UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

IDENTIFICAÇÃO DE CORRESPONDÊNCIAS ENTRE PRODUTOS A PARTIR DE DESCRIÇÕES TEXTUAIS CURTAS

ANDRÉ LUIZ FIRMINO ALVES

Campina Grande - Paraíba, dezembro de 2024

ANDRÉ LUIZ FIRMINO ALVES

IDENTIFICAÇÃO DE CORRESPONDÊNCIAS ENTRE PRODUTOS A PARTIR DE DESCRIÇÕES TEXTUAIS CURTAS

Tese submetida à **Programa de Pós-Graduação em Ciência da Computação** da **Universidade Federal de Campina Grande**, como requisito parcial à obtenção do título de **Doutor em Ciência da Computação**.

Orientador:

Prof. Cláudio de Souza Baptista, Ph.D.

Campina Grande - Paraíba, dezembro de 2024

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secpg@computacao.ufcg.edu.br

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

ANDRÉ LUIZ FIRMINO ALVES

IDENTIFICAÇÃO DE CORRESPONDÊNCIAS ENTRE PRODUTOS A PARTIR DE DESCRIÇÕES TEXTUAIS CURTAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Doutor em Ciência da Computação.

Aprovada em: 03/12/2024

Prof. Dr. CLÁUDIO DE SOUZA BAPTISTA, Orientador, UFCG

Prof. Dr. CARLOS EDUARDO SANTOS PIRES, Examinador Interno, UFCG

Prof. Dr. DALTON CÉZANE GOMES VALADARES, Examinador Interno, IFPE

Prof. Dr. GERALDO BRAZ JUNIOR, Examinador Externo, UFMA

Prof. Dr. LUCIANO DE ANDRADE BARBOSA, Examinador Externo, UFPE

Prof. Dr. FABIO GOMES DE ANDRADE, Examinador Externo, IFPB

Documento assinado eletronicamente por **CLAUDIO DE SOUZA BAPTISTA**, **PROFESSOR 3 GRAU**, em 05/12/2024, às 20:02, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Geraldo Braz Junior**, **Usuário Externo**, em 06/12/2024, às 13:58, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **CARLOS EDUARDO SANTOS PIRES**, **PROFESSOR 3 GRAU**, em 06/12/2024, às 14:30, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Luciano de Andrade Barbosa**, **Usuário Externo**, em 06/12/2024, às 14:30, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Fabio Gomes de Andrade**, **Usuário Externo**, em 06/12/2024, às 15:17, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Dalton Cézane Gomes Valadares**, **Usuário Externo**, em 10/12/2024, às 12:30, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **5083384** e o código CRC **A5A5F500**.

A474i Alves, André Luiz Firmino.
Identificação de correspondências entre produtos a partir de descrições textuais curtas / André Luiz Firmino Alves. – Campina Grande, 2024.
145 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2024.
"Orientação: Prof. Dr. Cláudio de Souza Baptista".
Referências.

1. Processamento de Linguagem Natural. 2. Recuperação da Informação. 3. Product Matching. 4. Integração de Dados. 5. Cross-Lingual Learning. I. Baptista, Cláudio de Souza. II. Título.

CDU 004.432.45(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

À minha mãe, Vera Lúcia Alves, exemplo de superação e resiliência, que me ensinou a nunca desistir dos meus sonhos.

AGRADECIMENTOS

À Deus, Senhor absoluto, por Sua infinita bondade e amor, por me sustentar nos momentos em que me faltavam forças para continuar e por me guiar em cada passo desta jornada.

À minha querida esposa, Rívia Diana, meu amor e minha gratidão, por ser meu porto seguro, oferecendo-me apoio, paciência e compreensão ao longo de todos os desafios deste percurso. À minha filha, Luiza, e ao meu filho, Rian, que são minha maior inspiração. Vocês são a razão pela qual persevero e busco ser sempre melhor. Cada conquista é também de vocês, pois o amor e a alegria que trazem à minha vida me impulsionam a seguir em frente.

À minha querida mãe, Vera Lúcia Alves, que, com coragem e amor, dedicou-se inteiramente à educação dos filhos, assumindo sozinha ambos os papéis após a perda precoce de meu pai. Sou profundamente grato por tudo que fez por mim e por ser o alicerce sobre o qual construí meus sonhos. Este trabalho é também uma forma de honrá-la e de reconhecer todo o seu esforço e dedicação. Te amo!

Ao meu pai, Bernardo Firmino do Nascimento (*in memoriam*), homem estudioso, trabalhador e fonte constante de inspiração. Seu sonho era ver-me doutor — não médico, como imaginou, mas batalhei para me tornar doutor em computação. Embora não esteja aqui para testemunhar essa conquista, acredito que estaria orgulhoso do "seu doutor". Seu caráter e exemplo sempre foram meu norte.

Aos meus irmãos, Breno Firmino e Weber Firmino, que sempre torceram pelas minhas conquistas. Mano Dr. Weber, muito obrigado por, em meio a tantas ocupações, ter conseguido um tempo para a revisão final desta tese.

Ao meu orientador, Prof. Cláudio de Souza Baptista, Ph.D., com quem aprendi e me inspirei, tanto como ser humano quanto como um excelente profissional. Um verdadeiro incentivador que, com dedicação exemplar, me orientou desde 2012, quando iniciei minha carreira acadêmica no mestrado, e que, de certa forma, me adotou como um pai. Foi ele quem abriu meus olhos para a importância de seguir em frente e realizar o doutorado. Minha eterna gratidão, professor. Que nossa amizade e respeito mútuo sejam sempre cultivados. Que o Eterno bom Deus lhe conceda sabedoria e graça na formação de novos profissionais e cidadãos.

Aos professores Dr. Luciano Barbosa (UFPE) e Dr. Anselmo Paiva (UFMA), que atuaram como coorientadores em diversas fases da minha pesquisa. Suas contribuições foram fundamentais para o desenvolvimento deste trabalho. Agradeço pelo conhecimento compar-

tilhado, pela paciência em responder minhas dúvidas e pela inspiração que proporcionaram ao longo dessa jornada acadêmica.

Ao Laboratório de Sistemas de Informação (LSI/UFCG) pela oportunidade de desenvolver pesquisas, incluindo a disponibilidade de infraestrutura para a execução dos experimentos desta tese. Agradeço também a todos os pesquisadores do LSI com quem pude compartilhar dúvidas e explorar oportunidades de pesquisa.

Ao IFPB, instituição da qual tenho orgulho de ser professor, agradeço pelo afastamento das atividades docentes regulares em um momento tão crucial, que me permitiu dedicar-me integralmente ao estudo e à pesquisa. Sem essa oportunidade, o desenvolvimento deste trabalho teria sido quase impossível. Estou convicto de que o investimento na educação é o melhor caminho para o progresso de nossa nação.

Aos meus alunos dos projetos de pesquisa (PIBIC-EM), especialmente ao Clécio Bruno, pela participação em experimentos que constituíram etapas iniciais desta pesquisa de doutorado. A colaboração de vocês foi fundamental para o sucesso deste trabalho.

Aos professores e funcionários do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande (PPGCC/UFCG) por contribuírem para a minha formação. Quero expressar uma menção especial à funcionária Paloma, cuja gentileza e apoio foram fundamentais na resolução de algumas intercorrências na fase final deste doutorado.

À CAPES pelo recebimento das bolsas durante um período de 5 meses na fase final do doutorado. Esse apoio foi crucial para auxiliar nos custos da publicação do artigo na ACM-SAC 2024, contribuindo significativamente para a concretização deste trabalho.

À Banca, composta pelos professores Cláudio de Souza Baptista, Ph.D. (PPGCC/UFCG), Carlos Eduardo Santos Pires (PPGCC/UFCG), Dr. Luciano de Andrade Barbosa (CIn/UFPE), Dalton Cézane Gomes Valadares (PPGCC/UFCG e IFPE), Dr. Geraldo Braz Junior (PPGCC/UFMA) e Dr. Geraldo Pereira Rocha Filho (DCEN/UESB), pelas valiosas críticas e contribuições durante o exame de qualificação. Meu agradecimento se estende também ao professor Dr. Fábio Gomes de Andrade (IFPB), pela honra de se juntar à comissão examinadora nesta fase final de defesa de tese. Sou profundamente grato por todas as contribuições que enriqueceram minha tese e impactaram positivamente minha trajetória acadêmica e profissional.

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho. Palavras de apoio e encorajamento foram fundamentais ao longo dessa jornada.

"Você não pode voltar e mudar o começo, mas pode começar onde está e mudar o final."

C. S. Lewis

RESUMO

O processo decisório nas organizações depende cada vez mais de dados. Contudo, problemas relacionados à qualidade desses dados, como informações incompletas, inconsistentes e redundantes, representam desafios significativos. A integração de dados surge como uma área de pesquisa fundamental para combinar e unificar informações provenientes de diferentes fontes e formatos, mesmo em ambientes heterogêneos e autônomos, de modo a proporcionar uma visão abrangente e consistente das informações. No contexto de transações comerciais de compra e venda, as empresas emitem notas fiscais para comprovar as transações realizadas. Entretanto, os dados dos produtos presentes nessas notas fiscais não possuem padronização, podendo apresentar descrições curtas, variadas e inconsistências. Esta pesquisa aborda os desafios técnicos de integração de dados e *Product Matching* em cenários com dados limitados ou incompletos, como os presentes em notas fiscais. A abordagem proposta, denominada STEPMatch, utiliza técnicas de Recuperação da Informação e Processamento de Linguagem Natural para realizar a correspondência entre textos curtos, como as descrições de produtos encontradas nas notas fiscais. Os resultados obtidos demonstram a eficácia do STEPMatch na correspondência entre produtos, alcançando uma acurácia de 98,11% em um cenário de teste. Técnicas de *Cross-Lingual Learning* também foram exploradas de forma inovadora na área de *Product Matching*, aprimorando a generalização dos modelos de aprendizado de máquina em contextos com escassez de dados anotados, com resultados promissores na adaptação entre idiomas e domínios.

Palavras-chave: Product Matching, Integração de Dados, Processamento de Linguagem Natural, Cross-Lingual Learning, Recuperação da Informação

ABSTRACT

Decision-making processes in organizations increasingly depend on data. Therefore, issues related to data quality, such as incomplete, inconsistent, and redundant information, represent significant challenges. Data integration emerges as a critical research area, focused on combining and unifying information from different sources and formats, even in heterogeneous and autonomous environments, aiming to provide a comprehensive and consistent data view. For commercial transactions, companies issue invoices to document sales and purchases. However, the product data within these invoices often lack standardization, potentially presenting short, varied, and inconsistent descriptions. This research addresses the technical challenges of data integration and Product Matching in scenarios with limited or incomplete data, such as those in invoices. Our proposed approach, STEPMatch, leverages Information Retrieval and Natural Language Processing techniques to match short texts, such as invoice product descriptions. The results demonstrated the effectiveness of STEPMatch, achieving an accuracy of 98.11% in a test scenario. Additionally, we present a novel approach by adopting cross-lingual learning techniques within the Product Matching field, enhancing the generalization of machine learning models in contexts with limited labeled data and yielding promising results in cross-lingual and cross-domain adaptation. Our primary contribution lies in adopting machine learning techniques for product-matching, training in scenarios targeting low-resource language data, and demonstrating the feasibility of improving product-matching quality in large volumes of data from distinct languages.

Keywords:Product Matching, Data Integration, Natural Language Processing, Cross Lingual Learning, Information Retrieval.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

AutoML	Automated Machine Learning
B2B	Business to Business
B2C	Business to Consumer
BERT	Bidirectional Encoder Representations from Transformers
BM25	Best Match 25
CEST	Código Especificador da Substituição Tributária
CL	Cascade Learning
CLL	Cross-Lingual Learning
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
EAN	European Article Number
GTIN	Global Trade Item Number
KNN	K-Nearest Neighbors
IA	Inteligência Artificial
JL	Joint Learning
LLM	Large Language Model
MAP	Mean Average Precision
MLM	Masked Language Model
MLPT	Modelo de Linguagem Pré-Treinado
MRR	Mean Reciprocal Rank
NCM	Nomenclatura Comum do Mercosul
NDCG	Normalized Discounted Cumulative Gain
NLP	Natural Language Processing
NSP	Next Sentence Prediction
RE	Resolução de Entidades
RI	Recuperação de Informação
RNN	Recurrent Neural Network
SKU	Stock Keep Unit
STEPMatch	Short Text Product Matching

SVM	Support Vector Machine
UPC	Universal Product Code
ZST	Zero-Shot Transfer

LISTA DE SÍMBOLOS

θ	Letra grega Teta, usada para representar ângulos ou variáveis
τ	Letra grega Tau, frequentemente usada para representar constantes
$\emptyset$	Indica o conjunto vazio, ou seja, um conjunto sem elementos
$\cap$	Representa a interseção entre conjuntos (elementos comuns)
$\cup$	Representa a união entre conjuntos (todos os elementos combinados)
$\in$	Significa “pertence a”, indicando que um elemento faz parte de um conjunto
$\subset$	Significa “está contido” e é usado para indicar que um conjunto é um “subconjunto próprio” de outro

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, a crescente disponibilização de plataformas online, como os sites de comércio eletrônico, tem aumentado a complexidade da análise e da compreensão das informações, predominantemente não estruturadas, impulsionando pesquisas no campo da Mineração de Textos (HAN et al., 2023; MANNING, 2022). Esse cenário digital descentralizado, caracterizado por dados heterogêneos e frequentemente redundantes, apresenta desafios significativos para a identificação, vinculação e integração dessas informações, aspectos fundamentais para a tomada de decisões baseadas em dados.

A Internet contém informações que descrevem entidades do mundo real. Uma entidade pode ser um produto, uma pessoa, um local, uma organização ou qualquer outro objeto nomeado. Essas entidades são frequentemente descritas de formas distintas em diferentes plataformas. Por exemplo, a Figura 1.1 ilustra como um mesmo produto pode ser representado de maneiras diferentes em diversas plataformas de comércio eletrônico. Nesse contexto, torna-se necessário aprimorar os mecanismos de busca e integração de dados para fornecer aos usuários informações consistentes e completas.

Figura 1.1 – Exemplos de representações distintas de um mesmo produto em plataformas de comércio eletrônico

A tomada de decisões nas organizações contemporâneas é cada vez mais orientada por dados (BOUSDEKIS et al., 2021; PROVOST; FAWCETT, 2013). Contudo, o processo de integração de dados enfrenta desafios complexos, como a presença de informações redundantes, conflitantes (inconsistências) ou incompletas, o que exige processos de limpeza, padronização e validação para garantir a qualidade das informações (CHRISTOPHIDES et al., 2020). A integração de dados, enquanto área de pesquisa, busca combinar e unificar informações de diferentes fontes e formatos, provenientes de ambientes heterogêneos e autônomos, oferecendo aos usuários, sejam pessoas ou sistemas, uma visão abrangente e consistente (DOAN et al., 2012). Esse processo é fundamental para assegurar a precisão e coerência dos dados, elementos essenciais para uma tomada de decisão eficaz. Nesse contexto, a Resolução de Entidades desempenha um papel crucial, pois identifica e elimina duplicações, mesmo quando os dados estão representados de forma diversa em diferentes

sistemas (CHRISTEN, 2012; DONG et al., 2009).

A Resolução de Entidades (RE) (CHRISTOPHIDES et al., 2015; ELMAGARMID et al., 2007; CHRISTEN, 2012), também conhecida na literatura como *entity resolution*, *record linkage* (interligação entre registros), *duplicate detections* (detecção de duplicatas) ou *reference reconciliation* (reconciliação de referências), é uma tarefa que busca resolver o problema de identificar diferentes representações de entidades que referenciam os mesmos objetos no mundo real, possibilitando a integração real dos dados em diversas aplicações. Pesquisas em RE têm despertado o interesse da comunidade científica em diversas áreas (BINETTE; STEORTS, 2022). Para realizar a tarefa de RE de forma eficaz, a abordagem deve ser capaz de lidar com casos em que uma mesma entidade é representada de maneiras diferentes, ou entidades distintas são apresentadas de maneiras semelhantes em fontes de dados distintas, ainda que os identificadores únicos das entidades não sejam conhecidos (CHRISTOPHIDES et al., 2020).

Aplicações que requerem RE podem enfrentar desafios complexos, especialmente quando os textos são ambíguos ou apresentam variações contextuais, como diferenças geográficas, culturais ou temporais. A correspondência de produtos, ou *Product Matching* em inglês, é uma subcategoria específica da RE que visa a identificar e agrupar produtos semelhantes. Esse tema tem recebido grande atenção na literatura, impulsionado pelo crescimento do comércio eletrônico e pelo crescente interesse comercial (CHRISTEN, 2008a; BHATTACHARYA; GETOOR, 2006; FIRMANI et al., 2016; PEETERS; BIZER, 2022; ŁUKASIK et al., 2021). A tarefa de correspondência de produtos em ofertas de comércio eletrônico envolve vários desafios, entre os quais: catálogos de produtos extensos e em evolução, dados heterogêneos (com diferentes fornecedores que descrevem produtos de formas distintas, incluindo variações nas imagens e características), descrições imprecisas que contêm sinônimos ou abreviações, erros ortográficos e a ausência de informações cruciais, como o nome do fabricante, embalagem, quantidade ou volume (ENAMOTO et al., 2021).

A correspondência de produtos tem um impacto significativo em diferentes áreas, elevando o interesse de soluções em diversos *stakeholders*, como consumidores, empresas do comércio (varejistas) e governo. Para os consumidores, ferramentas como os agregadores de produtos permitem comparar ofertas de produtos similares, facilitando a escolha da melhor opção de compra, ao considerarem não apenas o preço, mas também as características específicas dos produtos (LI et al., 2020) e as opiniões de outros consumidores (VIEIRA et al., 2022). No entanto, pequenas variações nas descrições dos produtos podem confundir os consumidores, levando-os a escolhas equivocadas. A Figura 1.2 ilustra esse ponto, mostrando que produtos com configurações distintas podem ser apresentados como equivalentes, induzindo erros de interpretação. Nesse caso específico, a ferramenta

agregadora de produtos Zoom[1] destaca o produto com o melhor preço de oferta. Contudo, o produto dessa oferta possui uma característica inferior, com apenas 16GB de memória RAM, enquanto a especificação pesquisada[2] pelo usuário inclui 32GB de memória RAM, disponível em outras ofertas listadas.

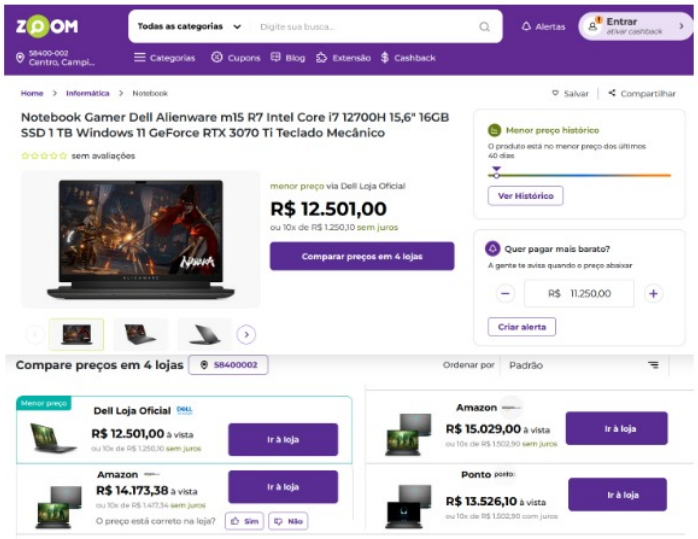

Figura 1.2 – Exemplo de Comparação de Ofertas de Produtos em um Agregador de Produtos.[2]

Por outro lado, empresas e governos também se beneficiam da correspondência de produtos. No setor privado, ela é essencial para otimizar estratégias de sortimento e precificação, permitindo que empresas identifiquem produtos similares nos catálogos dos concorrentes, o que ajuda a ajustar ofertas de produtos e definir preços competitivos (BERNSTEIN et al., 2015; ITO; FUJIMAKI, 2016). Essas estratégias são fundamentais para ampliar a receita e proporcionar uma posição competitiva no mercado.

No setor público, a correspondência de produtos oferece benefícios significativos para melhorar a eficiência dos gastos governamentais e auditar corretamente a cobrança de impostos. Ao comparar os preços de produtos e serviços comprados pelo governo com aqueles oferecidos no mercado privado, é possível garantir que as aquisições públicas sejam feitas de forma econômica e transparente. Além disso, essa tarefa facilita a auditoria de taxas de impostos, identificando inconsistências nos registros fiscais, o que contribui para a fiscalização mais eficaz das transações comerciais (SCHULTE et al., 2022).

A automatização do processo de correspondência de produtos é uma necessidade em diversos setores. Os dados de produtos registrados nas transações comerciais (*B2B - Business to Business* ou *B2C - Business to Consumer*) representam uma fonte valiosa

[1] <https://www.zoom.com.br>

[2] <https://www.zoom.com.br/notebook/notebook-gamer-dell-alienware-m15-r7-intel-core-i7-12700h-15.6-32gb-ssd-1-tb-windows-11-geforce-rtx-3070-ti-teclado-mecanico?utm_source=share&utm_medium=share_web&utm_campaign=share_prod&lc=305>. Acesso realizado em 20 de junho de 2023.

para a tarefa de correspondência de produtos. No Brasil, as notas fiscais eletrônicas são documentos obrigatórios que registram a comercialização de produtos e serviços, e servem como base para a cobrança de impostos, de acordo com a natureza do item comercializado, conforme a Lei N° 8.846, de 21 de janeiro de 1994. No entanto, esses documentos estão sujeitos a erros ou omissões nos códigos e descrições dos produtos, o que dificulta a correta associação entre itens semelhantes e aumenta o risco de fraudes fiscais, uma vez que os produtos podem ser cadastrados de forma incorreta, resultando em tributação inadequada (KIECKBUSCH et al., 2021; SCHULTE et al., 2022).

O Quadro 1.1 apresenta dados extraídos das notas fiscais utilizadas neste trabalho, destacando os principais problemas encontrados nos identificadores de produtos. Esse quadro inclui uma coluna que identifica os principais tipos de problemas associados aos identificadores dos produtos, a saber: a) identificador inválido ("código inválido"); b) ausência de identificação ("sem código"); e c) produto com código identificador de outro produto ("código de outro produto"). Há também uma coluna que indica o código correto do produto para referência. O GTIN - *Global Trade Item Number* - é o código que identifica de forma única um produto ou serviço. Portanto, soluções que envolvem a integração de dados de notas fiscais precisam tratar adequadamente esses dados, garantindo a correspondência correta dos produtos aos respectivos GTIN's.

Quadro 1.1 – Exemplo de Títulos de Produtos

GTIN	Descrição do Produto	Problema de associação	Código Correto
7893000949355	Empanado De Frango Tradicional Nuggets Sadia 300g	-	7893000949355
7893000949355	Nuggets De Frango Sadia 300g Tradicional.	-	
7893000475038	Empanado De Frango Tradicional Nuggets S 300g	código de outro produto	
SEM GTIN	Empanado De Frango Tradicional Nuggets Sadia	sem código	
NULL	Emp. de Frango Tradicional Nuggets Sadia	sem código	
78930009493	Nuggets De Frango Sadia	código inválido	
7893000474901	Empanado Sadia Nuggets de Frango com Queijo 300 g	-	7893000474901
7893000949355	Empanado De Frango Queijo Sadia Nuggets Pacote 300g	código de outro produto	
7893000475038	Nuggets Sadia Crocante De Frango 300 G	-	7893000475038
7891515472436	Empanado De Frango Perdigão 100g	-	7891515472436
789151597	Empanado De Frango Perdigão 1000g	código inválido	7891515975920
7891515544041	Mini Chicken Frango Perdigão 1kg Tradicional	-	7891515544041
7891515786298	Big Chicken Perdigão 1kg Tradicional	-	7891515786298

Em alguns estados do Brasil, há exemplos de ferramentas que necessitam de dados integrados de notas fiscais, como o Banco de Preço[3] ou o Preço de Referência[4]. Essas ferramentas agregam informações de produtos a partir do GTIN para calcular os preços médios praticados no mercado, fornecendo informações valiosas para empresas, consumidores e governo. A Figura 1.3 ilustra o histórico de preço de um produto gerado pela ferramenta Preço da Hora[5], revelando uma dispersão atípica de preços que se manifesta em variações abruptas registradas entre os dias 19/04/2023 e 10/05/2026. Essa anomalia é atribuída a inconsistências nos dados relacionados aos produtos.

[3] <https://bancodeprecos.tceac.tc.br/banco-precos/>
[4] <https://precodereferencia.tce.pb.gov.br/>
[5] <https://precodahora.pb.gov.br>

Figura 1.3 – Exemplo de inconsistências no histórico de preços de um produto em uma ferramenta de monitoramento de preços

Diante dessa vulnerabilidade nos dados de produtos, torna-se essencial uma solução automatizada que não apenas processe os códigos de identificação, mas também verifique se as descrições fornecidas pelos estabelecimentos correspondem corretamente a esses códigos. Uma solução eficiente de correspondência de produtos capaz de lidar com descrições curtas e inconsistentes pode melhorar significativamente a auditoria e a fiscalização tributária, assegurando maior precisão nas cobranças e reduzindo a possibilidade de fraudes. Além disso, essa solução também traz benefícios para as ferramentas agregadoras de preços, ao fornecer dados mais consistentes para a comparação dos preços praticados no mercado.

Esta tese propõe uma abordagem para a tarefa de correspondência de produtos em descrições curtas, como as que aparecem em notas fiscais eletrônicas. A abordagem denominada STEPMatch (*Short Text Product Matching*) abrange a verificação e a associação correta entre identificadores e descrições de produtos. Essa abordagem permite corresponder os produtos corretamente, promovendo a integração e o enriquecimento dos dados oriundos de diferentes fontes.

Muitos trabalhos na literatura exploram o problema de correspondência de produtos utilizando técnicas de Processamento de Linguagem Natural (PLN), incluindo modelos de linguagem baseados em *Transformers*, para melhorar a identificação de correspondências e através da captura do contexto e da semântica nas descrições de produtos (LI et al., 2020; PEETERS et al., 2020; NARARATWONG et al., 2020; KIM et al., 2022; QIU et al., 2018). Esses trabalhos frequentemente empregam modelos *Cross-Encoders* para tarefas de correspondência ou similaridade entre textos, pois permitem processar os textos simultaneamente. Entretanto, essas abordagens se concentram predominantemente em

dados no idioma inglês e em produtos do comércio eletrônico que apresentam informações estruturadas e detalhadas, como especificações técnicas, preços, marcas, modelos e até imagens. Em sua maioria, tais soluções não enfrentam os desafios impostos por dados mais limitados, como as descrições curtas presentes em notas fiscais eletrônicas, especialmente no contexto do idioma português (SCHULTE et al., 2022; ROMUALDO et al., 2021).

A escassez de dados rotulados de produtos no idioma português com descrições curtas restringe a aplicação de técnicas tradicionais de aprendizado supervisionado. Uma possível abordagem para mitigar esse problema é o uso de técnicas de aprendizado que aproveitam dados anotados em outros idiomas para treinar modelos, como as abordagens de *Cross-Lingual Learning* (CLL) (PIKULIAK et al., 2021). Essa estratégia, que busca contornar a falta de dados anotados em um idioma específico, já demonstrou resultados promissores em campos como análise de sentimentos, reconhecimento de entidades nomeadas e detecção de discurso de ódio (STAPPEN et al., 2020; OLIVEIRA et al., 2024). Contudo, na área de correspondência de produtos, o potencial do CLL ainda não foi devidamente explorado.

A busca de produtos correspondentes em cenários que envolvem descrições curtas, frequentemente sujeitas a ruídos, ambiguidades e informações incompletas ou insuficientes, representa um grande desafio para métodos tradicionais de Recuperação de Informação (RI), como TF-IDF e BM25 (RATERIA; SINGH, 2024; HAMBARDE; PROENçA, 2023; MANNING, 2009). Essas técnicas apresentam limitações importantes, como a incapacidade de capturar o contexto das palavras e sua semântica, o que compromete a relevância nos resultados de busca em situações nas quais produtos idênticos podem ser descritos de formas distintas (CHOI et al., 2020a; LADANAVAR et al., 2024). Nesse sentido, a adoção de modelos de linguagem treinados para o domínio específico de produtos de notas fiscais pode melhorar a precisão da recuperação de informações relevantes (PAN et al., 2023; WANG et al., 2020).

Diante desse contexto, esta tese propõe uma abordagem inovadora, adaptada às características específicas de descrições curtas de produtos, visando a superar as limitações dos atributos disponíveis nos dados de notas fiscais eletrônicas.

1.1 OBJETIVOS

O objetivo principal desta pesquisa é propor uma abordagem para identificar correspondências entre produtos, baseando-se em descrições textuais curtas, utilizando técnicas de processamento de linguagem natural e recuperação da informação.

A partir desse objetivo principal, foram definidos os seguintes objetivos específicos:

- Comparar os resultados de modelos de classificação de correspondência de produtos que utilizam técnicas tradicionais de aprendizagem de máquina e modelos de linguagens pré-treinados;
- Propor estratégias de *Cross-Lingual Learning* no contexto de análise de correspondências entre produtos com descrições curtas;
- Utilizar modelos de classificação de produtos para melhorar a relevância de resultados de algoritmos tradicionais de buscas utilizados no contexto de Recuperação da Informação para encontrar produtos correspondentes;
- Construir um corpus de produtos do idioma português que contenham pares de descrições curtas de produtos anotados para avaliação de técnicas de correspondência de produtos.

1.2 QUESTÕES DE PESQUISA

Para atender aos objetivos desta tese, buscou-se responder às seguintes questões de pesquisa:

- **Q1:** A utilização de Modelos de Linguagens Pré-Treinados apresenta resultados melhores na tarefa de correspondência de produtos com descrições curtas quando comparados com técnicas tradicionais de aprendizagem de máquina em que as *features* são medidas de similaridade de textos?
- **Q2:** O uso de técnicas de aprendizagem de máquina utilizando técnicas de CLL melhora os resultados de classificação de correspondência entre produtos?
- **Q3:** Estratégias de aprendizado contrastivo baseadas em similaridade são eficazes para gerar corpora rotulados que possam treinar modelos capazes de identificar correspondências entre produtos?
- **Q4:** Técnicas de aprendizagem de máquina supervisionada podem melhorar a relevância dos resultados de busca de produtos, quando usadas em conjunto com o algoritmo BM25 em um ambiente de RI?

Partindo dessas questões de pesquisa, este trabalho estabelece a hipótese de que a combinação de técnicas de RI e aprendizagem de máquina, especialmente com o uso de Cross-Encoders e técnicas de CLL, pode ser empregada para realizar a correspondência de produtos em um ambiente com limitações de dados, descrições curtas, ruídos, heterogeneidade e ausência de informações detalhadas.

