Large Language Model como Suporte à Tarefa de *Entity Matching*

Rodolfo Bolconte Donato[1], **Tiago Brasileiro Araújo**[1],
Carlos Eduardo S. Pires[2], **Demetrio Gomes Mestre**[2]

[1] Instituto Federal da Paraíba *campus* João Pessoa (IFPB)
João Pessoa, Paraíba, Brasil

[2]Universidade Federal de Campina Grande (UFCG)
Campina Grande, Paraíba, Brasil

rodolfo.donato@ifpb.edu.br, tiago.brasileiro@ifpb.edu.br

cesp@dsc.ufcg.edu.br, demetrio@computacao.ufcg.edu.br

Abstract. *Fundamental to data integration, Entity Matching aims to identify records that reference the same real-world entity. Pre-trained models dominate the state of the art but face limitations due to the need for large volumes of labeled data, whereas Large Language Models (LLMs) demonstrate superior generalization capabilities without requiring such volumes, despite their high computational cost. In this scenario, the present work proposes a hybrid approach to balance the best of both model types, performing new classifications with LLMs only on samples where the pre-trained model shows low confidence, resulting in up to an 8% increase in F1-Score.*

Resumo. *Fundamental para a integração de dados, a tarefa de Entity Matching visa identificar registros que referenciam a mesma entidade no mundo real. Modelos pré-treinados dominam o estado da arte, mas enfrentam limitações quanto à necessidade de grandes volumes de dados rotulados, enquanto que Large Language Models (LLMs) demonstram capacidade superior de generalização, sem a necessidade de grandes volumes, apesar do alto custo computacional. Diante deste cenário, o presente trabalho propõe uma abordagem híbrida para equilibrar o melhor dos dois mundos de modelos, realizando novas classificações com os LLMs apenas em amostras em que o modelo pré-treinado apresente baixa confiança, gerando um aumento de até 8% de F1-Score nos resultados.*

1. Introdução

A integração de dados é um desafio central em diversos domínios, pois informações sobre o mesmo objeto ou entidade costumam estar dispersas em múltiplas fontes e formatos [Araújo et al. 2020]. Nesse contexto, surge a necessidade de técnicas capazes de identificar e reconciliar essas descrições; logo, é apresentada a *Entity Matching (EM)*, ou Resolução de Entidades, sendo a tarefa de identificar descrições de objetos/entidades em diferentes fontes que referenciam a mesma entidade no mundo real [Christophides et al. 2019]. Essencial para a integração de dados, a realização de *EM* enfrenta desafios relacionados ao volume e à variedade dos dados, como a dispersão das informações em formatos estruturados ou não. Tradicionalmente, sua realização evoluiu

de regras manuais para métodos de Aprendizagem de Máquina Supervisionada, culminando no uso extensivo de modelos pré-treinados que requerem grandes volumes de dados para treino, como os modelos *BERT* e *RoBERTa*, que se tornaram a base dos sistemas modernos para a realização de *EM* [Peeters et al. 2024, Araújo et al. 2025a]. Paralelamente à consolidação dos modelos pré-treinados, a ascensão dos *Large Language Models (LLMs)*, como *ChatGPT*, *Copilot* e *Gemini*, introduziu novas possibilidades devido à sua capacidade de generalização e raciocínio semântico avançado [Wei et al. 2022].

Embora os *LLMs* demonstrem um desempenho superior ao lidar com informações não vistas anteriormente, sua utilização em larga escala impõe desafios consideráveis, como o alto custo computacional e a latência associados ao processamento de grandes volumes de dados [Brown et al. 2020, Zhao et al. 2025]. Já para os modelos pré-treinados, a principal desvantagem reside na dependência de grandes quantidades de dados do contexto para o treinamento, de modo que, sem um volume suficiente, os modelos podem apresentar quedas de desempenho quando confrontados com dados nunca antes vistos [Peeters et al. 2024]. Essa limitação contrasta com a capacidade dos *LLMs* obterem resultados satisfatórios ao interpretar sutilezas com pouca ou nenhuma quantidade de exemplos relacionados ao contexto, o que poderia mitigar erros comuns em casos complexos [Akbarian Rastaghi et al. 2022, Peeters et al. 2023].

Diante deste cenário, o presente trabalho propõe uma abordagem híbrida que visa combinar a eficiência dos modelos pré-treinados com a capacidade dos *LLMs* em trabalhar com nenhuma ou poucas informações do contexto específico. A solução consiste na utilização de *LLMs* para a realização de *EM* sobre amostras em que modelos pré-treinados possuem certa dúvida em identificar o que de fato representa uma única entidade do mundo real ou não. Ao direcionar apenas uma fração das amostras mais desafiadoras para os *LLMs*, busca-se superar a necessidade excessiva de dados rotulados e a falta de robustez dos modelos pré-treinados em casos que gerem dúvida, com foco em um custo computacional não tão alto enquanto se maximiza a precisão final da tarefa de *EM*.

2. *Background*

Nesta seção, são discutidos conceitos e abordagens que fundamentam o desenvolvimento do trabalho, situando o leitor no contexto em que a pesquisa se insere. O foco recai sobre dois temas principais: a atividade de *Entity Matching* e o papel dos Modelos de Aprendizagem Pré-Treinados e *Large Language Models*.

