Deduplicação de Indivíduos A Partir de *Record Linkage* Baseado em Busca Vetorial

Pablo Luiz Lemos Pita
Instituto Federal da Bahia
Departamento de Computação
Salvador - BA
pablolemos2004@hotmail.com

Robespierre Dantas da Rocha Pita
Universidade Federal da Bahia
Instituto de Computação
Salvador, Bahia
robespierre.pita@ufba.br

Grinaldo Lopes de Oliveira
Instituto Federal da Bahia
Departamento de Computação
Salvador, Bahia
grinaldolopes@ifba.edu.br

***Resumo*—Nos bancos de dados modernos, a recuperação de informações nem sempre pode ser realizada por meio de identificadores únicos. Em bases heterogêneas e sujeitas a inconsistências, torna-se necessário empregar técnicas capazes de identificar registros pertencentes à mesma entidade, mesmo na presença de variações textuais, dados incompletos ou erros de preenchimento. Nesse contexto, destaca-se o *record linkage*, técnica utilizada para vincular registros de uma mesma entidade a partir da comparação de atributos equivalentes.**

Tradicionalmente, o *record linkage* utiliza funções de similaridade textual, como Jaro, Jaro-Winkler e Levenshtein. Entretanto, essas abordagens podem apresentar limitações diante de variações semânticas e inconsistências mais complexas.

Este trabalho propõe uma abordagem de deduplicação de indivíduos baseada em *record linkage* utilizando busca vetorial aproximada. Para isso, empregam-se modelos de linguagem para representar registros por meio de *embeddings*, permitindo capturar relações semânticas entre diferentes formas de escrita, abreviações e erros comuns em bases de dados reais. A indexação e a recuperação dos vetores são realizadas por meio da estrutura *Hierarchical Navigable Small Worlds* (HNSW), possibilitando a busca eficiente por registros similares.

Por fim, a abordagem proposta é comparada a métodos tradicionais baseados em similaridade textual, considerando métricas de desempenho e qualidade, como precisão, acurácia, sensibilidade e F1-score. Os resultados obtidos permitem avaliar o potencial da busca vetorial aproximada como alternativa para aprimorar processos de deduplicação e *record linkage* em cenários com dados heterogêneos e inconsistentes.

***Palavras-chave*—*Information Retrieval*, *Record Linkage*, *Embeddings*, *Hierarchical Navigable Small Worlds*, *Deduplication*.**

I. Introdução

Atualmente, a recuperação da informação, definida como o processo de localizar documentos ou registros relevantes a partir de uma consulta, tem ganhado crescente importância em razão do aumento do volume de dados produzidos pelos sistemas modernos, especialmente aqueles de natureza não estruturada [2], [3].

Diferentemente dos dados estruturados, organizados em tabelas e manipulados por bancos de dados relacionais, os dados não estruturados não seguem um esquema rígido de representação, abrangendo informações como textos livres, imagens e áudios [2]. Em bancos de dados relacionais, a recuperação de informações normalmente é realizada por meio de chaves primárias que permitem relacionar registros de uma mesma entidade. Entretanto, em diversos cenários reais, registros referentes ao mesmo indivíduo podem apresentar variações de grafia, abreviações, erros de preenchimento ou inconsistências entre diferentes bases, inviabilizando sua identificação apenas por identificadores ou comparações exatas.

Nesse contexto, técnicas de *record linkage* são amplamente utilizadas para identificar registros que representam uma mesma entidade por meio da comparação de atributos equivalentes. Tradicionalmente, essas técnicas baseiam-se em funções de similaridade textual, que podem apresentar limitações diante de variações semânticas e inconsistências mais complexas.

Considerando esse cenário, o presente trabalho investiga a aplicação da busca vetorial aproximada para a deduplicação de indivíduos brasileiros, utilizando registros representados por *embeddings* gerados por modelos de linguagem natural pré-treinados e indexados por meio da estrutura HNSW [4]. O objetivo é avaliar o potencial dessa abordagem como alternativa às técnicas tradicionais de *record linkage* baseadas em similaridade textual.

A. Problema de Pesquisa

O processo de *record linkage* tradicional baseia-se na recuperação de candidatos e na comparação de registros por meio de funções de similaridade textual. Com a popularização de modelos de linguagem e de técnicas de busca vetorial aproximada, tornou-se pertinente investigar sua aplicação nesse contexto. Assim, este trabalho buscou responder ao seguinte problema de pesquisa: em que medida a utilização de *embeddings* e busca vetorial aproximada baseada em HNSW influencia a qualidade da deduplicação de indivíduos e o desempenho computacional do processo de *record linkage*, em comparação com abordagens tradcionais.

B. Objetivos Gerais

O presente trabalho teve como objetivo desenvolver e avaliar uma abordagem de deduplicação de indivíduos baseada em técnicas de *record linkage* com busca vetorial aproximada, investigando sua capacidade de identificar registros semanticamente semelhantes mesmo na presença de variações textuais, erros de digitação e outras inconsistências comuns em bases de dados. Além disso, buscou-se comparar seu desempenho com

métodos tradicionais de *record linkage*, analisando a qualidade dos resultados e o custo computacional, a fim de avaliar a viabilidade do uso de *embeddings* e busca vetorial aproximada como uma alternativa escalável e eficiente para processos de deduplicação de indivíduos.

C. Objetivos Específicos

Para atingir o objetivo geral deste trabalho, desenvolveu-se um conjunto de dados sintético capaz de representar desafios comuns em aplicações de record linkage, incluindo registros duplicados, erros ortográficos, variações fonéticas e informações ausentes.

Em seguida, utilizou-se a ferramenta CIDACS-RL [1] para executar processos de vinculação de registros baseados no algoritmo BM25 [16] e em funções tradicionais de similaridade textual, estabelecendo uma linha de base (baseline) para comparação. Adicionalmente, implementou-se uma funcionalidade de busca vetorial aproximada baseada em embeddings e similaridade do cosseno.

Para essa abordagem, avaliaram-se os modelos BERTimbau [14] e Multilingual E5 Base [15] quanto à capacidade de representar registros cadastrais e capturar similaridades semânticas.

Por fim, compararam-se três configurações de vinculação: (i) a abordagem tradicional do CIDACS-RL (baseline); (ii) o fluxo alternativo 1 (FA1), baseado em busca vetorial aproximada e similaridade do cosseno entre embeddings; e (iii) o fluxo alternativo 2 (FA2), que combina busca vetorial aproximada com funções tradicionais de similaridade textual. A comparação considerou o desempenho computacional e a qualidade das vinculações, por meio dos tempos de processamento e das métricas de precisão, sensibilidade, acurácia e F1-score, permitindo avaliar o impacto da incorporação de embeddings e recuperação vetorial ao fluxo do CIDACS-RL.