1.3 CONTRIBUIÇÕES

Esta tese apresenta o STEPMatch, uma abordagem para a resolução de entidades em descrições curtas de produtos, com foco em dados de notas fiscais eletrônicas. Ao explorar técnicas de Processamento de Linguagem Natural e Recuperação da Informação, a abordagem enfrenta os desafios de lidar com dados limitados, ruidosos e não estruturados, característicos desse cenário. Ao realizar as correspondências entre produtos de forma adequada, o STEPMatch permite a integração de dados de produtos para auxiliar processos gerenciais dependentes de informações consistentes sobre os produtos. As principais contribuições incluem:

- **Proposta de uma abordagem específica para realizar correspondências entre produtos para o domínio de notas fiscais:** a abordagem proposta considera as características específicas das descrições de produtos em notas fiscais, utilizando técnicas de Processamento de Linguagem Natural e de Aprendizagem de máquina para lidar com os ruídos nas descrições dos produtos;
- **Avaliação de CLL no domínio de correspondência de produtos:** este trabalho apresenta uma abordagem inovadora, utilizando técnicas de CLL para aprimorar a correspondência de produtos. Foram explorados diferentes modelos de linguagem (LLMs), tanto monolíngues quanto multilíngues, avaliando a aplicação de dados anotados em inglês para treinar modelos voltados à verificação de correspondências de produtos em português. Os resultados indicam que as estratégias de CLL são abordagens promissoras para melhorar o desempenho dos modelos de correspondência de produtos em idiomas com recursos limitados, otimizando, assim, o uso dos dados disponíveis;
- **Busca de Produtos Correspondentes com Auxílio de Modelos de Linguagem Treinados:** a busca de produtos correspondentes foi proposta com a aplicação de técnicas de aprendizagem de máquina em um ambiente de RI. Inicialmente, o algoritmo BM25 é utilizado para recuperar um conjunto de produtos candidatos. Posteriormente, um modelo de linguagem *cross-encoder* - treinado especificamente para a tarefa de correspondência de produtos - realiza o reordenamento desses candidatos, priorizando os produtos correspondentes no topo da lista;
- **Criação de corpora de produtos para treinar modelos de classificação em correspondência de produtos:** foi proposta uma metodologia contrastiva para a criação de corpora anotados. A criação de pares de produtos distintos, mas com descrições similares, permitiu o treinamento de modelos mais robustos, capazes de capturar nuances específicas das características dos produtos, para melhorar a tarefa de correspondência de produtos;

- **Desenvolvimento de uma solução completa para correspondência entre produtos aplicada a dados de notas fiscais eletrônicas no Brasil:** como uma aplicação prática, foi proposto o STEPMatch, uma solução que integra dados de produtos, promovendo a correta associação entre produtos comerciais e seus identificadores globais (GTIN/EAN). Essa solução foi validada em cenários reais, demonstrando sua aplicabilidade para aprimorar a fiscalização e a auditoria de transações comerciais, bem como para mitigar a ocorrência de fraudes fiscais.

Além das contribuições teóricas e experimentais discutidas, destaca-se a aplicação da metodologia proposta nesta tese, que já está em uso em um ambiente de produção. A ferramenta Banco de Preços[6], desenvolvida por meio de um acordo de parceria para Pesquisa, Desenvolvimento e Inovação (PD&I) entre a Universidade Federal de Campina Grande e o Tribunal de Contas do Estado do Acre, emprega algumas das técnicas de correspondência entre produtos apresentadas neste trabalho para integrar dados de produtos comercializados no estado do Acre, oferecendo suporte à tomada de decisões no contexto de compras públicas.

1.4 PUBLICAÇÕES

As publicações resultantes deste doutorado abrangem tanto os trabalhos diretamente relacionados ao tema principal da tese quanto estudos que exploram técnicas correlatas aplicadas a outras áreas. Nesta seção, as publicações são organizadas em duas categorias: (i) publicações que tratam diretamente das questões de pesquisa e contribuições centrais da tese; e (ii) publicações relacionadas a temas correlatos, em que técnicas exploradas ao longo deste trabalho foram aplicadas em contextos distintos, como análise de sentimentos e visão computacional.

1.4.1 Publicações Relacionadas ao Tema da Tese

Esta subseção apresenta os artigos diretamente relacionados às questões de pesquisa abordadas nesta tese. Esses trabalhos investigam estratégias para lidar com desafios no domínio do *product matching*, como a heterogeneidade dos dados, a escassez de dados anotados e a avaliação de métodos de busca em catálogos de produtos. Os artigos listados contribuem diretamente para responder às perguntas de pesquisa e são baseados nos experimentos e nas soluções que foram propostas ao longo deste trabalho.

1. de Santana, M.A., de Souza Baptista, C., Alves, A.L.F., Firmino, A.A., da Silva Januário, G., da Silva Caldera, R.W. (2023). *Using Machine Learning and NLP for*

[6] <https://bancodeprecos.tceac.tc.br/banco-precos/#/>

*the Product Matching Proble*m. In: Nagar, A.K., Singh Jat, D., Mishra, D.K., Joshi, A. (eds) Intelligent Sustainable Systems. Lecture Notes in Networks and Systems, vol 579. Springer, Singapore. <https://doi.org/10.1007/978-981-19-7663-6_41>

Esse artigo apresenta a estratégia de criação de corpora com produtos de notas fiscais e compara o desempenho de modelos tradicionais de aprendizagem de máquina com modelos de linguagem baseados no BERT. O artigo está relacionado às questões de pesquisa Q1 e Q3 desta tese.

2. Andre Luiz Firmino Alves, Claudio de Souza Baptista, Luciano Barbosa, and Clecio B. M. Araujo. 2024. *Cross-Lingual Learning Strategies for Improving Product Matching Quality*. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24). Association for Computing Machinery, New York, NY, USA, 313–320. <https://doi.org/10.1145/3605098.3636001>

   Esse artigo investiga o uso de técnicas de CLL para a correspondência de produtos no comércio eletrônico, abordando o desafio da heterogeneidade dos dados e a escassez de dados rotulados em determinados idiomas. O artigo está relacionado à questão de pesquisa Q2.

3. Francisco Igor de Lima Mendes, Melquisedeque Carvalho Silva, André Luíz Firmino Alves, Eniedson Fabiano Pereira da Silva Júnior, Mateus Queiroz Cunha, and Cláudio de Souza Baptista. 2025. *Comparative Study of Lexical and Semantic Approaches in Closed-domain Product Search*. In The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31–April 04, 2025, Catania, Italy. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3672608.3707948>

   Este artigo compara métodos de busca léxica e semântica no domínio de produtos, considerando descrições presentes em notas fiscais eletrônicas e em um catálogo de produtos governamental. A pesquisa apresentada está diretamente relacionada à questão de pesquisa Q4.

4. André Luíz Firmino Alves, Cláudio de Souza Baptista, José Itallo Martins Silva Diniz, Francisco Igor de Lima Mendes, Mateus Queiroz Cunha. 2025. *An Approach for Product Record Linkage Using Cross-Lingual Learning and Large Language Models*. In The 27th International Conference on Enterprise Information Systems (ICEIS'25), Porto, Portugal.

   Este artigo apresenta o STEPMatch como uma abordagem para Record Linkage aplicada às descrições de produtos em notas fiscais. O trabalho analisa métodos léxicos, semânticos e híbridos para a busca de produtos e propõe um método de *re-ranking* baseado em um modelo *cross-encoder* treinado com técnicas de CLL, visando aprimorar a classificação e a precisão dos resultados obtidos.

1.4.2 Outras Publicacões

Além das publicações diretamente relacionadas ao tema da tese, também foram realizadas pesquisas em áreas correlatas, utilizando técnicas similares às aplicadas no *product matching*. Estes trabalhos exploram o uso de modelos baseados em aprendizagem de máquina e *transformers* em diferentes contextos, como análise de sentimentos e diagnóstico por imagens médicas. As contribuições apresentadas nesses estudos refletem a versatilidade das abordagens investigadas neste doutorado.

1. Alves, André Luiz Firmino, Baptista, Cláudio de Souza, Serrano de Andrade, Davi Oliveira , de Oliveira, Maxwell Guimarães, de Oliveira, Aillkeen Bezerra de. 2021. A spatiotemporal approach for social media sentiment analysis. First Monday, 26(8). <https://doi.org/10.5210/fm.v26i8.10757>

2. de Oliveira, Aillkeen Bezerra , Alves, André Luiz Firmino, Baptista, Cláudio de Souza. 2021. Using Opinion Mining in Student Assessments to Improve Teaching Quality in Universities. In: Abraham, A., Siarry, P., Ma, K., Kaklauskas, A. (eds) Intelligent Systems Design and Applications. ISDA 2019. Advances in Intelligent Systems and Computing, vol 1181. Springer, Cham. <https://doi.org/10.1007/978-3-030-49342-4_22>

3. Negreiros, Ramoni Reus Barros, Silva, Heloíse Santos Silva, Alves, André Luiz Firmino, Valadares, Dalton Cézane Gomes, Perkusich, Angelo, Baptista, Cláudio de Souza. 2023. COVID-19 Diagnosis Through Deep Learning Techniques and Chest X-Ray Images. SN COMPUT. SCI. 4, 613. <https://doi.org/10.1007/s42979-023-02043-1>

1.5 ESTRUTURA

O restante desta tese está estruturado da seguinte forma: no Capítulo 2, apresenta-se a Fundamentação Teórica, abordando os principais conceitos utilizados na tese. No Capítulo 3, discutem-se os trabalhos relacionados. O Capítulo 4 foca na proposta da tese, apresentando o STEPMatch e descrevendo as principais técnicas utilizadas. A metodologia utilizada na construção de corpora de produtos anotados do idioma português é discutida no Capítulo 5. No capítulo 6, avaliam-se os resultados dos experimentos realizados, discutindo e respondendo às questões de pesquisa propostas. Finalmente, no Capítulo 7, realizam-se as considerações finais sobre os resultados obtidos e as indicações de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os conceitos fundamentais que embasam o presente trabalho. Inicialmente, na Seção 2.1, discutem-se os conceitos de aprendizagem de máquina, incluindo modelos supervisionados e por transferência. Em seguida, na Seção 2.2, são destacados os conceitos essenciais relacionados ao Processamento de Linguagem Natural (PLN). Na Seção 2.3, são apresentados os principais conceitos sobre Modelos de Linguagem Pré-Treinados. Os conceitos de Cross-Lingual Learning (CLL) são abordados na Seção 2.4. Na Seção 2.5, apresentam-se os principais conceitos de resolução de entidades, incluindo os desafios de Correspondência de Produtos. Além disso, na Seção 2.6, discutem-se os conceitos centrais relacionados à recuperação da informação. Finalmente, na Seção 2.7, são apresentadas as métricas de avaliação utilizadas no contexto deste trabalho.

2.1 APRENDIZAGEM DE MÁQUINA

A aprendizagem de máquina é uma área da Inteligência Artificial(IA) que possibilita que as máquinas aprendam a partir de dados (MITCHELL, 1997). A ideia de ensinar máquinas por meio de exemplos remonta a 1959, quando Arthur L. Samuel propôs que as máquinas pudessem aprender com exemplos e experiências passadas, em vez de serem explicitamente programadas para fazer algo específico (SAMUEL, 1959). A combinação do aumento da capacidade computacional e a disponibilização de grandes volumes de dados produzidos nessa era digital, associada ao desenvolvimento de técnicas computacionais avançadas, têm impulsionado as pesquisas do campo da IA, revolucionando várias áreas do conhecimento (LECUN et al., 2015; AALST, 2020; LUDERMIR, 2021).

O aprendizado de máquina é fundamentado no princípio de que sistemas podem aprender a partir de exemplos e experiências acumuladas, permitindo-lhes generalizar o conhecimento adquirido para tomar decisões ou realizar previsões em novos cenários (KOTSIANTIS et al., 2007). Algoritmos de aprendizado de máquina têm sido classificados, de forma clássica, em três tipos: não supervisionado, supervisionado e por reforço.

No aprendizado supervisionado, modelos são treinados com os conjuntos de dados rotulados, aprendendo o relacionamento entre os dados e os rótulos (KOTSIANTIS et al., 2007). Um exemplo clássico de aprendizado supervisionado é a análise de sentimentos, em que o modelo é treinado com um conjunto de textos rotulados como positivos, negativos ou neutros, permitindo que ele identifique a polaridade de novos textos (ALVES et al., 2021). No aprendizado não supervisionado, os algoritmos exploram dados não rotulados para identificar padrões e estruturas subjacentes. Um exemplo de aprendizado não supervi-

sionado é o agrupamento (do inglês, *clustering*), em que os dados são agrupados conforme similaridades, estruturas, padrões e relações intrínsecas (GHAHRAMANI, 2004). Por fim, o aprendizado por reforço envolve a interação do modelo com um ambiente, no qual o modelo aprende com tentativas e erros, baseado em *feedbacks* (positivos ou negativos), sendo recompensado quando atinge determinados objetivos (SUTTON; BARTO, 2018). Além dessas categorias clássicas, surgem abordagens híbridas como o aprendizado semi-supervisionado, que combina dados rotulados e não rotulados, e o aprendizado auto-supervisionado, em que o algoritmo aprende a encontrar padrões nos dados e, em seguida, gera rótulos para exemplos não rotulados (MRABET et al., 2021).

2.1.1 Aprendizado supervisionado

No aprendizado supervisionado a máquina aprende um padrão a partir de dados rotulados, possibilitando generalizar o conhecimento e fazer previsões com base nos exemplos aprendidos. Na fase de treinamento, os parâmetros do modelo de aprendizagem são ajustados para minimizar o erro entre rótulos dos dados de treinamento e as previsões obtidas, através de uma função de custo e de técnicas de otimização (Gradiente Descendente).

A aprendizagem supervisionada é amplamente utilizada em uma variedade de aplicações, podendo resolver problemas relacionados à regressão e classificação.

Em problemas de classificação, a variável de saída é categórica com os dados de entrada associados às categorias. Exemplos clássicos de problemas de classificação incluem Análise de Sentimentos ou Emoções (NANDWANI; VERMA, 2021; ALVES et al., 2021), Detecção de *Fake News* (CHOUDHARY; ARORA, 2021) ou Discurso de Ódio (AYO et al., 2020; OLIVEIRA et al., 2023) e Classificação de Produtos (LI et al., 2023; SANTANA et al., 2023).

Em problemas de regressão, a saída é uma variável numérica contínua, na qual o modelo estima um valor numérico com base nos dados de entrada. Um exemplo de aplicação de regressão pode ser a previsão de cotação do *Bitcoin* (CAI et al., 2022) ou de ações da bolsa de valores (KUMAR et al., 2020).

2.1.2 Aprendizado por transfêrencia

Nos últimos anos, o campo da aprendizagem de máquina tem avançado significativamente, especialmente com a introdução de técnicas que aproveitam o potencial de grandes volumes de dados e a capacidade computacional crescente. Dentre essas técnicas, destacam-se as redes neurais artificiais, em particular as que utilizam múltiplas camadas, conhecidas como aprendizado profundo ou *deep learning*. Essas abordagens têm

ganhado destaque na solução de problemas complexos e estão revolucionando várias áreas, incluindo visão computacional e processamento de linguagem natural (IBA; NOMAN, 2020; GOODFELLOW et al., 2016).

Diferentemente das técnicas tradicionais de aprendizagem de máquina, a aprendizagem profunda dispensa a extração manual de características dos dados para os modelos, pois as camadas neurais conseguem aprender e extrair características relevantes dos dados de forma autônoma, identificando padrões complexos e abstratos (CICHY; KAISER, 2019). No entanto, esses modelos demandam um grande volume de dados rotulados, o que pode limitar seu uso em situações com escassez de dados rotulados (TAN et al., 2018).

Nesse contexto, surge o aprendizado por transferência, uma técnica de aprendizagem de máquina supervisionada na qual o conhecimento adquirido em um modelo pré-treinado é reutilizado para treinar outro modelo. Ao utilizar um modelo pré-treinado em um conjunto de dados abrangente e diversificado, é possível adaptá-lo para diversas tarefas, permitindo a transferência de conhecimento para um modelo especializado em uma tarefa específica. Nesse processo de transferência de conhecimento, a necessidade de um grande conjunto de dados para o novo modelo é reduzida e a rede neural não precisa ser treinada do zero, resultando, assim, em uma diminuição significativa no tempo de treinamento e no custo computacional associados à nova tarefa (WEISS et al., 2016; GOODFELLOW et al., 2016).

A aprendizagem por transferência pode ser dividida em três categorias, dependendo da forma como o conhecimento é transferido (WEISS et al., 2016; ZHUANG et al., 2020):

- Aprendizado por Transferência Indutivo: este é o cenário em que a tarefa ou domínio alvo é diferente da tarefa ou domínio de origem, exigindo o reajuste (refinamento) dos parâmetros. Nesse caso, o modelo é induzido a aprender um novo conhecimento para a tarefa específica, necessitando apenas de alguns novos dados rotulados;
- Aprendizado por Transferência Transdutivo: aqui, o conhecimento é transferido entre instâncias individuais para melhorar a generalização do modelo. Então, nessa categoria as tarefas dos modelos de origem e destino são as mesmas;
- Aprendizado por Transferência Não-supervisionado: nesta categoria, o conhecimento prévio adquirido em uma tarefa de aprendizagem não supervisionada é utilizado para melhorar o desempenho em uma tarefa relacionada. Isso é frequentemente usado para extração de características ou representações.

Este trabalho utilizou a aprendizagem por Transferência Indutiva, em que os

modelos de linguagens pré-treinados foram refinados para um domínio específico (Correspondência de Produtos). Essa abordagem é particularmente adequada para cenários em que o domínio de aplicação apresenta características distintas do contexto original de treinamento dos modelos, exigindo ajustes para capturar nuances específicas com um volume limitado de dados rotulados. Dessa forma, a transferência indutiva permite explorar a riqueza das representações aprendidas previamente, enquanto direciona o modelo para as particularidades da tarefa proposta.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

A área de Processamento de Linguagem Natural (do inglês, *Natural Language Processing* - NLP) é um campo de pesquisa dedicado ao estudo e desenvolvimento de técnicas que permitem às máquinas compreender e interagir com a linguagem humana de maneira eficaz (CHOWDHURY, 2003). A história do NLP remonta à década de 1950, quando Alan Turing realizou o teste proposto que imaginava a possibilidade de um sistema processar a linguagem humana de forma indistinguível de um ser humano (TURING, 1950). Nos primórdios do NLP, na década de 1970, surgiram os primeiros sistemas, embora ainda baseados em abordagens de regras gramaticais e estruturas sintáticas incipientes (ALLEN, 2003). Desde então, houve um progresso significativo no desenvolvimento de sistemas de NLP, impulsionando avanços em diversas áreas que necessitam de processamento automático de textos (JACOB, 2019; PIKULIAK et al., 2021).

O crescimento exponencial de dados não estruturados na *Web*, impulsionado pela proliferação de redes sociais, blogs, wikis e outras plataformas colaborativas, tem sido um dos principais motores do desenvolvimento de pesquisas em NLP (ALLEN, 2003). À medida em que os usuários produzem conteúdos de forma colaborativa na *Internet*, contribuindo para uma inteligência coletiva (O'REILLY, 2007), torna-se fundamental o desenvolvimento de técnicas avançadas para analisar, interpretar e extrair informações valiosas desses dados (CHOWDHARY, 2020; PAJILA et al., 2023). NLP é uma área interdisciplinar que une conceitos da computação, linguística e inteligência artificial, com o objetivo de desenvolver sistemas capazes de compreender, gerar e traduzir a linguagem humana (MANNING, 2022). Com aplicações em diversas áreas, o NLP desempenha um papel fundamental em tarefas como análise de sentimentos em redes sociais (VASHISHTHA; SUSAN, 2019; PANG; LEE, 2008; ALVES et al., 2021), categorização de texto (SAQUETE et al., 2020; P. et al., 2019), detecção de notícias falsas (FIRMINO et al., 2021) e discurso de ódio (OLIVEIRA et al., 2024), tradução automática (BAHDANAU et al., 2015), sumarização de textos (BOORUGU; RAMESH, 2020; WAHAB et al., 2024), reconhecimento de entidades nomeadas (LAMPLE et al., 2016), geração de texto (RADFORD et al., 2019; LAI et al., 2024), respostas automáticas a perguntas (ADLOUNI et al., 2019) e assistentes de

conversação (HERBERT; KANG, 2018).

O pré-processamento de texto é uma etapa essencial nas aplicações de NLP, pois visa a preparar os dados textuais brutos para análise e extração de informações (VIJAYARANI et al., 2015; DOAN et al., 2012). As técnicas de pré-processamento incluem processos como radicalização (*Stemming*) e lematização (*Lemmatization*), que reduzem as palavras às suas formas básicas, remoção de *stopwords* (palavras comuns, como "e", "a", "de", que não agregam significado relevante para a análise), remoção de caracteres especiais e URLs, normalização do texto (transformando-o para maiúsculas ou minúsculas) e tokenização, que divide o texto em unidades menores, como palavras ou frases.

Para permitir que as máquinas processem e compreendam textos, é necessário representá-los numericamente. Isso é realizado por meio de técnicas de vetorização que transformam os textos em vetores numéricos. Essas técnicas se dividem em duas categorias principais: representações esparsas e representações densas (EISENSTEIN, 2018; DENG; LIU, 2018; PATIL et al., 2023).

As representações esparsas incluem métodos como *Bag of Words* (BoW) e *Term Frequency - Inverse Document Frequency* (TF-IDF). O BoW cria vetores de alta dimensionalidade, nos quais cada dimensão representa a presença ou a contagem de uma palavra específica (*n-grams*) no vocabulário. No TF-IDF, os pesos atribuídos a cada palavra são ajustados para refletir sua importância tanto no contexto do documento quanto na coleção como um todo. O peso de uma palavra é calculado levando em conta sua frequência no documento (TF) e sua raridade na coleção (IDF), resultando em vetores esparsos mais refinados que capturam de maneira mais eficaz a relevância das palavras.

De outro modo, as representações densas (*Word Embeddings*) abrangem técnicas como Word2Vec (MIKOLOV et al., 2013), GloVe (*Global Vectors for Word Representation*) (PENNINGTON et al., 2014), FastText (BOJANOWSKI et al., 2016) e abordagens baseadas em modelos de linguagem, como o ELMo (*Embeddings from Language Models*) (PETERS et al., 2018), BERT (*Bidirectional Encoder Representations from Transformers*) (DEVLIN et al., 2018) e GPT (*Generative Pre-Training*) (BROWN et al., 2020). Word2Vec, GloVe e FastText geram vetores contínuos e densos que capturam relações semânticas e contextuais entre palavras. Já as técnicas de vetorização baseadas em modelos de linguagem representam o estado da arte, pois produzem vetores altamente densos e contextuais que capturam o significado das palavras em contextos específicos (PATIL et al., 2023).

A comparação da similaridade entre textos é uma tarefa muito importante em NLP, pois permite, por exemplo, avaliar a proximidade entre documentos e identificar tópicos semelhantes. Essa tarefa envolve uma variedade de técnicas, desde abordagens baseadas

em características léxicas até métodos mais avançados, como o cálculo de distâncias entre vetores de representação textual e o uso de modelos de linguagem de grande escala, que permitem capturar nuances semânticas e contextuais mais complexas (LU et al., 2013; PAPADAKIS; NEJDL, 2011; BINETTE; STEORTS, 2022; SANTANA et al., 2023).

Entre as abordagens léxicas, existem basicamente três principais tipos de algoritmos para medir a similaridade entre *strings* (CHRISTEN, 2012; XU; LU, 2019): algoritmos baseados em edição, que focam nas alterações necessárias para transformar uma *string* em outra; algoritmos baseados em *tokens*, que comparam a presença ou ausência de palavras específicas; e algoritmos baseados em sequência, que avaliam a ordem e a posição dos caracteres.

Entre os algoritmos baseados em edição, destacam-se três distâncias: de Levenshtein, de Damerau-Levenshtein e de Hamming (NAVARRO, 2001). A distância de Levenshtein mede o número mínimo de edições necessárias para transformar uma *string* em outra, considerando as operações de inserção, exclusão e substituição de caracteres. A distância de Damerau-Levenshtein é uma variação da distância de Levenshtein que inclui a operação de transposição. A distância de Hamming, por outro lado, conta as diferenças entre duas *strings* de igual comprimento.

Nos algoritmos baseados em *tokens*, pode-se incluir as similaridades de Jaccard e Dice (NAVARRO, 2001). A similaridade de Jaccard calcula a proporção entre a interseção e a união de conjuntos de palavras. A similaridade de Dice é semelhante, mas duplica o peso da interseção dos conjuntos, oferecendo uma sensibilidade maior a pequenas variações nos textos.

2.3 MODELOS DE LINGUAGENS PRÉ-TREINADOS

No escopo de NLP, Modelos de Linguagem Pré-Treinados (MLPTs), também conhecidos como *Large Language Models* (LLMs), são modelos de aprendizagem de máquina que utilizam técnicas de aprendizado profundo. Esses modelos são treinados com conjuntos de textos muito grandes para capturar padrões e representações linguísticas. Atualmente, os MLPTs são amplamente utilizados em várias tarefas de NLP, principalmente devido à habilidade desses modelos em realizar a transferência de conhecimento para tarefas específicas (WANG et al., 2018).

MLPTs utilizam a arquitetura de *Transformers*, que tem revolucionado muitas aplicações de aprendizagem de máquina (VASWANI et al., 2017; MIN et al., 2023). O *Transformer* é uma arquitetura de redes neurais profundas baseada em codificadores (*encoder*) e decodificadores (*decoder*) que faz uso do mecanismo de Atenção. No *Transformer*,

o *encoder* recebe uma sequência de entrada e gera uma representação dela, enquanto o *decoder* é responsável por traduzir essa representação gerada pelo *encoder*. A Figura 2.1 ilustra a arquitetura de um Transformer.

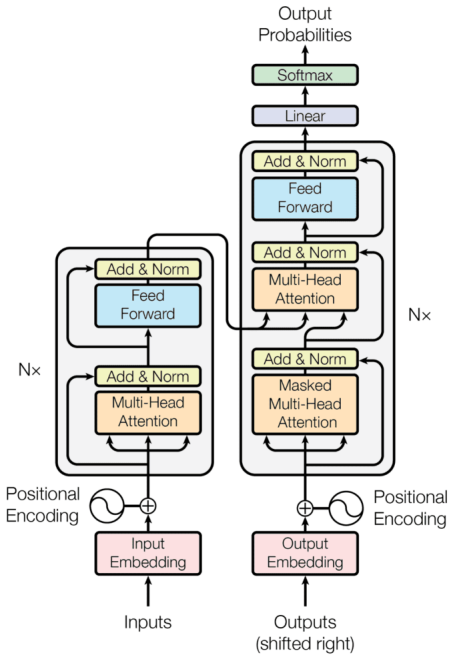

Figura 2.1 – Arquitetura do Transformer.

Fonte: Extraído de Vaswani et al. (2017).

O mecanismo de Atenção permite que o modelo de linguagem considere todo o contexto da sequência. Isso é realizado associando, a cada *token* de entrada da sequência, um vetor chave-valor que captura a relevância dos demais *tokens* de todo o contexto. Esse mecanismo possibilita aprender relacionamentos complexos e capturar dependências de longo alcance em uma sequência.

Diferentemente das tradicionais Redes Neurais de Recorrência (do inglês *Recurrent Neural Network* - RNN) e Redes Neurais de Convolução (do inglês *Convolutional Neural Network* - CNN), os *Transformers* não utilizam mecanismos recorrentes ou convolucionais para processar sequências. Sua arquitetura paralela, baseada em mecanismos de atenção, permite um processamento mais eficiente e rápido dos dados. A arquitetura original do *Transformer* apresenta seis codificadores e seis decodificadores, que, por sua vez, dispõem de camadas de Atenção e Rede Neural *Feed Forward*. No entanto, cada decodificador possui uma camada adicional de Atenção que o auxilia a ponderar partes relevantes dos dados de entrada.

O BERT (*Bidirectional Encoder Representations from Transformers*), um dos primeiros modelos a utilizar a arquitetura de *Transformers*, revolucionou o campo do NLP, superando os resultados em diversas tarefas de aprendizagem de máquina (DEVLIN et al.,

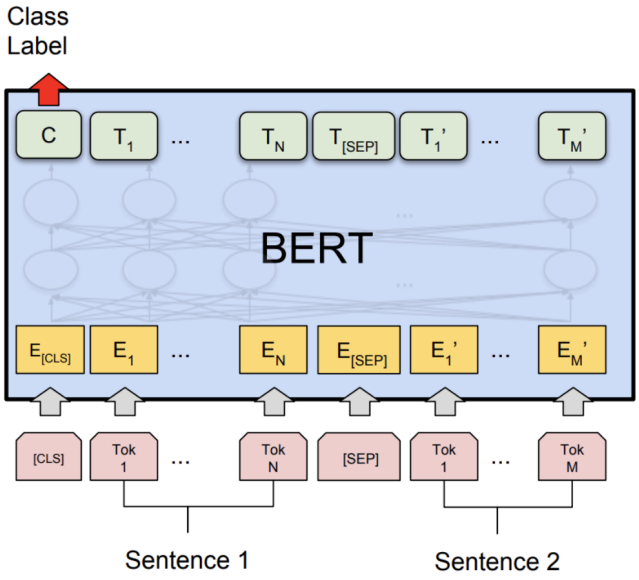

Figura 2.2 – BERT para classificação de pares de sentenças.

Fonte: Extraído de Devlin et al. (2018).

2018). Com doze camadas (modelo base) ou vinte e quatro camadas (modelo grande), o BERT inovou ao utilizar uma arquitetura bidirecional apenas com a camada de *Encoder*, permitindo capturar o contexto em relação às palavras que antecedem e sucedem em uma sequência de texto. Essa capacidade foi aprimorada por meio de duas tarefas de treinamento: Modelo de Linguagem Mascarada (*Masked Language Model*) e Predição da Próxima Sentença (*Next Sentence Prediction*).

No Modelo de Linguagem Mascarada, são descartados aleatoriamente 15% dos *tokens* de entrada para, em seguida, prevê-los utilizando o contexto. Essa tarefa melhora a compreensão da semântica das palavras dentro do contexto. Na tarefa de Predição da Próxima Sentença, o BERT foi treinado com pares de sentenças em que o modelo, durante o treinamento, prevê se uma determinada sentença sucede outra na sequência lógica dos pares de sentenças. Isso capacita o modelo a capturar a lógica de como as sentenças se relacionam umas com as outras.

Na Figura 2.2, é apresentada a arquitetura do BERT aplicada a uma tarefa de classificação que considera pares de sentenças como entrada. O modelo recebe as duas sentenças separadas pelo token especial "[SEP]". O token "[CLS]", inserido no início da entrada, informa ao modelo que a tarefa a ser realizada é de classificação. Em seguida, o BERT gera representações vetoriais (*embeddings*) para cada palavra nas sentenças, que são então utilizadas para determinar a relação entre elas.

Modelos de linguagens do tipo *encoder*, como os baseados na arquitetura do BERT, são frequentemente utilizados para tarefas de classificação devido à sua habilidade em

capturar nuances semânticas e contextuais em textos. Essa capacidade é particularmente vantajosa para lidar com os desafios encontrados em descrições de produtos, como variações linguísticas e dados não padronizados, comuns em problemas de classificação de correspondências.