2.1. *Entity Matching*

Sendo a tarefa de identificação de entidades referentes ao mesmo objeto do mundo real, a partir da mesma ou de diferentes fontes de dados [Mestre et al. 2017], a *Entity Matching* se assemelha a um problema de classificação binária, com o resultado dentro de dois valores únicos: ***match***, quando dois ou mais pares de informações de objetos representam a mesma entidade real; e ***non-match***, sendo o caso contrário. Formalmente, tem-se um conjunto de dados $A = \{a_1, a_2, ..., a_n\}$, onde cada $a_i = \{r_{i1}, r_{i2}, ..., r_{in}\}$ contém descrições de objetos, em que *EM* é a realização de comparações similares a um produto cartesiano (gerando pares de cada amostra de uma fonte, com cada amostra das demais fontes de dados): $EM = A \times B = \{(a_i, b_j) \mid a_i \in A,\ b_j \in B\}$, a fim de descobrir o que é *match* e *non-match*. Por exemplo, assumindo um cenário onde se tem as entidades "Lucas Henri-

que da Costa", "Lucas Henrique Costa" e "Lucas H da Costa", tal processo deve ser capaz de sinalizar tais informações como potencialmente referentes à mesma pessoa.

Ademais, por conta da semelhança de *EM* com um problema de classificação binária, é possível a aplicação de métricas da área para o entendimento do desempenho dos modelos utilizados para a tarefa, como: *Precision*, o índice de pares de entidades corretamente classificados como *match* em relação ao total de pares preditos como *match*; *Recall*, o percentual de pares de entidades corretamente classificados como *match* ante o total de pares de entidades que são de fato *match* no mundo real; e *F1-Score*, a média harmônica entre *Precision* e *Recall*, em que quanto mais alto seu valor, os valores das duas outras métricas também serão altos de forma equilibrada [Mestre et al. 2017].

2.2. Modelos de Aprendizagem

Pesquisas recentes sobre *EM* optam pela utilização de modelos de linguagem pré-treinados para a classificação dos pares de entidades, cujo núcleo é a similaridade semântica das informações fornecidas [Zeakis et al. 2025]. Tais modelos tendem a operar da seguinte forma: um processo de vetorização transforma cada entidade em um vetor de *embeddings*. Em seguida, os vetores são indexados e uma consulta aos vizinhos mais próximos para cada entidade é realizada a fim de determinar os pares candidatos a *match* de acordo com suas distâncias [Araújo et al. 2025a]. Logo, tal distância é considerada como o grau de similaridade entre as entidades que, apesar de ser um valor contínuo, por meio de um *threshold* é possível definir valores binários para um resultado final do modelo [Ebraheem et al. 2018, Papadakis et al. 2023], indicando assim quais amostras são *match* e *non-match*.

2.3. *Large Language Models*

Capazes de compreender e gerar linguagem humana, os *Large Language Models* conseguem prever a probabilidade de sequências de palavras ou gerar novos textos com base nas entradas recebidas, permitindo a paralelização e a captura de dependências de longo alcance no texto [Chang et al. 2024]. Sua principal característica é o aprendizado contextual, em que o modelo é treinado para gerar texto a partir de um contexto ou estímulo fornecido, resultando em respostas mais coerentes e contextualmente relevantes [Brown et al. 2020]. Os humanos podem ainda participar de interações de perguntas e respostas ou realizar interações de diálogo com os modelos, mantendo conversas em linguagem natural, semelhantes às conversas entre humanos. Ao abrangerem estas capacidades, os *LLMs* se mostraram importantes no processamento de linguagem natural, sendo promissores em diversas aplicações [Chang et al. 2024], incluindo a tarefa de *Entity Matching*.

3. Trabalhos Relacionados

A literatura referente à utilização de *Large Language Models* sobre resultados obtidos por modelos pré-treinados ao realizar *EM* é escassa, concentrando-se em estudos comparativos dessas diferentes abordagens [Christophides et al. 2019, Araújo et al. 2025b]. Sendo assim, é importante a apresentação de tais trabalhos que mostrem de fato a realização de *EM* mediante o uso de *LLMs*, como também de pesquisas que tenham como foco a diminuição do custo computacional na execução destes modelos, tendo em evidência também o bom desempenho classificatório dos mesmos na tarefa.

Duas das principais referências sobre a realização de *Entity Matching* utilizando *LLMs* são os trabalhos propostos em [Peeters et al. 2024] e [Xia et al. 2024]. Ambos adotam ideias semelhantes de execução de diferentes modelos (*LLMs* e pré-treinados) para diferentes conjuntos de dados, desenvolvendo técnicas de ajuste de *prompts* a fim de alcançar os melhores resultados possíveis. Os *LLMs* se destacam em ambos os trabalhos, atingindo os maiores resultados de classificação, seguidos pelos modelos pré-treinados, porém, são evidenciados alguns problemas relacionados aos *LLMs*, como o tempo e custo elevados para realizar *EM* nas bases inteiras de dados. Outro fator que impacta a utilização somente dos *LLMs* na tarefa de *EM* é que nem sempre estes modelos obterão os melhores resultados, como evidenciado na pesquisa realizada em [Donato and Araújo 2025], com um modelo pré-treinado sendo até 17% melhor que um *LLM*. Tais fatores então possibilitam uma preferência pelos modelos pré-treinados que realizam a mesma atividade em menor tempo e custo computacional, apesar do desempenho classificatório abaixo em certos casos.

Alinhada com a ideia de equilíbrio de qualidade e custo do presente trabalho, a pesquisa realizada em [Zhang et al. 2025] demonstra que *Small Language Models*, modelos com arquitetura mais leve e número menor de parâmetros, podem rivalizar com *LLMs* quando adotadas abordagens centradas nos dados. Porém, estes tipos de modelos acabam apresentando uma variância maior nos resultados, diminuindo o desempenho classificatório e, consequentemente, tendo um equilíbrio menor entre qualidade e custo. Por fim, alinhado à ideia de suporte a modelos pré-treinados do presente trabalho, tem-se o trabalho realizado em [Maciejewski et al. 2025], que, apesar de não utilizar *LLMs*, realiza diversas etapas de organização dos dados para a execução dos modelos em ambientes com restrições computacionais, o que pode tornar a atividade de *EM* mais complexa.