II. Recursos Computacionais

Os experimentos foram executados em um computador equipado com processador AMD Ryzen 7 5700U (8 núcleos e 16 threads), 8 GB de memória RAM e 512 GB de armazenamento SSD. O ambiente de software foi composto pelo CIDACS-RL 3.0, utilizando o Apache Spark 3.4.1 em modo local para o processamento distribuído e uma única instância do Elasticsearch 8.9 como mecanismo de indexação e recuperação dos registros.

III. Limitações

A infraestrutura computacional disponível e a ausência de acesso a bases de dados reais constituem limitações relevantes deste trabalho. A restrição de recursos computacionais impôs desafios relacionados à execução de experimentos em maior escala, limitando a avaliação das abordagens propostas em cenários com volumes mais elevados de registros e maior demanda de processamento.

Além disso, a indisponibilidade de bases de dados reais adequadas para experimentação restringiu a avaliação dos métodos em cenários que reproduzissem integralmente os desafios encontrados em aplicações práticas de *record linkage*.

IV. Fundamentação Teórica

A. record linkage

A recuperação da informação desempenha um papel central em muitos sistemas modernos, tais como comércio eletrônico e plataformas de pesquisa acadêmica. Tais sistemas, em seu núcleo, utilizam mecanismos de busca capazes não apenas de recuperar informação, mas de recuperar informação relevante [6]. A simples recuperação da informação não é suficiente: o mecanismo de busca deve apresentar-se como um autômato capaz de interpretar a intenção do usuário [6], isto é, daquele que fornece a consulta. Tal relevância encontra-se fortemente acoplada à lógica de negócio e depende de características capazes de direcionar a busca a resultados satisfatórios ao usuário [6].

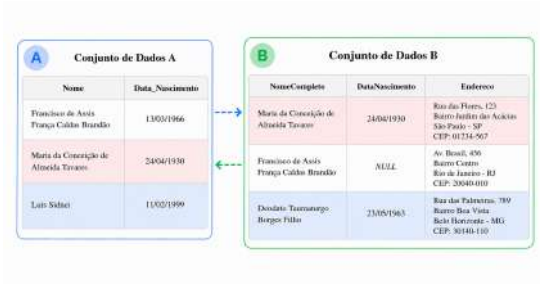

Fig. 1. *record linkage* como Enriquecedor de Dados

O *record linkage* consiste no processo de recuperação e associação de registros referentes a uma determinada entidade a partir da comparação de atributos semanticamente equivalentes entre um conjunto de dados A e um conjunto de dados-alvo B [1]. Nesse sentido, o *record linkage* consolida-se como um esforço de engenharia voltado à busca e recuperação massiva de registros em um ou mais conjuntos de dados, no qual um conjunto A atua como cliente, consultando outros conjuntos ou a si próprio em busca de entidades potencialmente equivalentes.

Uma das aplicações do *record linkage* é o enriquecimento de dados. Um conjunto de dados A pode consultar e comparar registros em um conjunto de dados B para recuperar informações adicionais acerca de uma entidade [1]. A Figura 1, por exemplo, demonstra a recuperação do campo “Endereco” de um indivíduo.

Outra aplicação, e especificamente objeto de estudo do presente trabalho, é a deduplicação de registros referentes à mesma entidade dentro de um mesmo conjunto de dados [1], os quais, entretanto, não possuem um identificador universal capaz de vinculá-los de maneira determinística, conforme demonstrado na figura de número 2.

Fig. 2. Exemplo de Duplicações em Conjuntos de Dados

Duas abordagens podem ser adotadas no contexto de *record linkage*: uma primeira, determinística, na qual um campo único ou um conjunto de campos [1] é utilizado para determinar se um par de registros, pertencentes ao mesmo ou a diferentes conjuntos de dados, refere-se à mesma entidade.

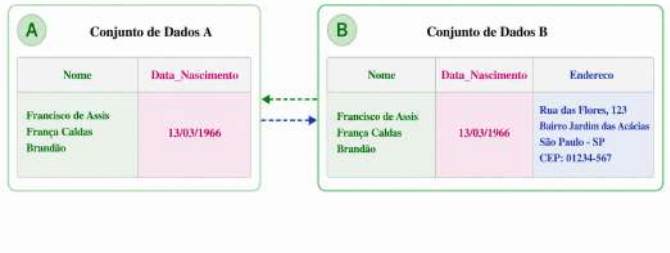

Fig. 3. *record linkage* Determinístico

Uma segunda abordagem, probabilística, lança mão de técnicas de cálculo de similaridade entre os dados [1], subsidiando o suporte à diversidade no que se refere à qualidade dos registros, que pode comportar situações como deleção, inserção e transposição de termos e caracteres em informações de indivíduos, como nomes, nomes maternos ou municípios de residência. O presente trabalho centrou seu experimento em torno desta segunda abordagem. Tal abordagem visa resolver problemas como o da figura 4. Ao olho humano, registros como ”Francisco de A. F. C. Brandão” e “Franscisco de Assis França Caldas Brandão” podem se referir ao mesmo indivíduo, apesar das diferenças textuais decorrentes de abreviações, ausência de acentuação, erros de digitação ou preenchimento incompleto. Dessa forma, a abordagem probabilística busca identificar correspondências mesmo quando os registros não são exatamente iguais, concretizando o problema relacionado à qualidade e padronização dos dados.

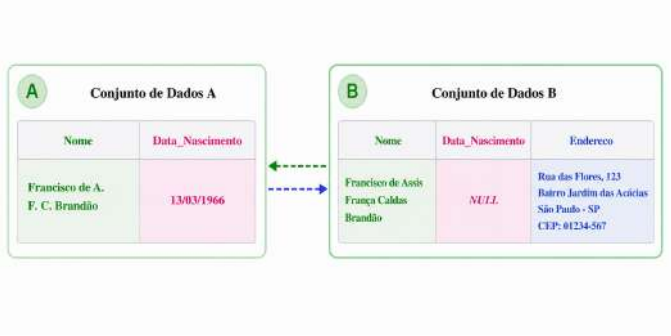

Fig. 4. *record linkage* Probabilístico

O *record linkage* é um processo estruturado em cinco etapas principais: pré-processamento, indexação, comparação de pares, classificação e avaliação de acurácia [1].

Na primeira etapa, o pré-processamento, como o nome sugere, realizam-se transformações sobre os conjuntos de dados com o objetivo de minimizar problemas de qualidade. Entre essas transformações, destacam-se a padronização de campos textuais (por exemplo, conversão para letras minúsculas ou maiúsculas), a remoção de caracteres especiais e a criação de colunas auxiliares, como códigos fonéticos, entre outras estratégias [1].

