B. (2020). The citation advantage of linking publications to research data. *PLOS ONE*, 15(4):1–18. 48, 53

Correa, P. R. and Cruz, R. G. (2005). Meta-análisis sobre la implantación de sistemas de planificación de recursos empresariales (erp). *JISTEM-Journal of Information Systems and Technology Management*, 2:245–273. 37

Correia, A., Guimarães, D., Paulino, D., Jameel, S., Schneider, D., Fonseca, B., and Paredes, H. (2021). Authcrowd: Author name disambiguation and entity matching using crowdsourcing. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 150–155. 58, 62

Cota, R. G., Ferreira, A. A., Nascimento, C., Gonçalves, M. A., and Laender, A. H. F. (2010). An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. *Journal of the American Society for Information Science and Technology*, 61(9):1853–1870. 48, 49

Crispim, R. T., Netto, C. O., Camboim, G. F., and Camboim, F. F. (2022). Capabilities for service innovation: Bibliometric analysis and directions for future research. *RAM. Revista de Administração Mackenzie*, 23. 37

da Silva, M. E. V. (2007). Xsimilarity: uma ferramenta para consultas por similaridade embutidas na linguagem xquery. 3

D'Angelo, C. A. and van Eck, N. J. (2020). Collecting large-scale publication data at the level of individual researchers: a practical proposal for author name disambiguation. *Scientometrics*, 123(2):883–907. 55, 60

DBLP (2025). Statistics. Disponível em `https://dblp.org/statistics/index.html`. 1

Debarshi, K. S., Plaban, K. B., and Partha, P. D. (2019). A review of author name disambiguation techniques for the pubmed bibliographic database. *Journal of Information Science*. 3, 53

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852. 27

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. viii, 13, 14, 15

Eler, D. M. (2022). *Qualidade em conjuntos de dados rotulados: uso do BERT para revisão de anotações e aplicação de saliência para a identificação de vieses*. PhD thesis, Universidade Estadual Paulista (Unesp). 14

Fang, Z., Zhuo, Y., Xu, J., Tang, Z., Jia, Z., and Zhang, H. (2023). Automatic author name disambiguation by differentiable feature selection. *Journal of Information Science*, 0(0):01655515231193859. 55, 64

Ferreira, A. A., Gonçalves, M. A., and Laender, A. H. F. (2012). A brief survey of automatic methods for author name disambiguation. *Acm Sigmod Record*, 41(2):15–26. viii, x, 3, 11, 38, 48, 49, 50, 53, 60, 61, 62, 63, 64, 65

Ferreira, A. A., Gonçalves, M. A., and Laender, A. H. F. (2020). *Automatic Disambiguation of Author Names in Bibliographic Repositories*. Morgan & Claypool Publishers. viii, ix, x, 1, 2, 3, 4, 11, 12, 28, 30, 38, 53, 60, 61, 62, 63, 64, 65, 66, 67

Ferreira, A. A., Veloso, A., Gonçalves, M. A., and Laender, A. H. F. (2014). Self-training author name disambiguation for information scarce scenarios. *Journal of the Association for Information Science and Technology (JASIST)*, 65(6):1257–1278. 47, 48, 49

Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 72

Firdaus, Lestari, S. D., Nurmaini, S., Malik, R. F., Rachmatullah, M. N., Darmawahyuni, A., Sapitri, A. I., and El Qiliqsandy, M. (2021a). Author matching classification on a highly imbalanced bibliographic data using cost-sensitive deep neural network. In *2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS*, pages 86–89. 58, 63

Firdaus, Nurmaini, S., Malik, R. F., Darmawahyuni, A., Rachmatullah, M. N., Juliano, A. H., Nugraha, T. A., and Putra, V. O. K. (2021b). Author identification in bibliographic data using deep neural networks. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, 19(3):911–919. 57, 61

Färber, M. and Ao, L. (2022). The Microsoft Academic Knowledge Graph enhanced: Author name disambiguation, publication classification, and embeddings. *Quantitative Science Studies*, 3(1):51–98. 58, 63

Färber, M. and Lamprecht, D. (2021). The data set knowledge graph: Creating a linked open data source for data sets. *Quantitative Science Studies*, 2(4):1324–1355. 63