Dado o contexto apresentado, a partir das amostras que geram dúvidas em modelos pré-treinados sendo utilizadas nos *LLMs*, o presente trabalho se destaca dos demais na área de *Entity Matching* como uma abordagem que foca no melhor equilíbrio possível entre classificação e custo computacional, beneficiando-se do melhor dos dois mundos: o baixo custo computacional de modelos pré-treinados em relação aos *LLMs* com o alto desempenho classificatório destes.

4. Suporte à *Entity Matching* com *Large Language Model*

O presente trabalho tem como proposta a utilização de *Large Language Models* como suporte a modelos pré-treinados que executam a atividade de *Entity Matching*, por meio da realização de uma classificação final em amostras que gerem dúvidas nos modelos pré-treinados. O objetivo com a proposta é a obtenção de melhores resultados de desempenho classificatório e computacional para as execuções.

Na Figura 1, é mostrado o fluxograma de execução da proposta dividido em quatro etapas: **1) Conjunto de Dados**: com a divisão do conjunto em três subconjuntos (treino, validação e teste); **2) Modelo Pré-Treinado**: contemplando o treino, validação e teste de modelo, a criação de Janelas de Incerteza para a divisão das amostras que geram dúvida ou não em serem *match* e *non-match* para o modelo; **3) *Large Language Model***: com a execução de nova classificação das amostras duvidosas por meio de diferentes *prompts*, seguido da união das novas classificações com as classificações das amostras não duvidosas; e **4) Análise de Resultados**: calculando as métricas de tempo total e por amostra

para os dois tipos de modelos, como também as métricas de classificação para avaliação final das amostras, estas calculadas a partir dos resultados do modelo pré-treinado (com amostras não duvidosas) combinados com os resultados do *LLM* (com amostras duvidosas).

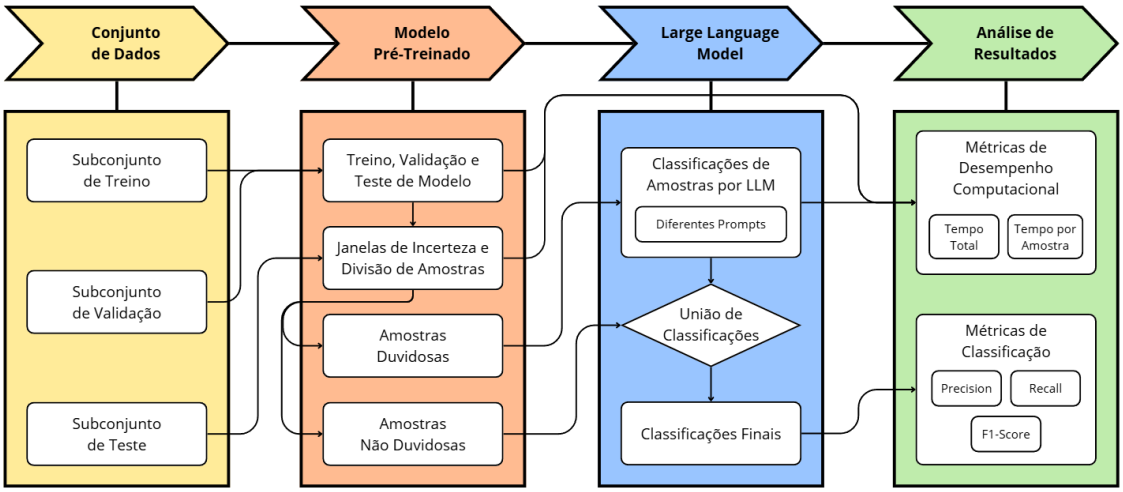

Figura 1. Fluxograma de execução da presente proposta.

4.1. Modelo de Aprendizagem

Para a primeira etapa de realização de *Entity Matching*, foi escolhido o *Ditto*, uma ferramenta baseada em modelos de linguagem pré-treinados como o *RoBERTa*, que adota a utilização de *embeddings* para a representação textual através de vetores numéricos. O *Ditto* possui três técnicas de otimização: 1) inserir conhecimento do domínio ao destacar informações importantes de entrada que podem ser de interesse ao tomar decisões; 2) resumir cadeias que são muito longas para que apenas a informação essencial seja retida e usada no reconhecimento de entidades; e 3) realizar um aumento de dados para complementar o volume de treinamento com exemplos considerados mais difíceis [Li et al. 2020]. Após as etapas tradicionais de modelos de aprendizagem, sendo treino, validação e teste, o *Ditto* é capaz de definir um valor contínuo de classificação, que, quanto mais próximo ou igual a 0, indica *non-match*, quando as informações não possuem similaridade para representar a mesma entidade real, e, quanto mais próximo ou igual a 1, indica o *match*, com as informações tendo maior similaridade na representação da mesma entidade real.

Para a definição em apenas dois valores finais, de fato, se *non-match* ou *match*, é utilizado um limiar *(threshold)* gerado a partir do cálculo de *F1-Score* durante a etapa de validação do treino que, por exemplo, se o limiar for igual a 0,6, amostras com valores abaixo deste recebem o valor final 0, indicando o *non-match*, e amostras com valor igual ou acima do limiar recebem o valor final 1, indicando o *match*.

4.2. Janelas de Incerteza

Como apresentado na última Seção, o modelo pré-treinado gera como resultado valores contínuos de 0 a 1, em que quanto mais próximo de 0 indica que o par de amostras passadas é um *non-match*, e quanto mais próximo de 1, indica o *match* de fato. O resultado

binário do modelo se dá através de um limiar de decisão que é obtido através da etapa de validação, a partir do *F1-Score* do mesmo para com as amostras de validação, logo, se o resultado contínuo está abaixo do limiar, o valor é convertido para 0 (*non-match*), e se estiver acima, é convertido para 1 (*match*).