A partir da arquitetura do BERT, outros MLPTs surgiram, como o RoBERTa (LIU et al., 2019), o DistilBERT (SANH et al., 2020) e o ALBERT (LAN et al., 2020). A biblioteca Hugging Face[1] disponibiliza uma variedade de modelos de linguagens baseados na arquitetura Transformer. Nesta pesquisa, destacam-se os seguintes modelos: BERT monolíngue e Multilíngue (DEVLIN et al., 2019), XLM-RoBERTa (CONNEAU et al., 2021), BERTimbau (SOUZA et al., 2020), Albertina-PT-BR (RODRIGUES et al., 2023) e e-CommerceBERT[2]. Essa seleção abrange tanto modelos monolíngues, como o BERT em inglês, o BERTimbau e o Albertina-PT-BR em português, quanto modelos multilíngues, como o BERT Multilíngue, o XLM-RoBERTa e o e-CommerceBERT.

O XLM-RoBERTa (XLM-R) (CONNEAU et al., 2020), lançado em 2019 , é um modelo multilíngue treinado em 100 idiomas diferentes, desenvolvido a partir do modelo RoBERTa do Facebook. O modelo utiliza a técnica de Codificação de Par de *Bytes* (*Byte Pair Encoding - BPE*) para aprimorar a representação e compreensão de textos em várias línguas. O BPE é uma técnica de codificação de subpalavras que segmenta as informações em unidades menores, como palavras ou partes delas, com base na frequência de ocorrência dos pares de caracteres (*bytes*) mais comuns (SENNRICH et al., 2016). Essa abordagem permite capturar a estrutura e complexidade das palavras em diferentes idiomas, aumentando o vocabulário compartilhado entre eles. Como resultado, o XLM-RoBERTa demonstra excelentes desempenhos em tarefas de Cross-Lingual Learning (CONNEAU et al., 2021).

Os modelos Albertina PT-BR e BERTimbau foram desenvolvidos para processar dados do idioma português e apresentam características distintas em relação ao tamanho, complexidade e desempenho. O BERTimbau, em sua versão grande, possui 335 milhões de parâmetros e 12 camadas. Por outro lado, o Albertina PT-BR destaca-se por sua robustez, uma vez que foi treinado com um volume maior de dados, apresentando 900 milhões de parâmetros e 24 camadas, o que lhe confere uma maior capacidade de captura de informações linguísticas complexas. Em várias tarefas de NLP para dados em língua portuguesa, o Albertina PT-BR tem demonstrado excelentes resultados (RODRIGUES et al., 2023).

Por fim, o modelo e-CommerceBERT, derivado do modelo BERT Multilíngue,

[1] <https://huggingface.co>
[2] <https://huggingface.co/EZlee/e-commerce-bert-base-multilingual-cased>

foi treinado usando um conjunto de dados que inclui informações específicas do comércio eletrônico. Esse treinamento especializado torna o e-CommerceBERT particularmente eficaz em tarefas de correspondência de produtos.

2.4 CROSS-LINGUAL LEARNING

O *Cross-Lingual Learning* (CLL) ou Aprendizagem por Cruzamentos de Idiomas é uma abordagem que visa a aprimorar o desempenho de tarefas em um determinado idioma, utilizando conhecimentos e recursos provenientes de outro idioma, por meio de um processo de transferência de conhecimento (PAN; YANG, 2010).

Essa abordagem apresenta-se como uma solução viável para enfrentar a escassez de dados em idiomas com recursos limitados (PIKULIAK et al., 2021). Em essência, o CLL busca explorar dados rotulados de outros idiomas para construir novos modelos de NLP ou melhorar o desempenho dos modelos já existentes.

Pikuliak et al. (2021) define os paradigmas de transferências, classificando-os em quatro categorias:

- Transferência de rótulo: as anotações ou rótulos das amostras de dados no idioma de origem são transferidos diretamente para amostras correspondentes no idioma de destino. Essa transferência de rótulos ou anotações ocorre entre amostras correspondentes em idiomas distintos e serve como base para treinar o modelo no idioma de destino;
- Transferência de características: é semelhante à transferência de rótulos, mas a transferência ocorre com as características das amostras, as quais são utilizadas para treinar o modelo no idioma de destino;
- Transferência de representação: é semelhante à transferência de características, pois também transfere conhecimento sobre as características das amostras. No entanto, a transferência de representação ensina o modelo a gerar as representações desejadas. Dessa forma, representações semânticas de um idioma fonte podem ser transferidas para o idioma de destino;
- Transferência de parâmetros: nessa categoria, os valores dos parâmetros são transferidos entre os modelos paramétricos, transferindo o comportamento do modelo treinado no idioma de origem para um modelo no idioma de destino.

Esta pesquisa explora o paradigma de Transferência de Parâmetros entre modelos, que pode ocorrer de três maneiras: a) *Zero-Shot Transfer* (ZST), quando não se utilizam

dados do idioma de destino na indução do modelo; b) *Joint Learning* (JL), em que uma porção dos dados do idioma de destino é utilizada no treinamento inicial; e c) *Cascade Learning* (CL), que ocorre quando um treinamento adicional é realizado com os dados do idioma destino. Essa abordagem mostra-se particularmente relevante para a tarefa de classificação de correspondência de produtos, pois possibilita o aproveitamento dos valores dos parâmetros de modelos pré-treinados em idiomas com maior disponibilidade de dados para o treinamento de modelos em idiomas com recursos limitados. Assim, o conhecimento linguístico adquirido durante o pré-treinamento é transferido, promovendo uma compreensão mais profunda e generalizável entre idiomas.

2.5 RESOLUÇÃO DE ENTIDADES

Resolução de Entidades (RE) (*Entity Resolution*), também conhecida como Correspondência de Entidades (*Entity Matching*) ou Vinculação de Registros (*Record Linkage*) e Identificação de Duplicatas (*Duplicate Detection*), é uma área de atuação que tem por objetivo identificar entidades que representam os mesmos objetos no mundo real (CHRISTOPHIDES et al., 2015; BINETTE; STEORTS, 2022). A primeira menção ao termo "Vinculação de Registros" ocorreu em 1946, quando Dunn associou o registro de informações de um indivíduo ao longo de sua vida às páginas de um livro que se referiam ao mesmo indivíduo (DUNN, 1946).

Na era de *Big Data*, em ambientes que integram dados de diversas fontes, é possível encontrar problemas associados à qualidade dos dados, como incompletude (dados parciais), redundância (dados sobrepostos), inconsistência (dados conflitantes) ou simplesmente incorreção (erros de dados). Nesse contexto, uma tarefa típica de RE é melhorar a qualidade dos dados (CHRISTEN, 2012).

2.5.1 Etapas de Resolução de Entidades

Normalmente, os problemas de RE passam por duas grandes etapas (CHRISTEN, 2012; BARLAUG; GULLA, 2021), conforme ilustrado na Figura 2.3: 1) Blocagem (*Blocking*) ou Seleção (*Selection*), que é responsável por determinar o escopo das comparações a serem feitas com o objetivo de minimizar o número de operações necessárias; e 2) Correspondência (*Matching*), que é a etapa responsável por comparar as entidades selecionadas e decidir, efetivamente, se um determinado par de entidades representa a mesma entidade.

A etapa de blocagem visa a tornar o processo de RE mais eficiente e escalável, evitando uma complexidade algorítmica de ordem quadrática, $O(n^2)$, com operações de comparações entre todos os objetos (NASCIMENTO et al., 2019; PAPADAKIS et al., 2021). Limitar o escopo das comparações de objetos é fundamental para reduzir a complexidade

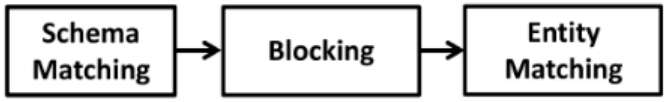

Figura 2.3 – Visão Geral de um Processo de Resolução de entidades

Fonte: Extraído de Papadakis e Nejdl (2011).

computacional, permitindo resolver problemas reais que necessitam integrar dados com um desempenho que atenda às necessidades dos usuários. Normalmente, deseja-se dividir os dados em conjuntos de blocos, em que cada bloco contém entidades que compartilham uma propriedade comum. No entanto, a quantidade e o tamanho dos blocos também interferem no desempenho e na qualidade das soluções propostas (ARAÚJO, 2020).

A etapa de correspondência realiza comparações de pares de entidades, que normalmente envolvem operações de custo computacional elevado através de cálculos de medidas de similaridade de textos (CHRISTEN, 2012). Nesta etapa, duas entidades serão consideradas correspondentes quando os resultados das funções de similaridade forem maiores que um limiar determinado. Para evitar comparações desnecessárias, especialmente quando as técnicas envolvidas são complexas e exigem alto esforço computacional, a verificação é realizada apenas entre pares de entidades candidatas que foram agrupadas previamente na etapa de blocagem. Essa etapa de correspondência é geralmente tratada como um problema de classificação, em que, após o cálculo da similaridade dos atributos das entidades, o par de entidades é classificado como correspondente (*match*) ou não correspondente (*non-match*) (CHRISTOPHIDES et al., 2020).

Atualmente, com o advento da era do *Big Data*, muitas pesquisas de RE estenderam os modelos das primeiras soluções de RE para atender aos 5 V's do *Big Data*: Volume, Velocidade, Variedade, Veracidade e Valor. Se antes o foco estava no desenvolvimento de técnicas para garantir a veracidade dos dados estruturados, atualmente as pesquisas estão preocupadas também com o volume e a variedade dos dados, bem como com a velocidade com que esses dados são processados (CHRISTOPHIDES et al., 2020). Ou seja, é necessário estender as soluções para lidar com dados semiestruturados, com informações ruidosas e heterogêneas, focando na eficácia e eficiência de tempo (PAPADAKIS et al., 2021). Assim, soluções de RE aptas para lidar com Big Data transformam dados em informações valiosas que podem ser usadas no processo de tomada de decisões em tempo hábil nas organizações.

Em Papadakis et al. (2021), os autores categorizam os principais métodos de RE em quatro gerações. Na Figura 2.4, é apresentada uma visão geral de soluções de RE, enfatizando as atividades envolvidas para tratar com *Big Data*. As soluções de RE utilizam

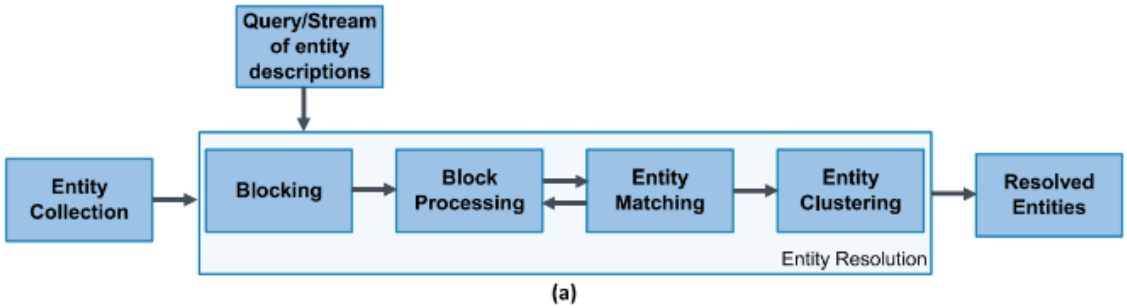

Figura 2.4 – Visão Geral das etapas de uma abordagem para resolução de entidades

Fonte: Extraído de Christophides et al. (2020).

ao máximo os atributos disponíveis das entidades. Normalmente, na primeira etapa, é realizado o mapeamento dos atributos das entidades da coleção, buscando identificar atributos e valores semanticamente semelhantes, de forma a facilitar a próxima etapa de blocagem (*Blocking*).

A segunda etapa, Blocagem, é utilizada para tratar os aspectos de velocidade e volume de dados, evitando a complexidade $O(n^2)$ com as comparações entre todas as entidades da coleção. Nesta etapa, criam-se mecanismos para possibilitar o paralelismo computacional, ou seja, blocos de entidades (*clusters*) que possuem atributos similares. A ideia é dividir a carga de dados em diferentes blocos que possam ser processados em paralelo de forma distribuída. A etapa de processamento de blocos interage com a etapa de correspondência (*Entity Matching*) para refinar os blocos que foram criados na etapa anterior, removendo comparações repetidas (desnecessárias) e que envolvem entidades não correspondentes. Essas estratégias de blocagem aplicam-se a dados estruturados ou semiestruturados, com vistas a tratar a Variedade dos dados. Diferentes estratégias de blocagem, como *Standard Blocking*, *Sorted Neighborhood*, *MultiBlock* e *LIMES*, são utilizadas para agrupar entidades de forma eficaz, levando em consideração diferentes tipos de dados e requisitos de comparação (PAPADAKIS; NEJDL, 2011).

Na etapa de Correspondência, são executadas as comparações das entidades contidas nos conjuntos que foram refinados na etapa anterior, resultando em um grafo de similaridade, em que um par de entidades é interligado por uma aresta ponderada, indicando o nível de similaridade. Durante essa etapa, são aplicadas técnicas de comparação de atributos e cálculo de similaridade para determinar se duas entidades são correspondentes ou não. Os métodos de correspondências podem variar desde comparações simples de igualdade até algoritmos mais complexos que levam em consideração a semântica dos dados e a tolerância a erros.

Métodos de correspondências baseados em regras utilizam conjuntos de regras definidas manualmente para verificar correspondências e apresentam desafios em termos de escalabilidade e adaptação a grandes volumes de dados. As abordagens baseadas em similaridade de *strings*, como Distância de Levenshtein, Similaridade de Jaccard e Similaridade do Cosseno, podem comparar diretamente atributos textuais com pequenas variações ou erros tipográficos; no entanto, elas não capturam a semântica subjacente dos textos. Métodos probabilísticos, por outro lado, utilizam modelos estatísticos para calcular a probabilidade de correspondência entre registros, oferecendo flexibilidade e capacidade de lidar com incertezas. Por outro lado, os métodos baseados em aprendizagem de máquina, que incluem métodos supervisionados e não supervisionados, são capazes de capturar relações complexas entre atributos, além de oferecer alta precisão. Finalmente, abordagens híbridas combinam múltiplas técnicas para melhorar a precisão e a robustez do sistema.

A etapa de *Entity Clustering* constitui um dos estágios finais no processo de Resolução de Entidades. Ela envolve agrupar registros que foram identificados como referentes à mesma entidade, consolidando informações duplicadas e garantindo uma representação única e coerente de cada entidade no conjunto de dados.

2.5.2 Correspondência de Produtos

A Correspondência de Produtos, do inglês *Product Matching*, é um caso particular de RE, em que as entidades são produtos, ou seja, o objetivo é verificar se dois produtos de fontes de dados diferentes fazem referência ao mesmo objeto. Em um ambiente de comércio eletrônico dinâmico e em constante expansão, onde milhões de produtos são disponibilizados por diversos vendedores, a tarefa de correspondência de produtos torna-se essencial para melhorar a experiência do usuário, garantir a precisão nas recomendações de produtos e facilitar a comparação de preços.

O crescimento do comércio eletrônico ocorrido nos últimos anos tem trazido alguns desafios, como os elencados por Jovanovic e Bagheri (2016):

1. Dados de produtos não estruturados, dificultando automatização de processos;
2. Dificuldade em identificar produtos equivalentes em diferentes sites de compras;
3. Dificuldade em lidar com a grande quantidade de dados de produtos disponíveis;
4. Heterogeneidade de taxonomias das categorias de produtos;
5. Descrições de produtos incompletas, inconsistentes ou desatualizadas; e
6. Dificuldade em lidar com a grande quantidade de dados de transações de comércio eletrônico.

Para facilitar a identificação e o rastreamento de produtos no varejo, os fabricantes normalmente atribuem um identificador único universal. O GTIN (*Global Trade Item Number*) é amplamente utilizado para identificar produtos de forma inequívoca. Gerenciado pela GS1[3], o GTIN substituiu os antigos identificadores EAN (*European Article Number*) e UPC (*Universal Product Code*). A quantidade de dígitos do GTIN varia de acordo com a natureza do produto, sendo que a maioria dos produtos comerciais no varejo utiliza a codificação de 13 dígitos (EAN-13 ou GTIN-13). No entanto, o GTIN nem sempre está disponível em algumas plataformas. Em alguns casos, empresas ou varejistas utilizam identificadores internos, como o SKU (*Stock Keeping Unit*), para gerenciamento de estoque em seus sistemas, o que pode dificultar a integração com outras fontes de dados.

Diante dos desafios associados à correspondência de produtos e impulsionadas pelo avanço das pesquisas em NLP, muitas soluções inovadoras surgiram nos últimos anos (PETROVSKI et al., 2014; VANDIC et al., 2020; CHOI et al., 2020b; PEETERS et al., 2020; WILKE; RAHM, 2021; LI et al., 2020; KIM et al., 2022; FOXCROFT et al., 2021). A necessidade de pesquisas em correspondência de produtos deve-se principalmente à ausência de identificadores de produtos em muitas plataformas e catálogos, o que dificulta a integração dos dados. Algumas soluções de correspondência de produtos utilizam técnicas de aprendizagem de máquina, mas essas técnicas exigem conjuntos de dados anotados com informações precisas sobre os produtos, o que pode ser um desafio significativo.

No contexto do comércio eletrônico, dados de produtos podem ser extraídos de páginas da *Internet* utilizando as anotações do schema.org[4] e a *Web* Semântica, nas quais é possível obter, de forma automática, informações de produtos como títulos, descrições, imagens ou vídeos, detalhes técnicos, marcas, modelos, preços e GTINs (PEETERS et al., 2020). Quando a informação de identificadores únicos dos produtos está disponível, é possível criar de forma automática corpora anotados para serem utilizados em treinamento de modelos de aprendizagem de máquina (PEETERS et al., 2020).

Na literatura, existem diversos corpora disponíveis que servem como *benchmarks* para a avaliação de tarefas relacionadas à correspondência de produtos, como o WDC Products (PRIMPELI et al., 2019) e o VLDB2010 (KöPCKE et al., 2010). Esses *benchmarks* foram elaborados com dados reais de produtos extraídos da Web e de sites de comércio eletrônico. Entretanto, a disponibilidade de corpora contendo dados de produtos em outros idiomas é limitada na Internet, o que torna imprescindível a criação de novos corpora para soluções de correspondência de produtos em outros idiomas.

No Brasil, no contexto de transações comerciais B2B ou B2C, há a emissão de um

[3] <www.gs1br.org>
[4] <www.schema.org>

NCM	DESC	QT	VAL	TIP	DATE	MUNICIPALITY
30049099	#$ACETO.TRIA+SULFNEO+GRAM+NIST POM G LE	1	20	UN	2020-07-07	Nova Cruz
30049069	ZYXEM GTS 5MG20ML UCB Lote: 19I19 Vcto: 08/21	1	58.72	CX	2020-06-30	BELEM
30059090	ALGODAO HIDROFILO NEVOA 500GR	3000	8.35	RL	2020-04-23	MOSSORO
30049099	DICLOFENACO POTASSIO 50MG GEO GEOLAB	6000	0.07	CP	2020-05-14	Campina Grande
90183929	SCALP CANULA 21G C/DISP. SEG. WILTEX	15,000	0.4	UN	2020-04-22	MOSSORO

Legend: NCM—Product Category Identifier; DESC—Product Description, QT—Quantity Bought, VAL—Price, TIP—Product's Measure (e.g., unity, box or gallon), DATE—Purchase Date.

Figura 2.5 – Exemplos de dados em uma nota fiscal

Fonte: Extraído de Schulte et al. (2022).

documento denominado Nota Fiscal, que discrimina cada operação realizada, registrando a circulação de mercadorias ou a prestação de serviços entre as partes envolvidas. As informações contidas em uma nota fiscal necessitam de validação, pois podem apresentar informações incorretas, como categorias ou códigos de produtos errados, pontuações inadequadas para preços e erros ortográficos na descrição do produto (SCHULTE et al., 2022; LUCENA et al., 2022).

Os atributos relacionados aos produtos comercializados em uma nota fiscal incluem: GTIN/EAN, Descrição do Produto, NCM[5], quantidade e valor do item comercializado, bem como a unidade de medida do produto. A validação das informações dos produtos nas notas fiscais pode ser realizada por técnicas de correspondência de produtos, auxiliando na detecção de sonegação de impostos (SCHULTE et al., 2022; LUCENA et al., 2022). Na Figura 2.5 são apresentados exemplos de dados de produtos comercializados disponíveis em uma nota fiscal.

2.6 RECUPERAÇÃO DA INFORMAÇÃO

A Recuperação de Informação (RI) é uma área da computação voltada para a busca de informações não estruturadas em documentos, como textos, em grandes coleções de dados, com o objetivo de satisfazer necessidades específicas de informação (MANNING, 2009). Os sistemas de RI objetivam recuperar todos os documentos relevantes para uma busca, enquanto minimizam a recuperação de documentos irrelevantes (BAEZA-YATES; RIBEIRO-NETO, 2013).

Os Motores de Busca da Web representam os serviços de RI mais populares, utilizados para localizar uma ampla gama de informações. No entanto, a RI também pode ser aplicada em contextos específicos, como a busca de produtos no comércio eletrônico. A RI abrange uma diversidade de técnicas e algoritmos para indexar, representar, armazenar, recuperar e classificar informações. Para recuperar documentos relevantes de maneira

[5] A Nomenclatura Comum do Mercosul (NCM) é utilizada para determinar os tributos envolvidos nas operações de comércio exterior e na saída de produtos industrializados.

eficiente em buscas de usuários em grandes volumes de dados, é necessário representar os dados de forma adequada, muitas vezes por meio de representações numéricas. Os modelos clássicos de RI são categorizados em três principais grupos (MANNING, 2009):

- Modelo booleano: Baseado na teoria dos conjuntos e na álgebra booleana, este modelo foi amplamente utilizado no passado. No entanto, ele não permite o ranqueamento dos resultados de busca, fornecendo apenas uma correspondência binária (sim/não);
- Modelo vetorial: Baseado na álgebra vetorial, este modelo representa documentos e os termos das buscas como vetores. Ele considera a frequência dos termos para mensurar o grau de similaridade e realizar o ranqueamento dos resultados, permitindo uma classificação mais refinada dos documentos recuperados;
- Modelo probabilístico: Baseado na teoria probabilística, este modelo estima a relevância de um documento para uma busca através de cálculos de probabilidades. Isso possibilita o ranqueamento dos resultados, com os documentos de maior probabilidade de relevância aparecendo no topo da lista.

O modelo Best Match 25 (BM25) é um modelo probabilístico muito utilizado em motores de busca e ferramentas de pesquisa textual (ROBERTSON; ZARAGOZA, 2009; SCHUTH et al., 2014). O BM25 estima a relevância de um documento baseado nas distribuições dos termos das buscas nos documentos, considerando fatores como: a frequência do termo nos documentos e na própria busca; o comprimento do documento; e a frequência média do termo na coleção de documentos. A importância desses fatores para a obtenção do resultado final da busca pode ser ajustada através de parâmetros que controlam a pontuação final dos documentos recuperados. A Equação 1 apresenta a função de similaridade utilizada pelo BM25 (SCHUTH et al., 2014; QIN et al., 2010):

$$BM25(q,d) = \sum_{q_i : tf(q_i,d)>0} \frac{idf(q_i).tf(q_i,d).(k_1+1)}{tf(q_i,d) + k_1.(1-b+b.\frac{|d|}{avgdl})} . \frac{(k_3+1).qtf(q_i,q)}{k_3 + qtf(q_i,q)}, \quad (1)$$

onde:

- $idf(q_i)$ representa a frequência inversa do documento calculada por $idf(q_i) = \log\left(\frac{N-df(q_i)+0.5}{df(q_i)+0.5}\right)$, onde N é o número de documentos da coleção e $df(q_i)$ é o número de documentos contendo o termo q_i;
- $tf(q_i,d)$ é a frequência do termo, representando o número de vezes que o termo q_i ocorre no documento d;

- $qtf(q_i, q)$ é frequência do termo de busca, representando o número de vezes que o termo q_i ocorre na busca q;
- $\frac{|d|}{avgdl}$ é o comprimento do documento d, normalizado pelo comprimento médio dos documentos na coleção;
- k_1, b e k_3 são os parâmetros ajustáveis que possibilitam otimizar os resultados conforme objetivos definidos. Normalmente, k_1 é definido como um valor entre 1 e 3, b é definido com um valor aproximado a 0,8 e k3 é definido como 0. Se $k3 = 0$, então o algortimo não considera a frequência do termo da busca.

Diversas adaptações da equação do modelo BM25 foram propostas na literatura com o objetivo de aprimorar sua eficácia em diferentes tipos de coleções de documentos e cenários de busca, visando aumentar a precisão e a relevância dos resultados (GHAWI; PFEFFER, 2019; SCHUTH et al., 2014; QIN et al., 2010).

Diversas adaptações da equação do modelo BM25 foram propostas na literatura com o objetivo de aprimorar sua eficácia em diferentes tipos de coleções de documentos e cenários de busca (GHAWI; PFEFFER, 2019; SCHUTH et al., 2014; QIN et al., 2010; TROTMAN et al., 2014). O BM25F (ROBERTSON et al., 2004), por exemplo, incorpora pesos para diferentes campos dos documentos, permitindo uma busca mais refinada em coleções com múltiplos atributos. O BM25+ (LV; ZHAI, 2011b), por sua vez, ajusta a normalização para evitar penalizar indevidamente documentos curtos, enquanto o BM25L (LV; ZHAI, 2011c) corrige a normalização excessiva em documentos longos. O BM25T (LV; ZHAI, 2012) utiliza técnicas de expansão de termos para melhorar a recuperação de documentos relevantes em casos de vocabulário restrito. Já o BM25-Adaptive (LV; ZHAI, 2011a) adapta dinamicamente os parâmetros do modelo conforme as características da coleção ou da consulta.

2.7 MÉTRICAS DE AVALIAÇÃO PARA TAREFAS DE CLASSIFICAÇÃO E RECUPERAÇÃO DA INFORMAÇÃO

As métricas de avaliação são ferramentas essenciais para medir o desempenho de modelos computacionais. Além de permitirem a comparação de diversas abordagens, elas possibilitam a identificação de aspectos que necessitam de aprimoramento nos modelos avaliados. Essas métricas orientam o desenvolvimento e a otimização de soluções, garantindo que os modelos atendam aos requisitos de precisão e relevância em diversas aplicações.

Em problemas de classificação, as métricas comumente usadas para a avaliação dos modelos são acurácia, precisão, *recall* e *F1-score* (GOUTTE; GAUSSIER, 2005). A

acurácia representa a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões realizadas. A precisão indica a proporção de verdadeiros positivos dentro do conjunto das previsões positivas, ou seja, avalia quantas das previsões positivas são realmente positivas. O *recall*, também conhecido como sensibilidade, mede a proporção de verdadeiros positivos identificados em relação ao total de amostras que são realmente positivas. O *F1-score* é a média harmônica da precisão e do *recall*, que proporciona uma avaliação equilibrada do desempenho de um modelo, especialmente em conjuntos de dados desbalanceados.

As métricas acurácia, precisão, *recall* e *F1-score* são definidas, respectivamente, pelas Equações 2, 3, 4 e 5.

$$Accuracy = \frac{(TN + TP)}{(N + P)}, \tag{2}$$

$$Precision = \frac{TP}{(FP + TP)}, \tag{3}$$

$$Recall = \frac{TP}{(FN + TP)}, \tag{4}$$

$$F1_{score} = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)}, \tag{5}$$

onde:

- TP (verdadeiros positivos): é o número de instâncias classificadas de forma correta pelo modelo para a classe positiva;
- FP (falsos positivos): é o número de instâncias classificadas de forma errada pelo modelo para a classe positiva;
- TN (negativos verdadeiros) é o número de instâncias classificadas de forma correta pelo modelo para a classe negativa;
- FN (falsos negativos) é o número de instâncias classificadas de forma errada pelo modelo para a classe negativa.

A acurácia é uma métrica útil quando as classes estão equilibradas, ou seja, quando o número de amostras em cada classe é aproximadamente o mesmo. No entanto,

em conjuntos de dados desbalanceados, em que há uma disparidade significativa entre o número de amostras das diferentes classes, um modelo pode obter uma alta acurácia simplesmente classificando a maioria das amostras na classe majoritária. Neste caso, além de analisar todas as métricas individualmente por classe, é importante analisar as médias ponderadas, que consideram os pesos de cada classe por quantidade de instâncias.

No contexto da RI, as métricas são normalmente computadas considerando a relevância dos resultados recuperados por meio de uma busca realizada. Uma busca recupera uma lista de itens ordenada conforme critérios de relevância definidos. Por exemplo, um item relevante, no contexto deste trabalho, representa um produto que corresponde a um produto pesquisado no sistema de RI. A precisão é a fração de itens relevantes no total recuperado. Por outro lado, o *recall* é a fração dos itens relevantes recuperados em relação a todos os itens verdadeiramente relevantes.

Os resultados mais relevantes para um sistema de RI devem estar no topo da lista de documentos recuperados. Assim, surge a necessidade de considerar as métricas de avaliação sob a ótica dos *top-K* resultados, em que K representa a quantidade dos primeiros documentos retornados pelo sistema. Essa perspectiva permite uma análise mais granular do desempenho, concentrando-se nos resultados que têm maior probabilidade de serem visualizados e utilizados pelos usuários.

As métricas *Precision@K*, *Recall@K* e *F1-Score@K* são utilizadas para avaliar o desempenho de modelos de recuperação de informações, considerando os *top-K* como os resultados da lista recuperada. A métrica *Precision@K* mede a proporção de itens relevantes entre os *top-K* itens recuperados, enquanto a *Recall@K* avalia a proporção de itens relevantes recuperados em relação ao total de itens relevantes disponíveis. O *F1-Score@K* é a média harmônica de *Precision@K* e *Recall@K*, que proporciona uma visão equilibrada da precisão e completude dos resultados recuperados.

A métrica Mean Average Precision (mAP), definida pela Equação 6, avalia o desempenho de um sistema de recuperação em um conjunto de consultas (VOORHEES; HARMAN, 2005). Esta métrica oferece uma visão mais abrangente ao calcular a média para todas as consultas em uma coleção de teste. A mAP assume valores entre 0 e 1, em que os valores mais próximos de 1 indicam que o sistema é capaz de identificar e classificar os itens relevantes com alta precisão.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i, \tag{6}$$

onde N é o total de buscas, e AP_i é a média da precisão de cada busca i realizada.

As métricas baseadas em precisão e *recall*, embora úteis para a avaliação de RI, apresentam a limitação de não capturar a importância da ordem dos itens recuperados. Além disso, essas métricas são sensíveis à quantidade de itens relevantes por consulta. Para superar essas limitações, métricas como *Mean Reciprocal Rank* (MRR) e *Normalized Discounted Cumulative Gain* (nDCG) (JäRVELIN; KEKäLäINEN, 2002) são frequentemente utilizadas na avaliação da ordem dos elementos relevantes recuperados em sistemas de RI (LILLIS, 2020; LI, 2015).