Garakhanova, N. (2023). Bibliometric analysis on digital diplomacy studies. *Korkut Ata Türkiyat Araştırmaları Dergisi*, 1(Özel Sayı 1 (Cumhuriyetin 100. Yılına)):1325–1338. 37

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR. 22

Gong, J., Fang, X., Peng, J., Zhao, Y., Zhao, J., Wang, C., Li, Y., Zhang, J., and Drew, S. (2024). More: toward improving author name disambiguation in academic knowledge graphs. *International Journal of Machine Learning and Cybernetics*, 15(1):37–50. 55

Gong, J., Fang, X., Wang, C., Ju, J., Bao, Y., Zhang, J., and Xu, J. (2023). Author name disambiguation based on capsule network via semantic and structural features. In *Proceedings of the 2023 6th International Conference on Signal Processing and Machine Learning*, SPML '23, page 293–300, New York, NY, USA. Association for Computing Machinery. 55, 64

Grauwin, S. (2022). Bibliotools / bibliomaps. `http://www.sebastian-grauwin.com/bibliomaps/index.html`. 36

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*. 13

Hamilton, W. L. (2020). *Graph representation learning*. Morgan & Claypool Publishers. 22, 24, 25, 27, 28

Han, H., Giles, L., Zha, H., Li, C., and Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.*, pages 296–305. 50

Han, H., Xu, W., Zha, H., and Giles, C. L. (2005a). A hierarchical naive bayes mixture model for name disambiguation in author citations. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, page 1065–1069, New York, NY, USA. Association for Computing Machinery. 50

Han, H., Zha, H., and Giles, C. L. (2005b). Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '05, page 334–343, New York, NY, USA. Association for Computing Machinery. 50

Heradio, R., Fernandez-Amoros, D., Cerrada, C., and Cobo, M. J. (2020). Group decision-making based on artificial intelligence: A bibliometric analysis. *Mathematics*, 8(9):1566. 34

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Q.*, 28(1):75–105. 6

Huang, L., Zhang, J., Wang, B., Li, Z., Wang, S., and Zhang, R. (2025). A framework for global role-based author name disambiguation. *Pattern Recognition*, 166:111703. 58, 64

Huang, Z., Zhang, H., Hao, C., Yang, H., and Wu, H. (2024). A cross-domain transfer learning model for author name disambiguation on heterogeneous graph with pretrained language model. *Knowledge-Based Systems*, 305:112624. 2, 54, 64

Hung, N. T., Huynh, T., and Do, T. (2014). Author name disambiguation by using deep neural network. In Nguyen, N. T., Attachoo, B., Trawiński, B., and Somboonviwat, K., editors, *Intelligent Information and Database Systems. ACIIDS 2014. Lecture Notes in Computer Science, vol. 8397*. Springer, Cham. 3

Hussain, I. and Asghar, S. (2017). A survey of author name disambiguation techniques: 2010-2016. *Knowledge Eng. Review*, 32:e22. 1, 3

Jhawar, K., Sanyal, D. K., Chattopadhyay, S., Bhowmick, P. K., and Das, P. P. (2020). Author name disambiguation in pubmed using ensemble-based classification algorithms. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, page 469–470, New York, NY, USA. Association for Computing Machinery. 54, 60

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. (2020). Tinybert: Distilling bert for natural language understanding. 18

Jinqi, Q., Luoyi, F., Xiaoying, G., and Xinbing, W. (2020). A network maximum flow based approach for author name disambiguation. *Journal of Shanghai Jiaotong University*, 54(2). 56, 60

Kang, I.-S., Na, S.-H., Lee, S., Jung, H., Kim, P., Sung, W.-K., and Lee, J.-H. (2009). On co-authorship for author disambiguation. *Information Processing & Management*, 45(1):84–97. 50

Khider, H., Hammoudi, S., Meziane, A., and Cuzzocrea, A. (2023). Bpm in the era of industry 4.0: A bibliometric analysis. In *ICEIS (2)*, pages 651–659. 37

Kim, J. (2019). A fast and integrative algorithm for clustering performance evaluation in author name disambiguation. *Scientometrics*, 120(2):661–681. 48, 49

Kim, J. and Kim, J. (2024). Andez: An open-source tool for author name disambiguation using machine learning. *SoftwareX*, 26:101719. 55, 64