Contextualizados os tipos de resultados do modelo pré-treinado, é possível obter a distribuição dos valores contínuos, e para uma melhor compreensão, agrupá-los a cada 0,05 ou 0,1 em um histograma, por exemplo. Logo, a partir da distribuição dos resultados obtidos, pode-se agrupar as amostras dentro de uma faixa específica de valores contínuos.

Visto que o foco do trabalho é obter o melhor equilíbrio entre classificação e custo na execução dos modelos, não é viável a execução de *LLMs* utilizando todo o subconjunto de testes, dadas as suas problemáticas em relação ao custo computacional na execução de grandes quantidades de amostras. E é a partir deste pensamento que surge o conceito de janela empregado aqui, que é a seleção de não todas, mas de parte das amostras já executadas pelo modelo pré-treinado, para que possam ser re-executadas através do *LLM*.

Intituladas Janelas de Incerteza, suas construções se dão a partir do limiar de decisão do modelo, através do entendimento de que quanto mais o valor obtido para a amostra está próximo do limiar, mais a amostra pode gerar dúvida para o modelo quanto a representar um *non-match* ou *match*. E com um valor fixo acrescentado ao limiar (taxa de crescimento), é possível definir uma faixa de valores para a seleção de amostras de acordo com os seus resultados obtidos, se apresentando então como as Janelas de Incerteza. Por exemplo, ao definir uma taxa de crescimento de 0,1 a cada nova iteração, a primeira janela partindo de um limiar 0,6 seria [0,5 - 0,7], aumentando e diminuindo 0,1 do limiar, com as janelas seguintes sendo [0,4 - 0,8], [0,3 - 0,9] e assim por diante.

É importante salientar, porém, que nem sempre o ponto de partida das Janelas de Incerteza será o limiar de decisão do modelo, uma vez que tal valor pode se apresentar próximo de 0 ou 1, impossibilitando um crescimento simétrico da janela para ambos os lados. Logo, para que a janela possa crescer de forma simétrica gradualmente, o ponto de partida das janelas é direcionado para a média entre os valores 0 e 1, como sendo 0,5.

4.2.1. Escolha de *Large Language Model*

Foi realizado preliminarmente um teste com dezesseis *LLMs*, sendo escolhidas versões com quantidades razoáveis de parâmetros que não exigem tanto poder computacional: *deepseek-r1:7b, dolphin3:8b, falcon3:7b, gemma:7b, gemma3:12b, llama3.1:8b, llava:7b, mistral-openorca:7b, mistral:7b, olmo2:7b, openhermes:v2.5, orca-mini:7b, phi3:3.8b, qwen3:8b, wizardlm2:7b* e *zephyr:7b*. A ideia destas execuções é a escolha do melhor modelo que apresente bom equilíbrio entre desempenho classificatório e tempo de execução, para que o modelo final escolhido possa receber as amostras das Janelas de Incerteza geradas pelo modelo pré-treinado.

Para a escolha do *LLM* e para a execução final, utilizando amostras das Janelas de Incerteza, foram idealizados quatro *prompts* para a execução dos *LLMs*, com o objetivo de classificar as amostras recebidas como *match* ou *non-match*, construídos a partir de diferentes técnicas de Engenharia de *Prompt*. Três *prompts* foram construídos com base em técnicas de *persona, few-shots* e *output constraint*, variando apenas a quanti-

dade de exemplos rotulados (*few-shots*): *Prompt 1* com apenas um exemplo para cada classificação, *match* e *non-match*; *Prompt 2* com dois exemplos de cada; e *Prompt 3* com cinco exemplos de cada classificação. Já o *Prompt 4*, além das técnicas já citadas, possui também a combinação das técnicas *self-consistency* e *chain-of-thought*, em que é solicitada ao modelo a geração de três possíveis resultados e, em seguida, a definição de um resultado final com base nas opções anteriores alcançadas. Para todos os *prompts*, especifica-se que a saída deve ser unicamente 1 (*match*) ou 0 (*non-match*).

Após a execução dos *LLMs* com subconjuntos de testes voltados para a atividade de *EM*, são calculados os tempos de execução (totais e por amostra), bem como as métricas de classificação dos resultados gerados pelos modelos. O modelo mais rápido em executar as amostras de teste dos conjuntos foi o *phi3:3.8b*, levando 6 minutos e 36 segundos para classificar 1916 amostras, porém obteve um baixo desempenho classificatório, atingindo apenas 48,05% de *F1-Score* utilizando o *Prompt 3* para o mesmo conjunto de dados. Em se tratando de desempenho classificatório, o *zephyr:7b* obteve os melhores resultados para os conjuntos, obtendo *F1-Score* de 88,40% utilizando o *Prompt 2* em determinada execução; porém, em relação ao tempo de processamento das amostras, consumiu quase o dobro do tempo do modelo mais rápido, levando cerca de 11 minutos e 50 segundos.

Com tais informações obtidas, optou-se por seguir com o *zephyr:7b* para o recebimento das amostras selecionadas pelas Janelas de Incerteza que, apesar de um maior tempo de processamento geral, conseguiu atingir o maior valor de *F1-Score* no menor tempo entre os cinco melhores modelos no desempenho classificatório.

5. Execução da Proposta e Resultados

5.1. Configurações

O *Ditto* (modelo pré-treinado) foi executado com os valores padrão definidos por seus autores, `64` de comprimento máximo, `32` de tamanho do *batch* e `3e-5` de taxa de aprendizagem. Para o *Zephyr:7b* (*LLM*), são utilizados *prompts* determinísticos e restritos (apresentados no repositório do trabalho[1]), visando a garantia de resultados reprodutíveis, e a temperatura do modelo é definida como `0` para que a resposta seja a mais direta possível, sendo executado utilizando o *framework Ollama*.

Conforme apresentado na Tabela 1, são utilizados três conjuntos de informações sobre produtos de sites de venda, comumente utilizados em trabalhos relacionados a *EM*. A obtenção de tais conjuntos se deu por meio de um repositório[2] com os mesmos já estruturados e separados em subconjuntos de treino, validação e teste.