A métrica MRR, definida pela Equação 7, assume valores entre 0 e 1 e avalia a relevância do primeiro elemento relevante recuperado para cada consulta. O MRR concentra-se na precisão do sistema em identificar e classificar o item mais relevante no topo da lista de resultados, independentemente da posição dos demais itens relevantes.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}, \tag{7}$$

onde Q é a quantidade de buscas e $rank_i$ é a posição do primeiro elemento relevante da *i-ésima* busca.

Em contraste com a métrica MRR, que se concentra apenas no primeiro item relevante, a métrica $NDCG$, definida pela Equação 8, considera a relevância de todos os itens recuperados e sua posição na lista de resultados. Isso significa que o NDCG atribui maior peso aos itens mais relevantes que estão posicionados no topo da lista, enquanto itens relevantes em posições inferiores recebem um peso menor. Essa característica torna o $NDCG$ uma métrica mais completa para avaliar o desempenho geral de sistemas de RI, pois leva em consideração a relevância e a ordenação de todos os itens recuperados.

$$NDCG@K = \frac{DCG@K}{IDCG@K} = \frac{\sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}}{\sum_{i=1}^{rel@k} \frac{2^{rel_i}-1}{log_2(i+1)}}, \tag{8}$$

onde k representa a quantidade de elementos da lista do *ranking* a serem avaliados, rel_i é o ganho (*score*) de relevância do resultado na posição i e $rel@k$ é a lista de elementos relevantes no *ranking* ideal.

2.8 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo abordou os principais conceitos que fundamentam esta tese e que são necessários para a compreensão da abordagem de correspondência de produtos proposta.

Foram discutidos conceitos de aprendizagem de máquina, incluindo aprendizado por transferência e aprendizado por cruzamento de idiomas, além de modelos de linguagem pré-treinados. Também foram abordadas as temáticas relacionadas ao processamento de linguagem natural e à recuperação da informação, além dos conceitos de Resolução de Entidades e Correspondência de Produtos. No próximo capítulo, serão discutidos os trabalhos relacionados a esta tese.

3 TRABALHOS RELACIONADOS

Neste capítulo, analisam-se os trabalhos relacionados à esta pesquisa. Inicialmente, na seção 3.1, são apresentadas as técnicas e *frameworks* utilizados em RE. Os principais trabalhos na área de Correspondência de Produtos são apresentados na seção 3.2. Finalmente, na seção 3.3, são discutidos trabalhos que visam a resolver problemas relacionados às descrições de produtos em notas fiscais.

3.1 TÉCNICAS E *FRAMEWORKS* PARA RESOLUÇÃO DE ENTIDADES

A RE tem sido amplamente estudada na literatura, principalmente pelo interesse comercial, com diversas abordagens já propostas (CHRISTEN, 2008a; FIRMANI et al., 2016; BARLAUG; GULLA, 2021). Os problemas de RE, geralmente, passam por duas etapas (CHRISTOPHIDES et al., 2015): 1) Blocagem (*Blocking*) (EFTHYMIOU et al., 2017), que é a etapa responsável por minimizar o número de comparações necessárias; e 2) correspondência (KöPCKE et al., 2010), que é a etapa que efetivamente decide se um determinado par de entidades representa a mesma entidade.

Existem, pelo menos, duas abordagens principais para tratar os problemas de correspondência (KöPCKE et al., 2010): as abordagens não baseadas em aprendizagem de máquina, que utilizam análises léxicas e medidas de similaridade para compor a função de correspondência, e as abordagens baseadas em aprendizagem de máquina, que fornecem diferentes tipos de medidas de similaridade através das características aprendidas pelos modelos de classificação. Esses modelos são responsáveis por identificar padrões e correspondências entre objetos (CHRISTEN, 2012).

Os problemas de RE geralmente envolvem registros textuais, fazendo com que as técnicas de RE evoluam junto com os avanços do NLP (FOXCROFT et al., 2021). A representação textual tem sido um foco central nessas pesquisas, desde modelos tradicionais como o *Bag-of-Words* com ponderação TF-IDF até representações mais sofisticadas como *embeddings*, que capturam relações semânticas entre palavras. Recentemente, com o advento da arquitetura *Transformers* (VASWANI et al., 2017), surgiram avanços ainda mais expressivos, especialmente no uso de aprendizado profundo e modelos de linguagem pré-treinados (BARLAUG; GULLA, 2021).

Nos últimos anos, foram propostos vários *frameworks* para resolver problemas de RE. Em Christophides et al.(2020), Papadakis et al (2021) e Barlaug e Gulla (2021) são sintetizadas as principais técnicas e *frameworks* utilizados em RE. Aqui, serão destacados alguns *frameworks* utilizados em trabalhos específicos de Correspondência de Produtos,

tais como: Dedupe (BILENKO; MOONEY, 2003) , Febrl (CHRISTEN, 2008b), Magellan (KONDA, 2018), DeepER (EBRAHEEM et al., 2017), DeepMatcher (MUDGAL et al., 2018) e Ditto (LI et al., 2020).

Febrl (Freely Extensible Biomedical Record Linkage) (CHRISTEN, 2008b) é uma ferramenta de código aberto para RE, que oferece uma interface gráfica para o usuário que implementa vários mecanismos nas etapas de blocagem (indexação) e classificação. Inicialmente projetada para registros biomédicos, é extensível para outros tipos de entidades. A ferramenta implementa 26 funções de similaridade para tratar vários tipos de dados. Os resultados das funções aplicadas a pares de entidades permitem gerar vetores que serão utilizados na etapa de classificação, podendo empregar técnicas supervisionadas e não supervisionadas.

Dedupe (BILENKO; MOONEY, 2003) conhecido também por MARLIN (*Multiply Adaptive Record Linkage with Induction*) é um *framework* que incorpora técnicas de blocagem e classificação de registros. A blocagem é baseada na função de distância de Jaccard. Para a classificação, Dedupe utiliza métricas de similaridade de strings adaptativas para identificar registros duplicados. Os autores definem duas métricas de similaridade textual: uma baseada no algoritmo Expectation-Maximization (EM) para estimar os parâmetros de um modelo baseado na distância de edição (similaridade entre duas strings) e outra que aplica o algoritmo de classificação SVM à representação vetorial dos atributos. As métricas de similaridade são então codificadas em um vetor de características utilizado para treinar um classificador SVM, responsável por identificar os registros duplicados.

Magellan (KONDA, 2018) é um sistema abrangente aplicável a diversos cenários de RE. Inclui processos não somente para blocagem e correspondência, mas também recursos para limpeza de dados, extração e visualização de dados, entre outros recursos para análise de dados. Na blocagem, oferece técnicas baseadas em atributos e sobreposição, regras definidas pelo usuário e técnicas de aprendizagem de máquina. Na etapa de correspondência, Magellan extrai características que descrevem as diferenças e similaridades entre as tuplas (entidades), incluindo similaridade de *strings*, *tokens*, n-gramas e fonética. A etapa de correspondência é realizada por meio de algoritmos como árvores de decisão, SVM e regressão logística.

DeepER (EBRAHEEM et al., 2017) explora técnicas de *deep learning* para criar representações vetoriais das entidades para capturar similaridades entre as tuplas de entidades. Para as representações vetoriais, o *framework* possibilita utilizar *Word Embeddings* pré-treinados como word2vec, GloVe, ou fastText, ou a criação de representações vetoriais utilizando RNNs, que consideram o contexto sequencial das palavras. Para verificar a correspondência entre entidades, calcula-se a similaridade entre os vetores, utilizando, por

exemplo, o cosseno do ângulo formado pelos pares de vetores que representam os textos comparados. Os autores indicam que a blocagem pode ser realizada utilizando técnicas de busca aproximada de vizinho mais próximo (*Approximate Nearest Neighbor-ANN*) em um espaço de similaridade, a partir da indexação dos vetores (*Locality Sensitive Hashing-LSH*).

DeepMatcher (MUDGAL et al., 2018) é uma extensão do DeepER, que utiliza as representações vetoriais para analisar similaridades dos atributos e os valores dos atributos a partir de um classificador que aprende através dessas similaridades se as entidades são correspondentes. Adicionalmente, o *framework* possibilita configurar a função de similaridade e os *embeddings*, seja em nível de caracteres ou de palavras.

Ditto (LI et al., 2020) é um sistema de RE baseado em modelos de linguagem pré-treinados. A tarefa de RE passa a ser uma tarefa de classificação de sequências de pares. A ideia aqui é utilizar a capacidade de MLPTs de compreender o contexto das palavras para resolver problemas de valores ausentes ou corrompidos. Ditto possibilita três otimizações: injeção explícita de exemplos de treinamento do domínio específico; aumento de dados de treinamento; e sumarização de textos longos para manter as informações essenciais das entidades, principalmente devido às limitações de quantidades de tokens dos MLPTs. A etapa de blocagem é realizada de forma semelhante à ideia proposta em DeepER, utilizando técnicas de busca aproximada do vizinho mais próximo. Para isso, ajustam o MLPT Sentence-BERT com os dados rotulados. Comparações realizadas em outros estudos indicam que o framework Ditto representa o estado da arte em tarefas de RE (PEETERS; BIZER, 2022; BARLAUG; GULLA, 2021; PAGANELLI et al., 2022; MOZDZONEK et al., 2022)

O Quadro 3.1 apresenta uma comparação dos principais *frameworks* utilizados em trabalhos de Correspondência de Produtos, sintetizando as técnicas implementadas em cada *framework*, incluindo blocagem, características (*features*) para a representação das entidades e técnicas de correspondência. *Frameworks* mais tradicionais, como *Febrl* e *Magellan*, utilizam uma variedade de técnicas de blocagem, diferentes métricas de similaridade textual e algoritmos de classificação, permitindo identificar e realizar correspondências entre entidades. Em contrapartida, *frameworks* mais recentes, como DeepER, DeepMatcher e Ditto, exploram técnicas de *deep learning* para a verificação de correspondências, utilizando representações vetoriais (*embeddings*) que capturam relações semânticas entre as entidades. Além disso, essas abordagens recentes não implementam técnicas específicas de blocagem, sugerindo o uso de busca vetorial como alternativa.

Quadro 3.1 – Resumo dos principais *frameworks* utilizados em trabalhos de Correspondência de Produtos

Framework	**Téc. de Blocagem**	**Características**	**Téc. Correspondência**
Febrl (2008b)	Standard Blocking, Sorted Neighborhood, StringMap, Sorted Blocks	funções de similaridade	SVM, Naive Bayes,...
Dedupe (2003)	Standard Blocking	EM e similaridade por um classificador SVM	SVM
Magellan (2018)	Standard Blocking, Sorted Neighborhood e regras do usuário	Funções para strings, tokens, n-gramas e fonéticas	SVM, árvores de decisão, regressão logística, ...
DeepER (2017)	ANN*	Embeddings pré-treinados (word2vec, GloVe ou fastText)	Distância entre os embeddings
DeepMatcher (2018)	ANN*	Embeddings pré-treinados ou gerados a partir de uma RNN	Rede Neural (Multi-Layer Perceptron)
Ditto (2020)	ANN*	Embeddings Contextualizados (BERT)	BERT, DistilBERT, RoBERTa

*Autores sugerem a indexação vetorial para técnicas de busca aproximada de vizinho mais próximo (ANN).

3.2 TRABALHOS ESPECÍFICOS DE PRODUCT MATCHING

Nesta seção, serão apresentados os principais trabalhos que objetivam resolver problemas de REs relacionados a produtos. De forma geral, os trabalhos tratam o problema de correspondência de produtos como um problema de classificação, em que se deseja verificar se dois produtos, com seus atributos e valores, correspondem ao mesmo produto. Ou seja, o termo *Product Matching* frequentemente é utilizado como um sinônimo da etapa de correspondência dos *frameworks* REs.

Em Ristoski et al. (2018), os autores propõem uma abordagem multimodal para a correspondência e categorização de produtos, combinando características visuais e textuais. As características visuais de imagens dos produtos são extraídas através de uma rede CNN pré-treinada. Para as características textuais, utilizam tanto modelos estatísticos, como o CRF (Conditional Random Fields), quanto técnicas baseadas em dicionários. Essas características são então concatenadas em um único vetor representativo do produto. A similaridade entre os produtos é calculada utilizando métricas adequadas ao tipo de dado, como a distância do cosseno para dados contínuos (imagens e textos longos) e a similaridade de Jaccard para dados categóricos (palavras). Os autores empregam diversos algoritmos de aprendizagem de máquina, como *Naive Bayes, SVM, Random Forest* e *KNN*, para classificar os produtos com base nessas representações vetoriais. Nos experimentos, testes foram realizados para avaliar a importância das características incluídas nos vetores, e os resultados indicaram que a complementaridade das características visuais e textuais, com a combinação das duas, resultou em um desempenho superior na tarefa de classificação.

Barbosa (2019) utiliza diferentes representações de textos (*embeddings e Bag of Words*) para a tarefa de RE em descrições de produtos utilizando técnicas de *deep learning*. A utilização de várias representações dos textos possibilita a captura de padrões de

distâncias entre as diversas representações, em que um classificador binário é aplicado para resolver o problema de RE. Neste trabalho, o autor avalia que sua abordagem apresenta melhores resultados quando comparada com outros baselines, como a distância do cosseno, e os *frameworks* como Febrl e Dedupe. Os *frameworks* Febrl e Dedupe, em particular, apresentaram os piores resultados com os dados de produtos.

Kertkeidkachorn e Ichise (2020) propuseram o PMap, um sistema de correspondência de títulos de produtos que combina múltiplos MLPTs baseados em BERT, ajustados especificamente para essa tarefa específica. A combinação desses modelos visou explorar a complementaridade de suas representações, buscando um desempenho superior. Os resultados obtidos indicam que o PMap supera os modelos individuais, embora a diferença em relação ao modelo Roberta-large seja relativamente pequena, conforme evidenciado pelo F1-score.

Tracz et al. (2020) investigaram o impacto de diferentes estratégias de amostragem na correspondência de produtos com modelos BERT. Eles compararam três abordagens: amostragem aleatória, por categoria e de pares difíceis. A amostragem por categoria ajuda a discriminar produtos semelhantes, enquanto a de pares difíceis desafia o modelo a distinguir entre itens visualmente ou semanticamente próximos. Ao variar a composição dos conjuntos de treinamento, os autores demonstraram que a escolha da estratégia de amostragem exerce um impacto significativo no desempenho dos modelos, com a amostragem de pares difíceis mostrando-se particularmente eficaz em melhorar a capacidade de generalização dos modelos.

Peeters et al. (2020) realizaram ajustes finos em modelos baseados em BERT, comparando os resultados obtidos com os *frameworks* Magellan e Deepmatcher. Semelhante a Tracz et al. (2020), os autores adotaram uma abordagem para formar o corpus de treinamento com o objetivo de melhorar o desempenho dos modelos. Eles combinaram critérios de similaridade e aleatoriedade para gerar pares de descrições de produtos. Para a estratégia de similaridade, utilizaram a distância do cosseno das cinco primeiras palavras dos títulos dos produtos (Bag of Words), gerando 50% dos pares de produtos mais semelhantes (classes positiva e negativa), enquanto os outros 50% dos pares foram gerados aleatoriamente. Essa estratégia foi aplicada com os dados do *benchmark* WDC Products, alcançando o melhor desempenho de classificação, com um F1-Score de 96,53%.

Łukasik et al. (2021) desenvolveram um classificador XGBoost para verificar a correspondência entre produtos, utilizando atributos como títulos, tipo, marca e preço. O classificador emprega um vetor de características que inclui: a similaridade de Jaccard entre os títulos dos produtos, a distância cosseno entre os embeddings (fastText) dos tipos de produtos, a similaridade das marcas medida pela distância de Damerau-Levenshtein, a

diferença relativa de preços e a Análise de Componentes Principais (PCA) dos tipos de produtos. A avaliação do classificador foi realizada com dados provenientes do comércio eletrônico.

Romaldo et al. (2021) investigaram a similaridade entre títulos de produtos utilizando tanto MLPTs quanto *Word Embeddings*. O estudo envolveu cerca de 7,5 milhões de títulos de produtos extraídos de um *marketplace* de comércio eletrônico brasileiro, a partir dos quais foram gerados cinco conjuntos de *Word Embeddings* específicos para esse domínio. A similaridade entre os títulos foi calculada com base na distância do cosseno entre os *embeddings*. Os autores então compararam os resultados dos embeddings específicos de domínio com modelos de propósito geral, como Word2Vec, FastText e GloVe, além de MLPTs, como BERT e BERTimbau. Apesar de não terem realizado o ajuste fino nos MLPTs, os resultados mostraram que, embora os *embeddings* específicos de domínio tenham obtido bom desempenho, o modelo pré-treinado multilíngue BERT destacou-se como a abordagem mais eficaz para avaliar a similaridade de títulos de produtos.

Em Peeters e Bizer (2022), os autores avaliaram o uso de CLL em correspondências de produtos para treinar modelos para o idioma alemão. A estratégia foi utilizar dados de produtos do idioma inglês, adicionando uma fração menor de dados do idioma alemão. Os autores demonstraram que essa estratégia melhora o desempenho dos MPLTs avaliados. Em contrapartida, usando um classificador SVM não houve melhorias nos resultados. O trabalho avaliou as estratégias *zero-shot* e *joint learning* de CLL.

Um resumo dos principais trabalhos relacionados a técnicas de Correspondência de Produtos é apresentado no Quadro 3.2, destacando as representações utilizadas, os classificadores empregados e os resultados mais relevantes de cada estudo. Observa-se uma evolução das abordagens, desde o uso de métodos tradicionais de aprendizagem de máquina até a incorporação mais recente de *deep learning* e Modelos de Linguagem Pré-Treinados, com ênfase na utilização de técnicas avançadas de representação de dados e de estratégias específicas de amostragem para o treinamento de modelos. Todos os estudos analisados utilizaram dados de produtos do comércio eletrônico e focaram exclusivamente na tarefa de classificar a correspondência entre pares de produtos, sem abordar técnicas de busca para localizar corretamente essas correspondências. Destaca-se, ainda, o trabalho de Peeters e Bizer (2022), que realiza uma investigação inicial de duas estratégias de CLL para o treinamento de modelos com dados em outros idiomas.

3.3 CLASSIFICAÇÃO DE PRODUTOS EM NOTAS FISCAIS

Na literatura, são escassos os trabalhos que buscam resolver problemas relacionados às descrições de produtos em notas fiscais, especialmente no cenário brasileiro. A

Quadro 3.2 – Resumo dos principais trabalhos de Correspondência de Produtos

Trabalho	Representação	Classificador	Resultados Destacados
Ristoski et al. (2018)	CNN para imagens, CRF e técnicas baseadas em dicionários para texto	SVM, KNN, Random Forest	Combinação de características visuais e textuais
Łukasik et al. (2021)	Embeddings (fastText), PCA	XGBoost	Utilização de múltiplos atributos
Barbosa (2019)	Embeddings e Bag-of-Words	Deep learning (classificador binário)	Superioridade de técnicas com deep learning
Kertkeidkachorn e Ichise (2020)	BERT	Múltiplos MLPTs	Combinação de múltiplos MLPTs (Ensemble)
Tracz et al. (2020)	BERT	BERT	Eficácia da amostragem de pares difíceis
Peeters et al. (2020)	BERT	BERT	Melhor desempenho com combinação de critérios de similaridade e aleatoriedade
Romaldo et al. (2021)	Word Embeddings (específicos e gerais) e BERT	Dist. dos Vetores BERT, BERTimbau	Superioridade de BERT multilíngue
Peeters e Bizer (2022)	BERT	BERT	Melhora do desempenho com CLL (Joint Learning)

classificação desses produtos é particularmente desafiadora devido à heterogeneidade das descrições, que muitas vezes são curtas, ambíguas e inconsistentes. A falta de padronização e a variabilidade nos formatos adotados por diferentes empresas também agravam o problema. Nesse contexto, técnicas de NLP e aprendizagem de máquina têm sido exploradas como soluções promissoras para aumentar a precisão na identificação e categorização de produtos, contribuindo para auditorias fiscais e detecção de fraudes. A seguir, são discutidos alguns dos principais trabalhos que investigam essas abordagens para a classificação de produtos em notas fiscais.

Santana et al. (2023) compararam técnicas de NLP para a verificação de correspondências em títulos de produtos de notas fiscais. Os autores propuseram uma metodologia para a construção de um corpus de treinamento, combinando informações do código GTIN com uma heurística baseada na similaridade semântica das descrições dos produtos. Em seguida, avaliaram o desempenho de modelos de classificação tradicionais, como Naive Bayes, SVM e XGBoost, além de um modelo de aprendizado profundo baseado em BERT. Os resultados obtidos indicam que o modelo BERT superou os demais, demonstrando a eficácia das representações de linguagem contextualizadas para a tarefa de Correspondência de Produtos.

Schulte et al. (2022) propuseram o ELINAC, uma ferramenta para auxiliar na auditoria de notas fiscais, baseada no agrupamento de descrições textuais de produtos. A técnica empregada utiliza redes neurais *autoencoders* para aprender representações latentes compactas das descrições dos produtos. Essas representações latentes são obtidas através de um processo de codificação e decodificação. Nesse processo, a rede neural comprime a entrada original em um espaço de representação de menor dimensão, capturando as características mais relevantes dos dados. Em seguida, a rede tenta reconstruir a entrada

original a partir dessa representação comprimida. O erro de reconstrução, calculado como a diferença entre a entrada original e a reconstruída, serve como uma medida da qualidade da representação aprendida. Quanto menor o erro, mais similar a representação latente é à entrada original. Além do mais, produtos com descrições semelhantes tenderão a gerar erros de reconstrução similares, permitindo assim a formação de grupos de produtos com características comuns. Assim, esse agrupamento de produtos permite identificar fraudes e inconsistências nas notas fiscais.

Em Kieckbusch et al. (2021), os autores propuseram o SCAN-NF, um classificador desenvolvido para produtos de notas fiscais, utilizando as descrições dos produtos e o código NCM. O sistema é baseado em redes neurais convolucionais (CNN) e tem como objetivo auxiliar os auditores fiscais na identificação de inconsistências nos cadastros de produtos, visando à detecção de possíveis fraudes no sistema de tributação, através do uso inadequado do código NCM. A proposta do trabalho é categorizar adequadamente os produtos de notas fiscais nos respectivos NCMs, possibilitando a cobrança adequada dos tributos. O SCAN-NF apresenta duas arquiteturas: um modelo multiclasse e um modelo *ensemble* composto por classificadores binários especializados em categorias específicas do NCM. Os resultados demonstraram que o modelo *ensemble*, apesar de apresentar maior precisão, obteve um *recall* ligeiramente inferior em comparação ao modelo multiclasse. Ao comparar as CNNs com um classificador SVM utilizando representações TF-IDF, os autores concluíram que as CNNs superaram significativamente o SVM, evidenciando a superioridade das CNNs na tarefa de classificação de produtos em notas fiscais.

Lima et al. (2022) propõem a utilização de modelos BERT para automatizar a classificação de códigos NCM (Nomenclatura Comum do Mercosul) na importação de produtos no Brasil. A identificação do código NCM é necessária para a arrecadação precisa de impostos de importação, sendo um processo complexo devido à grande quantidade de códigos envolvidos (mais de 10.000 códigos) e à necessidade de uma descrição detalhada dos produtos. Inconsistências na classificação podem levar a penalidades financeiras e implicações legais. Os autores focaram nos produtos fotográficos e cinematográficos (Capítulo 90 do NCM), realizando o ajuste fino de modelos BERT multilíngues e em português. Os resultados indicaram que o modelo BERTimbau obteve o melhor desempenho na tarefa de classificação.

O Quadro 3.3 apresenta uma síntese dos principais trabalhos relacionados à classificação de produtos em notas fiscais no Brasil, destacando as técnicas utilizadas e os resultados alcançados. Observa-se que apenas o trabalho de Santana et al. (2023) abordou o problema de correspondência entre produtos de notas fiscais, explorando tanto técnicas tradicionais de aprendizado quanto modelos baseados em BERT. Os demais trabalhos tratam da classificação de produtos com foco na categorização de NCM. Todos eles aplicam

técnicas avançadas de aprendizado profundo, como BERT e CNNs, que demonstram uma boa capacidade de lidar com a complexidade das descrições dos produtos.

Quadro 3.3 – Resumo dos principais trabalhos com abordagens para a classificação de produtos em notas fiscais

Trabalho	Objetivo	Técnicas	Resultados Destacados
Santana et al. (2023)	Classificação de correspondências em produtos	Naive Bayes, SVM, XGBoost, BERT	Superioridade do modelo BERT
Schulte et al. (2022)	Classificação de produtos pelo NCM	Autoencoders	Clusterização de produtos mais rápida que métodos como DBSCAN, K-means e hierárquico.
Kieckbusch et al. (2021)	Classificação de produtos pelo NCM	CNN, SVM	O modelo ensemble (MLP) apresentou maior precisão
Lima et al. (2022)	Classificação de produtos pelo NCM	BERT e BERTimbau	Modelo BERTimbau apresentou um desempenho superior ao BERT multilíngue

3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo, foram discutidos os principais trabalhos relacionados à proposta desta tese, discorrendo sobre as várias técnicas utilizadas para resolver problemas de RE. O STEPMatch, proposto nesta tese, tem como objetivo corresponder produtos com base em descrições curtas, como as encontradas em notas fiscais. A abordagem parte da premissa de que há um conjunto de descrições de produtos vinculadas aos seus identificadores (GTIN/EAN), buscando corrigir associações incorretas entre descrições e identificadores, melhorando a qualidade dos dados com informações mais consistentes. O diferencial do STEPMatch em relação aos trabalhos analisados está na proposta de uma abordagem completa de RE específica para contextos como o de produtos de notas fiscais - os processos envolvidos abrangem desde a blocagem e verificação de correspondências até a busca de correspondências, utilizando Modelos de Linguagem Pré-Treinados (MLPTs). Além disso, na etapa de verificação de correspondências, a abordagem proposta se diferencia do trabalho de Peeters e Bizer (2022) ao explorar várias estratégias de CLL, comparando o desempenho de diversos MLPTs no treinamento de classificadores utilizando dados de produtos em inglês. Essas estratégias analisadas permitem aprimorar a tarefa de classificação.

4 STEPMATCH: UMA ABORDAGEM PARA IDENTIFICAÇÃO DE CORRESPONDÊNCIAS ENTRE PRODUTOS

Este capítulo apresenta a abordagem proposta nesta tese para a correspondência entre produtos. Inicialmente, na seção 4.1, é realizada uma formalização do problema a ser resolvido. A seção 4.2 apresenta o STEPMatch - *aa* (Correspondência de Produtos com Textos Curtos) proposto, detalhando os principais componentes e métodos envolvidos para resolver o problema de correspondência de produtos. Na seção 4.3, são apresentadas as técnicas de aprendizagem de máquina utilizadas na solução do problema. As métricas utilizadas na avaliação das técnicas são descritas na seção 4.4. Por fim, a seção 4.5 apresenta as considerações finais do capítulo.

4.1 DEFINIÇÃO DO PROBLEMA

Uma entidade P de uma fonte de dados F pode ser representada como um conjunto de pares formados pelo nome do atributo e o seu respectivo valor:

$$P^F = \{(a_1, v_1), ..., (a_n, v_n)\},$$

onde a_i e v_i representam respectivamente um atributo e um valor do atributo da entidade, com $1 \leq i \leq n$, sendo n a quantidade de atributos da entidade.

Seja $P'^{F'} = \{(a'_1, v'_1), ..., (a'_m, v'_m)\}$ uma entidade de outra fonte de dados F', com $F \neq F'$, onde m é a quantidade de atributos da entidade P', então, o problema de RE é verificar se há correspondências entre os atributos e valores das entidades P^F e $P'^{F'}$ referindo-se a essas mesmas entidades, mesmo que os atributos e valores estejam descritos de formas distintas e $m \neq n$.

Assim, $P^F = P'^{F'}$, se e somente se, os valores de $a_i \in P^F$ carregam a mesma informação semântica de $a'_j \in P'^{F'}$.

O objetivo da RE é encontrar a maior combinação possível da relação $R \subseteq F \times F'$ em que P^F e $P'^{F'}$ referenciam a mesma entidade para todas as entidades com (P^F e $P'^{F'}$) $\in R$. Ou seja, deseja-se encontrar todos os pares de registros das fontes de dados que fazem referência à mesma entidade.

Em particular, neste trabalho, foi experimentado o caso de produtos que contêm os atributos GTIN/EAN e título. Nesse caso, assume-se que $n = m = 2$. A escolha desses

dois atributos permite avaliar a eficácia de técnicas de *matching* em cenários com dados restritos, além de generalizar as contribuições do trabalho para diferentes cenários práticos que compartilhem as mesmas limitações de dados, tais como catálogos comerciais ou bases de notas fiscais, em que descrições curtas e inconsistentes apresentam desafios para a correspondência de produtos.

4.2 ABORDAGEM PARA IDENTIFICAÇÃO DE CORRESPONDÊNCIAS ENTRE PRODUTOS

Nesta seção, a abordagem para identificar correspondências entre produtos é detalhada, apresentando a visão geral do STEPMatch, com as descrições dos algoritmos e as principais técnicas utilizadas.

4.2.1 Visão da Geral do STEPMatch

Um cenário de aplicação para o problema definido na seção anterior (seção 4.1) envolve registros de produtos com descrições textuais curtas e sem padronização, podendo apresentar muitos ruídos e inconsistências nos dados, como é o caso de notas fiscais. Nesse cenário, um mesmo identificador de produto pode estar associado a várias descrições diferentes do mesmo item, além de ocorrerem erros de associação entre códigos de produto e descrições. Deseja-se associar adequadamente as descrições dos produtos aos seus respectivos identificadores, resolvendo problemas de inconsistência nos registros. Isso inclui identificar produtos não correspondentes que compartilham os mesmos identificadores e localizar os identificadores mais adequados para produtos com cadastros inconsistentes. Esta tese propõe o *STEPMatch - Short Text Product Matching*, uma solução para problemas de inconsistência de dados semelhantes a esse cenário.

A Figura 4.1 apresenta uma visão geral do *STEPMatch*. Considerando o conjunto de todos os produtos $P = (p_1, p_2, \ldots, p_n)$, provenientes de diversas fontes de dados, o processamento do *STEPMatch* inicia-se na Etapa 1, com a realização de um agrupamento inicial dos produtos. Os produtos com valores de atributos semelhantes são agrupados em $G = (g_1, g_2, \ldots, g_m)$. Cada grupo $g_i \in G$ contém um subconjunto de produtos semelhantes, ou seja, $g_i = \{p_j \mid p_j \in P\}$, onde $g_i \subset P$. O blocking é fundamental nesta etapa, pois permite reduzir significativamente o espaço de busca ao agrupar produtos com valores de atributos semelhantes, criando subconjuntos mais homogêneos. Dessa forma, produtos que são mais propensos a serem correspondentes são processados juntos, aumentando a eficiência e a precisão da correspondência. Na Etapa 2, os grupos de produtos $g_i \in G$ são processados e a verificação de correspondência é feita internamente entre os produtos de cada grupo, resultando em grupos correspondentes (G_{Matchs}) e grupos não correspondentes (G_{noMatchs}). Esses grupos são, então, passados para a Etapa 3, na qual se busca identificar

as correspondências restantes dos produtos que não foram encontradas na Etapa 2. Essa etapa é crucial para explorar de forma abrangente as possíveis correspondências, com ênfase nos produtos em G_{noMatchs}.