A comparação de registros é, sem dúvida, o principal gargalo do processo. Isso ocorre porque a comparação exaustiva de todos os registros de um conjunto de dados A com todos os registros de um conjunto B resulta em uma complexidade de ordem Nºb × Nºb, o que torna o processo inviável em larga escala. Nesse contexto, a etapa de indexação propõe a aplicação de uma ou mais regras para reduzir o número de pares candidatos à comparação. Essa etapa é altamente dependente do domínio do problema e do conhecimento sobre os dados. [1]

A terceira etapa consiste na comparação propriamente dita, na qual os campos equivalentes entre os registros são analisados. No *record linkage* determinístico, verifica-se se os valores dos campos são exatamente iguais. Já no *record linkage* probabilístico, mede-se o grau de similaridade entre os valores, utilizando métricas específicas. [1]

Após a comparação, os pares são classificados de acordo com critérios definidos pela lógica de negócio. Por exemplo, pode-se estabelecer um ponto de corte baseado em uma média ponderada das similaridades obtidas, classificando os pares como correspondentes ou não correspondentes. Essa constitui a quarta etapa do processo. [1]

Por fim, na quinta etapa, realiza-se a avaliação da acurácia dos resultados, que pode incluir revisão manual dos pares classificados, especialmente em casos limítrofes ou de maior incerteza. [1]

B. Vetores como Representação de Dados - embeddings

Nos mais diversos conjuntos de dados é possível identificar uma diversidade de formas de se referir a um mesmo indivíduo, conforme ilustrado na Figura 4. Este trabalho propõe uma abordagem que reconhece essa diversidade e pretende

fazer uso de um instrumental técnico capaz de lidar com tais variações.

Uma abordagem relevante no campo da recuperação da informação consiste na representação de dados como vetores, uma ferramenta matemática amplamente utilizada no aprendizado de máquina [7]. Seu objetivo é mapear as características dos dados em dimensões [7], de modo que, ao serem inseridos em um espaço de alta dimensionalidade, cada elemento seja representado por um ponto definido por suas coordenadas.

Atualmente, modelos de linguagem natural têm sido utilizados para avaliar dados em múltiplas dimensões e produzir vetores de características que capturam a semântica, as regras sintáticas de uma determinada língua e o contexto [8], os chamados *embeddings*.

Conforme exemplificado na Figura 5, o dado é fornecido a um modelo de linguagem natural, que o codifica como um vetor de números reais, em que cada posição representa uma dimensão.

Fig. 5. Codificação Textual em *embeddings*

A ideia central é que dados semanticamente e sintaticamente semelhantes produzem *embeddings* igualmente, de modo que, no espaço de alta dimensionalidade, seus pontos estejam posicionados próximos entre si [8] e possam ter suas distâncias avaliadas por meio de uma função de distância entre vetores [8].

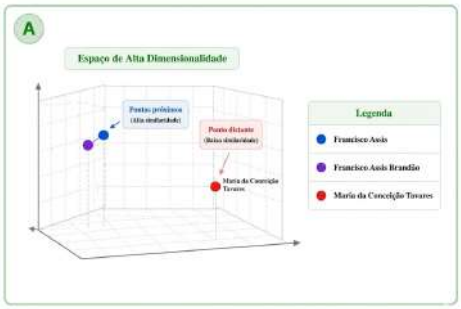

Fig. 6. Representação Textual no Espaço de Alta Dimensionalidade

Desse modo, a abordagem específica adotada por este trabalho para lidar com a diversidade de dados consiste na representação de indivíduos por meio de *embeddings*.

C. Aproximate Nearest Neighbour Search

Tratando-se do problema de busca, busca-se sempre evitar que todos os elementos de uma determinada estrutura de dados sejam percorridos até que o elemento desejado seja encontrado.

Muito utilizada na resolução de problemas em aprendizado de máquina, visão computacional e bancos de dados [9], o *Nearest Neighbour Search* (NNS) consiste em um problema no qual, dado um conjunto de pontos em um espaço, para uma determinada consulta retornam-se os vizinhos mais próximos de acordo com uma função de distância definida [9].

No contexto da abordagem adotada neste trabalho, a Figura 7 ilustra um exemplo aplicado ao caso de indivíduos. A busca por ”Raul Sei”, representado como um ponto no espaço, seria comparada aos demais pontos desse mesmo espaço, os quais poderiam corresponder a registros armazenados em uma estrutura de dados de um banco de dados que implementa o NNS como método de busca.

Um fator problemático encontra-se na abordagem mais simples para esse tipo de problema: a comparação entre a consulta e todos os pontos do espaço. Caso a quantidade de pontos seja muito grande, o processo resultaria em uma complexidade O(N), uma vez que todos os elementos precisariam ser comparados com a consulta [9].

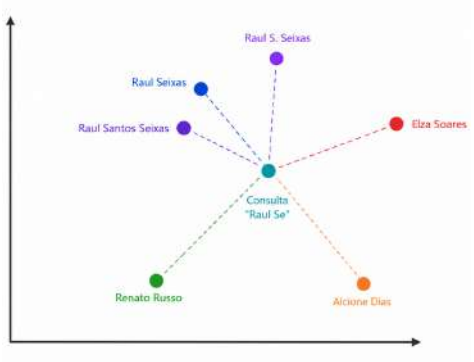

Fig. 7. Busca pelos Vizinhos mais Próximos

Entretanto, em diversas aplicações, é aceitável barganhar levemente a acurácia dos resultados em favor de uma solução computacionalmente mais viável, tanto em termos de tempo de execução quanto de utilização de memória. Nesse contexto, os algoritmos pertencentes à classe *Approximate Nearest Neighbor Search* (ANNS) destacam-se por oferecer um equilíbrio entre desempenho e recuperação de informações relevantes.

De forma semelhante ao problema de NNS, esses algoritmos têm como objetivo identificar os K vizinhos mais similares a uma determinada consulta, considerando uma função de distância previamente definida [10]. Contudo, em vez de garantir a obtenção dos vizinhos exatos mais próximos através da comparação com os N vizinhos, empregam estratégias heurísticas que permitem reduzir significativamente o custo computacional da busca.

Frequentemente, tais abordagens baseiam-se na construção de índices especializados, isto é, estruturas de dados projetadas para viabilizar buscas eficientes em grandes volumes de dados. Essas estruturas procuram equilibrar dois objetivos fundamentais: a relevância dos resultados recuperados e a velocidade da consulta. Dentre as soluções existentes, o HNSW [13], objeto de estudo deste trabalho, destaca-se por sua capacidade de conciliar elevada qualidade dos resultados com tempos de busca reduzidos.

D. Hierarchical Navigable Small Worlds

Um dos principais desafios abordados no campo da Recuperação da Informação está relacionado à estrutura de dados utilizada para realizar a busca e recuperação de registros pertencentes a uma determinada coleção. A escolha dessa estrutura impacta diretamente o desempenho das operações de consulta, influenciando fatores como tempo de resposta, consumo de memória e escalabilidade do sistema.

Em aplicações modernas de busca e recuperação de documentos, como o Elasticsearch e o Solr, é amplamente utilizada uma estrutura denominada Índice Invertido. Sua principal finalidade é reorganizar os dados de forma a possibilitar consultas mais eficientes sobre grandes volumes de informação. Para isso, os termos presentes nos documentos são indexados individualmente e associados a listas de identificadores dos registros nos quais ocorrem. Dessa forma, em vez de percorrer toda a coleção durante uma consulta, o sistema pode acessar diretamente os documentos relacionados aos termos pesquisados, reduzindo significativamente o custo computacional da operação [6].