A máquina utilizada nos experimentos possui a seguinte configuração: GPU NVIDIA GeForce RTX 3060 6GB, CPU AMD Ryzen 7 6800H e RAM 16GB DDR5.

5.2. Conjunto de Dados *Abt-Buy*

A Tabela 2 contém os resultados obtidos para os dois modelos utilizando o *Abt-Buy*. A etapa de treino do *Ditto* é realizada em cerca de 39 minutos e 52 segundos para um total de 7.659 amostras (subconjuntos de treino e teste), resultando em um processamento

[1]Repositório do trabalho no GitHub: https://github.com/rodolfobolconte/SBBD26-paper-em

[2]Repositório do *Ditto* com os conjuntos utilizados: https://github.com/megagonlabs/ditto

Tabela 1. Quantidades e Atributos dos Conjuntos de Dados, sendo m=match e nm=non-match

Conjunto de Dados	Treino	Validação	Teste	Atributos
Abt-Buy	5743 (616 m - 5127 nm)	1916 (206 m - 1710 nm)	1916 (206 m - 1710 nm)	Nome, Preço e Descrição
Amazon-Google	6874 (699 m - 6175 nm)	2293 (234 m - 2059 nm)	2293 (234 m - 2059 nm)	Nome, Preço e Fabricante
Walmart-Amazon	6144 (576 m - 5568 nm)	2049 (193 m - 1856 nm)	2049 (193 m - 1856 nm)	Nome, Preço, Categoria, Marca e Modelo

de 0,4166 segundos por amostra, com o modelo possuindo um limiar de 0,6, ou seja, amostras com classificação abaixo ou igual a este valor são do tipo *non-match* e acima deste são do tipo *match*. Já a classificação de fato do modelo utilizando o subconjunto de teste atingiu 83,39% de *F1-Score* para classificação geral, com desempenho maior para as amostras do tipo *non-match*, cerca de 96,33% e desempenho menor para as amostras do tipo *match*, atingindo 70,45%. O maior valor métrico foi 96,65%, atingido em *Precision* para as amostras *non-match*. A fase de teste foi executada em apenas 40,6 segundos, cerca de 0,0212 segundos por amostra (1.916 ao todo). Ainda com relação ao *Ditto*, a Figura 2 contém a distribuição das classificações obtidas pelo modelo, sendo possível visualizar uma concentração de amostras próximas ao limiar e, para a realização da execução do *zephyr:7b*, são realizadas cinco seleções de amostras por Janelas de Incerteza aumentadas a cada 0,05 como taxa de crescimento, sendo elas: [0,55 - 0,65], [0,50 - 0,70], [0,45 - 0,75], [0,40 - 0,80] e [0,35 - 0,85].

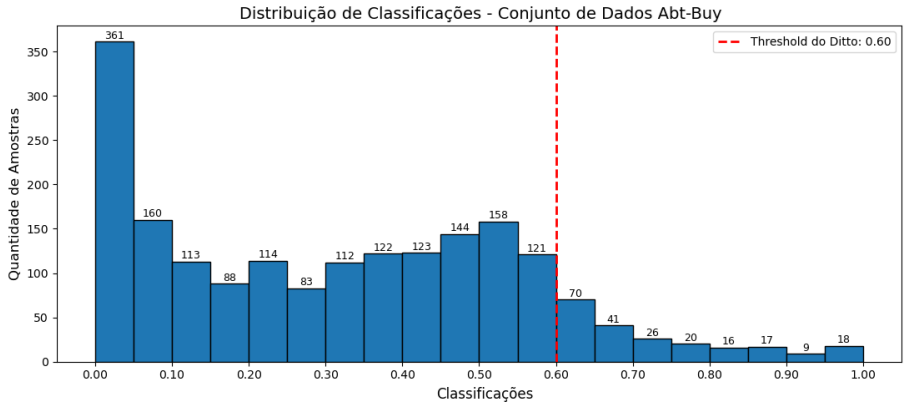

Figura 2. Distribuição dos resultados do *Ditto* para o *Abt-Buy*.

Ao executar o *zephyr:7b* com as amostras das Janelas de Incerteza, observa-se que o aumento das métricas de classificação é gradual a cada acréscimo de amostras. Na primeira janela [0,55 - 0,65], em que o modelo é executado com 191 amostras, a classificação final resultou em 87,73% de *F1-Score* geral, 97,57% para as amostras *non-match* e 77,89% para as amostras *match*, levando cerca de 1 minuto e 31 segundos para processar as amostras. A quarta janela ([0,40 - 0,80]) obteve os maiores resultados de classificação: com a execução de cerca de 703 amostras, a classificação final atingiu um *F1-Score* geral de 91,15%, evidenciando um aumento de mais de 7% em relação à execução do *Ditto* com o subconjunto de teste. Na quinta e última janela, [0,35 - 0,85], o

comportamento foi inverso; os valores métricos de classificação diminuíram.

Tabela 2. Resultados das execuções do *Ditto* e *Zephyr:7b* para o *Abt-Buy*.