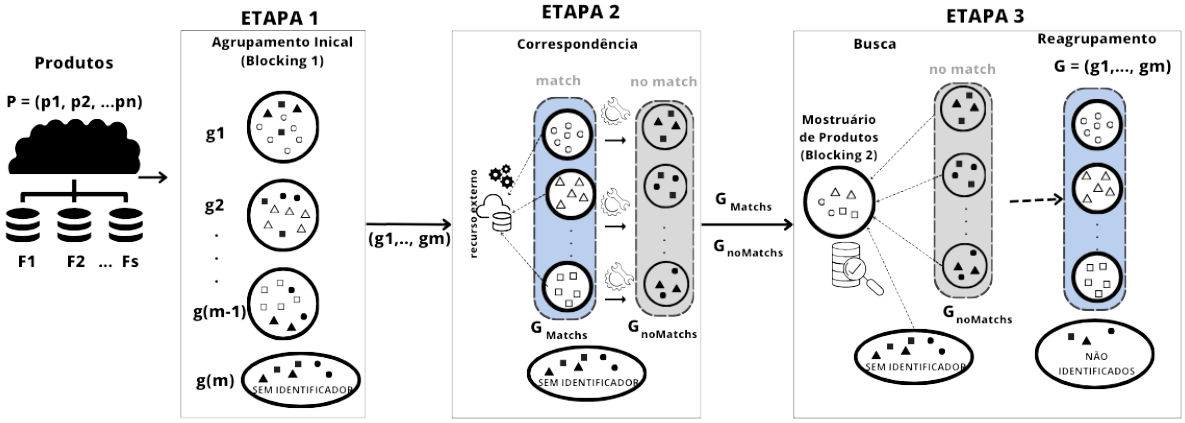

Figura 4.1 – Visão Geral do STEPMatch

Funções de similaridade são empregadas para definir os grupos de produtos nas diferentes etapas do *STEPMatch*. Essas funções variam em comportamento e tratamento de dados conforme a etapa e são utilizadas para analisar a similaridade entre pares de produtos. Em geral, a similaridade entre dois produtos quaisquer, p_i e p_j, é determinada por uma função de similaridade $F_{Sim}(p_i, p_j) \geq \theta$, onde θ representa o limiar de similaridade. As funções de similaridade utilizadas neste trabalho são detalhadas nas seções 4.2.2 e 4.2.3.

O Algoritmo 1, denominado Identificador de Correspondências, descreve as operações realizadas em cada uma das etapas do *STEPMatch*. Esse algoritmo é o núcleo do *STEPMatch* e seu objetivo principal é identificar e agrupar produtos correspondentes a partir da entrada de produtos oriundos de diversas fontes de dados. As descrições de cada etapa serão complementadas nas próximas seções com detalhes específicos de implementação e funcionamento. De forma geral, o algoritmo apresenta as seguintes tarefas:

1. agrupamento inicial de produtos (Etapa 1): linhas 1-8;
2. verificação de correspondências das descrições dos produtos (Etapa 2): linha 9;
3. definição de mostruário de produtos: linha 10;
4. busca de correspondências para os produtos (Etapa 3); e
5. unificação dos grupos de produtos correspondentes (linha 13).

Algorithm 1: ICProdutos: Identificador de Correspondências

Entrada : $P = (p_1, p_2, ...p_n)$; /*conjuntos de todos os produtos */
Saída : $G = (g_1, g_2, ..., g_m)$; /*grupos de produtos agrupados pelo ID */

$G \leftarrow \emptyset$;
$P_{unknown} \leftarrow \emptyset$;
foreach p_i *in* P **do**
  if $p_i.id$ *is null* **then**
    $P_{unknown}.add(p_i)$;
  else
    $G[p_i.id].add(p_i)$; /*Agrupamento dos produtos pelo ID */
end
$G_{Matchs}, G_{noMatchs} \leftarrow AgruparProdutos(G)$;
$Products_{showcase} \leftarrow getShowCase(G_{Matchs})$; /*mostruário de produtos */
$G_{noMatchs}['unknown'].addAll(P_{unknown})$;
$G_{new}, G_{unknown} \leftarrow LocalizarCorrespondentes(G_{noMatchs}, Products_{showcase})$;
$G \leftarrow G_{Matchs} \cup G_{new}$; /*unificar grupos de produtos correspondentes */
return G

A Figura 4.2 ilustra um exemplo das operações realizadas nas etapas do Algoritmo 1. O processo começa com a entrada de um conjunto de produtos provenientes de diversas fontes de dados. Na Etapa 1, após a análise dos dados de entrada, o algoritmo identifica dois grupos de produtos. Na Etapa 2, são detectados os produtos que não pertencem a nenhum dos grupos identificados. Esses produtos com erros de correspondência são, então, encaminhados para a Etapa 3, que é responsável por associá-los corretamente aos seus respectivos grupos. Produtos que não foram associados a nenhum grupo são separados e, juntamente com futuras cargas de dados, serão processados novamente pelo *STEPMatch*.

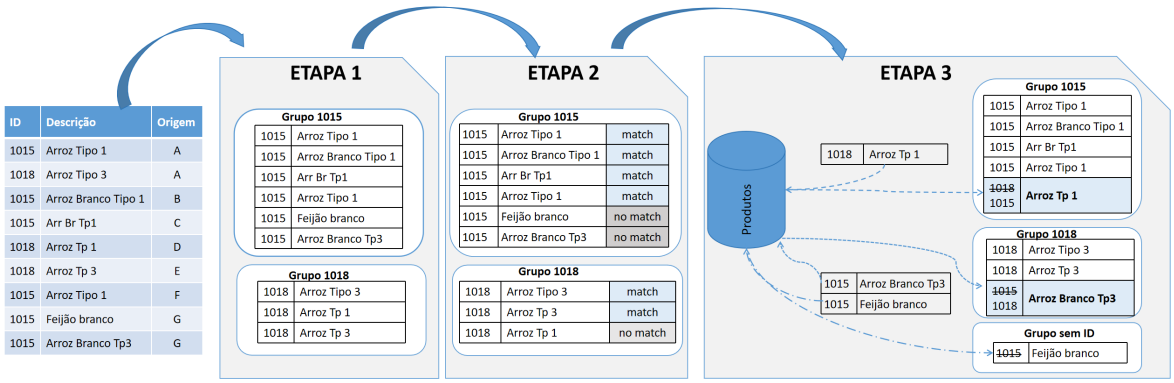

Figura 4.2 – Exemplo ilustrativo do funcionamento do Algoritmo 1.

As próximas seções deste capítulo irão detalhar as técnicas utilizadas nas principais etapas do Algoritmo 1.

4.2.2 Etapa 1: Agrupamento Inicial

Esta é primeira etapa de *blocking* adotada pelo *STEPMatch*, que envolve a divisão do conjunto de dados de produtos em blocos ou grupos menores com base em critérios específicos. Esses grupos são criados para selecionar os produtos que são potencialmente candidatos a serem comparados durante a etapa de correspondência. O objetivo do *blocking* é reduzir o espaço de comparação, limitando as comparações apenas às entidades dentro do mesmo grupo, o que ajuda a melhorar a eficiência do processo de correspondência de produtos. Esta etapa é fundamental para evitar a complexidade algorítmica $O(N^2)$ na fase de correspondência (PAPADAKIS; NEJDL, 2011; CHRISTOPHIDES et al., 2015).

O método *Standard Blocking* (SB) (FELLEGI; SUNTER, 1969) é uma estratégia baseada em *hash* utilizada em sistemas de resolução de entidades. Sua simplicidade, eficiência e flexibilidade o tornam uma opção popular para agrupar dados em diferentes domínios. Neste método, especialistas selecionam atributos relevantes e definem uma função de transformação para criar chaves de *blocking*. Essas chaves são geradas concatenando partes dos valores dos atributos selecionados. Cada chave única forma um grupo que contém todas as entidades que correspondem a essa chave.

O agrupamento inicial dos produtos, realizado na primeira etapa do Algoritmo 1, utiliza o método *Standard Blocking*. O atributo "identificador do produto", presente nos dados, foi utilizado para representar a chave de *blocking*. No entanto, devido aos erros inerentes aos cadastros dos produtos, as etapas 2 e 3 realizam verificações para confirmar as correspondências entre os produtos previamente agrupados, permitindo corrigir possíveis inconsistências nos grupos formados. As linhas de 1 a 8 do algoritmo definem os grupos de produtos com base nos identificadores disponíveis. Produtos que não apresentam identificadores (linha 5) são adicionados a um grupo específico (linha 9).

Nesta etapa, a função de similaridade $F_{Sim}(p_i, p_j)$, utilizada para agrupar dois produtos p_i e $p_j \in P$, baseia-se no identificador único de cada produto, definido por $id(p_j)$. Para o agrupamento de produtos, utiliza-se a função F^{SB}_{Sim}, embasada no método *Standard Blocking*, definida como:

$$F^{SB}_{Sim}(p_i, p_j) = \begin{cases} 1 & \text{se } id(p_i) = id(p_j) \\ 0 & \text{se } id(p_i) \neq id(p_j) \end{cases}$$

Dois produtos p_i e p_j são agrupados se $F_{Sim}(p_i, p_j) \geq \theta$. Nesse caso, a similaridade definida pelo limiar θ é igual a 1, ou seja, dois produtos são considerados semelhantes, se e somente se seus identificadores forem iguais.

4.2.3 Etapa 2: Verificação de Correspondências

Na Etapa 2, Verificação de Correspondências, verifica-se se as descrições dos produtos correspondem aos identificadores dos produtos. Para realizar essa tarefa, o Algoritmo 1, na linha 9, emprega a função *AgruparProdutos*, passando o agrupamento inicial dos produtos $G = \{g_1, \ldots, g_m\}$ definido na Etapa 1. A função *AgruparProdutos*, definida pelo Algoritmo 2 - denominado de Agrupador de Descrições de Produtos - verifica as correspondências dos produtos dentro dos grupos fornecidos.

Algorithm 2: AgruparProdutos: Agrupador de Descrições de Produtos

```
Entrada: G = (g_1, g_2, ..., g_m) ;                          /* grupos de produtos */
Saídas : G_Matchs = (g'_1, g'_2, ..., g'_m) ;   /* grupos de produtos correspondentes */
       1 G_noMatchs = (g''_1, g''_2, ..., g''_m) ;  /* produtos não correspondentes em G */
2  foreach g_i in G do
       // Definir descrição canônica de cada grupo de produtos
3      g_i.canonicalDesc ← findCanonicalDesc(g_i);
4  end
5  G_noMatchs ← ∅ ;                          /* grupos de produtos com ids inválidos */
6  G_Matchs ← clone(G) ;
   // Verificar correspondências de produtos por grupo
7  foreach g_i in G_Matchs do
8      P_noMatchs ← ∅ ;                       /* produtos com ids inválidos no grupo */
9      foreach p_j in g_i.products do
10         if not isMatch(p_j.desc, g_i.canonicalDesc) then
11             P_noMatchs.add(p_j) ;
12             g_i.delete(p_j);
13     end
14     G_noMatchs[g_i.id].add(P_noMatchs) ;                  /* manter ID do grupo */
15 end
16 return (G_Matchs, G_noMatchs);
```

A função *AgruparProdutos*, ao receber como entrada os grupos de produtos, verifica as correspondências dos produtos por grupo e retorna dois conjuntos, com a mesma quantidade de grupos de produtos, em que o primeiro conjunto representa os grupos de produtos intrinsecamente correspondentes, enquanto que o segundo conjunto representa os grupos de produtos que não corresponderam no agrupamento inicial.

Seja $G = \{g_1, g_2, \ldots, g_m\}$ o conjunto de grupos de produtos, a função *AgruparProdutos* processa cada grupo $g_i \in G$ e retorna dois conjuntos $G_{\text{Matchs}} = \{g'_1, \ldots, g'_m\}$ e $G_{\text{noMatchs}} = \{g''_1, \ldots, g''_m\}$, onde G_{Matchs} representa os produtos do grupo g_i que foram proces-

sados pelo algoritmo e identificados como correspondentes, enquanto G_{noMatchs} representa os produtos do grupo g_i identificados como não correspondentes. Assim, para cada grupo $g_i \in G$, tem-se que $g_i = g'_i \cup g''_i$ e $g'_i \cap g''_i = \emptyset$, onde $g'_i \in G_{\text{Matchs}}$ e $g''_i \in G_{\text{noMatchs}}$, garantindo que todos os produtos sejam categorizados de forma exclusiva em um dos dois conjuntos, preservando a estrutura do agrupamento inicial G. Ou seja, $G_{\text{Matchs}} \cup G_{\text{noMatchs}} = G$ e $G_{\text{Matchs}} \cap G_{\text{noMatchs}} = \emptyset$.

Para evitar a complexidade $O(N^2)$ nas comparações de todos os produtos dos grupos formados, define-se inicialmente uma descrição válida para cada grupo, denominada aqui de descrição canônica (linha 3 do Algoritmo 2). A verificação de correspondência dos produtos do grupo é realizada apenas com esta descrição canônica, resultando em uma complexidade $O(N)$ por agrupamento. Para definir a descrição canônica do grupo, foi adotada uma estratégia de voto majoritário, ou seja, escolhe-se a descrição com maior número de ocorrências. Em situações de empate, quando múltiplas descrições possuem o mesmo número de ocorrências, propõe-se a utilização de um critério secundário para desempate, como a escolha da descrição com mais palavras ou caracteres. Em caso de disponibilidade de um recurso externo confiável, este pode ser adotado para obter uma descrição canônica válida de um determinado produto.

Definida a descrição canônica de cada grupo de produtos, o Algoritmo 2 deve identificar e separar as associações incorretas dos produtos, mantendo os grupos cujos produtos são de fato correspondentes (G_{Matchs}), e criando grupos de produtos não correspondentes (G_{noMatchs}) (linhas 7 a 15). Essa identificação de produtos é realizada através da função de similaridade $F_{Sim}(p_i, p_j) \geq \theta$, onde $\{p_i, p_j\} \in g_i$, p_i é o produto que contém a descrição canônica do grupo g_i e θ representa o limiar de similaridade, com $0 \leq \theta \leq 1$. Formalmente, têm-se:

$$g'_i = \{p_j \mid F_{Sim}(p_i, p_j) \geq \theta\}$$
$$g''_i = \{p_j \mid F_{Sim}(p_i, p_j) < \theta\},$$

onde $g'_i \in G_{\text{Matchs}}$ e $g''_i \in G_{\text{noMatchs}}$. Então, esses dois conjuntos de grupos de produtos, G_{Matchs} e G_{noMatchs}, são retornados para o algoritmo principal (Identificador de Correspondências).

A função de similaridade $F_{Sim}(p_i, p_j)$ é representada através da função *isMatch()*, identificada na linha 10 do Algoritmo 2. As técnicas utilizadas na implementação da função *isMatch()* utilizam abordagem de aprendizagem de máquina e são detalhadas na seção 4.3.

4.2.4 Etapa 3: Busca de Produtos Correspondentes

A Etapa 2, Verificação de Correspondências, identifica produtos com correspondências inválidas (G_{noMatchs}) no agrupamento inicial (G). Na Etapa 3, Busca de Produtos Correspondentes, o objetivo é associar os produtos identificados como não correspondentes (G_{noMatchs}) a outros produtos que representam a mesma entidade, estabelecendo as correspondências corretamente.

Para viabilizar essa associação, define-se um mostruário de produtos, que representa um conjunto de itens indexados em um sistema de RI, o qual serve como base para associações subsequentes. Esse mostruário é criado a partir dos produtos que possuem correspondências válidas (G_{Matchs}), previamente identificadas na Etapa 2. Assim, o mostruário funciona como um repositório consolidado de produtos validados, permitindo que, nesta Etapa 3, sejam realizadas buscas de correspondências para os produtos pertencentes a G_{noMatchs}. As descrições dos produtos pertencentes a G_{noMatchs} são utilizadas como chaves de busca no sistema de RI, com o objetivo de localizar os produtos mais adequados e estabelecer as associações corretas.

O processo realizado na Etapa 3 pode ser formalmente descrito da seguinte forma:

1. Indexação: os produtos $p \in G_{\text{Matchs}}$ são indexados no sistema RI para possibilitar uma recuperação mais eficiente;
2. Busca: para cada produto $p'' \in G_{\text{noMatchs}}$, realiza-se uma busca no sistema RI utilizando a descrição de p'' como chave;
3. Correspondência: o sistema de RI retorna um conjunto de produtos $\{p_i\}$ para cada p'' pesquisado, onde $\{p_i\} = \text{findSimilarity}(p'')$; e
4. Associação: determina-se a correspondência correta entre p'' e $\{p_i\}$ com base na similaridade, $F_{Sim}(p'', \{p_i\}) \geq \theta$. A associação é realizada considerando o maior valor da função de similaridade F_{Sim}. Ou seja, para cada $p'' \in G_{\text{noMatchs}}$, encontra-se p^* tal que:

$$p^* = \arg\,{}_{p_i \in G_{\text{Matchs}}} \max\, F_{Sim}(p'', \{p_i\}).$$

   Nesse caso, o produto p^* é aquele que maximiza a função de similaridade F_{Sim} entre p'' e os produtos G_{Matchs}.

A função findSimilarity(p'') foi otimizada para localizar os produtos mais adequados para realizar as associações corretas. Para isso, são utilizados dois mecanismos de ordenação:

1. Algoritmo BM25:

   - Inicialmente, o algoritmo BM25 é usado para calcular a relevância dos produtos indexados ($p_i \in G_{\text{Matchs}}$) em relação ao produto de consulta ($p'' \in G_{\text{noMatchs}}$);
   - A função de similaridade do BM25, $F_{Sim}^{BM25}(p'', p_i)$, é utilizada para ordenar os produtos candidatos com base na similaridade textual. Formalmente, a busca inicial pode ser representada como:

$$\{p_i\} = \text{findSimilarity}_{BM25}(p''), \text{onde } p_i \in G_{\text{Matchs}} \text{ e } F_{Sim}^{BM25}(p'', p_i) > \theta$$

2. Reordenamento com *Cross-Encoder*:

   - Após a ordenação inicial com BM25, um reordenamento é realizado utilizando um modelo de linguagem de *cross-encoder*.
   - O modelo de linguagem avalia a relevância dos pares (p'', p_i) de maneira mais precisa, gerando uma pontuação de similaridade refinada $F_{Sim}^{\text{Cross-Encoder}}(p'', p_i)$. Esta pontuação é calculada considerando a contextualização e a semântica dos textos associados aos produtos. Assim, o reordenamento dos pares (p'', p_i) é realizado de forma que $F_{Sim}^{\text{Cross-Encoder}}(p'', p_i)$ seja maior ou igual a $F_{Sim}^{\text{BM25}}(p'', p_i)$.
   - Formalmente, o reordenamento pode ser representado como:

$$\{p_i\}_{\text{final}} = \text{Reorder}_{Cross-Encoder}(\{p_i\}, p''),$$

     onde $F_{Sim}^{\text{Cross-Encoder}}(p'', p_i) = \text{Cross-Encoder}(p'', p_i)$ e Cross-Encoder representa um modelo de linguagem treinado para calcular a similaridade entre dois produtos.

Assim, para cada produto $p'' \in G_{\text{noMatchs}}$, a função findSimilarity(p'') realiza uma busca inicial utilizando o algoritmo BM25 e um reordenamento subsequente com *cross-encoder*, retornando os produtos mais relevantes $\{p_i\}_{\text{final}}$ para cada produto p''.

A Figura 4.3 ilustra este processo de busca de produtos correspondentes implementado no *STEPMatch*.

O algoritmo BM25 é utilizado como uma estratégia de filtro (*Blocking*) para reduzir a complexidade $O(N^2)$ no reordenamento com *Cross-Encoder*. A indexação dos produtos foi implementada utilizando o *Elasticsearch*[1].

Os processos envolvidos nesta Etapa 3 são detalhados no Algoritmo 3, denominado Localizador de Correspondências. Esse algoritmo recebe como entrada dois parâmetros:

[1] <https://www.elastic.co/>

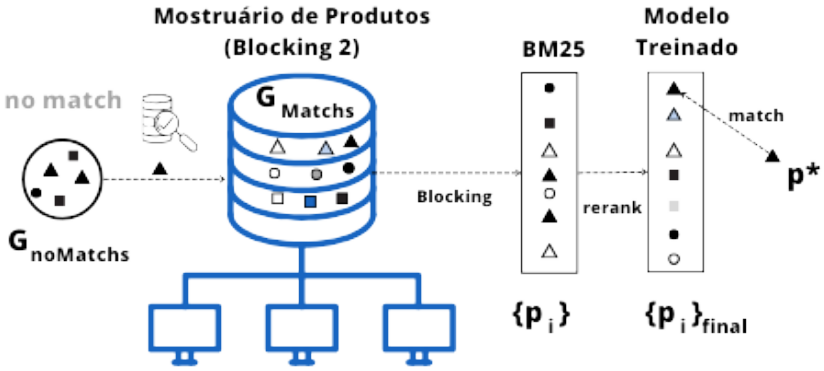

Figura 4.3 – Busca por correspondência de produtos

- G_{noMatchs}: um conjunto de grupos de produtos sem correspondências, identificados na Etapa de Correspondência (Etapa 2) pelo Algoritmo 2; e
- G_{Matchs}: um mostruário de produtos cujas correspondências foram validadas pelo Algoritmo 2. Esse mostruário representa os produtos que estão indexados no sistema de RI.

Na linha 3 do Algoritmo 3 são instanciados dois conjuntos vazios: G_{newMatch} e G_{unknown}. O conjunto G_{newMatch} representa os grupos de produtos nos quais foi possível realizar novas correspondências com produtos do mostruário, enquanto G_{unknown} representa um conjunto com os produtos para os quais não foi possível determinar correspondências. Esses conjuntos constituem o resultado final do Algoritmo 3.

Em seguida, o Algoritmo 3 itera sobre cada produto de cada grupo em G_{noMatchs} para realizar as buscas no mostruário G_{Matchs} (linhas 5-17). A função *findSimilarity* (linha 7) é responsável por retornar uma lista ordenada por relevância, considerando o grau de similaridade do item pesquisado $p_j \in G_{\text{noMatchs}}$ com os produtos $p_i \in G_{\text{Matchs}}$. O primeiro elemento da lista, p^*, representa o produto p_i com o maior nível de similaridade com o produto utilizado na busca p_j (linhas 9-10). A função *isMatch()*, apresentada no Algoritmo 2, é utilizada novamente para verificar se, de fato, há correspondência entre p_j e o primeiro elemento da lista p^* (linha 11). Confirmada a correspondência dos itens, o produto p_j é adicionado ao mesmo grupo do elemento p^* do topo do resultado da busca (linhas 12-14). Caso a função *isMatch()* não confirme a correspondência, o item p_j é adicionado ao grupo G_{unknown} de produtos não correspondentes (linhas 15 e 18). Os produtos em G_{unknown} são separados para serem processados posteriormente com novas recargas de dados no *STEPMatch*, permitindo novas tentativas de associações. Finalmente, os conjuntos G_{newMatch}, que incluem grupos de produtos correspondentes, e G_{unknown}, com produtos sem correspondências (linha 19), representam o resultado final do processamento do Algoritmo 3 e são retornados.

Algorithm 3: LocalizarCorrespondentes: Localizador de Correspondências

Entradas: $G_{noMatchs} = (g_1, g_2, ..., g_m)$; /* grupos de produtos não correspondentes */

1 $Products_{showcase}$, ; /* mostruário de produtos */

Saídas : $G_{newMatch} = (g_1, g_2, ..., g_n)$; /* produtos reagrupados */

2 $G_{unknown} = (g_{unknown})$; /* grupo produtos não correspondentes */

3 $G_{newMatch} \leftarrow G_{unknown} \leftarrow \emptyset$;

4 $Products_{unknown} \leftarrow \emptyset$; /* produtos sem correspondências */

5 **foreach** g_{aux} *in* $G_{noMatchs}$ **do**

6 **foreach** p_j *in* $g_{aux}.products$ **do**

 // busca de produtos correspondentes

7 $products_{result} \leftarrow findSimilarity(p_j, Products_{showcase})$;

8 $flag_{match} \leftarrow False$;

9 **if** $products_{result}.size() > 0$ **then**

10 $p^* = products_{result}[0]$; /* produtos mais relevantes estão no topo */

11 **if** $isMatch(p^*.desc, p_j.desc)$ **then**

12 $p_j.id \leftarrow p^*.id$; /* Corrigir o id do produto */

13 $flag_{match} = True$;

14 $G_{newMatch}[p_j.id].add(p_j)$;

15 **if** $(not\ flag_{match}$ **then**

16 $Products_{unknown}.add(p_j)$; /* produtos sem correspondências */

17 **end**

 // redefinição do grupo de produtos sem ids

18 $G_{unknown}['unknown'].add(Products_{unknown})$;

19 **return** $(G_{newMatch}, G_{unknown})$

20 **end**

4.3 CORRESPONDÊNCIA DE PRODUTOS COM APRENDIZAGEM DE MÁQUINA

O *STEPMatch* utiliza uma função de similaridade F_{Sim} - formalizada na seção 4.2.3 - para analisar e realizar as correspondências dos produtos. Essa função é empregada tanto no Algoritmo 2, Agrupador de Descrições de Produtos, quanto no Algoritmo 3, Localizador de Correspondências. Nesta pesquisa, a função F_{Sim} foi implementada utilizando duas abordagens distintas que empregam técnicas de aprendizagem supervisionada de máquina: uma baseada em algoritmos tradicionais de classificação e outra que utiliza modelos de linguagens pré-treinados (MLPTs). A Figura 4.4 ilustra as abordagens, destacando o processo de treinamento dos modelos. Em uma das abordagens, os MLPTs passam por ajustes finos para a tarefa específica, enquanto na outra, as características textuais

são extraídas inicialmente para, em seguida, serem usadas no treinamento dos modelos. Pretende-se, com isso, comparar os resultados dessas abordagens no contexto de títulos curtos de produtos.

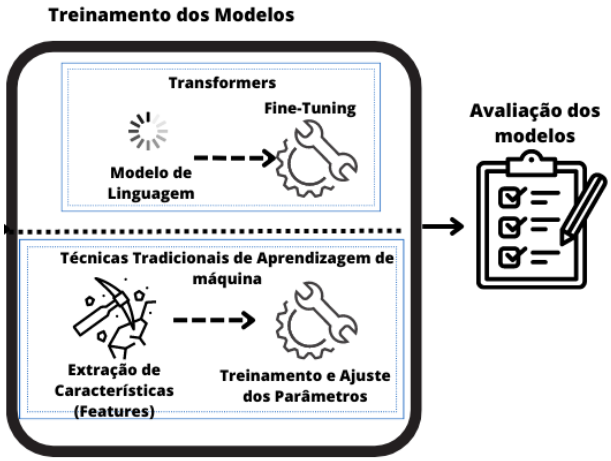

Figura 4.4 – Técnicas utilizadas para análise de Correspondência de Produtos

Os modelos tradicionais de classificação oferecem uma abordagem clássica da literatura e são frequentemente utilizados em tarefas de classificação (RISTOSKI et al., 2018; FOXCROFT et al., 2021; CHRISTEN, 2012). Por outro lado, os MLPTs, que utilizam técnicas de aprendizagem profunda, representam o estado da arte em diversas tarefas de NLP (KIM et al., 2022; BARLAUG; GULLA, 2021; MOŻDŻONEK et al., 2022).

4.3.1 Modelos Tradicionais de Classificação

Os algoritmos tradicionais de classificação supervisionada, tais como Árvores de Decisão, *Random Forest*, XGBoost, *Naive Bayes* e SVM, têm apresentado bons resultados em muitas tarefas de classificação na área de NLP (BIRJALI et al., 2021; SANTANA et al., 2023; CHRISTEN, 2012).

A comparação dos algoritmos que utilizam técnicas tradicionais de classificação no contexto de *correspondência de produtos* com descrições curtas foi realizada por meio do *framework* AutoML (*Automated Machine Learning*) auto-sklearn[2], a fim de automatizar a seleção de modelos e a otimização de hiperparâmetros. Utilizando os dados de treinamento e teste, o auto-sklearn analisa diversos algoritmos de aprendizagem supervisionada, ajusta seus hiperparâmetros e seleciona a melhor combinação de modelos e configurações (FEURER et al., 2015; FEURER et al., 2022). Isso permite identificar automaticamente o algoritmo mais adequado e as melhores configurações para a tarefa específica, sem a necessidade de intervenção manual intensiva.

[2] <https://automl.github.io/auto-sklearn>

Quando aplicados a dados textuais, os classificadores de aprendizagem de máquina tradicionais requerem que os textos sejam transformados em vetores numéricos que representem as características (*features*) relevantes para o processamento e classificação. Neste trabalho, foi adotada uma abordagem semelhante à utilizada por Magellan (KONDA, 2018), Foxcroft et al. (2021) e Santana et al. (2023), na qual as características são representadas por medidas de similaridades entre os textos. Nesta pesquisa, as medidas utilizadas para calcular as similaridades dos títulos dos produtos foram: distâncias de Levenshtein, Damerau–Levenshtein e Hamming, similaridades de Jaccard, Jaro e Jaro-Winkler, taxas de similaridades de *tokens* e de números, além de métricas de similaridade entre palavras da biblioteca de Python fuzzwuzy (H.GOMAA; FAHMY, 2013).

O vetor de características das descrições de dois produtos P_1 e P_2 é dado por $V(P_1, P_2) = \{S_1^{similarity}, .., S_n^{similarity}\}$, onde $S_k^{similarity}$ representa o valor obtido por uma técnica que mensura a similaridade entre as descrições dos produtos, com $1 \leq k \leq n$ e n representando a quantidade de técnicas utilizadas. Por exemplo, considerando as distâncias de Levenshtein e Hamming, e similaridade Jaccard, o vetor de características dos produtos P_1 e P_2 poderia ser representado por:

$$V(P_1, P_2) = \{F_{Levenshtein}(P_1, P_2), F_{Hamming}(P_1, P_2), F_{Jaccard}(P_1, P_2)\},$$

onde $F_{Levenshtein}(P_1, P_2)$, $F_{Hamming}(P_1, P_2)$ e $F_{Jaccard}(P_1, P_2)$ correspondem às funções que mensuram a similaridade entre as descrições dos produtos P_1 e P_2 utilizando, respectivamente, os algoritmos de distância de Levenshtein, distância de Hamming e similaridade Jaccard.