Nos anos 1990, William Pugh propôs a estrutura de dados denominada *skip lists* [12] como uma alternativa probabilística às árvores balanceadas, oferecendo desempenho semelhante com menor complexidade de implementação.

As *skip lists* organizam os elementos em múltiplos níveis hierárquicos, nos quais parte dos elementos é promovida probabilisticamente para camadas superiores. Essa organização permite que as buscas realizem grandes ”saltos” nos níveis mais altos e sejam refinadas nos níveis inferiores, alcançando complexidade temporal média logarítmica.

Os princípios introduzidos pelas *skip lists* influenciaram o desenvolvimento de estruturas para busca aproximada em grafos, como a *Navigable Small World* (NSW) e, posteriormente, a *Hierarchical Navigable Small World* (HNSW), ilustradas na Figura X.

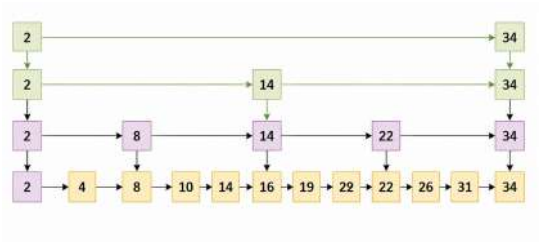

Fig. 8. *skip lists*

O HNSW adota uma abordagem que combina conceitos dos grafos de navegação com a estrutura hierárquica das *skip lists*. Sua ideia central consiste em organizar os elementos em múltiplas camadas de grafos, nas quais os níveis superiores possuem menos vértices e conexões de longo alcance, enquanto os níveis inferiores apresentam uma conectividade progressivamente maior [11]. De maneira semelhante às *skip lists* propostas por William Pugh, todos os elementos estão presentes na camada mais inferior, que representa o grafo completo da estrutura [11].

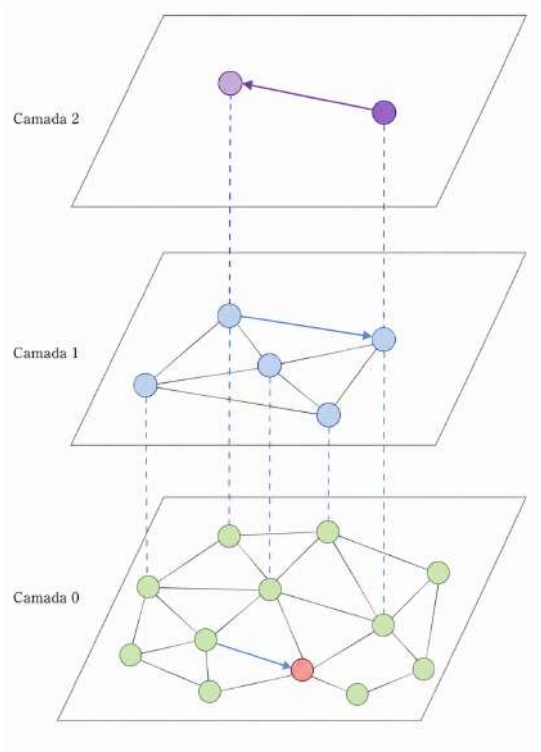

Fig. 9. Hierarchical Navigable Small World

O processo de busca é realizado por meio de comparações de distância entre o elemento consultado e os vértices visitados, reduzindo a necessidade de explorar regiões pouco promissoras do espaço de busca. A navegação inicia-se no nível mais alto da hierarquia e prossegue de forma descendente, camada por camada [12]. Em cada nível, o algoritmo percorre os vértices conectados ao ponto de entrada atual até encontrar um vértice cuja distância para o elemento consultado seja menor do que a dos demais candidatos disponíveis [12]. Esse vértice passa então a ser utilizado como ponto de entrada para a camada imediatamente inferior [12].

Ao repetir esse procedimento sucessivamente até alcançar o nível mais baixo, o HNSW consegue localizar vizinhos aproximados de forma eficiente [12], explorando inicialmente conexões de longo alcance e refinando gradualmente a busca em regiões mais próximas da consulta. Assim como as *skip lists*, essa estratégia permite que o algoritmo apresente complexidade média logarítmica para operações de busca,

tornando-o altamente escalável para coleções de grande dimensão [12].

V. Metodologia

A. Conjunto de Dados

Para o presente trabalho, foram extraídos dados públicos da primeira chamada de aprovados nos cursos da Universidade Federal da Bahia (UFBA) em 2026. A partir desses dados, foi construído um conjunto de dados base contendo os nomes dos candidatos, mantendo apenas registros com nomes únicos.

Com o objetivo de compor um conjunto de dados sintético mais rico, foram geradas datas de nascimento fictícias e utilizado um conjunto de bairros do município de Salvador para a criação do atributo fictício bairro. Dessa forma, o conjunto de dados passou a ser composto pelos atributos nome, bairro e data de nascimento, representando informações frequentemente utilizadas em tarefas de *record linkage*. Adicionalmente, foi criado um identificador único para cada indivíduo, permitindo a posterior avaliação da qualidade das soluções propostas por meio de métricas de desempenho comumente utilizadas no *record linkage*.

Em seguida, foi desenvolvido um *script* Python responsável pela geração de um conjunto de dados alvo. Esse processo incluiu a duplicação de registros, com um limite máximo de dez duplicações por indivíduo, bem como a inserção controlada de erros sintéticos. Nos campos textuais de nome completo e bairro, foram aplicadas operações de inserção, deleção, substituição e transposição de caracteres, além da remoção de termos, simulando inconsistências comuns em processos de entrada e integração de dados. Também foram introduzidos erros fonéticos, por meio da substituição de grafias por variantes foneticamente semelhantes, como a troca de ”T” por ”TH”, reproduzindo variações de escrita frequentemente encontradas em registros reais. Adicionalmente, foram gerados valores nulos para os campos de data de nascimento e bairro, reproduzindo situações de ausência de informação frequentemente observadas em bases de dados reais. Dessa forma, buscou-se construir um cenário experimental mais próximo das condições encontradas em aplicações práticas de *record linkage*.

A construção desse cenário experimental teve como objetivo criar um ambiente desafiador para a avaliação das soluções propostas neste trabalho, permitindo analisar sua robustez diante de diferentes tipos de erros e variações nos dados.