Kim, J., Kim, J., and Owen-Smith, J. (2019a). Generating automatically labeled data for author name disambiguation: an iterative clustering method. *Scientometrics*, 118(1):253–280. 48, 49

Kim, J., Kim, J., and Owen-Smith, J. (2021). Ethnicity-based name partitioning for author name disambiguation using supervised machine learning. *Journal of the Association for Information Science and Technology (JASIST)*, 72(8):979–994. 54, 62

Kim, J. and Owen-Smith, J. (2020). Model reuse in machine learning for author name disambiguation: An exploration of transfer learning. *IEEE Access*, 8:188378–188389. 54, 60

Kim, J. and Owen-Smith, J. (2021). Orcid-linked labeled data for evaluating author name disambiguation at scale. *Scientometrics*, 126(3):2057–2083. 54, 62

Kim, K., Rohatgi, S., and Giles, C. L. (2019b). Hybrid deep pairwise classification for author name disambiguation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2369–2372, New York, NY, USA. Association for Computing Machinery. 48, 52

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, page 1. Poster Presentations. 78

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*. 21, 22, 25, 26, 27, 85

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:1–26. 32

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15. 32

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942. 15

Lee, D.and Kang, J., Mitra, P., Giles, C. L., and On, B. (2007). Are your citations clean? *Communications of the ACM*, 50(12):33–38. 9

Levin, M., Krawczyk, S., Bethard, S., and Jurafsky, D. (2012). Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology*, 63(5):1030–1047. 48, 49

Li, H., Cui, Y., and Wang, T. (2020). An effective approach for automatic author name disambiguation based on multiple strategies. In *Proceedings of the 3rd International Conference on Computer Science and Software Engineering*, CSSE '20, page 169–175, New York, NY, USA. Association for Computing Machinery. 55, 60

Li, Y. (2024). Research on chinese-english author name disambiguation based on semantic fingerprint. In *2024 4th International Conference on Computer Science and Blockchain (CCSB)*, pages 390–394. 58, 65

Liu, D., Zhang, R., Chen, J., and Chen, X. (2024a). Author name disambiguation via paper association refinement and compositional contrastive embedding. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2193–2203, New York, NY, USA. Association for Computing Machinery. 56, 64

Liu, T., Zeng, X., Wu, Q., and Wang, M. (2024b). Andi: a joint disambiguation framework integrating author name disambiguation goals. In *International Conference on Database Systems for Advanced Applications*, pages 219–229. Springer. 57

Liu, W., Islamaj Doğan, R., Kim, S., Comeau, D. C., Kim, W., Yeganova, L., Lu, Z., and Wilbur, W. J. (2014). Author name disambiguation for PubMed. *Journal of the Association for Information Science and Technology (JASIST)*, 65(4):765–781. 51

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*. 15

Lotka, A. J. (1926). The frequency distribution of scientific productivity. *Journal of the Washington Academy of Sciences*, 16(12):317–323. 34

Ma, X., Wang, R., Zhang, Y., Jiang, C., and Abbas, H. (2020a). A name disambiguation module for intelligent robotic consultant in industrial internet of things. *Mechanical Systems and Signal Processing*, 136:106413. 57, 61

Ma, Y., Wu, Y., and Lu, C. (2020b). A graph-based author name disambiguation method and analysis via information theory. *Entropy*, 22(4):416. 56, 60

Makinde, A. (2024). Optimizing time series forecasting: A comparative study of adam and nesterov accelerated gradient on lstm and gru networks using stock market data. *arXiv preprint arXiv:2410.01843*. 78

Manzoor, A., Asghar, S., and Amjad, T. (2022). Toward a new paradigm for author name disambiguation. *IEEE Access*, 10:76055–76068. 58, 63

Mariano, A. M., Reis, A. C. B., Althoff, L. d. S., and Barros, L. B. (2019). A bibliographic review of software metrics: Applying the consolidated meta-analytic approach. In Reis, J., Pinelas, S., and Melão, N., editors, *Industrial Engineering and Operations Management I*, pages 243–256, Cham. Springer Int. Publishing. 32, 33

Mariano, A. M. and Rocha, M. S. (2017). Revisão da literatura: apresentação de uma abordagem integradora. In *AEDEM Int. Conf.*, volume 26, pages 427–442, Reggio di Calabria, Italy. Economy, Business and Uncertainty: ideas for a European and Mediterranean industrial policy? 32, 37