Os algoritmos utilizados no *framework* AutoML recebem esses vetores $V(P_i, P_j)$ como entrada para treinar e avaliar modelos de classificação. Para cada vetor $V(P_i, P_j)$, há um rótulo associado y_{ij} que indica se os produtos são correspondentes ($y_{ij} = 1$) ou não ($y_{ij} = 0$). Deseja-se, então, encontrar uma função de classificação f_c que mapeia o vetor de características $V(P_i, P_j)$ ao rótulo correspondente y_{ij}, isto é,

$$f_c \colon V(P_i, P_j) \rightarrow y_{ij}.$$

Além disso, o modelo de classificação deve ser treinado para produzir um índice de similaridade, denotado por $Sim(P_i, P_j)$ que indica o grau de correspondência entre os produtos P_i e P_j. Formalmente, a saída do modelo é:

$$\hat{y}_{ij} = f_c(V(P_i, P_j))$$

$$\hat{S}im(P_i, P_j) = g_c(V(P_i, P_j)),$$

onde $\hat{y}_{ij}$ é a classificação binária (0 ou 1) e $Sim(P_i, P_j)$ representa o índice de similaridade calculado a partir da função g_c aplicada às características $V(P_i, P_j)$ dos produtos, isto é,

$$g_c \colon V(P_i, P_j) \rightarrow [0, 1].$$

Esse índice de similaridade é utilizado na Etapa 3 do *STEPMatch*, cuja descrição detalhada pode ser encontrada na Seção 4.2.4.

4.3.2 Modelos de Linguagens para Classificação

Esta seção apresenta duas abordagens principais relacionadas ao uso de MLPTs nesta pesquisa. Primeiro, discute-se o ajuste fino dos modelos para a tarefa de classificação de pares de produtos, com ênfase no refinamento dos parâmetros utilizando os corpora rotulados. Em seguida, explora-se o aprendizado por transferência com cruzamento de idiomas, uma estratégia para expandir o uso de MLPTs para idiomas com poucos recursos, aproveitando dados de um idioma fonte para melhorar o desempenho em outro idioma destino.

4.3.2.1 Ajuste Fino de Modelos de Linguagem Pré-Treinados

A utilização de MLPTs na tarefa de análise de correspondência de produtos em textos curtos foi realizada através de ajuste fino (*fine-tuning*). Esse ajuste fino é efetuado por meio do refinamento dos parâmetros dos modelos com base em corpora específicos. A abordagem adotada visa ao aprendizado por transferência indutivo, no qual o modelo é aprimorado mediante a inserção de entradas específicas da tarefa-alvo. Então, o modelo ajusta seus parâmetros de acordo com essas entradas para melhor atender às necessidades da tarefa. Especificamente, para a correspondência de produtos, o modelo recebe duas descrições de produtos P_i e P_j como entrada, classificando-as como Correspondentes ($y = 1$) ou Não Correspondentes ($y = 0$).

Formalmente, um MLPT pode ser representado por M_θ, em que θ são os parâmetros do modelo. Inicialmente, M_θ foi treinado em uma grande quantidade de dados de texto para aprender representações linguísticas gerais. O ajuste fino do modelo M_θ para a tarefa de classificação de correspondência de produtos consiste em refinar os parâmetros θ em um corpus específico de produtos rotulados. Então, seja $D = \{(p_1^{(i)}, p_2^{(i)}, y^{(i)}\}_{i=1}^{N}$ o conjunto de dados rotulados, onde N é o número de pares de produtos do corpus específico, $p_1^{(i)}$ e $p_2^{(i)}$ são as descrições dos pares de produtos e $y^{(i)}$ é o rótulo correspondente (1 para Correspondentes, 0 para Não Correspondentes). Dessa forma, o ajuste fino é realizado otimizando os parâmetros θ do modelo para minimizar o erro na classificação dos pares de produtos, ou seja, o objetivo é encontrar os parâmetros θ^* que minimizam a função de

perda:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta),$$

onde θ^* são os parâmetros ajustados do modelo para a tarefa específica de correspondência de produtos. A função de perda $\mathcal{L}(\theta)$ é utilizada para quantificar o erro, guiando o ajuste dos parâmetros do modelo. A perda de entropia cruzada (*Cross-Entropy Loss*) é geralmente utilizada em tarefas de classificação, medindo a diferença entre as previsões do modelo ($\hat{y}^{(i)}$) e os rótulos verdadeiros ($y^{(i)}$).

Após o ajuste fino, o modelo ajustado $M_{\theta*}$ recebe duas descrições de produtos P_i e P_j e prevê se são Correspondentes ($y = 1$) ou Não Correspondentes ($y = 0$). Semelhante aos modelos tradicionais de aprendizagem de máquina, o modelo $M_{\theta*}$ fornece um índice de similaridade $Sim_m(y_{ij} = 1 \mid (p_i, p_j))$ que indica a confiança de ocorrência de correspondência entre os produtos P_i e P_j. Formalmente, a saída do modelo é representada por:

$$Sim_m = f_m(M_{\theta*}(P_i, P_j))$$

$$\hat{y} = \begin{cases} 1 & \text{se } M_{\theta*}(P_i, P_j) \geq \tau \\ 0 & \text{se } M_{\theta*}(P_i, P_j) < \tau, \end{cases}$$

onde Sim_m representa o grau de correspondência entre os produtos P_i e P_j. Este índice é calculado a partir da função f_m, que recebe o valor retornado da função de ativação *softmax*, utilizada na camada final do MLPT $M_{\theta*}$. Finalmente, $\hat{y}$ representa a classificação binária (0 ou 1) predita através de um limiar τ. Por padrão, o limiar τ é igual a 0,5, mas pode ser ajustado conforme a otimização desejada.

É importante ressaltar que, quando algum MLPT é utilizado, em geral, o próprio modelo contém um módulo responsável pela vetorização dos textos (BHASKARAN; BHALLAMUDI, 2019; HEIJDEN et al., 2021), conforme discutido no Capítulo 3. Desta forma, o processo de vetorização das descrições dos produtos é realizado pelo MLPT específico. As entradas para os modelos foram preparadas de forma que os títulos dos produtos de cada par fossem concatenados em uma única sequência de texto. Assim, a sequência final, uma vez pronta para ser processada pelo modelo, segue o formato padrão: "[CLS] Título do Produto 1 [SEP] Título do Produto 2 [SEP]". Os tokens especiais "[CLS]" e "[SEP]" representam, respectivamente, o início da sequência, indicando uma tarefa de classificação, separação entre os títulos e o fim da sequência.

4.3.2.2 Transferência de Aprendizado entre Idiomas na Correspondência de Produtos

No contexto de correspondência de produtos, este trabalho também contribui com o estado da arte ao avaliar técnicas de transferência de aprendizado entre idiomas por meio da exploração de diversas estratégias de CLL. Nessa abordagem, são avaliados diversos MLPTs, incluindo tanto modelos monolíngues quanto multilíngues. As estratégias de CLL têm o objetivo de aprimorar o desempenho dos MLPTs utilizando corpora rotulados de um idioma específico para construir modelos de classificação aplicáveis a diferentes idiomas, através do aprendizado por transferência. A estratégia envolve a indução de modelos de classificação a partir de um idioma fonte, realizando ajuste fino com uma porção menor de dados do idioma alvo. Essa abordagem também permite o aproveitamento de modelos de linguagens em idiomas com recursos limitados, potencializando a eficiência e a precisão na tarefa de correspondência de produtos.

A Figura 4.5 fornece uma visão geral da metodologia proposta, que compreende quatro etapas distintas: aquisição de corpora, seleção de MLPTs, determinação da estratégia de ajuste fino para CLL e avaliação do modelo final usando dados do corpus destino.

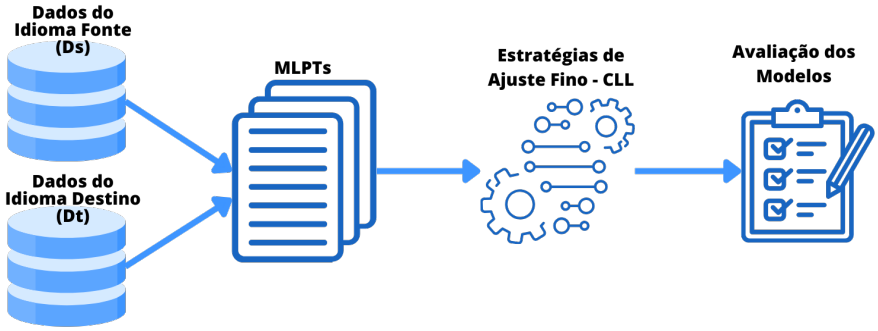

Figura 4.5 – Visão geral do uso de CLL

A utilização de estratégias de CLL pressupõe o uso de pelo menos dois corpora de idiomas distintos com o objetivo de desenvolver, por meio de transferência de aprendizado, um modelo de classificação que aproveite dados de um idioma fonte D_s para melhorar a classificação de correspondência de produtos em um idioma alvo D_t. O modelo treinado M_θ^{CLL} é ajustado utilizando uma combinação dos dados dos corpora D_s e D_t, controlada pelos parâmetros α e β, que ajustam a proporção de dados de cada idioma no processo de ajuste fino. Formalmente, têm-se:

$$D = \alpha D_s + \beta D_t,$$

onde α e β são parâmetros que controlam a quantidade de dados do idioma fonte e do idioma alvo, respectivamente.

Nesta pesquisa, foram utilizados dados em inglês como idioma fonte (D_s) e dados em português como idioma alvo (D_t), cujos detalhes são apresentados no 5. No que diz respeito à transferência de aprendizagem com cruzamento de idiomas, foram exploradas as estratégias ZST (*Zero-Shot Transfer*), JL (*Joint Learning*) e CL (*Cascade Learning*), detalhadas na Seção 2.4. Além disso, foram implementadas combinações dessas abordagens, denominadas JL/CL e JL/CL+.

As estratégias de transferência de aprendizagem com cruzamento de idiomas utilizadas nesta tese são descritas a seguir:

- Na estratégia ZST, foram utilizados exclusivamente os dados de treinamento do idioma fonte para ajustar os parâmetros dos modelos, ou seja, $\alpha = 1$ e $\beta = 0$. Na etapa de teste, os modelos foram avaliados apenas com os dados de teste do corpus do idioma alvo;
- Em relação à estratégia JL, foi mantida a mesma proporção de dados apresentada na estratégia ZST. No entanto, avaliou-se incrementos de um subconjunto dos dados de treinamento do idioma alvo no ajuste fino inicial dos modelos, variando entre 10% e 50%, ou seja, $\alpha = 1$ e $0,1 \leq \beta \leq 0,5$. Assim, foi possível avaliar o comportamento dos modelos com diferentes tamanhos de subconjuntos do idioma alvo durante o treinamento inicial;
- Na estratégia CL, foram utilizados apenas os dados do idioma fonte para o treinamento inicial do modelo ($\alpha = 1$ e $\beta = 0$). Em seguida, realizou-se um novo ajuste fino do modelo, incorporando frações dos dados de treinamento do idioma alvo, variando de 10% a 50% ($\alpha = 0$ e $0,1 \leq \beta \leq 0,5$). O modelo avaliado corresponde ao ajustado com os hiperparâmetros definidos nesse segundo ajuste fino;
- Na estratégia CL/JL, as estratégias JL e CL foram combinadas, utilizando frações dos dados de treinamento do idioma alvo em 50% e 100%. Essa abordagem visou a avaliar o impacto do uso de diferentes frações do idioma alvo ao aplicar as estratégias JL e CL, em que uma parte dos dados do idioma alvo é utilizada no treinamento inicial e as demais frações são empregadas no segundo ajuste fino; e
- Na estratégia JL/CL+, realizou-se treinamentos adicionais de CL em variações dos modelos previamente treinados com a estratégia JL/CL, utilizando todo o conjunto de treinamento do idioma alvo.

4.4 AVALIAÇÃO

Para avaliar o *STEPMatch* proposto neste trabalho, decidiu-se analisar as técnicas utilizadas na Etapa 2, Correspondência de Produtos, e na Etapa 3, Busca de Produtos

Correspondentes.

Na Etapa 2, relativa à tarefa de classificação, foram avaliados os modelos que analisam as correspondências entre as descrições de produtos. Para isso, foram utilizadas as métricas de Precisão, Revocação (*Recall*) e Medida F1 (*F1 Score*).

No que tange à análise da qualidade do ranqueamento dos itens recuperados na Etapa 3, objetivando recuperar produtos correspondentes em sistema de recuperação da informação, foram utilizadas as métricas Acurácia@n, MAP@n, MRR@n e NDCG@n, onde n denota o número de posições consideradas na lista recuperada pelo sistema.

4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, foi apresentado o *STEPMatch*, proposto nesta tese para realizar a correspondência de produtos em cenários em que as descrições são curtas, variadas e frequentemente inconsistentes, como ocorre nas notas fiscais. Foram detalhadas as principais etapas da metodologia proposta, incluindo o agrupamento inicial dos produtos por identificadores, a verificação de correspondências e a busca de produtos correspondentes. Para a análise de correspondência de produtos, foram adotadas abordagens clássicas de aprendizagem de máquina e técnicas avançadas de NLP, incluindo o uso de MLPTs e estratégias de CLL. Para localizar os identificadores de produtos, foi proposto um método de RI que realiza a busca de produtos correspondentes com o auxílio de modelos de linguagem treinados especificamente para essa tarefa.

Os próximos capítulos descreverão os dados utilizados na pesquisa, a abordagem adotada para a construção dos corpora de produtos e os resultados dos experimentos realizados.

5 CONSTRUÇÃO DOS CORPORA

Este capítulo tem como objetivo apresentar a metodologia utilizada na construção de corpora de produtos de forma automática. Inicialmente, na seção 5.1, são descritos os dados de produtos empregados na pesquisa. Na seção 5.2, é detalhada a metodologia adotada para a geração de corpora de produtos, visando ao treinamento de modelos de aprendizagem de máquina na tarefa de correspondência de produtos.

5.1 DADOS DE PRODUTOS

Os conjuntos de dados utilizados nesta tese incluem informações de produtos em português e inglês, e são compostos por três bases de dados distintas: 1) Notas Fiscais, que contêm dados de produtos oriundos de notas fiscais emitidas no estado do Acre; 2) eCommerce, que abrange dados de produtos coletados em sites de comércio eletrônico no Brasil; e 3) WDC_products, que contém pares anotados de produtos em inglês, utilizado em estudos anteriores de correspondência de produtos (PEETERS et al., 2020; PRIMPELI et al., 2019)

A base de dados de Notas Fiscais utilizada nesta pesquisa abrange registros de produtos comercializados no estado do Acre durante um período de três meses, de maio a julho de 2023, totalizando aproximadamente 6,6 milhões de registros. Esses dados foram disponibilizados por meio de um acordo de cooperação para Pesquisa, Desenvolvimento e Inovação (PD&I) estabelecido entre a Universidade Federal de Campina Grande e o Tribunal de Contas do Estado do Acre. As informações disponíveis na base de dados incluem apenas o código de identificação (GTIN), uma descrição curta do produto e o NCM (NCM-Nomenclatura Comum do Mercosul ou CEST[1]). Essa base de dados apresenta uma diversidade significativa de produtos, contabilizando cerca de 578.640 códigos de barras (GTINs) distintos e 942.447 descrições únicas. A Figura 5.1 ilustra o quantitativo diário de produtos comercializados durante o período analisado. Em média, são comercializados aproximadamente 71.637 produtos por dia.

Os cadastros dos produtos são realizados pelos próprios estabelecimentos, o que resulta em problemas comuns no contexto de notas fiscais, como identificadores inválidos, produtos sem identificação e a atribuição incorreta de códigos de identificador a produtos diferentes. A Figura 5.2 apresenta exemplos de produtos provenientes da base de dados de Notas Fiscais.

[1] CEST - Código Especificador da Substituição Tributária

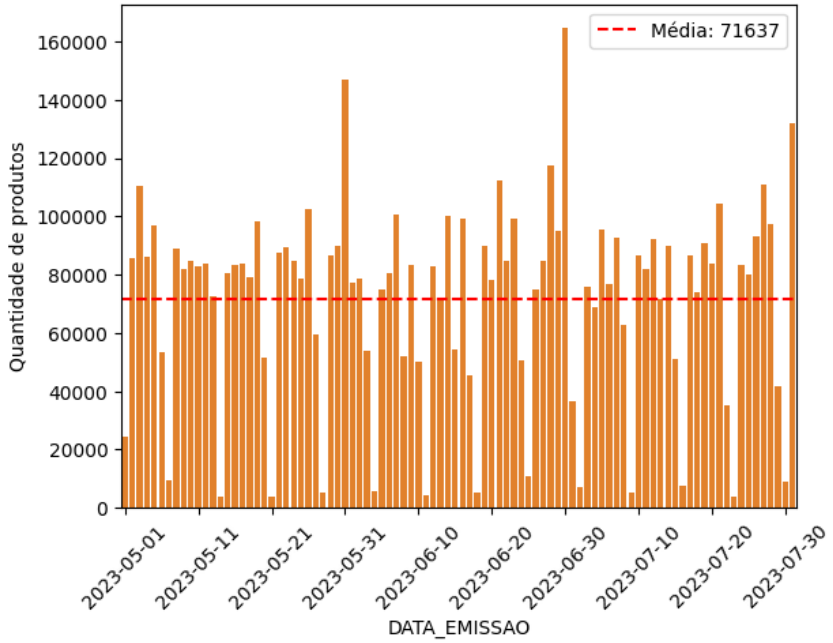

Figura 5.1 – Distribuição Diária da Quantidade de Produtos da base de dados de Notas Fiscais

GTIN/EAN	NCM	DESCRIÇÃO
7897424080762	96082000	PINCEL ATOMICO 1100 P PILOT PRETO
7896572001124	96081000	CANETA COMPACTOR TOP 2000 AZUL
7896123402059	39241000	COPO DESC COPOZAN 250 MILILITROS C 100 UNIDADES
7896021623402	73231000	ESPONJA LIMPANO LA DE ACO C 8 60 GRAMAS
7896026838245	48181000	PAPEL HIGIENICO PALOMA 4X30 METROS NEUTRO
...	...	...
7891634329345	48025899	PAPEL ESP FILIPERSON VERGE 180 GRAMAS BRANCO 5...
7891173022868	48025610	PAPEL A 4 75 GRAMAS CHAMEX VERDE 210X297 C 500FLS
7896303600015	83059000	CLIPS ACC GALVANIZADO N 8 0 C 25
7897849608091	83059000	CLIPS BACCHI GALVANIZADO NO6 0
7897256222156	84729040	PERFURADOR METAL C 2 FUROS S20 P 20FLS JOCAR

Figura 5.2 – Exemplos de Produtos de Notas Fiscais

A base de dados eCommerce contém informações de produtos extraídas de sites de comércio eletrônico brasileiros, como Americanas[2] e Amazon[3]. Foram coletadas informações sobre os títulos e os códigos GTIN de produtos das categorias Notebooks, Celulares e Televisores. Considerando que esses sites de comércio eletrônico atuam como *marketplaces*, que não apenas comercializam seus próprios produtos, mas também intermediam vendas de diversos vendedores, é comum encontrar um mesmo produto (identificado pelo código GTIN) com diferentes descrições, refletindo as diversas formas como os vendedores o apresentam. No total, foram coletados cerca de sete mil produtos, contendo cerca de 4.500 códigos GTIN distintos.

[2] <http://www.americanas.com.br>
[3] <https://www.amazon.com.br>

A base de dados WDC Products (*Web Data Commons Training and Test Sets for Large-Scale Product Matching*) contém dados de produtos do idioma inglês, coletados de diversos sites de comércio eletrônico (PEETERS et al., 2024). O corpus é composto por pares de produtos anotados com os rótulos correspondentes "match" (correspondência) e "non match" (não correspondência), sendo utilizado como referência (*benchmark*) para tarefas de correspondência de produtos (PEETERS et al., 2020; PRIMPELI et al., 2019). Além disso, a base de dados já possui os conjuntos de treinamento, validação e teste separados, o que permite uniformizar as comparações entre as diversas soluções desenvolvidas pela comunidade. O WDC Products inclui diversas configurações de diferentes tamanhos, abrangendo quatro categorias de produtos: computadores, câmeras, relógios e sapatos (PRIMPELI et al., 2019). Nesta tese, utilizou-se a base de dados de tamanho médio da categoria Computadores[4], que é composta por aproximadamente oito mil pares de produtos rotulados.

A Tabela 5.1 apresenta uma amostra de produtos de cada base de dados. Os exemplos ilustram as diferenças entre as bases: na base de Notas Fiscais, observam-se descrições mais curtas, como em "DESOD ROLLON FANTASY 1X50ML" e "VINAGRE VIRROSAS ALCOOL 750ML MACA", que refletem a linguagem simplificada utilizada por estabelecimentos comerciais. Nas bases de dados eCommerce e WDC Products, as descrições são mais detalhadas e incluem características do produto que variam conforme a apresentação de cada vendedor, como em "Refrigerador Philco 434 Litros Side By Side Eco Inverter Inox PRF533ID – 220 Volts" e "Kingston Digital 64GB Data Traveler Micro Duo USB 3.0 OTG".

5.2 CONSTRUÇÃO DOS CORPORA - ANOTAÇÃO DE PARES DE PRODUTOS

Observa-se uma predominância de corpora em inglês nas bases de dados disponíveis para a tarefa de correspondência de produtos, o que destaca a necessidade de desenvolver corpora em outros idiomas. Para esta pesquisa, foi necessário construir um corpus em português que atendesse às características específicas dos produtos nas bases de dados analisadas, especialmente na base de produtos provenientes de notas fiscais. A criação do corpus consiste na anotação de pares de produtos, indicando se os itens comparados correspondem ao mesmo produto ou são distintos.

Deseja-se, portanto, construir um conjunto de tuplas $T = \{(P_i, P_j, r_{ij})\}$, onde P_i e P_j representam os títulos dos produtos, e $r_{ij} \in \{0, 1\}$ é uma variável indicadora que denota se P_i e P_j referem-se ao mesmo produto. Mais especificamente, $r_{ij} = 1$ indica que P_i e P_j são correspondentes (*match*), classificando-se como uma instância positiva, enquanto

[4] <http://webdatacommons.org/largescaleproductcorpus/v2/>

Base de Dados	Identificador (GTIN)	Título do Produto
Notas_Fiscais	7892509120975	Celular Samsung Galaxy A03 64GB Dual SM-A035MZBSZTO AZUL QUADRIBAND
	7896044960065	DESOD ROLLON FANTASY 1X50ML
	7897006310140	VINAGRE VIRROSAS ALCOOL 750ML MACA
	7898215157441	LEITE DESNATADO MOLICO 1L
eCommerce	7891360000000	Refrigerador Philco 434 Litros Side By Side Eco Inverter Inox PRF533ID – 220 Volts
	609964000000	Smartphone Red Mobile Volt L S51, Tela 5, Câmera 8MP + 5MP Frontal, Memória 16GB + Expansível até 512GB - Preto/Azul
	7891130000000	Geladeira Brastemp BRM44HK Frost Free Duplex 375L com Compartimento Extrafrio Fresh Zone Inox - 110V
	195553000000	ASUS Laptop fino e leve VivoBook 15 F515, tela FHD de 15,6 polegadas, processador Core i7-1165G7, gráficos Iris Xe, RAM DDR4 8GB, SSD 512GB, impressão digital, Windows 11 Home, cinza ardósia, F515EA-DH75
WDC_ Products	4276619	Kingston Digital 64GB Data Traveler Micro Duo USB 3.0 OTG (DTDUO3/64GB)"@en-US "Data Storage - Page 51 | Laptops Outlet Direct"@en-US"
	17034093	4th Generation Intel® Core™ i3 4160 3.6GHz Socket LGA1150 "@en CoreAndtrade; | BX80646I34160 Novatech"@en"
	10999920	Kingston 16GB PC3-14900 240-pin DDR3 SDRAM DIMM Kit@en "Buy Kingston Kit at Connection Public Sector Solutions"@en
	10589695	XPS 12 Convertible Touch Ultrabook™@en, Ultrabook™"@en "XPS 12"Ultrabook™ Details | Dell Thailand"@en"

Tabela 5.1 – Exemplos de produtos das três bases de dados utilizadas

$r_{ij} = 0$ indica que não são correspondentes (*non match*), caracterizando-se, então, como uma instância negativa.

No processo de criação do conjunto de tuplas $T = \{(P_i, P_j, r_{ij})\}$, o desafio não se limita apenas à identificação correta das correspondências, mas também à inclusão de pares contrastivos que ajudem o modelo a aprender a diferenciar corretamente entre instâncias positivas e negativas. O uso de pares contrastivos — que envolvem a comparação de produtos similares e dissimilares — é fundamental a fim de refinar a capacidade do modelo para distinguir entre descrições que representam o mesmo produto de forma diferente e aquelas que se referem a produtos distintos. Essa abordagem contrastiva possibilita um aprendizado mais robusto, ajudando a capturar nuances específicas da língua portuguesa e das características dos produtos (EMBAR et al., 2020; PEETERS et al., 2020; PEETERS; BIZER, 2022).

Neste trabalho, foi utilizada uma abordagem contrastiva para a criação dos corpora de pares de produtos anotados. A Figura 5.3 ilustra os mecanismos empregados para a definição de instâncias positivas e negativas, possibilitando a composição de corpora de pares de produtos diversificados. As subseções 5.2.1 e 5.2.2 detalham, respectivamente, os processos de definição de instâncias positivas (pares de produtos correspondentes) e de instâncias negativas (pares de produtos não correspondentes).

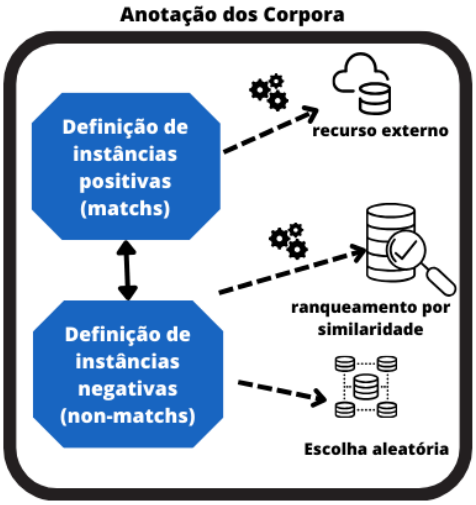

Figura 5.3 – Módulo de Criação de Corpora

5.2.1 Definição de Pares de Produtos Correspondentes

As bases de dados de produtos contêm informações sobre os identificadores únicos (GTIN), permitindo a construção de pares de produtos a partir da combinação de diferentes descrições associadas ao mesmo identificador. Além disso, esta tese contempla o uso de recursos externos para o enriquecimento da base de dados, possibilitando a busca de descrições adicionais de produtos disponíveis na Web. Essa abordagem amplia e diversifica as combinações de descrições de um mesmo produto, tornando o corpus ainda mais representativo.

Neste trabalho, para enriquecer as descrições correspondentes de produtos da base de dados, em algumas categorias de produtos, foram utilizados *crawlers* para realizar *web scraping* no site Bluesoft Cosmos[5], que é um Catálogo Online de Produtos, onde é possível consultar produtos comercializados no Brasil por código de barras (GTIN) e por classificação fiscal (NCM ou CEST).

Para a base de dados Notas_Fiscais, com o intuito de reduzir o risco de erros na construção do corpus, como correspondências incorretas entre produtos distintos com o mesmo identificador único (GTIN), decorrentes de falhas no cadastro de produtos pelos estabelecimentos de venda, foram estabelecidos os seguintes critérios:

- seleção dos itens mais comercializados; e
- uso de heurística baseada em métricas de similaridade de textos: mínimo de duas palavras idênticas no título do produto (SANTANA et al., 2023).

[5] <https://cosmos.bluesoft.com.br/>

Ademais, em uma das categorias definidas (Leites e Laticínios), foi realizada uma revisão manual das tuplas positivas por pelo menos três pesquisadores, selecionando apenas os itens em que há concordância de correspondências, com base no critério do voto majoritário.

5.2.2 Definição de Pares de Produtos não Correspondentes

A seleção de tuplas negativas difere da seleção de tuplas positivas, que considera as descrições distintas de produtos com identificadores iguais. Nesse caso, outros critérios são necessários para a formação de tuplas negativas que apresentem uma diversificação de similaridades. Em classificação supervisionada, o sucesso da aprendizagem depende de dados representativos que permitam compreender as relações entre os dados de entrada e de saída, possibilitando uma generalização eficaz do classificador (JAIN et al., 2020).

Nesta tese, foram analisadas duas abordagens para a seleção de tuplas negativas: a seleção aleatória e a seleção por similaridade de descrição dos produtos.

A seleção aleatória de tuplas negativas pode ser realizada dentro das categorias dos produtos ou em categorias distintas. Nessa abordagem, basta escolher aleatoriamente produtos que não possuem os mesmos identificadores (GTIN). Para a base de dados Notas_Fiscais, a categorização dos produtos pode ser realizada por meio do NCM. Essa abordagem de seleção aleatória pode resultar em um corpus que não representa os dados reais encontrados em uma aplicação prática de correspondência de produtos, o que pode levar ao treinamento de modelos enviesados. Esse fator torna-se ainda mais relevante ao se considerar títulos curtos de produtos, nos quais uma simples palavra, letra ou número pode representar descrições distintas, como, por exemplo, “leite desnatado molico 1l” e “leite desnatado parmalat 1l”.

A outra abordagem de seleção de tuplas negativas é baseada em critérios de similaridade das descrições dos produtos. Nesse caso, almeja-se formar tuplas negativas com o maior grau de similaridade possível, permitindo que os algoritmos de aprendizagem de máquina captem nuances nas descrições dos produtos, diferenciando produtos distintos com descrições similares. Na literatura, o termo *hard negative* é utilizado para descrever os dados que são mais difíceis de serem classificados corretamente (KIM et al., 2022; ZHUANG et al., 2020).

A similaridade de duas descrições de produtos P_1 e P_2 é calculada por uma função de similaridade $F_{Sim} : P_1 \times P_2 \mapsto R$, com um limiar de similaridade $0 \leq \theta \leq 1$, onde $F_{Sim}(P_1, P_2) = 0$ indica nenhuma similaridade e $F_{Sim}(P_1, P_2) = 1$ indica similaridade máxima. Assim, é possível identificar a similaridade de todos os pro-

dutos, gerando as tuplas negativas com os respectivos graus de similaridade, isto é, $(P_i, P_j) : F_{Sim}(P_i, P_j) \geq 0$ e $P_i \neq P_j$, onde, nesse contexto, $P_i \neq P_j$ indica produtos com identificadores distintos.

Assim, para um determinado produto, pode-se obter um conjunto de itens similares, gerando tuplas negativas com um elevado grau de similaridade para serem usadas no treinamento, validação e teste dos modelos de classificação de produtos. Essa estratégia de tuplas *hard negative* contribui para aumentar a capacidade do modelo de identificar corretamente produtos não correspondentes em cenários reais, em que as diferenças entre descrições podem ser sutis e difíceis de detectar.

Neste trabalho, foi utilizado o algoritmo BM25 implementado no motor de busca Elasticsearch, para indexar as bases de dados de todos os produtos. Nesse caso, a função de similaridade F_{Sim} representa o algoritmo BM25. A busca de produtos neste ambiente tem por objetivo selecionar as tuplas negativas com maior grau de similaridade entre os produtos de interesse. Ou seja, para cada tupla de correspondências (tuplas positivas) do corpus, foram geradas **k** tuplas negativas contendo os produtos mais semelhantes da base de dados.