B. Limitações do Conjunto de Dados

Uma análise complementar realizada neste trabalho consistiu em investigar o quão desafiador o conjunto de dados sintéticos se mostrou para as diferentes abordagens de *record linkage*. Em vez de restringir a avaliação apenas às métricas tradicionais de desempenho, como precisão, sensibilidade, acurácia e F1-Score, buscou-se compreender a distribuição das pontuações (*match scores*) atribuídas aos pares de registros ao longo do processo de recuperação e re-ranqueamento. Essa análise permite identificar, considerando o ponto de corte ótimo determinado por meio da curva ROC (apresentada na

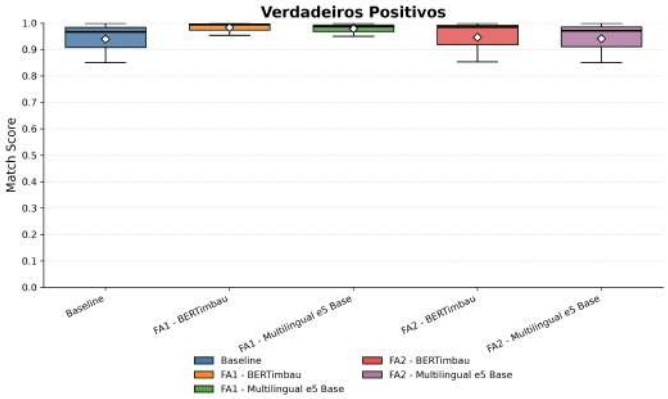

Fig. 10. Distribuição de Verdadeiros Positivos em Relação ao Match Score

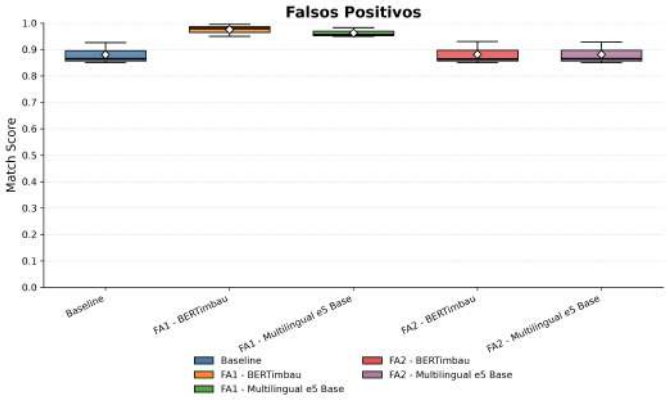

Fig. 11. Distribuição de Falsos Positivos em Relação ao Match Score

Seção de Resultados), em quais faixas de similaridade se concentraram os pares corretamente e incorretamente classificados, proporcionando uma compreensão mais aprofundada do grau de dificuldade imposto pelo conjunto de dados às abordagens avaliadas.

Para esse fim, foram coletados, para cada abordagem os valores mínimo, médio e máximo do *match score* dos pares pertencentes às quatro categorias da matriz de confusão: Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Negativos. A partir dessas estatísticas, foram construídos gráficos cujo objetivo é sintetizar a distribuição das pontuações obtidas por cada grupo, permitindo visualizar a região do espaço de similaridade ocupada por cada tipo de classificação.

Embora o conjunto de dados sintéticos tenha proporcionado uma avaliação controlada e reprodutível, a análise sugere que sua capacidade de representar a diversidade e a complexidade dos erros presentes em cenários práticos é limitada. Dessa forma, entende-se que, como continuidade da presente pesquisa, as abordagens propostas devem ser avaliadas em bases de dados reais, permitindo verificar se o desempenho observado neste trabalho se mantém diante de inconsistências naturalmente presentes em aplicações reais. Nesse contexto, a análise da distribuição dos *match scores* poderá ser novamente

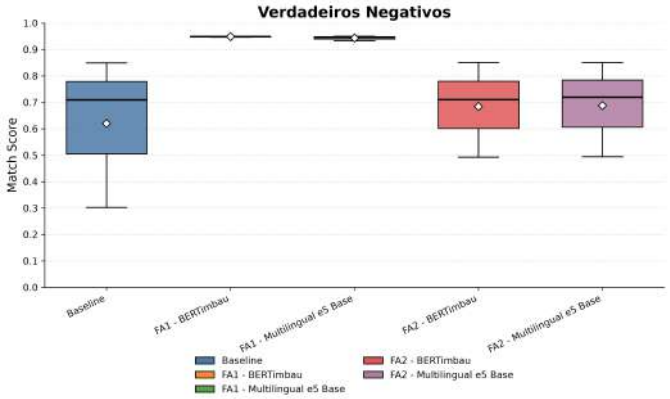

Fig. 12. Distribuição de Verdadeiros Negativos em Relação ao Match Score

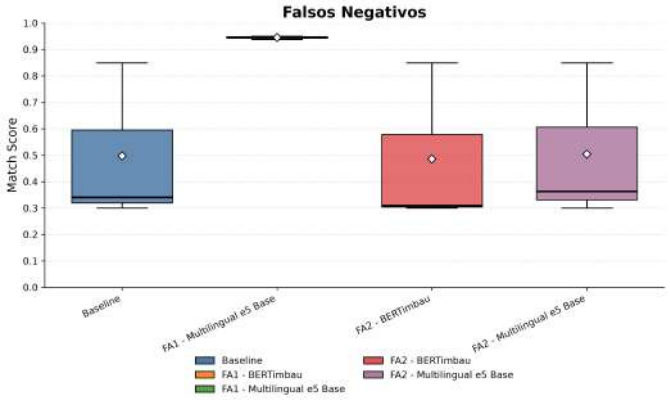

Fig. 13. Distribuição de Falsos Negativos em Relação ao Match Score

empregada para investigar se os padrões identificados neste estudo permanecem consistentes, ampliando as evidências acerca da robustez, da capacidade de generalização e da aplicabilidade dos métodos propostos.

C. CIDACS-RL e o Fluxo de Vinculação Original

O CIDACS-RL [1] é uma ferramenta de vinculação de registros desenvolvida pelo Centro de Integração de Dados e Conhecimentos para Saúde (CIDACS/Fiocruz Bahia), projetada para realizar a integração de grandes bases de dados administrativas e de saúde. Na arquitetura proposta, o processo de vinculação é estruturado em três componentes principais: (i) a indexação dos registros da base alvo no Elasticsearch, (ii) a recuperação de candidatos por meio do Elasticsearch e (iii) um núcleo de comparação e re-ranqueamento baseado em funções clássicas de similaridade textual. Dessa forma, o fluxo de vinculação é organizado em duas etapas principais: recuperação de candidatos e comparação dos registros.

Na primeira etapa, os registros da base fonte foram utilizados para consultar a base alvo indexada no Elasticsearch, buscando registros potencialmente correspondentes.

Em seguida, os candidatos recuperados foram submetidos à etapa de comparação. Inicialmente, realizou-se uma vinculação baseada em correspondências exatas. Posteriormente, os candidatos remanescentes foram avaliados com relaxamento nas regras de similaridade, permitindo a identificação de pares mesmo na presença de significativas divergências decorrentes de erros de digitação, variações ortográficas, abreviações e alterações fonéticas.