MEDLINE (1993-2025). Pubmed. `https://pubmed.ncbi.nlm.nih.gov/`. National Library of Medicine, 8600, Rockville Pike Bethesda, MD 20894. 41, 50

Mihaljević, H. and Santamaría, L. (2021). Disambiguation of author entities in ads using supervised learning and graph theory methods. *Scientometrics*, 126(5):3893–3917. 54, 62

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 12

Mozafari, N. (2021). A genetic-based approach for author name disambiguation problem. *Iranian Journal of Information processing and Management*, 36(3):791–816. 56, 62

Müller, M. (2023). Pyblionet–software for the creation, visualization and analysis of bibliometric networks. *SoftwareX*, 24:101565. 37

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. 20

Nadkarni, P. M., Ohno-Machado, L., and Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551. 12

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2024). A comprehensive overview of large language models. 105, 111

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press. 19

Pooja, K. M., Mondal, S., and Chandra, J. (2020). A graph combination with edge pruning-based approach for author name disambiguation. *Journal of the Association for Information Science and Technology (JASIST)*, 71(1):69–83. 57, 61, 93, 94, 95, 96, 97, 107, 108, 109

Pooja, K. M., Mondal, S., and Chandra, J. (2021). Exploiting similarities across multiple dimensions for author name disambiguation. *Scientometrics*, 126(9):7525–7560. 57, 93, 94, 95, 96, 97, 107, 108, 109

Pooja, K. M., Mondal, S., and Chandra, J. (2022a). Exploiting higher order multi-dimensional relationships with self-attention for author name disambiguation. *ACM Transactions on Knowledge Discovery from Data*, 16(5). 57, 62, 63, 93, 95, 97

Pooja, K. M., Mondal, S., and Chandra, J. (2022b). Online author name disambiguation in evolving digital library. *Neurocomputing*, 493:1–14. 57

Pratama, M., Zhou, J., Liu, Y., and Lim, E.-P. (2019). Deep graph convolutional neural network for author name disambiguation. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 21

Qian, Y., Zheng, Q., Sakai, T., Ye, J., and Liu, J. (2015). Dynamic author name disambiguation for growing digital libraries. *Information Retrieval Journal*, 18(5):379–412. 51, 88

Qiao, Z., Du, Y., Fu, Y., Wang, P., and Zhou, Y. (2019). Unsupervised author disambiguation using heterogeneous graph convolutional network embedding. In *2019 IEEE international conference on big data (Big Data)*, pages 910–919. IEEE. 20, 79, 80

Qiping, D., Weijing, C., Ling, J., and Yu'e, Z. (2022). Author name disambiguation based on heterogeneous information network. *Data Analysis and Knowledge Discovery*, 6(4):60–68. 56, 63

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*. 18

Rastogi, C., Agarwal, P., and Singh, S. (2023). Exploring graph based approaches for author name disambiguation. 2

Rehs, A. (2021). A supervised machine learning approach to author disambiguation in the web of science. *Journal of Informetrics*, 15(3):101166. 54, 63

Rodrigues, N. and Ralha, C. (2024). A hybrid machine learning method to author name disambiguation. In *Anais do XV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 108–117, Porto Alegre, RS, Brasil. SBC. 91

Rodrigues, N. D. S. and Ralha, C. G. (2025). A flexible and configurable system to author name disambiguation. *IEEE Access*, pages 1–1. Early Access. 91

Rodrigues, N. S., Costa, A. R., Lemos, L. C., and Ralha, C. G. (2021). Multi-strategic approach for author name disambiguation in bibliography repositories. In Lossio-Ventura, J., Valverde-Rebaza, J., Díaz, E., and Alatrista-Salas, H., editors, *Information Management and Big Data. SIMBig 2020. Communications in Computer and Information Science, vol. 1410*. Springer, Cham. 55, 61

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. 15, 18

Santana, A. F., Gonçalves, M. A., Laender, A. H., and Ferreira, A. A. (2015). On the combination of domain-specific heuristics for author name disambiguation: the nearest cluster method. *International Journal on Digital Libraries*, 16(3):229–246. 50