Métricas			Ditto	Zephyr:7b - Janelas de Incerteza				
				[0,55 - 0,65] 191 Amostras	[0,50 - 0,70] 390 Amostras	[0,45 - 0,75] 560 Amostras	[0,40 - 0,80] 703 Amostras	[0,35 - 0,85] 841 Amostras
Precision (%)		Non-Match	96,65	96,67	96,96	97,14	**97,25**	97,20
		Match	68,66	85,06	90,00	92,86	92,94	**93,45**
		Geral	82,65	90,86	93,48	95,00	95,10	**95,32**
Recall (%)		Non-Match	96,02	98,48	99,01	99,30	99,30	**99,36**
		Match	72,33	71,84	74,27	75,73	**76,70**	76,21
		Geral	84,18	85,16	86,64	87,51	**88,00**	87,79
F1-Score (%)		Non-Match	96,33	97,57	97,97	98,21	**98,26**	**98,26**
		Match	70,45	77,89	81,38	83,42	**84,04**	83,96
		Geral	83,39	87,73	89,68	90,81	**91,15**	91,11
Tempo de Execução	Treino	Total (min)	39:52,7	-	-	-	-	-
		Por Amostra (seg)	0,4166	-	-	-	-	-
	Teste	Total (min)	**00:40,6**	01:31,5	01:23,4	01:57,7	02:25,5	02:53,8
		Por Amostra (seg)	**0,0212**	0,4791	0,2139	0,2101	0,2069	0,2066

5.3. Conjunto de Dados *Amazon-Google*

Como mostrado na Tabela 3, a etapa de treino do *Ditto* com o *Amazon-Google* precisou de 27 minutos e 25 segundos para processar as 9.167 amostras no total, equivalente a 0,2394 segundos para cada uma delas. Com relação ao limiar de classificação do modelo, o valor resultante foi 0,15, próximo ao valor definitivo que indica *non-match*, 0. Em relação às classificações utilizando o subconjunto de teste, o modelo obteve *F1-Score* geral de 86,24%, sendo 96,90% para as amostras *non-match* e 75,58% para as amostras *match*, além de 98,06% de *Precision* para as amostras *non-match*, o maior valor dentre os resultados para os três conjuntos de dados. Os tempos de execução dos dados de teste se mantiveram na mesma faixa observada pela instância do *Abt-Buy*, sendo executados em aproximadamente 12,92 segundos para as 2.293 amostras, resultando em 0,0056 segundos para cada uma delas. Na Figura 3, é possível visualizar uma concentração de amostras com classificações entre [0 - 0,05] e [0,95 - 1]. Assim como mencionado na Seção 4.2, nem sempre o ponto de partida das Janelas de Incerteza será o limiar do modelo; o valor inicial foi redirecionado para o meio do gráfico, começando na Janela [0,40 - 0,60] e finalizando em [0,10 - 0,90], com uma taxa de crescimento de 0,1 acima e abaixo, resultando em quatro janelas.

Ao contrário do *Abt-Buy*, a execução do *zephyr:7b* com as amostras do *Amazon-Google* gerou valores menores de *F1-Score* geral e para as amostras *match*. Somente para as amostras *non-match* a execução em conjunto com o *LLM* gerou valores maiores, sendo o maior *F1-Score* atingido na quarta execução, com a janela [0,10 - 0,90], atingindo 97,05%, e apenas 0,15 ponto percentual maior que a execução do *Ditto*. Além disso, na execução da quarta janela de incerteza, foi obtido o segundo maior tempo de execução, de cerca de 58 segundos. Outra informação importante para esta execução é que a primeira janela de incerteza, que possui a menor quantidade de amostras (36), apresentou o maior tempo de execução do *zephyr:7b*, cerca de 1 minuto e 8 segundos no processamento.

5.4. Conjunto de Dados *Walmart-Amazon*

Por fim, a instância do *Ditto* construída para o *Walmart-Amazon* levou cerca de 29 minutos e 12 segundos para ser finalizada, em torno de 0,2852 segundos para cada uma das

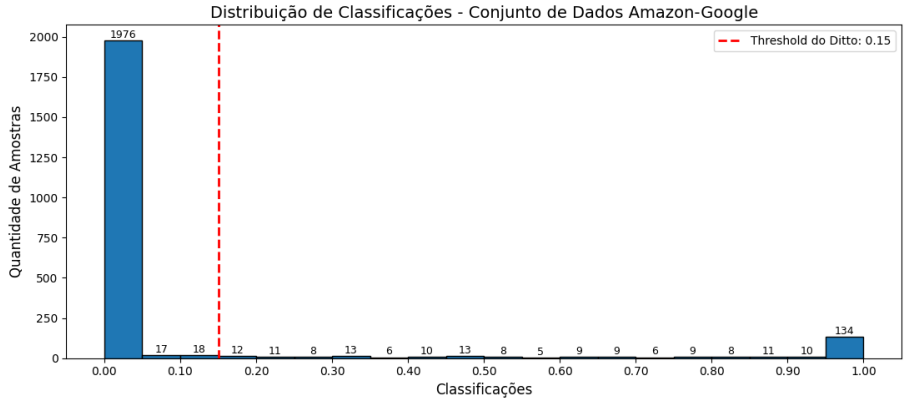

Figura 3. Distribuição dos resultados do *Ditto* para o *Amazon-Google*.

Tabela 3. Resultados das execuções do *Ditto* e *Zephyr:7b* para o *Amazon-Google*.

Métricas			Ditto	Zephyr:7b - Janelas de Incerteza			
				[0,40 - 0,60] 36 Amostras	[0,30 - 0,70] 73 Amostras	[0,20 - 0,80] 107 Amostras	[0,10 - 0,90] 156 Amostras
Precision (%)		Non-Match	**98,06**	97,45	96,66	96,26	95,96
		Match	69,15	71,37	73,01	75,73	**79,68**
		Geral	83,60	84,41	84,84	86,00	**87,82**
Recall (%)		Non-Match	95,77	96,45	97,04	97,57	**98,15**
		Match	**83,33**	77,78	70,51	66,67	63,68
		Geral	**89,55**	87,12	83,78	82,12	80,91
F1-Score (%)		Non-Match	96,90	96,95	96,85	96,91	**97,05**
		Match	**75,58**	74,44	71,74	70,91	70,78
		Geral	**86,24**	85,69	84,29	83,91	83,92
Tempo de Execução	Treino	Total (min)	27:26,0	-	-	-	-
		Por Amostra (seg)	0,2394	-	-	-	-
	Teste	Total (min)	**00:13,0**	01:08,6	00:29,9	00:39,8	00:58,8
		Por Amostra (seg)	**0,0056**	1,9047	0,4099	0,3725	0,3770