Além disso, foram atribuídos pesos aos atributos utilizados no processo de comparação, de forma a refletir sua importância relativa na identificação de correspondências. Os pesos associados aos campos nome, bairro e data de nascimento foram definidos a partir da execução de uma amostra de registros submetida a um procedimento de otimização baseado em algoritmos genéticos [13]. Esse processo buscou identificar combinações de pesos capazes de maximizar o desempenho da vinculação, fornecendo uma parametrização mais adequada às características do conjunto de dados analisado.

Ao combinar uma etapa inicial de recuperação de candidatos baseada em BM25 [16] por meio do Elasticsearch com fases de comparação exata e aproximada, o fluxo de vinculação tornou-se capaz de recuperar tanto registros idênticos quanto registros contendo inconsistências sintéticas. Essa configuração corresponde à abordagem de referência (baseline) adotada neste trabalho.

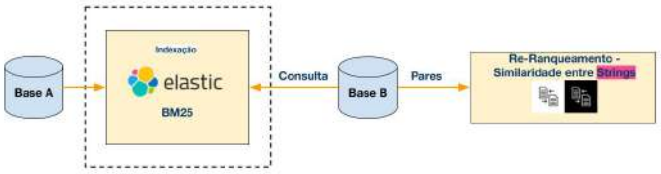

Fig. 14. Abordagem Baseline

D. Fluxo de Vinculação Baseado em Busca Vetorial Aproximada

No processo de *record linkage* baseado em busca vetorial aproximada, os atributos nome, bairro e data de nascimento foram inicialmente concatenados em uma única representação textual para cada registro. Em seguida, esses textos foram submetidos a modelos de linguagem para a codificação em *embeddings*, capazes de representar os registros em um espaço vetorial no qual registros semelhantes tenderiam a ocupar posições próximas.

Foram avaliados dois modelos *open-source* para a geração de *embeddings*: o BERTimbau [14], em sua versão base, desenvolvido especificamente para a língua portuguesa, e o Multilingual E5 [15], também em sua versão base, projetado para aplicações multilíngues de recuperação de informação. Para cada modelo, foram gerados *embeddings* para os conjuntos de dados fonte e alvo a partir da concatenação estruturada dos atributos de cada registro. Nessa representação, cada atributo foi precedido pelo respectivo nome da coluna, fornecendo contexto adicional ao modelo. Por exemplo, um registro contendo os valores (João Victor Souza, 08/12/2004, Pituaçu) foi representado como ”Coluna ’Nome’ Valor ’João Victor Souza’ Coluna ’Data de Nascimento’ Valor ’08/12/2004’ Coluna ’Bairro’ Valor ’Pituaçu’”, permitindo a geração de um vetor de *embeddings* representativo para cada registro.

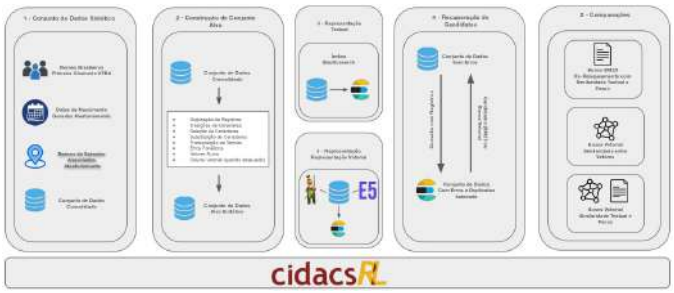

Fig. 17. Visão Geral da Metodologia

Após a etapa de vetorização, os registros passaram a ser representados por vetores em um espaço multidimensional. A recuperação de candidatos foi então realizada por meio de busca vetorial aproximada, utilizando a similaridade do cosseno como critério de ranqueamento para identificar os registros mais próximos entre as bases. Essa estratégia permite recuperar candidatos semanticamente semelhantes mesmo na presença de erros de digitação, variações ortográficas, abreviações, erros fonéticos e valores ausentes.

Uma vez recuperados os candidatos, foram avaliadas duas estratégias distintas para a etapa de comparação dos registros. Na primeira, a comparação foi realizada diretamente entre os *embeddings* por meio da similaridade do cosseno, mantendo tanto a recuperação de candidatos quanto a comparação fundamentadas em representações vetoriais (FA1). Na segunda, os candidatos recuperados pela busca vetorial foram comparados e re-ranqueados utilizando as funções clássicas de similaridade textual empregadas pelo CIDACS-RL, permitindo avaliar o impacto da recuperação vetorial de forma independente da estratégia de comparação adotada (FA2).

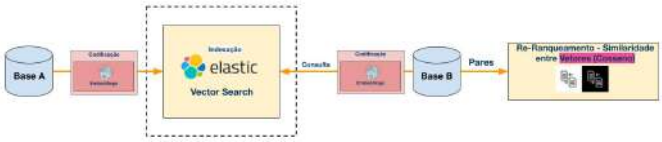

Fig. 15. Abordagem FA1

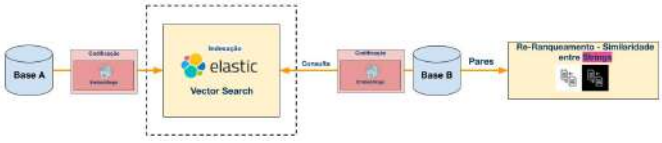

Fig. 16. Abordagem FA2

A utilização de diferentes modelos de *embeddings* e de distintas estratégias de comparação possibilitou avaliar o impacto das representações vetoriais tanto na recuperação de candidatos quanto na qualidade final das vinculações, permitindo comparar abordagens puramente vetoriais e híbridas em um cenário de *record linkage* aplicado a dados cadastrais.

E. Análise de Qualidade

Para cada processo de vinculação, os resultados recuperados foram submetidos a diferentes pontos de corte de similaridade, com o objetivo de analisar o impacto desses valores sobre a qualidade da vinculação. Foram avaliados limiares compreendidos entre 0,50 e 1,00, com incrementos de 0,05, permitindo observar o comportamento dos métodos em diferentes níveis de exigência para aceitação de correspondências.

Adicionalmente, foi construída a curva ROC (*Receiver Operating Characteristic*), a partir da qual foi calculado o Índice de Youden. O ponto de corte correspondente ao maior valor desse índice foi considerado o limiar ótimo, por representar o melhor equilíbrio entre sensibilidade e especificidade.

Para cada ponto de corte avaliado, foram construídos gráficos das métricas de qualidade da vinculação, incluindo precisão, sensibilidade, especificidade, acurácia e F1-score. A análise dessas métricas, em conjunto com a curva ROC e o Índice de Youden, permitiu avaliar o desempenho de cada abordagem e identificar o ponto de corte mais adequado para cada configuração experimental.

F. Análise de Desempenho

Ao final da execução dos processos de *record linkage*, foram realizadas análises de desempenho para ambas as abordagens estudadas. Para isso, foram registrados os instantes de início e término das etapas de indexação e vinculação por meio de arquivos de log gerados durante a execução dos experimentos.

Para cada configuração avaliada, o tempo de indexação foi obtido a partir da média de cinco execuções independentes, com o objetivo de reduzir a influência de variações ocasionais decorrentes do ambiente computacional. O tempo de processamento da etapa de vinculação foi mensurado durante a execução de cada configuração experimental.