Nos experimentos, os valores de **k** foram 1 e 5, ou seja, para cada tupla positiva, foram geradas proporções de tuplas negativas da seguinte forma:

- Balanceado: 1:1. Para cada tupla positiva, há uma tupla negativa contendo o produto diferente mais semelhante possível; e
- Desbalanceado: 1:5. Para cada tupla positiva, há cinco tuplas negativas contendo os produtos diferentes mais semelhantes possíveis.

Em determinadas situações de aplicações reais, principalmente associadas à limpeza de dados conflitantes, deseja-se maximizar a identificação de produtos não correspondentes. Por isso, optou-se por explorar os corpora com dados desbalanceados nos experimentos realizados.

5.2.3 Preparação dos Corpora - Pré-Processamento

Com o objetivo de uniformizar os dados rotulados na etapa de Construção dos Corpora, realiza-se o pré-processamento dos dados textuais, associados aos títulos de produtos. Nessa etapa, a tarefa de pré-processamento ficou restrita à normalização dos caracteres, conversão dos textos em caracteres minúsculos e remoção de caracteres especiais que apresentavam certa recorrência nas descrições e que não tinham importância

significativa para a diferenciação de palavras, tais como “(”, “)”, “[”, “]”, “-”, “;”, entre outros. O Apêndice B apresenta as funções utilizadas neste pré-processamento.

Ademais, considerando os produtos da base de dados das notas fiscais, foram necessárias algumas tarefas adicionais no pré-processamento, conforme o diagrama da Figura 5.4, a saber:

- Remoção de produtos com GTINs inválidos: o formato GTIN possui regras específicas de construção, tornando possível validar código de barras;
- Remoção de descrições repetidas: os estabelecimentos emitem várias notas fiscais dos seus produtos, resultando em registros repetidos de descrições de produtos nas notas fiscais, podendo, inclusive, serem de estabelecimentos diferentes. Nesse processo de remoção, foi adicionado um atributo para registrar a quantidade de repetições da mesma descrição por GTIN. Esse atributo pode ser utilizado para definir descrições mais relevantes em termos de frequência absoluta ou relativa (considerando descrições por estabelecimento); e
- Remoção de NCMs inválidos: produtos que contêm NCMs que são compostos unicamente por zeros.

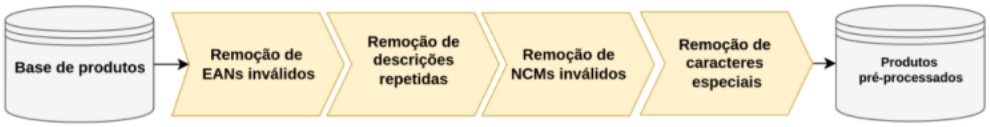

Figura 5.4 – Etapas de pré-processamento do corpus de produtos das notas fiscais

É importante ressaltar que as remoções realizadas nessa etapa foram apenas para a construção dos corpora, compostos por pares de produtos rotulados com as indicações de correspondências ou não, com o intuito de treinar e avaliar as técnicas utilizadas.

5.3 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo apresentou os dados de produtos utilizados na pesquisa, bem como a metodologia utilizada para a geração de corpora dos produtos. No próximo capítulo, serão detalhados os experimentos realizados com os corpora construídos, seguindo a metodologia apresentada, bem como as discussões dos resultados obtidos.

6 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados dos experimentos realizados para avaliar o STEPMatch - *Short Text Product Matching*, proposto nesta tese. A seção 6.1 descreve e caracteriza os conjuntos de dados utilizados nos experimentos. Em seguida, na seção 6.2, são apresentados os corpora construídos. A seção 6.3 apresenta as configurações do ambiente de execução dos experimentos e as bibliotecas utilizadas no treinamento dos modelos de aprendizagem de máquina. A seção 6.4 discute as avaliações de modelos de aprendizagem de máquina para realizar a tarefa de correspondência de produtos. Na seção 6.5, são avaliados os mecanismos de busca de produtos correspondentes em um ambiente de recuperação da informação. Na seção 6.6, é realizada uma avaliação integral do STEPMatch considerando uma situação real. Finalmente, a seção 6.7 apresenta as considerações sobre o capítulo.

6.1 CARACTERIZAÇÃO DOS CONJUNTOS DE DADOS

Nos experimentos realizados, foram utilizadas três bases de dados distintas, conforme descrito no Capítulo 5. A Tabela 6.1 apresenta um resumo das informações relativas à origem, idioma dos dados, quantidade de produtos únicos, que são identificados por meio do código GTIN, e o número de descrições distintas contidas em cada uma das bases de dados.

Tabela 6.1 – Bases de dados de produtos

Origem do Produtos	Idioma	Produtos Únicos	Descrições Distintas
Notas Fiscais	Português	578.640	942.447
eCommerce	Português	4.500	6.975
WDC Products medium	Inglês	745	3.337

A base de dados de Notas Fiscais abrange uma ampla diversidade de produtos comercializados no estado do Acre, ao longo de um período de três meses, conforme descrito no Capítulo 5. Para o escopo desta pesquisa, foi selecionado um subconjunto de categorias, priorizando aquelas com maior representatividade em termos de quantidade de produtos. A partir da classificação do NCM, foram escolhidas três categorias com o maior volume de itens, como apresentadas na Tabela 6.2, as quais foram utilizadas nos experimentos.

Para a categoria de Leites e Laticínios, que possui a maior quantidade de produtos, foi utilizado um recurso externo para enriquecer a base de dados, inserindo outros títulos de produtos da categoria. Os dados de produtos foram extraídos de forma automatizada do site Bluesoft, por meio da técnica de *web scraping*. O código desenvolvido para a extração desses dados encontra-se detalhado no Apêndice A.

Tabela 6.2 – Categorias de Produtos da Base de produtos de Notas fiscais

Categoria	NCM	Quantidade de Produtos
Cosméticos-Outros	3305.90.00	1.290
Leites e Laticínios	0401.10.10	1.849
Veículos-Suspensão	8708.80.00	648
	Total	**3.787**

Com o intuito de proporcionar uma melhor compreensão sobre as características dos dados, especialmente no que se refere ao tamanho dos títulos dos produtos, a Figura 6.1 apresenta os gráficos de *boxplot*, mostrando as distribuições do número de caracteres e de palavras nos títulos das bases de dados. Observa-se que, com exceção dos *outliers*, os títulos dos produtos nas notas fiscais são, de modo geral, mais curtos que os das bases de dados de *eCommerce* e *WDC Products*.

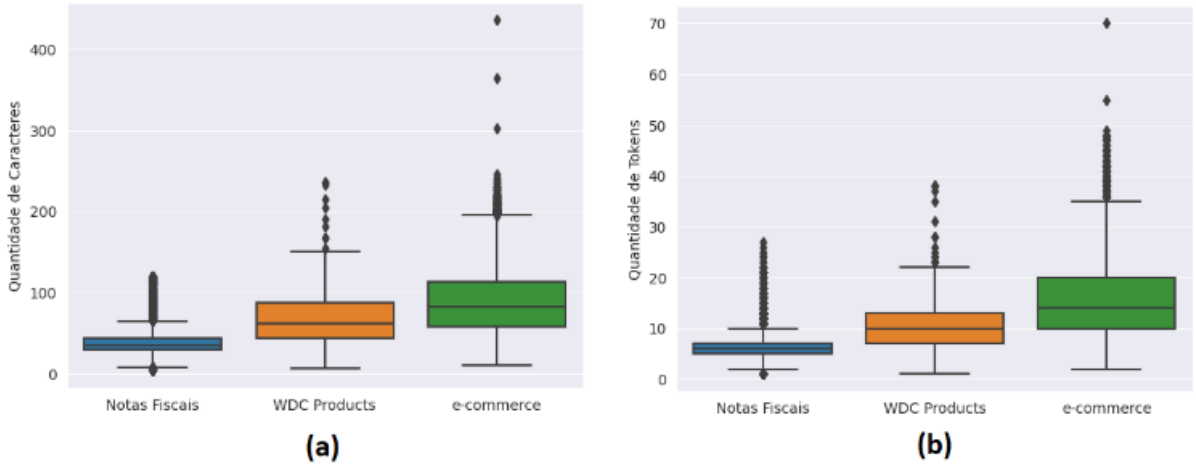

Figura 6.1 – BoxPlot das Distribuições dos Títulos de Produtos em Relação à Quantidade de: a) Caracteres; b) Palavras

Ainda relacionado aos tamanhos dos títulos dos produtos, os gráficos das Figuras 6.2 e 6.3 apresentam, respectivamente, as distribuições das quantidades de caracteres e palavras (*tokens*) dos produtos de cada uma das bases de dados, por meio dos histogramas de frequência relativa. Observam-se características semelhantes nos títulos de produtos contidos nas bases de dados WDC Products e eCommerce. De fato, ambos referem-se a títulos presentes em sites de comércio eletrônico, distinguindo-se apenas quanto ao idioma.

Essa caracterização de títulos dos produtos das bases de dados é importante para destacar as particularidades dos títulos presentes nas notas fiscais. Em geral, eles são mais curtos, apresentando um número menor de caracteres e palavras, quando comparados aos títulos de produtos que constam em sites de comércio eletrônico. Esse fato sugere que os produtos podem conter muitas abreviações, siglas, palavras truncadas e ausência de informações relevantes, o que aumenta ainda mais a complexidade da tarefa de correspondência de produtos.

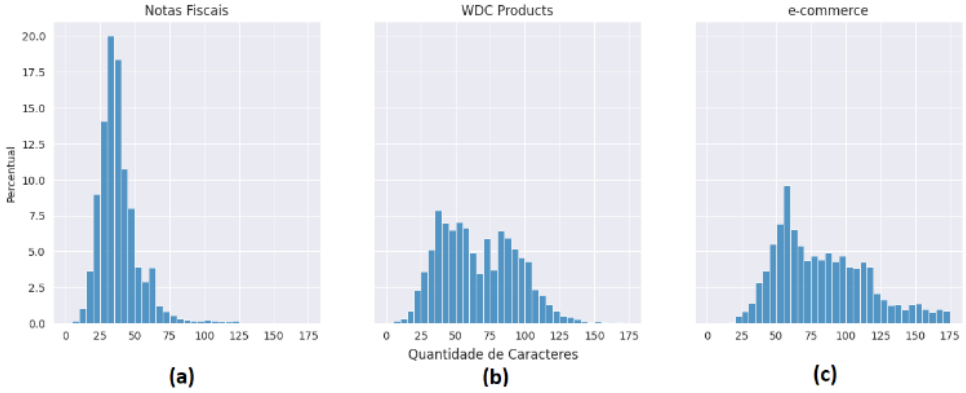

Figura 6.2 – Distribuição percentual da quantidade de caracteres nos títulos dos produtos: (a) Notas Fiscais (b) WDC Products (c) eCommerce

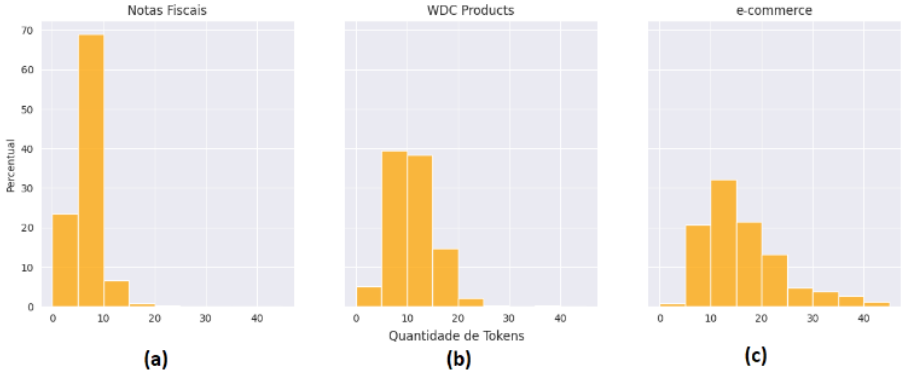

Figura 6.3 – Distribuição percentual da quantidade de palavras nos títulos dos produtos: (a) Notas Fiscais (b) WDC Products (c) eCommerce

6.2 CRIAÇÃO DOS CORPORA

Os corpora foram criados seguindo a metodologia proposta no Capítulo 5. Assim, considerando as bases de produtos definidas anteriormente, para a definição de tuplas negativas (não correspondentes), foram adotadas duas estratégias: seleção aleatória e por critérios de similaridades.

Na estratégia de seleção aleatória, formaram-se pares de produtos com identificadores diferentes dentro da mesma categoria. Por sua vez, na estratégia de seleção por critérios de similaridade, denominada aqui de *Hard Negative*, o algoritmo BM25, implementado no Elasticsearch[1], foi utilizado para formar os pares de produtos não correspondentes mais semelhantes possíveis, conforme o critério de ranqueamento do algoritmo.

O Quadro 6.1 apresenta exemplos de tuplas negativas geradas pelas estratégias de seleção aleatória e *Hard Negative* na base de dados de Notas Fiscais.

[1] <https://www.elastic.co/pt/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables>

Quadro 6.1 – Exemplo de Pares de Tuplas Negativas por estratégia

Estratégia	Título 1	Título 2
Hard Negative	leite condensado zero lactose itambé nolac caixa 395g	leite condensado zero lactose piracanjuba caixa 395g
	leite condensado zero lactose itambé nolac caixa 395g	leite condensado zero lactose frimesa caixa 395g
	leite condensado zero lactose itambé nolac caixa 395g	leite condensado semidesnatado zero lactose moça caixa 395g
	leite em pó integral 1kg italac	leite em pó integral 400g italac
	leite em pó integral 1kg italac	leite em pó integral 200g italac
	leite em pó integral 1kg italac	leite em pó integral camponesa – 1kg
	leite em pó integral 1kg italac	leite em pó itambé integral 1kg
	leite em pó integral 1kg italac	leite em pó integral instantâneo camponesa – 1kg
Aleatório	creme de leite leve itambé caixa 200g	leite italac uht tp 1l integral un qtd. 15.00 un
	creme de leite leve itambé caixa 200g	leite rosca italac 1l integral
	creme de leite leve itambé caixa 200g	leite lv nilza 1lt semidesnatado
	creme de leite leve itambé caixa 200g	leite uht semidesnat 1l piracanjuba
	leite condensado semidesnatado moça caixa 340g	leite ninho uht 1l semidesnatado un qtd. 12.00 un
	leite condensado semidesnatado moça caixa 340g	leite lv nilza 1lt semidesnatado

A definição de tuplas positivas (correspondentes) foi feita agrupando as diferentes descrições dos produtos das bases de dados, com base nos seus identificadores únicos.

O corpus WDC Products, por ser um *benchmark* para tarefas de *Product Matching*, contém pares de produtos anotados, incluindo a divisão dos dados em conjuntos de treinamento, validação e teste. No entanto, a partir desse corpus, foi criado um novo conjunto de dados utilizando a estratégia *Hard Negative*.

Os detalhes dos corpora construídos com base nas estratégias descritas na metodologia estão apresentados na Tabela 6.3. Foram construídos corpora tanto de forma balanceada quanto desbalanceada. Nos corpora desbalanceados, priorizou-se a classe de produtos não correspondentes, uma vez que essa configuração reflete cenários de aplicações reais, em que o foco é a busca de correspondências.

Para o processo de construção e avaliação dos classificadores, os dados foram divididos em 70% para treinamento, 15% para validação e 15% para teste.

6.3 CONFIGURAÇÕES DO AMBIENTE E PARÂMETROS DE TREINAMENTO DOS MODELOS DE CLASSIFICAÇÃO

Todos os experimentos realizados nesta pesquisa foram feitos em um computador dedicado com as seguintes configurações: Intel(R) Core(TM) i7-10700KF CPU @ 3.80GHz com 64GB de RAM, placa de vídeo NVIDIA GeForce RTX 4090 de 24 GB e com o Sistema Operacional Pop!_OS 20.04 LTS.

Tabela 6.3 – Corpora anotados

Base de Dados	Categoria	Estratégia	Tamanho	*Matches*	*No Matches*
WDC	computers	Benchmark Original	3.934	1.022	2.912
	computers	*Hard Negative*	33.370	12.656	20.714
eCommerce	Diversas Categorias	Aleatório 1:1	8.632	4.316	4.316
		Aleatório 1:5	25.896	4.316	21.580
		Hard Negative	69.740	9.972	59.768
Notas Fiscais	Leites e Laticínios	Aleatório 1:1	28.564	14.282	14.282
		Aleatório 1:5	85.692	14.282	71.410
		Hard Negative	18.462	3.729	14.733
	Cosméticos- Outros	Aleatório 1:1	3.662	1.831	1.831
		Aleatório 1:5	10.986	1.831	9.155
		Hard Negative	12.890	2.234	10.656
	Veículos-Suspensão	Aleatório 1:1	1404	702	702
		Aleatório 1:5	4.212	702	3.510
		Hard Negative	6.480	738	5.742

Os experimentos foram realizados utilizando a linguagem de programação Python (versão 3.7), utilizando-se de algumas bibliotecas, dentre as quais destacam-se: scikit-learn[2], xgboost[3], auto-sklearn[4], PyTorch[5] e *transformers*[6].

Para construir modelos de classificação utilizando *MLPTs,* foi utilizada a biblioteca *transformers* para gerar os *tokens* de entrada e carregar os modelos. Os vetores de entrada foram configurados com uma dimensão máxima de 64 *tokens*, atendendo à necessidade dos *MLPTs* de trabalharem com tamanhos fixos de *tokens*. Esse limite foi estabelecido considerando que a entrada corresponde à concatenação de dois títulos de produtos. Conforme a Seção 6.1, os títulos de produtos, com exceção dos *outliers*, apresentam até 30 *tokens* para produtos de comércio eletrônico (WDC e eCommerce) e até 12 *tokens* para produtos de notas fiscais. Entradas menores que esse limite foram preenchidas com zeros (0's), enquanto entradas maiores foram truncadas, descartando os elementos excedentes.

Foram avaliados seis diferentes MLPTs baseados em Transformers disponíveis na biblioteca HuggingFace: BERT em inglês e BERT-Multilingual (DEVLIN et al., 2019), XLM-RoBERTa (CONNEAU et al., 2021), BERTimbau (SOUZA et al., 2020), Albertina-PT-BR (RODRIGUES et al., 2023), e e-CommerceBERT[7]. Entre esses, os modelos monolíngues incluem o BERT em inglês e os modelos em português BERTimbau e Albertina-PT-BR. Enquanto isso, os modelos multilíngues são BERT-Multilingual, XLM-RoBERTa e e-Commerce-BERT. Além do mais, para estimar a incerteza do modelo, foi empregada uma estratégia de *Bootstrapping*. Assim, as métricas dos resultados de avaliação dos modelos

[2] https://scikit-learn.org/
[3] https://github.com/dmlc/xgboost
[4] https://automl.github.io/auto-sklearn
[5] https://pytorch.org/
[6] <https://github.com/huggingface/transformers>
[7] <https://huggingface.co/EZlee/e-commerce-bert-base-multilingual-cased>

foram obtidas a partir da execução de 10 repetições.

A configuração do treinamento dos modelos envolveu uma taxa de aprendizado ajustada para 1×10^{-5}, com o otimizador AdamW configurado com $epsilon = 1 \times 10^{-8}$. A função de perda utilizada foi a entropia cruzada binária, e a função de ativação empregada foi o *softmax*. Durante o ajuste fino, o número de épocas foi definido com base no corpus utilizado, priorizando os dados de validação. Essa abordagem visou evitar o enviesamento do modelo e garantir uma generalização adequada aos novos dados durante a fase final de testes.

6.4 AVALIAÇÃO DOS CLASSIFICADORES PARA CORRESPONDÊNCIA DE PRODUTOS

Os experimentos nesta seção objetivam avaliar técnicas de aprendizagem de máquina para realizar a classificação de correspondências de produtos. O propósito desta avaliação é escolher um modelo de classificação a ser utilizado na função *isMatch*(), utilizada no Algoritmo 2 da Etapa 2 do framework proposto no capítulo 4.

Os experimentos realizados nesta seção visam a:

- Comparar MLPTs e técnicas tradicionais de aprendizagem supervisionada de máquina;
- Avaliar a utilização de CLL no contexto de descrições de produtos para indução de modelos de classificação;
- Avaliar estratégias de criação de corpora.

6.4.1 Técnicas Tradicionais de Aprendizagem Supervisionada de Máquina para Correspondência de Produtos

O primeiro experimento realizado teve como objetivo identificar o modelo tradicional de classificação com o melhor desempenho na tarefa de classificar descrições de produtos correspondentes. Os vetores que representam as características utilizadas para treinar os modelos foram gerados conforme descrito na Seção 4.3.1. Em seguida, utilizou-se a biblioteca Auto-sklearn para a construção automatizada dos modelos de classificação.

O Auto-sklearn possibilitou a comparação entre diversos algoritmos tradicionais de aprendizagem de máquina, como Random Forest, XGBoost e SVM, e a otimização dos hiperparâmetros de cada algoritmo, simplificando o processo de seleção do modelo. As Tabelas 6.4 e 6.5 apresentam, respectivamente, os resultados das métricas obtidas a

partir dos melhores modelos tradicionais, treinados no corpus de Notas Fiscais — categoria "Leites e Laticínios" (*Hard Negative*) — e no corpus WDC Products (Benchmark Original).

Tabela 6.4 – Resultados do Modelo Tradicional de Aprendizagem de Máquina: Corpus Nota Fiscal — Categoria Leites e Laticínios — Hard Negative

Modelo	Acurácia	Precisão	Recall	F1-score
XGBClassifier	95,8%	95,1%	94,6%	94,9%

Tabela 6.5 – Resultados do Modelo Tradicional de Aprendizagem de Máquina - Corpus WDC (Benchmark original).

Modelo	Acurácia	Precisão	Recall	F1-score
XGBClassifier	89,3%	89,5%	88,6%	89,1%

A Tabela 6.6 apresenta os resultados detalhados por classe para o classificador XGBoost, obtidos por meio da avaliação do modelo com validação cruzada *k-fold* ($k = 5$).

Tabela 6.6 – Resultados do Classificador XGBoost - $5 - fold$

Corpus		Precisão	Recall	F1-score	Acurácia
WDC Producs: Original	*No Match*	93,4%	90,5%	91,4%	87,4%
	Match	74,1%	78,4%	76,2%	
	Média	87,7%	87,4%	87,5%	
	Desvio Padrão	1,0%	1,1%	1,1%	
Nota Fiscal: Leite e Laticínios - Hard Negative	*No Match*	98,2%	97,7%	97,9%	96,7%
	Match	91,1%	92,8%	91,8%	
	Média	96,7%	96,7%	96,7%	
	Desvio Padrão	0,3%	0,3%	0,3%	

6.4.2 Comparação de Modelos para Correspondência de Produtos

Este experimento visa a comparar o desempenho do modelo XGBoost com classificadores que utilizam MLPT. Os modelos foram construídos realizando ajustes finos dos MLPTs específicos de cada idioma do corpus.

A Tabela 6.7 apresenta a média do F1-score de cada modelo de classificação, treinado com alguns dos corpora apresentados na Tabela 6.3, considerando a aplicação de validação cruzada *k-fold*, com $k = 5$.

Na sequência, a questão de pesquisa Q1 será respondida com base na análise dos resultados dos experimentos apresentados nesta seção.

Questão de Pesquisa:

- **Q1:** A utilização de Modelos de Linguagens Pré-Treinados apresenta resultados melhores na tarefa de correspondência de produtos com descrições curtas quando

Tabela 6.7 – Comparação dos classificadores: XGBoost e MLPTs

Corpus		Classificador	F1-score (Média)	Desvio Padrão
WDC	computers Original	XGBoost	89.1%	1.1
		BERT	93.6%	0.7
		BERT-Multilingual	**92.6%**	0.8
	computers hard negative	XGBoost	94.3%	0.9
		BERT	**98.2%**	0.8
		BERT-Multilingual	97.9%	0.8
eCommerce	Diversas Categories hard negative	XGBoost	94.5%	0.7
		BERTimbau	**96.4%**	0.4
		BERT-Multilingual	95.7%	0.5
Nota Fiscal	Leites e Laticínios hard negative	XGBoost	94.3%	0.8
		BERTimbau	**98.5%**	0.5
		BERT-Multilingual	96.3%	0.5
	Cosméticos hard negative	XGBoost	93.4%	0.8
		BERTimbau	**97.3%**	0.4
		BERT-Multilingual	95.7%	0.5

comparados com técnicas tradicionais de aprendizagem de máquina em que as *features* são medidas de similaridade de textos?

Diante dessa questão de pesquisa, os resultados obtidos nos experimentos realizados confirmaram que, sim, a utilização de MLPT apresenta um desempenho superior na tarefa de correspondência de produtos. Destacaram-se duas principais considerações:

1. Para os títulos de produtos nos corpora WDC e eCommerce, que apresentam descrições mais detalhadas, os MLPTs mostraram-se mais eficazes em lidar com as variações na forma de descrição dos produtos, alcançando resultados superiores ao XGBoost. Além disso, os modelos de linguagens monolíngues tiveram melhor desempenho em corpora que correspondiam ao idioma para o qual os modelos foram treinados.
2. Para os títulos de produtos em português presentes nas notas fiscais, também foi observado um melhor desempenho dos MLPTs, com destaque para o BERTimbau, modelo treinado especificamente com dados do idioma português.

Os resultados observados na Tabela 6.7 indicam que os MLPTs ajustados para a tarefa de classificação apresentaram desempenho superior em todos os testes. Os MLPTs avaliados, baseados na arquitetura do BERT, demonstraram uma capacidade de capturar tanto as representações sintáticas quanto semânticas dos textos, proporcionando uma compreensão mais detalhada das descrições de produtos, inclusive das descrições mais curtas, como as que constam nas notas fiscais.

6.4.3 Avaliação de Abordagens de Aprendizagem por Cruzamentos de Idiomas no contexto de descrições de produtos

Esta subseção avalia o uso de Aprendizagem por Cruzamentos de Idiomas (CLL) no contexto de correspondências de produtos. O objetivo é avaliar o desempenho do uso de corpora anotados de um idioma específico para construir modelos de classificação para idiomas diferentes, por meio da transferência de aprendizado. Foram avaliadas as estratégias ZST, JL, CL, JL/CL e JL/CL+, conforme detalhadas na seção 4.3.2.

A abordagem a ser analisada é a indução de um modelo de classificação através do idioma inglês como fonte de dados, para aproveitar os corpora de produtos anotados disponíveis, realizando, assim, ajustes finos com uma porção menor de dados do idioma destino, o idioma português.

Inicialmente, para avaliação das estratégias de CLL, foram explorados os corpora eCommerce (português) e WDC Products (inglês), que são relacionados a produtos oriundos do comércio eletrônico e apresentam características semelhantes, conforme apresentado na seção 6.1. Os dados de teste foram fixados para possibilitar uma avaliação comparativa entre todos os modelos treinados. Assim, para os produtos do eCommerce (*Hard Negative*), uma amostra dos pares de produtos rotulados foi dividida aleatoriamente em 70%, 15% e 15%, respectivamente, para treinamento, validação e teste. Para o corpus WDC Products, foram utilizados os conjuntos fornecidos no *benchmark*. A Tabela 6.8 apresenta os dados utilizados nos experimentos.

Tabela 6.8 – Corpora de Produtos usados no Experimentos de CLL

Corpora	Treino		Validação		Teste	
	Match	No Match	Match	No Match	Match	No Match
WDC Products	1410	5065	352	1267	300	800
BR	1014	4962	195	1086	214	1067

Em relação ao ajuste dos parâmetros do modelo, dependendo da estratégia adotada, apenas uma fração específica do conjunto de treinamento é utilizada. Uma estratégia de *Bootstrapping* foi empregada para estimar a incerteza do modelo, o qual foi treinado e avaliado com 10 repetições.

Para a entrada dos modelos, foi utilizada uma tokenização limitada a 128 tokens. Após a análise dos dados na Figura 6.1, constatou-se que a maioria dos títulos de produtos continha menos de 60 palavras (considerando a entrada com Título 1 + Título 2). Essa escolha visou a otimizar o desempenho do experimento e reduzir os custos computacionais. Textos com comprimento inferior a 128 tokens foram preenchidos com 0's, enquanto textos maiores foram truncados para respeitar o limite estabelecido.

Para avaliar as estratégias de CLL na tarefa de correspondência de produtos, inicialmente, realizou-se um experimento base para comparar a eficácia das estratégias de CLL. Esse experimento envolveu o treinamento e a avaliação de modelos utilizando dados de um único idioma. Em seguida, foram conduzidos experimentos adicionais para avaliar o impacto das estratégias de CLL na correspondência de produtos. Em todos os experimentos, os modelos foram avaliados utilizando um conjunto de dados de teste do idioma alvo que não foi empregado durante o treinamento dos modelos.

O primeiro experimento analisou os resultados de diferentes MLPTs, monolíngues e multilíngues, utilizando corpora específicos de cada idioma. Os modelos criados neste experimento não utilizaram abordagens de CLL e os resultados obtidos constituem uma base para comparação com outros modelos que utilizaram estratégias de CLL.

A Tabela 6.9 apresenta os resultados do F1-Score para os modelos de referência, em que o ajuste dos modelos para cada MLPT foi realizado utilizando todos os dados de treinamento e validação dos corpora disponíveis em cada idioma, e os resultados foram obtidos com os dados de teste. Os resultados indicam que o modelo Albertina-PT-BR, ajustado e testado com dados do eCommerce, obteve o melhor desempenho, com um F1-Score de 95,3%. Observou-se que modelos monolíngues treinados com dados em inglês alcançaram bons resultados, mesmo quando comparados com outros modelos multilíngues. O modelo Albertina-PT-BR apresentou o melhor desempenho, alcançando um F1-Score de 92,8%, o que demonstra sua capacidade de adaptação também aos dados em inglês. Esse resultado reflete, em parte, o impacto do extenso volume de dados utilizados em seu treinamento, aliado à sua arquitetura robusta, composta por aproximadamente 900 milhões de parâmetros distribuídos em 24 camadas. Já o modelo eCommerceBERT obteve o segundo melhor resultado para os dados do eCommerce.

Tabela 6.9 – Modelos de Referência - Sem CLL

	F1 Score (%)	
MLPT / Dataset	WDC-Product English	E-commerce Portuguese
BERT-base	90.1 (±0.0048)	94.7 (±0.0053)
BERT-Multilingual	92.3 (±0.0064)	94.9 (±0.0061)
XLM-Roberta	89.2 (±0.0049)	94.7 (±0.0051)
BERTimbau	91.3 (±0.0048)	94.8 (±0.0047)
Albertina-PT-BR	**92.8 (±0.0054)**	**95.3 (±0.0051)**
e-commerce-BERT	91.0(±0.0065)	95.1 (±0.0068)

Os experimentos subsequentes concentraram-se na análise de estratégias de CLL para aprimorar os resultados obtidos pelos MLPTs com dados de produtos em português.

Nas estratégias JL e CL, foram examinados os resultados dos MLPTs por meio da utilização de dados limitados do idioma alvo no ajuste do modelo. Nesse cenário, foi avaliada a utilização de até 50% dos dados de treinamento disponíveis para o corpus em português (alvo), começando com 0% (ZST) e aumentando em incrementos de 10% até o limite estabelecido.