8.193 amostras dos subconjuntos de treino e validação, com um limiar de 0,9 para as classificações, similar ao conjunto *Amazon-Google* no quesito do valor estar próximo a uma extremidade de classificação final, neste caso, o valor 1, que indica o *match*. Quanto às classificações do subconjunto de teste, o modelo atingiu 83,26% de *F1-Score* geral, com 96,88% para as amostras *non-match* e 69,63% para as amostras *match*, levando cerca de 53 segundos para processamento, sendo 0,0259 por amostra, o maior tempo de processamento por amostra dentre os subconjuntos de teste, como mostrado na Tabela 4. Da mesma forma que o *Amazon-Google*, adotou-se a estratégia de iniciar a construção das janelas de incertezas no intervalo [0,40 - 0,60], seguindo até o intervalo [0,10 - 0,90] a cada 0,1 de taxa de crescimento. Na Figura 4, é possível visualizar a distribuição das classificações do *Ditto* para o *Walmart-Amazon*, com concentrações de amostras próximas a 0 e 1.

Ao executar as amostras com o *zephyr:7b*, foram obtidos resultados positivos, embora abaixo dos valores para o *Abt-Buy*. Os maiores valores obtidos em *Precision* e *F1-Score* foram alcançados na quarta janela de incerteza, [0,10 - 0,90]. Ao executar 716 amostras com o *LLM*, o *F1-Score* geral da classificação final foi de 85,64%, uma melhoria

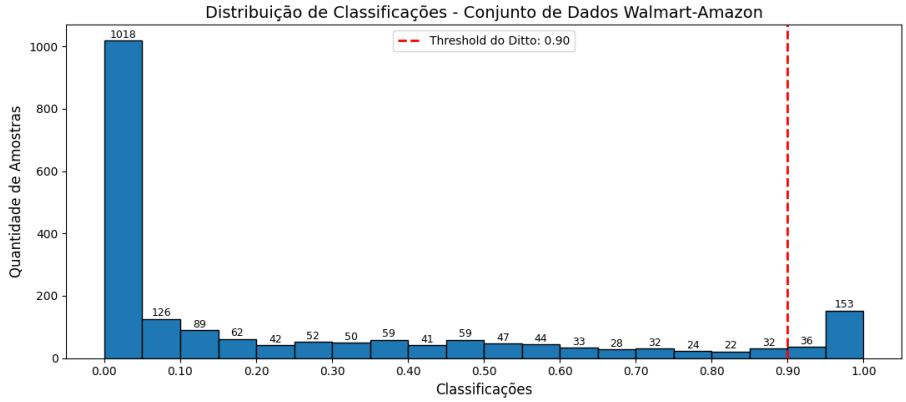

Figura 4. Distribuição dos resultados do *Ditto* para o *Walmart-Amazon*.

de apenas 2,38 pontos percentuais em relação à execução do *Ditto*. A maior melhoria foi obtida nas amostras do tipo *match*, com um aumento de 4,52 pontos percentuais na mesma métrica. Em relação ao tempo de execução, o valor se manteve relativamente próximo ao tempo por amostra, entre 0,2202 e 0,2354 segundos em todas as execuções, com a maior janela levando 2 minutos e 48 segundos para processar as amostras.

Tabela 4. Resultados das execuções do Ditto e Zephyr:7b para o Walmart-Amazon.

Métricas			Ditto	Zephyr:7b - Janelas de Incerteza			
				[0,40 - 0,60] 191 Amostras	[0,30 - 0,70] 361 Amostras	[0,20 - 0,80] 511 Amostras	[0,10 - 0,90] 716 Amostras
Precision (%)		Non-Match	96,77	96,97	97,24	97,45	**97,76**
		Match	**70,37**	69,19	69,61	69,86	70,05
		Geral	83,57	83,08	83,42	83,65	**83,90**
Recall (%)		Non-Match	**96,98**	96,71	96,66	96,61	96,50
		Match	68,91	70,98	73,58	75,65	**78,76**
		Geral	82,95	83,85	85,12	86,13	**87,63**
F1-Score (%)		Non-Match	96,88	96,84	96,95	97,02	**97,13**
		Match	69,63	70,08	71,54	72,64	**74,15**
		Geral	83,26	83,46	84,24	84,83	**85,64**
Tempo de Execução	Treino	Total (min)	29:13,0	-	-	-	-
		Por Amostra (seg)	0,2852	-	-	-	-
	Teste	Total (min)	00:53,0	**00:42,5**	01:19,5	01:58,1	02:48,6
		Por Amostra (seg)	**0,0259**	0,2223	0,2202	0,2310	0,2354

Em suma, após a apresentação dos resultados tanto para o *Ditto* quanto para a utilização do *zephyr:7b* como suporte, é perceptível a melhoria dos índices da classificação final em dois dos três conjuntos de dados. Para o *Abt-Buy*, foi possível observar a maior melhoria da solução proposta, com um aumento de *F1-Score* em quase 8 pontos percentuais em relação ao resultado do *Ditto*. Para o *Walmart-Amazon*, apesar de não atingir uma diferença próxima, a melhoria de fato é comprovada, com um aumento de 2,38 pontos percentuais para o *F1-Score* geral. Por fim, embora a execução do *zephyr:7b* não tenha melhorado os valores de *F1-Score* obtidos pelo *Ditto* para o *Amazon-Google*, é notável a melhoria em *Precision* para as amostras *match*, sendo um aumento de 10,53 pontos percentuais, evidenciando uma diminuição na quantidade de amostras classificadas como *match* quando sendo *non-match*, após a utilização do *zephyr:7b* em conjunto

com o *Ditto*. Com relação aos tempos de execução dos modelos, apesar de o *LLM* levar mais tempo para processar os subconjuntos de testes em relação ao modelo pré-treinado, quando se é levado em consideração o tempo de treinamento deste último, a adoção do *LLM* se torna viável em equilíbrio com menos quantidade de dados.