Os resultados obtidos foram apresentados na forma de gráficos comparativos, permitindo analisar o comportamento das diferentes abordagens em relação ao tempo de indexação e ao tempo de execução do processo de vinculação.

G. Reprodutibilidade

O código-fonte desenvolvido para a realização dos experimentos encontra-se disponível publicamente no repositório GitHub: PabloLemosPita/TCC.

VI. Resultados

A. Análise de Qualidadade da Vinculação

Os experimentos demonstraram que a substituição integral do fluxo tradicional do CIDACS-RL por uma abordagem baseada exclusivamente em busca vetorial e similaridade cosseno entre *embeddings* não produziu resultados satisfatórios em termos de qualidade da vinculação. Embora a busca vetorial tenha recuperado candidatos semanticamente semelhantes, a etapa de re-ranqueamento baseada apenas na similaridade cosseno dos vetores apresentou redução nos indicadores de precisão e sensibilidade quando comparada à abordagem tradicional.

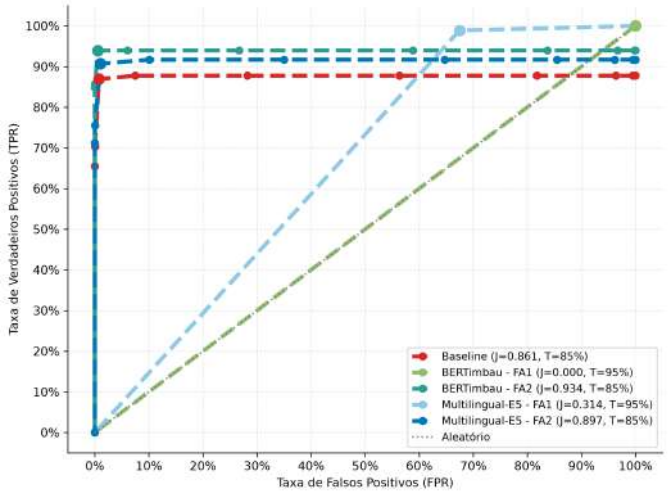

Fig. 18. Curva ROC e Índice de Youden em cada Abordagem

Em contrapartida, observou-se que a utilização da busca vetorial apenas como mecanismo de recuperação de candidatos, seguida da etapa de comparação baseada nos valores originais dos atributos e em funções clássicas de similaridade textual ponderadas por pesos previamente otimizados, produziu resultados significativos. Nessa configuração, a abordagem proposta apresentou desempenho competitivo em relação ao CIDACS-RL original, mantendo níveis de precisão semelhantes e alcançando ganhos em sensibilidade, acurácia e f1-score.

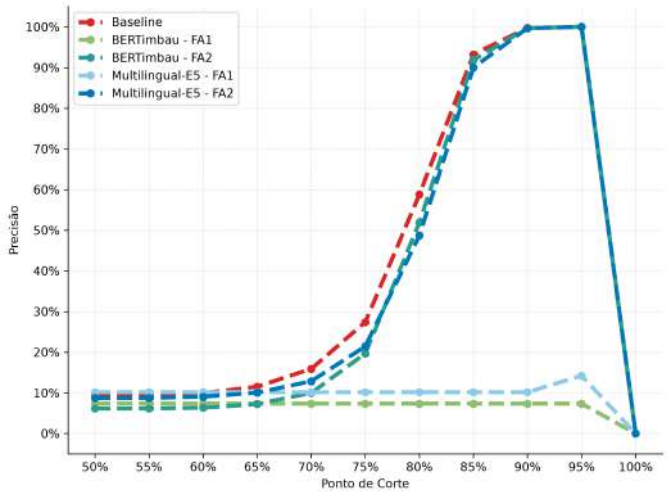

Fig. 19. Análise da Precisão ao Longo dos Pontos de Corte em cada Abordagem

B. Análise de Desempenho

Em relação ao desempenho computacional, observou-se um aumento no tempo de execução das abordagens baseadas em *embeddings* quando comparadas ao método tradicional baseado apenas em atributos textuais. Na etapa de indexação, o tempo necessário para a criação do índice vetorial foi aproximadamente três vezes superior ao da indexação textual.

Método	Tempo (Min)	Ponto de Corte (%)	Precisão (%)	Sensibilidade (%)	Acurácia (%)	F1-Score (%)
Baseline	0.57	85.0	93.24	86.91	97.90	89.96
FA1 (BERTimbau)	45.63	95.0	7.37	100.00	7.39	13.73
FA1 (Multilingual-E5 Base)	46.51	95.0	14.22	98.87	39.30	24.87
FA2 (BERTimbau)	45.85	85.0	91.85	93.93	99.06	92.88
FA2 (Multilingual-E5 Base)	46.64	85.0	90.01	90.70	98.17	90.35

Fig. 20. Resumo dos Pontos de Corte Indicados pela Curva ROC

Entretanto, essa diferença mostrou-se pouco significativa em termos práticos, uma vez que ambas as abordagens concluíram essa etapa em poucos segundos.

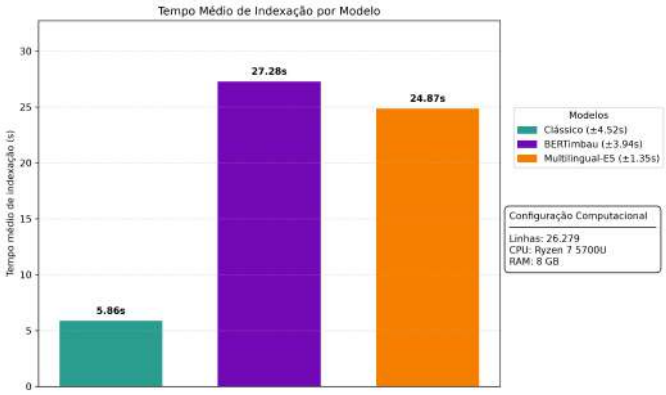

Fig. 21. Desempenho da Indexação

O maior impacto foi observado durante o processamento da vinculação. Enquanto a abordagem tradicional concluiu o processo em aproximadamente um minuto, as abordagens baseadas em busca vetorial demandaram pouco mais de quarenta minutos para completar a vinculação. Esse aumento expressivo decorre, principalmente, do custo adicional associado ao processamento das representações vetoriais, bem como das operações de recuperação e comparação envolvendo *embeddings*.