Santana, A. F., Gonçalves, M. A., Laender, A. H. F., and Ferreira, A. A. (2017). Incremental author name disambiguation by exploiting domain-specific heuristics. *Journal of the Association for Information Science and Technology*, 68(4):931–945. 29

Santini, C., Gesese, G. A., Peroni, S., Gangemi, A., Sack, H., and Alam, M. (2022). A knowledge graph embeddings based approach for author name disambiguation using literals. *Scientometrics*, 127(8):4887–4912. 56, 63

Shi, C., Li, Y., Zhang, J., Sun, Y., and Yu, P. S. (2017). A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37. 20

Shin, D., Kim, T., Choi, J., and Kim, J. (2014). Author name disambiguation using a graph model with node splitting and merging based on bibliographic information. *Scientometrics*, 100(1):15–50. 2, 47, 48

Smalheiser, N. R. and Torvik, V. I. (2009). Author name disambiguation. *Annual review of information science and technology*, 43(1):1–43. 48, 49

Strotmann, A. and Zhao, D. (2012). Author name disambiguation: What difference does it make in author-based citation analysis? *Journal of the American Society for Information Science and Technology*, 63(9):1820–1833. 51

Sun, Q., Peng, H., Li, J., Wang, S., Dong, X., Zhao, L., Philip, S. Y., and He, L. (2020). Pairwise learning for name disambiguation in large-scale heterogeneous academic networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 511–520. IEEE. 2, 9

Tang, J., Fong, A. C., Wang, B., and Zhang, J. (2012). A unified probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):975–987. 50, 88

Tang, L. and Walsh, J. (2010). Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps. *Scientometrics*, 84(3):763–784. 50

Torvik, V. I. and Smalheiser, N. R. (2009). Author name disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data*, 3(3):1–29. 48, 50

Torvik, V. I., Weeber, M., Swanson, D. R., and Smalheiser, N. R. (2005). A probabilistic similarity metric for medline records: A model for author name disambiguation. *Journal of the American Society for Information Science and Technology*, 56(2):140–158. 48, 50

Trueswell, R. L. (1969). Some behavioral patterns of library users: The 80/20 rule. *Wilson Libr Bull.* 36

van Lieshout, J. (2024). Author name disambiguation using large language models. Bachelor Thesis, Delft University of Technology, The Netherlands. 105, 111

Vera-Olivera, H., Guo, R., Huacarpuma, R. C., Da Silva, A. P. B., Mariano, A. M., and Holanda, M. (2021). Data modeling and nosql databases - a systematic mapping review. *ACM Comput. Surv.*, 54(6). 32

Vogel, R. and Güttel, W. H. (2013). The dynamic capability view in strategic management: A bibliometric review. *International Journal of Management Reviews*, 15(4):426–446. viii, 38

VOSviewer (2022). Vosviewer - visualizing scientific landscapes. `https://www.vosviewer.com/`. Centre for Science and Technology Studies, Leiden University, The Netherlands. 36

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*. 18

Wang, C., He, X., and Zhou, A. (2020a). HEEL: exploratory entity linking for heterogeneous information networks. *Knowledge and Information Systems (KAIS)*, 62:485–506. 57, 61

Wang, H., Wan, R., Wen, C., Li, S., Jia, Y., Zhang, W., and Wang, X. (2020b). Author name disambiguation on heterogeneous information network with adversarial representation learning. In *Proceedings of* 34^{th} *AAAI Conference on Artificial Intelligence*, pages 238–245. AAAI Press. 57, 61

Wang, J., Berzins, K., Hicks, D., Melkers, J., Xiao, F., and Pinheiro, D. (2012). A boosted-trees method for name disambiguation. *Scientometrics*, 93(2):391–411. 50

Wang, S., Li, Q., and Koopman, R. (2023). Co-attention-based pairwise learning for author name disambiguation. In *International Conference on Asian Digital Libraries*, pages 240–249. Springer. 55, 64

Wang, W., Bao, H., Huang, S., Dong, L., and Wei, F. (2021). MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics. viii, 17, 18

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020c). MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc. 16, 18, 83, 90

Waqas, H. and Qadir, A. (2022). Completing features for author name disambiguation (AND): An empirical analysis. *Scientometrics*, 127(2):1039–1063. 58, 64