6. Conclusões e Trabalhos Futuros

O presente trabalho investigou a viabilidade de uma abordagem híbrida para a tarefa de *Entity Matching*, integrando a eficiência de um modelo de aprendizagem pré-treinado com a capacidade de raciocínio generalista de um *Large Language Model*. E para que fosse possível a utilização de diferentes modelos em conjunto, foi idealizada a construção das Janelas de Incerteza, com o objetivo de selecionar aquelas amostras que mais possam gerar dúvidas (incertezas) ao modelo pré-treinado, para então serem re-executadas pelo *LLM* escolhido, visando melhorias em desempenho classificatório e computacional das execuções.

Os experimentos realizados trazem à tona diferentes níveis de resultados. Mostrando que a proposta é viável para a realização de *Entity Matching*, tem-se os resultados obtidos para os conjuntos *Abt-Buy*, com um aumento de quase 8 pontos percentuais (p.p.) em *F1-Score* geral, de 83,39% somente com o *Ditto*, para 91,11% com a execução do *Zephyr:7b*, e para o *Walmart-Amazon*, com um aumento de 2,38 p.p. no *F1-Score* geral, indo de 83,46% para 85,64%. Porém o cenário é invertido para o conjunto *Amazon-Google*, com o *Ditto* se saindo melhor em relação à utilização do *Zephyr:7b*. E com relação aos tempos de execução, o *Ditto* possui os melhores tempos ao comparar a utilização somente dos subconjuntos de teste, porém a etapa de treino apresenta valores altos, com execuções em torno de 40 minutos.

Como trabalho futuro, pretende-se a investigação de novas implementações de janelas, não se limitando a uma taxa de crescimento simétrica (acima e abaixo do limiar de decisão do modelo), como a adoção de uma janela deslizante de tamanho variável. Outra investigação inclui avaliar novos modelos pré-treinados de *Entity Matching*, visto que somente o *Ditto* foi utilizado para a realização do trabalho.

7. Agradecimentos

Os autores agradecem ao Instituto Federal da Paraíba *campi* João Pessoa, Soledade e Esperança por proporcionar apoio estrutural e financeiro à realização da pesquisa.

Nota sobre o uso de IA

Durante a produção da parte escrita deste artigo, o texto foi revisado linguisticamente com o auxílio do *Copilot* para a realização de correções manuais. Todo o conteúdo, resultados e conclusões foram integralmente desenvolvidos, verificados e validados pelos autores..

Referências

Akbarian Rastaghi, M., Kamalloo, E., and Rafiei, D. (2022). Probing the robustness of pre-trained language models for entity matching. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 3786–3790, New York, NY, USA. Association for Computing Machinery.

Araújo, T. B., Efthymiou, V., Christophides, V., Pitoura, E., and Stefanidis, K. (2025a). Treats: Fairness-aware entity resolution over streaming data. *Information Systems*, 129:102506.

Araújo, T. B., Efthymiou, V., and Stefanidis, K. (2025b). Fairness and explanations in entity resolution: An overview. *IEEE Access*.

Araújo, T. B., Stefanidis, K., Santos Pires, C. E., Nummenmaa, J., and Da Nóbrega, T. P. (2020). Schema-agnostic blocking for streaming data. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 412–419.

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

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2024). A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., and Stefanidis, K. (2019). End-to-end entity resolution for big data: A survey. *arXiv preprint arXiv:1905.06397*.

Donato, R. B. and Araújo, T. B. (2025). Entity matching com large language models: estudo comparativo com abordagem de entity blocking. In *Anais do XL Simpósio Brasileiro de Bancos de Dados*, pages 956–962, Porto Alegre, RS, Brasil. SBC.

Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., and Tang, N. (2018). Distributed representations of tuples for entity resolution. *Proc. VLDB Endow.*, 11(11):1454–1467.

Li, Y., Li, J., Suhara, Y., Doan, A., and Tan, W. (2020). Deep entity matching with pre-trained language models. *CoRR*, abs/2004.00584.

Maciejewski, J., Nikoletos, K., Papadakis, G., and Velegrakis, Y. (2025). Progressive entity matching: A design space exploration. *Proc. ACM Manag. Data*, 3(1).

Mestre, D. G., Pires, C. E. S., Nascimento, D. C., de Queiroz, A. R. M., Santos, V. B., and Araujo, T. B. (2017). An efficient spark-based adaptive windowing for entity matching. *Journal of Systems and Software*, 128:1–10.

Papadakis, G., Efthymiou, V., Thanos, E., Hassanzadeh, O., and Christen, P. (2023). An analysis of one-to-one matching algorithms for entity resolution.

Peeters, R., Der, R. C., and Bizer, C. (2023). Wdc products: A multi-dimensional entity matching benchmark.

Peeters, R., Steiner, A., and Bizer, C. (2024). Entity matching using large language models.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022). Emergent abilities of large language models.

Xia, Y., Chen, J., Li, X., and Gao, J. (2024). Aprompt4em: Augmented prompt tuning for generalized entity matching.

Zeakis, A., Papadakis, G., Skoutas, D., and Koubarakis, M. (2025). An in-depth analysis of pre-trained embeddings for entity resolution: An in-depth analysis of pre-trained embeddings for entity resolution: A. zeakis et al. *VLDB Journal International Journal on Very Large Data Bases*, 34(1).

Zhang, Z., Groth, P., Calixto, I., and Schelter, S. (2025). A deep dive into cross-dataset entity matching with large and small language models. In *Advances in Database Technology - EDBT*, number 3 in Advances in Database Technology - EDBT, pages 922–934. OpenProceedings.org.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2025). A survey of large language models.