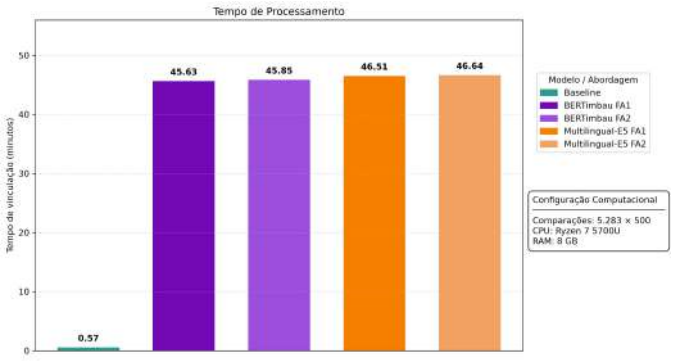

Fig. 22. Desempenho do Processamento da Vinculação

VII. Conclusão e Trabalhos Futuros

Os resultados obtidos indicam que a busca vetorial apresenta elevado potencial como mecanismo de ranqueamento na etapa inicial de recuperação de candidatos relevantes. Entretanto, uma etapa subsequente de re-ranqueamento baseada em funções clássicas de similaridade textual, associada à adequada parametrização do processo de vinculação, mostrou-se determinante para a obtenção dos melhores resultados na avaliação final dos pares candidatos. Além disso, os resultados evidenciam que os ganhos em qualidade da vinculação devem ser analisados em conjunto com as contrapartidas relacionadas ao desempenho computacional, uma vez que abordagens mais sofisticadas podem implicar aumento no custo de processamento.

O presente trabalho foi desenvolvido utilizando modelos de linguagem pré-treinados, dados sintéticos e uma infraestrutura computacional limitada. Embora essas condições tenham sido suficientes para a realização dos experimentos e para a análise comparativa das abordagens propostas, elas também impõem restrições quanto à generalização dos resultados obtidos. Nesse contexto, identificam-se algumas oportunidades relevantes para a continuidade desta pesquisa, visando ampliar sua aplicabilidade e aprofundar a compreensão acerca do uso de busca vetorial em processos de *record linkage*.

a) Realizar o *fine-tuning* dos modelos de linguagem empregados, especializando-os para a tarefa de deduplicação de nomes e vinculação de registros. Espera-se que essa adaptação permita a obtenção de representações vetoriais mais adequadas ao domínio do problema, aumentando a capacidade de discriminação entre registros verdadeiramente correspondentes e registros semelhantes, porém distintos.

b) Avaliar modelos de *Sentence Transformers* mais recentes, incluindo modelos da família *BGE*, bem como investigar o emprego de *rerankers* neurais para o refinamento dos candidatos recuperados durante o processo de *record linkage*.

c) Avaliar as abordagens propostas em bases de dados reais, compostas por registros provenientes de cenários práticos de aplicação. A utilização de dados reais possibilitará observar o comportamento dos métodos diante de inconsistências, erros de digitação, variações linguísticas e demais desafios característicos de bases administrativas, permitindo uma validação mais representativa dos resultados alcançados.

d) Investigar o uso de modelos de linguagem na etapa de pré-processamento, realizando a normalização de registros antes da vinculação, com o objetivo de reduzir inconsistências textuais e melhorar a qualidade dos candidatos recuperados.

e) Comparar o desempenho da abordagem proposta utilizando diferentes mecanismos de indexação e recuperação vetorial, incluindo bancos vetoriais como *Milvus*, *Qdrant* e *pgvector*, além da biblioteca *FAISS*, analisando aspectos relacionados ao desempenho computacional, escalabilidade e qualidade das vinculações.

f) Avaliar o desempenho das soluções em ambientes distribuídos e em infraestruturas computacionais mais robustas, contemplando maior capacidade de processamento, memória e paralelismo. Essa avaliação permitirá analisar aspectos relacionados à escalabilidade, ao tempo de execução e ao consumo de recursos computacionais em bases contendo milhões de registros.

g) Investigar o uso de classificadores para a decisão de vinculação entre registros, substituindo a utilização de pontos de corte fixos por modelos capazes de aprender automaticamente a distinguir pares correspondentes e não correspondentes.

Além dessas perspectivas, futuras investigações podem explorar outras estratégias de recuperação e re-ranqueamento de candidatos, bem como a utilização de novos modelos de representação textual e técnicas de otimização de parâmetros. Essas extensões têm potencial para ampliar tanto a eficiência computacional quanto a qualidade da vinculação, contribuindo para a evolução de soluções baseadas em busca vetorial aplicadas ao processo de *record linkage*.

Referências

[1] BARBOSA, George CG et al. CIDACS-RL: a novel indexing search and scoring-based *record linkage* system for huge datasets with high accuracy and scalability. BMC medical informatics and decision making, v. 20, n. 1, p. 289, 2020.

[2] SCHÜTZE, Hinrich; MANNING, Christopher D.; RAGHAVAN, Prabhakar. Introduction to information retrieval. Cambridge: Cambridge University Press, 2008.

[3] JAIN, Shubham; DE BUITLEIR, Amy; FALLON, Enda. A review of unstructured data analysis and parsing methods. In: 2020 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2020. p. 164-169.

[4] MALKOV, Yu A.; YASHUNIN, Dmitry A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE transactions on pattern analysis and machine intelligence, v. 42, n. 4, p. 824-836, 2018.

[5] ELMASRI, Ramez. Sistemas de banco de dados: fundamentos e aplicações. LTC, 2002.

[6] BERRYMAN, John; TURNBULL, Doug. Relevant search: with applications for Solr and Elasticsearch. Simon and Schuster, 2016.

[7] BRUCH, Sebastian. Foundations of vector retrieval. Springer, 2024.

[8] ZALAND, Obaidullah; ABULAISH, Muhammad; FAZIL, Mohd. A comprehensive empirical evaluation of existing word embedding approaches. arXiv preprint arXiv:2303.07196, 2023.

[9] ANDONI, Alexandr; INDYK, Piotr; RAZENSHTEYN, Ilya. Approximate nearest neighbor search in high dimensions. In: Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018. 2018. p. 3287-3318.

[10] LU, Kejing et al. Approximate Nearest Neighbor Search for Modern AI: A Projection-Augmented Graph Approach. arXiv preprint arXiv:2603.06660, 2026.

[11] PATELLA, Marco; CIACCIA, Paolo. Approximate similarity search: A multi-faceted problem. Journal of Discrete Algorithms, v. 7, n. 1, p. 36-48, 2009.

[12] UGH, William. *skip lists*: a probabilistic alternative to balanced trees. Communications of the ACM, v. 33, n. 6, p. 668-676, 1990.

[13] ITA, Pablo L. et al. Optimizing *record linkage* Parameters with Genetic Algorithms for Health Data Integration. In: Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS). SBC, 2026. p. 681-692.

[14] OUZA, Fábio; NOGUEIRA, Rodrigo; LOTUFO, Roberto. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: Brazilian conference on intelligent systems. Cham: Springer International Publishing, 2020. p. 403-417.

[15] ANG, Liang et al. Multilingual e5 text *embeddings*: A technical report. arXiv preprint arXiv:2402.05672, 2024.

[16] INEGA, Gesare Asnath; MWANGI, Waweru; RIMIRU, Richard. Text mining in digital libraries using okapi BM25 model. International Journal of Computer Applications Technology and Research, v. 7, n. 10, p. 398-406, 2018.