Waqas, H. and Qadir, M. A. (2021). Multilayer heuristics based clustering framework (MHCF) for author name disambiguation. *Scientometrics*, 126(9):7637–7678. 55, 63

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer Publishing Company, Incorporated. 6

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., and Wessln, A. (2012). *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated. 6

Wu, H., Li, B., Pei, Y., and He, J. (2014). Unsupervised author disambiguation using dempster–shafer theory. *Scientometrics*, 101(3):1955–1972. 50

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. 13

Xiong, B., Bao, P., and Wu, Y. (2021a). Learning semantic and relationship joint embedding for author name disambiguation. *Neural Computing and Applications*, 33(6):1987–1998. 1, 62

Xiong, B., Bao, P., and Wu, Y. (2021b). Learning semantic and relationship joint embedding for author name disambiguation. *Neural Computing and Applications*, 33:1987–1998. 57

Xu, J., Kim, S., Song, M., Jeong, M., Kim, D., Kang, J., Rousseau, J. F., Li, X., Xu, W., Torvik, V. I., et al. (2020). Building a PubMed knowledge graph. *Scientific data*, 7(1):1–15. 48, 52, 60

Yan, Q. and AsirAsir (2024). Synergizing large language models and tree-based algorithms for author name disambiguation. In *Submitted to KDD 2024 OAG-Challenge Cup*. 105

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019a). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32. 14

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 15

Ye, F., Xia, Z., Ling, Z., and Wu, L. (2025). Multi-view contrastive and cluster-guided learning for author name disambiguation. *Expert Systems with Applications*, 289:128324. 57, 65

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27. 13

Zelkowitz, M. V. and Wallace, D. (1997). Experimental validation in software engineering. *Information and Software Technology*, 39(11):735–743. Evaluation and Assessment in Software Engineering. 6

Zhang, L. and Ban, Z. (2020). Author name disambiguation based on rule and graph model. In Zhu, X., Zhang, M., Hong, Y., and He, R., editors, *Natural Language Processing and Chinese Computing*, pages 617–628, Cham. Springer International Publishing. 54, 56, 61

Zhang, L., Huang, Y., Yang, J., and Lu, W. (2021a). Aggregating large-scale databases for pubmed author name disambiguation. *Journal of the American Medical Informatics Association*, 28(9):1919–1927. 62

Zhang, L., Lu, W., and Yang, J. (2023). Lagos-and: A large gold standard dataset for scholarly author name disambiguation. *Journal of the Association for Information Science and Technology*, 74(2):168–185. xi, 89, 93, 99, 100, 107

Zhang, W., Yan, Z., and Zheng, Y. (2019). Author name disambiguation using graph node embedding method. In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 410–415. 48, 51, 52

Zhang, Z., Wu, C., Li, Z., Peng, J., Wu, H., Song, H., Deng, S., and Wang, B. (2021b). Author name disambiguation using multiple graph attention networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. 57, 62

Zhang, Z., Yu, B., Liu, T., and Wang, D. (2020). Strong baselines for author name disambiguation with and without neural networks. In Lauw, H. W., Wong, R. C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., and Pan, S. J., editors, *Advances in Knowledge Discovery and Data Mining*, pages 369–381, Cham. Springer International Publishing. 55, 61

Zhao, R. and Chen, Y. (2025). Scholar name disambiguation with search-enhanced llm across language. 105, 111

Zhou, A., Shi, M., and Yuan, R. (2024a). An effective author name disambiguation framework for large-scale publications. *IEEE Access*, 12:182086–182100. 2

Zhou, Q., Chen, W., Wang, W., Xu, J., and Zhao, L. (2021). Multiple features driven author name disambiguation. In *2021 IEEE International Conference on Web Services (ICWS)*, pages 506–515. 56, 61

Zhou, Q., Chen, W., Zhao, P.-P., Liu, A., Xu, J.-J., Qu, J.-F., and Zhao, L. (2024b). Towards effective author name disambiguation by hybrid attention. *Journal of Computer Science and Technology*, 39(4):929–950. 2, 55, 65

Zhu, Q., Du, B., and Yan, P. (2020). Self-supervised training of graph convolutional networks. `https://arxiv.org/abs/2006.02380`. Acessado em: Junho de 2025. 28, 86