A Tabela 6.10 apresenta os experimentos das estratégias ZST e JL. Na estratégia ZST, na qual nenhum dado do idioma português foi utilizado durante o treinamento do modelo, o melhor resultado foi alcançado pelo BERT-Multilingual, com um F1-Score de 78,9%. Ao empregar a estratégia de adicionar uma parte dos dados do idioma alvo no treinamento inicial do modelo (JL), observou-se que, com apenas 10% dos dados de treinamento do idioma alvo, correspondendo a aproximadamente 600 pares de produtos rotulados, todos os modelos alcançaram cerca de 90% de F1-Score. Notavelmente, observa-se uma melhoria significativa no desempenho do XLM-Roberta, que saltou de 57,8% para 90,4%, resultando em um ganho de 32,5% no F1-Score.

Em relação aos resultados das estratégias JL, observou-se que o melhor resultado foi obtido com o BERT-Multilingual utilizando 50% dos dados do idioma português no treinamento inicial, alcançando um F1-Score de 93,8%. No entanto, com apenas 30% dos dados, o Albertina-PT-BR atingiu um valor semelhante, resultando em um F1-Score de 93,3% (uma diferença de 0,3%).

Tabela 6.10 – Resultados de F1-Score: Estratégias ZST e JL

	Dados de Treinamento do Idioma Alvo					
MLPT / Base de Dados	ZST 0%	JL 10%	JL 20%	JL 30%	JL 40%	JL 50%
BERT-base	62.5	88.5	89.9	91.8	91.3	93.2
BERT-Multilingual	**78.9**	89.6	90.8	91.3	92.9	**93.8**
XLM-Roberta	57.8	**90.4**	**91.3**	90.3	91.7	92.8
BERTimbau	73.5	88.2	90.2	91.5	91.9	92.1
Albertina-PT-BR	75.0	89.4	91.0	**93.6**	**93.3**	93.6
e-commerce-BERT	68.3	87.8	90.0	92.4	92.5	92.4

A Tabela 6.11 apresenta os resultados da estratégia CL. De forma semelhante à estratégia JL, observou-se que, com 10% dos dados do idioma alvo, todos os modelos alcançaram resultados de F1-Score próximos a 90%. O melhor resultado foi obtido utilizando 50% dos dados do idioma alvo no segundo ajuste fino dos modelos BERT-base e BERT-Multilingual.

A Tabela 6.12 apresenta os resultados das combinações das estratégias JL e CL. Ao utilizar no máximo 50% dos dados de treinamento do idioma desetino, o melhor

Tabela 6.11 – Resultados de F1-Score: Estratégia CL

	Dados de Treinamento do Idioma Alvo				
MLPT / Estratégia	CL 10%	CL 20%	CL 30%	CL 40%	CL 50%
BERT-base	87.4	89.9	91.2	92.7	**94.2**
BERT-Multilingual	89.3	91.6	91.6	92.5	**94.2**
XLM-Roberta	**90.5**	**92.4**	92.3	91.7	92.6
BERTimbau	88.8	90.9	91.5	92.1	92.3
Albertina-PT-BR	89.6	92.2	**93.4**	**93.4**	93.2
e-commerce-BERT	86.7	89.8	91.2	92.3	93.2

resultado alcançado foi de 94,2%, com o uso de 30% JL e 20% CL no modelo XLM-Roberta. Resultados semelhantes foram obtidos com os modelos BERT-base e BERT-Multilingual utilizando apenas a estratégia CL. Por outro lado, ao utilizar todos os dados de treinamento do idioma alvo, o melhor resultado foi obtido com o BERT-Multilingual, alcançando um F1-Score de 96,5%, treinado com ambas as estratégias JL e CL, cada uma com 50% dos dados. Esse resultado representou uma melhoria de 1,3%, em comparação com o modelo de referência.

Tabela 6.12 – Resultados de F1-Score: Estratégia JL/CL

	50% Dados de Treinamento do Idioma Alvo			100% Dados de Treinamento do Idioma Alvo		
MLPT / Base de Dados	JL 10% CL 20%	JL 20% CL 30%	JL 30% CL 20%	JL 30% CL 70%	JL 40% CL 60%	JL 50% CL 50%
BERT-base	90.1	94.0	92.4	95.8	95.4	95.5
BERT-Multilingual	91.1	**94.1**	92.8	**96.2**	95.5	**96.5**
XLM-Roberta	**92.4**	93.4	**94.2**	95.6	95.5	95.4
BERTimbau	90.1	93.1	92.9	94.9	95.5	94.7
Albertina-PT-BR	91.5	93.4	93.4	95.5	**95.8**	96.4
e-commerce-BERT	91.1	92.0	92.3	95.9	95.7	95.3

A Tabela 6.13 apresenta os resultados de várias estratégias JL/CL+. O melhor resultado foi um F1-Score de 95,4%, alcançado pelos modelos Albertina-PT-BR e e-commerce-BERT. Esses resultados foram obtidos ajustando os modelos com um treinamento inicial que utilizou 30% dos dados na estratégia JL e, na estratégia CL, utilizando 20% e 50%. Nessa estratégia, foi utilizado todo o conjunto de dados de treinamento do idioma alvo.

A Tabela 6.14 oferece uma análise detalhada das métricas de avaliação dos modelos que alcançaram os melhores desempenhos nos experimentos. Esses modelos incluem:

Tabela 6.13 – Resultados de F1-Score: Estratégia JL/CL+

	100% dos Dados de Treinamento do Idioma Alvo			
MLPT / Base de Dados	JL 10% CL 20% CL 70%	JL 20% CL 30% CL 50%	JL 30% CL 20% CL 50%	JL 30% CL 20% CL 25% CL 25%
BERT-base	93.1	94.5	94.1	93.4
BERT-Multilingual	**94.9**	94.3	93.0	93.8
XLM-Roberta	94.6	94.3	94.1	93.4
BERTimbau	92.9	94.3	93.6	93.6
Albertina-PT-BR	94.5	94.6	**95.4**	94.6
e-commerce-BERT	94.5	**95.7**	**95.4**	**95.4**

1. O modelo de referência, que foi treinado sem a aplicação da técnica CLL, utilizando o MLPT Albertina-PT-BR;

2. O modelo treinado com o MLPT BERT-Multilingual utilizando a estratégia JL/CL. Nesse caso, o ajuste inicial utilizou os dados do idioma fonte combinados com 50% dos dados do idioma alvo, enquanto o segundo ajuste fino utilizou os 50% restantes dos dados do idioma alvo;

3. O modelo treinado com o MLPT e-commerce-BERT utilizando a estratégia JL/CL+. Nesse cenário, o ajuste inicial utilizou os dados do idioma fonte combinados com 20% dos dados do idioma alvo. Os ajustes subsequentes, especificamente o segundo e o terceiro, incorporaram, respectivamente, os 30% e 50% restantes dos dados do idioma alvo.

Tabela 6.14 – Comparação entre Modelo de Referência e o Modelo com uso de CLL que apresentou o melhor resultado

MLPT	Estratégia	Métricas*		
		F1	Recall	Precisão
Albertina-PT-BR	Modelo de Referência (sem CLL)	95.3 ±0.0053	95.3 ±0.0049	95.4 ±0.0052
BERT-Multilingual	CLL JL 50%+CL 50%	**96.5** ±0.0064	**96.6** ±0.0060	**96.5** ±0.0062
e-commerce -BERT	CLL JL 20%+CL 30% + CL 50%	94.5 ±0.0074	94.6 ±0.0068	94.5 ±0.0070

* Valores Médios com o desvio padrão - Nível de Confiança de 95% - Obtidos a partir de 10 Execuções.

Os valores apresentados na Tabela 6.14 foram obtidos a partir de 10 execuções dos experimentos, cada uma utilizando frações aleatórias dos conjuntos de dados de

treinamento. Esse procedimento foi essencial para avaliar a variabilidade dos modelos e proporcionou uma análise abrangente de seus desempenhos.

Além disso, a Figura 6.4 apresenta os *boxplots* das métricas F1-Score, Precisão e Recall, detalhando as distribuições dos valores gerados pelos modelos, considerando várias iterações do processo de treinamento e teste dos modelos avaliados. Observa-se que o modelo treinado com a estratégia CLL demonstrou desempenho superior. No entanto, para uma análise estatística das diferenças entre as médias dessas métricas, foi realizado um teste de hipóteses, considerando exclusivamente a métrica F1-Score. Para testar a hipótese nula de que as médias do F1-Score entre os dois melhores modelos (BERT-Multilingual utilizando a abordagem CLL e o modelo de referência) eram iguais, utilizou-se o teste *t de Student*, assumindo uma distribuição normal dos valores. O valor p obtido para o teste foi 0,0026, que é inferior ao nível de significância predefinido de 0,05. Consequentemente, a hipótese nula foi rejeitada, concluindo-se que os resultados do modelo BERT-Multilingual, utilizando a abordagem CLL, superaram os do modelo de referência.

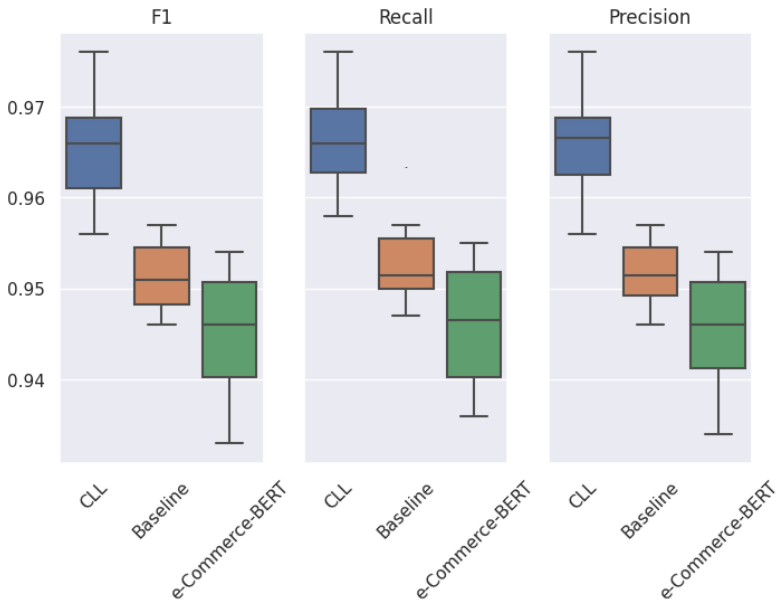

Figura 6.4 – Boxplot das Métricas de Avaliação Obtidas pelos Modelos considerando 10 Execuções

Finalmente, foi conduzido um experimento com o objetivo de avaliar o uso da técnica de CLL utilizando os dados de títulos de produtos WDC Products como fonte e os dados de produtos da Nota Fiscal como destino. O foco principal foi investigar, além da transferência de aprendizado entre idiomas, a capacidade do modelo de aprender características distintas de produtos.

Nesse experimento, o MLPT utilizado foi o BERT-Multilingual e as estratégias ZST e JL foram avaliadas. Na estratégia ZST, o ajuste do modelo foi realizado exclusivamente com os dados de treinamento e validação do WDC Products. Já na estratégia JL, foram

incorporadas adicionalmente frações (20% e 40%) dos dados de treinamento do corpus das notas fiscais no ajuste dos modelos. Os testes foram realizados exclusivamente com os dados de teste do corpus em português, que não foram utilizados na fase de treinamento dos modelos.

A Tabela 6.15 apresenta os resultados das estratégias de CLL utilizando os corpora WDC Products e Notas Fiscais.

Tabela 6.15 – Estratégias de CLLs no Contexto de Produtos das Notas Fiscais

Corpus	Dados do Idioma Fonte no Treino	Precisão	Recall	F1-score	Acurácia
Leites e Laticínios - *Hard Negative*	0% (ZST)	58%	62%	48%	50%
	20% (JL)	**84%**	**91%**	**87%**	**90%**
	40% (JL)	**89%**	**95%**	**92%**	**93%**
Cosméticos - *Hard Negative*	0% (ZST)	64%	74%	60%	64%
	20% (JL)	**76%**	**79%**	**78%**	**86%**
	40% (JL)	**82%**	**86%**	**84%**	**89%**

Os resultados apresentados na Tabela 6.15 indicam que o uso de CLL para os dados de notas fiscais proporciona melhorias consideráveis em relação à estratégia ZST. Ademais, observa-se que, mesmo com a utilização de apenas 40% dos dados de treinamento, o MLPT BERT-Multilingual alcançou resultados satisfatórios, com as médias F1-Score de 92% e 84%, respectivamente, para as categorias de Leites/Laticínios e Cosméticos, em comparação com os resultados anteriores de 96% e 95%, obtidos com 100% dos dados de treinamento (Tabela 6.7).

Com base na análise dos resultados dos experimentos apresentados nesta seção, a questão de pesquisa Q2 será respondida em seguida.

Questão de Pesquisa:

- **Q2:** O uso de técnicas de aprendizagem de máquina utilizando técnicas de CLL melhora os resultados de classificação de correspondência entre produtos?

Sim, os resultados dos experimentos apresentados indicam que o uso de CLL pode ser uma abordagem eficaz na indução de modelos para classificar correspondências de produtos, proporcionando melhorias nos resultados de classificação, especialmente em cenários com escassez de dados anotados. Destacam-se as seguintes observações:

1. Com apenas 10% dos dados anotados do idioma alvo, todos os modelos treinados com as estratégias de CLL alcançaram resultados de F1-Score próximos a 90%;

2. Ao adicionar mais dados do idioma alvo em diversas estratégias CLL, os modelos treinados demonstraram melhorias nas métricas de classificação, possibilitando superar os de referência. Por exemplo, na estratégia CL/JL+, em que o modelo de referência obteve um F1-Score de 95,3% com o MLPT Albertina-PT-BR, enquanto o modelo CLL (JL 50% + CL 50%) obteve 96,5% com o MLPT BERT-Multilingual; e

3. A abordagem CLL não apenas facilita a adaptação entre a aprendizagem de diferentes idiomas, mas também é eficaz na captura de características específicas de produtos, promovendo uma generalização robusta em cenários com variação linguística e de domínio, como as descrições de produtos de comércio eletrônico e aquelas que tratam de produtos de notas fiscais.

6.4.4 Avaliação das estratégias de criação de corpora

A qualidade dos dados utilizados no treinamento de modelos de aprendizagem de máquina é essencial para o sucesso de tarefas complexas, como a classificação em *product matching*. A representatividade e a precisão dos dados anotados influenciam diretamente o desempenho dos modelos, proporcionando maior acurácia e melhor capacidade de generalização. Os experimentos dessa subseção objetivam avaliar as estratégias de criação de pares de produtos não correspondentes, conforme detalhado na seção 5.2.

Para essa avaliação, adotou-se a abordagem de treinamento cruzado, na qual os modelos são treinados com dados de uma estratégia específica e avaliados com dados de teste de outra estratégia. Por exemplo, um modelo treinado com dados anotados pela estratégia "Aleatória" foi avaliado com os dados de teste da estratégia "Hard Negativa", e vice-versa. Considerando que os dados de produtos anotados foram criados em estratégias distintas, é importante ressaltar que pares de produtos anotados em um corpus de treinamento de uma estratégia podem também estar presentes no corpus de teste da outra estratégia. Adicionalmente, para evitar o enviesamento do modelo, foi garantida a não inserção de dados de treinamento na base de testes.

A Tabela 6.16 apresenta os resultados da comparação das estratégias de criação de corpora utilizando o MLPT BERT-Multilingual. Observa-se que, ao utilizar os dados de produtos da base de dados das notas fiscais, o modelo treinado com a estratégia "*Hard Negative*", quando avaliado com os dados da estratégia "Aleatório", obteve um F1-score de 97%, resultado semelhante ao obtido quando o modelo foi testado também com a estratégia "*Hard Negative*", conforme apresentado na Tabela 6.7. Em contrapartida, o modelo treinado com a estratégia "Aleatório" apresentou um F1-score inferior.

Em relação aos modelos gerados com as duas estratégias diferentes, utilizando os dados de referência originais do WDC Products e os dados gerados a partir da estratégia *Hard Negative*, foram observados comportamentos semelhantes aos modelos gerados com os dados de notas fiscais. Ou seja, o modelo treinado com os dados provenientes da estratégia "Hard Negative", quando testado com os dados originais de referência do WDC, apresentou aproximadamente 97% de F1-score. Esse resultado é semelhante ao melhor desempenho de um modelo classificador do estado da arte, que apresenta um F1-score de 96,53% (PEETERS et al., 2020). Além disso, o modelo ajustado com os dados de treinamento originais do WDC, quando testado com os dados de teste da estratégia "*Hard Negative*", também apresentou um resultado inferior.

Tabela 6.16 – Métricas de Desempenho de Modelos Treinados com Diferentes Estratégias de Anotação

Corpus	Treinamento	Teste	F1-score
Nota Fiscal: Leites e Laticínios	Aleatório	Hard Negative	**63%**
	Hard Negative	Aleatório	**96%**
WDC Products	Original	Hard Negative	86%
	Hard Negative	Original	**97%**

Assim, considerando os experimentos com corpora distintos, os resultados obtidos permitem inferir que os modelos treinados com os dados da estratégia "*Hard Negative*" possuem uma maior capacidade de generalização na aprendizagem.

Baseando-se na avaliação dos resultados dos experimentos descritos nesta seção, a questão de pesquisa Q3 será respondida em seguida.

Questão de Pesquisa:

- **Q3**: Estratégias de aprendizado contrastivo baseadas em similaridade são eficazes para gerar corpora anotados que possam treinar modelos capazes de identificar correspondências entre produtos?

Sim, os experimentos realizados nesta seção indicam que a abordagem de geração de pares anotados de produtos, baseada em similaridades, apresentada no Capítulo 5.2.1, permitiu o treinamento de modelos que produziram resultados consistentes, com boa capacidade de generalização (conforme Tabela 6.16), sugerindo que os modelos treinados são capazes de identificar novos produtos não vistos durante o treinamento.

6.5 AVALIAÇÃO DE MECANISMO DE BUSCA DE PRODUTOS CORRESPONDENTES

Os experimentos desta seção têm como objetivo avaliar o mecanismo de busca implementado pela função *findSimilarity()*, utilizada no Algoritmo 3 da Etapa 3 (Busca de Produtos Correspondentes). O foco principal é analisar a relevância dos resultados de busca na identificação de produtos correspondentes em um ambiente de recuperação da informação.

Conforme discutido no Capítulo 4, os modelos treinados não se restringem à tarefa de classificação, mas também são capazes de retornar um índice de correspondência entre pares de produtos. Esse índice de correspondência é empregado como critério para determinar a relevância de um resultado de busca no processo de reordenamento. Em outras palavras, quanto maior o índice de correspondência entre um produto pesquisado e os itens recuperados, maior será a relevância do resultado para o MLPT.

Para este experimento, todas as descrições distintas dos produtos presentes nos corpora foram indexadas no *Elasticsearch*. Em seguida, para cada descrição única de produto no conjunto de testes, foram realizadas buscas no sistema de RI. O algoritmo de referência utilizado para essas buscas foi o BM25, e a relevância dos top-N produtos recuperados foi avaliada. Os resultados obtidos pelo BM25 (*baseline*) foram comparados com os resultados gerados pela técnica de reordenamento *Cross-Encoder*, aplicada utilizando o MLPT, conforme descrito na seção 4.3.2.

Na sequência, são descritos os experimentos conduzidos com os dados de teste dos corpora Notas Fiscais (categoria Leites e Laticínios - *Hard Negative*) e WDC Products.

6.5.1 Análise do resultado da busca de produtos utilizando os dados de notas fiscais

O objetivo desta análise é avaliar os métodos de busca que recuperam os itens relevantes para um determinado produto do corpus de teste das notas fiscais. Um item retornado em uma busca é considerado relevante quando possui o mesmo GTIN do item pesquisado, mesmo que tenha descrições alternativas.

A Figura 6.5 apresenta a quantidade de itens relevantes, que estão indexados no *Elasticsearch*, para cada descrição de produto nos dados de teste do corpus de notas fiscais da categoria Leites e Laticínios. Observa-se que alguns produtos não possuem outras descrições correspondentes, ou seja, não há itens relevantes a serem recuperados nas buscas para esses produtos. Nesses casos, apenas os produtos com, pelo menos, um item relevante foram considerados neste experimento.

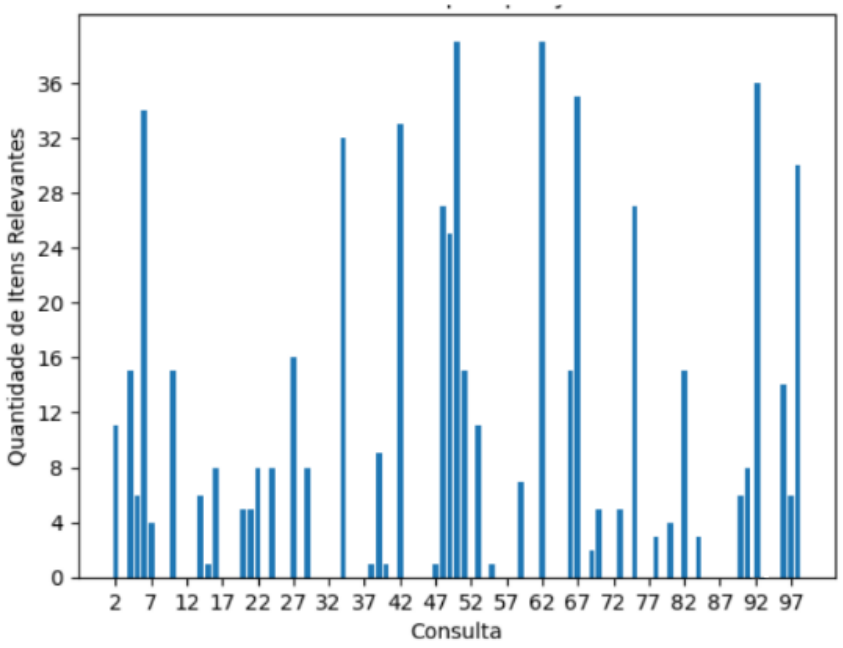

Figura 6.5 – Quantidade de itens relevantes por descrição de produto dos dados de testes

Por outro lado, há itens que apresentam mais de 30 correspondências. Por exemplo, o produto identificado no eixo X, pela consulta de número 92, possui cerca de 36 descrições correspondentes (relevantes). Assim, uma busca ideal para esse item deveria retornar essas 36 descrições, considerando os Top-36 itens resultantes dessa busca. Da mesma forma, o item identificado pelo número 2 deveria apresentar 10 itens relevantes, quando considerados os Top-10 itens resultantes da busca realizada.

Nesse contexto de busca por produtos correspondentes, a ordem exata dos itens relevantes recuperados não é um fator crucial, já que todas as descrições referem-se ao mesmo produto, representado pelo mesmo GTIN. O mais importante é que todas as variações da descrição do mesmo produto estejam incluídas entre os primeiros resultados da busca. Por exemplo, se o produto "Leite Integral XYZ" estiver registrado no sistema com três descrições distintas — "Leite Integral XYZ 1L", "Leite XYZ Integral 1000ml" e "XYZ Leite 1L" —, todas essas descrições devem ser consideradas relevantes. Assim, não importa se "Leite XYZ Integral 1000ml" aparece na primeira posição e "Leite Integral XYZ 1L" na terceira; o essencial é que ambas sejam reconhecidas como variações do mesmo produto (mesmo GTIN) e estejam entre as principais posições dos itens recuperados.

Dessa forma, para cada item do conjunto de testes, foi realizada uma busca no sistema de RI, computando as métricas de avaliação para os Top-50 itens retornados pelos métodos de busca. A escolha de considerar os Top-50 itens deve-se ao fato de que este valor corresponde ao número máximo de itens relevantes esperado para alguns produtos no conjunto de testes, conforme Figura 6.5. Isso garante que todas as variações de descrições

para um mesmo produto possam ser analisadas, sem excluir potenciais correspondências relevantes. Esses resultados foram analisados considerando a média de todas as consultas realizadas.

A Figura 6.6 apresenta as métricas de Precisão@N, Recall@N e F1-Score@N para os algoritmos avaliados. Um ponto de destaque é o desempenho no Top-1, onde o BM25 atingiu uma precisão de aproximadamente 50%, enquanto a técnica de reordenamento *Cross-Encoder* alcançou uma precisão de cerca de 95%. Avaliar a precisão na primeira posição dos resultados é crucial, pois reflete diretamente a eficácia da função *findSimilarity()*, utilizada para buscar produtos correspondentes. Além disso, observa-se que o *recall* da técnica de reordenamento é superior ao do BM25, especialmente quando analisados os primeiros itens do ranking. Isso indica que o *Cross-Encoder* consegue recuperar mais itens relevantes nos primeiros resultados, em comparação ao BM25.

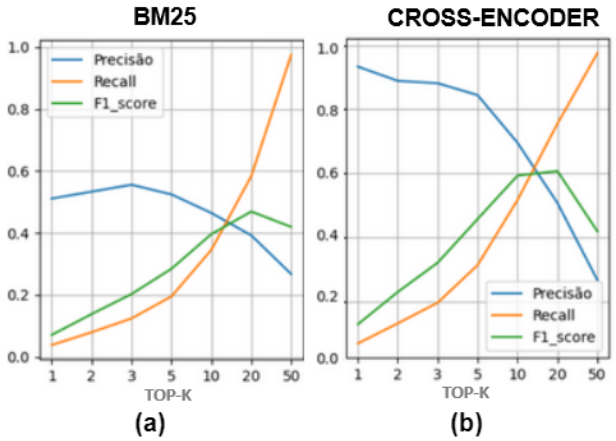

Figura 6.6 – Métricas Precisão@N, Recall@N e F1-Score@N: a) BM25; e b) Cross-Encoder (MLPT)

A Figura 6.7 apresenta as métricas de NDCG e MRR em relação aos elementos recuperados nos primeiros N itens do resultado. Nota-se que a técnica de reordenamento *Cross-Encoder* demonstra um desempenho superior, apresentando um NDCG superior a 95% desde o Top 1, enquanto o BM25 inicia com uma pontuação de apenas 51%. O NDCG é uma métrica que avalia a qualidade do ranking dos resultados, priorizando a relevância dos itens recuperados em função da posição em que aparecem na lista. Assim, quanto mais alto o NDCG, melhor é a qualidade do ranking em termos de relevância dos itens mais importantes. Por outro lado, o MRR mede a qualidade do ranking focando apenas no primeiro item relevante encontrado na lista de resultados. Devido a essa característica, é esperado que o MRR diminua após o Top 1, já que a métrica reflete a presença de itens relevantes apenas na primeira posição.

As métricas analisadas indicam que a técnica de reordenamento *Cross-Encoder*, aplicada com o auxílio de um MLPT treinado para a tarefa de *Product Matching*, melhora

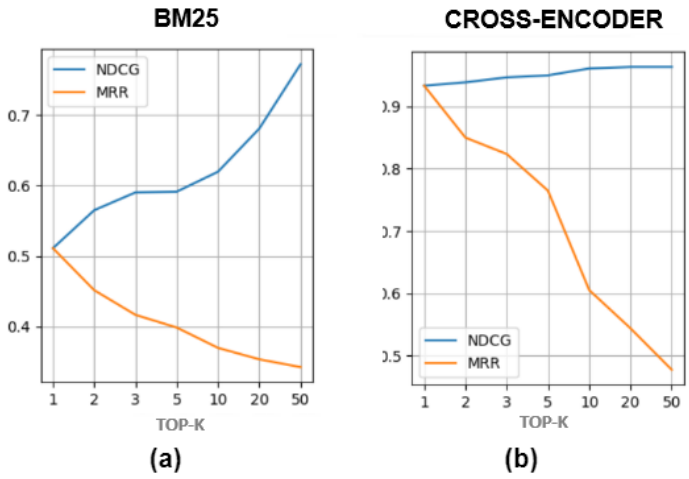

Figura 6.7 – Métricas NDCG e MRR: a) BM25; e b) Cross-Encoder (MLPT)

significativamente os resultados na busca por produtos correspondentes, em um contexto de RI. A Figura 6.8 reforça essa análise, permitindo a comparação entre os algoritmos, ao exibir a distribuição da quantidade de itens relevantes por posição nos resultados das buscas realizadas. Esta comparação revela que a técnica *Cross-Encoder* recupera muito mais itens relevantes nas posições iniciais dos resultados das consultas. Além disso, nota-se que o algoritmo BM25 (Figura 6.8a) apresenta um número considerável de itens relevantes entre as posições 30 e 50 na lista de resultados, enquanto a técnica *Cross-Encoder* (Figura 6.8b) recupera poucos itens relevantes nas posições acima de 30, concentrando-os nas primeiras posições, o que demonstra uma maior precisão no ordenamento dos itens relevantes recuperados. De fato, há poucos produtos contendo mais de 30 itens relevantes esperados em uma determinada consulta, conforme ilustra a Figura 6.5.

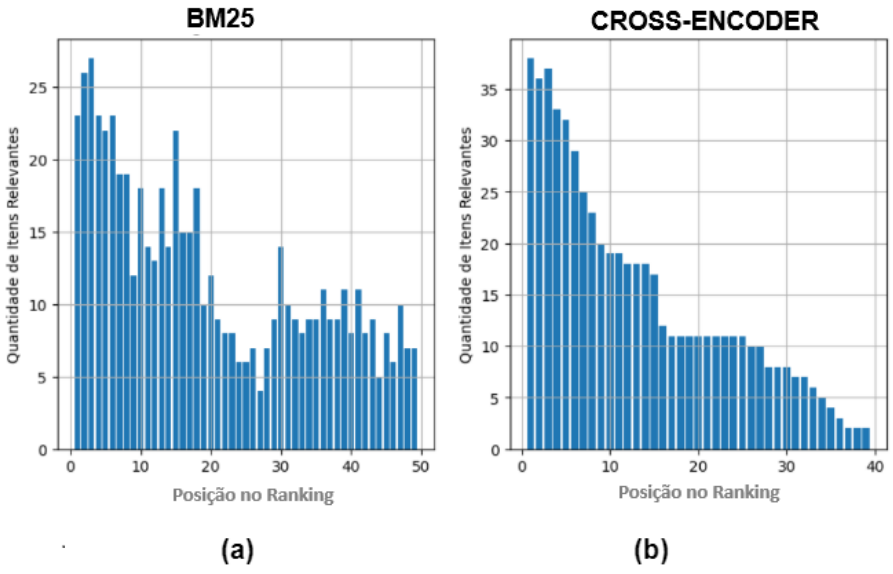

Figura 6.8 – Distribuição da quantidade de itens relevantes por posição nos resultados da busca: a) BM25; b) Cross-Encoder.

6.5.2 Análise do resultado da busca de produtos com os dados do WDC Products

Um experimento semelhante ao descrito na subseção anterior foi realizado utilizando a base de dados WDC Products. Para isso, foi utilizado o mesmo procedimento: 1) indexar os produtos do WDC no *Elasticsearch*; 2) realizar buscas utilizando os produtos contidos no conjunto de testes; 3) utilizar o MLPT treinado para reordenar os resultados da busca inicial realizada pelo algoritmo BM25; e 4) analisar os resultados relevantes obtidos pelo algoritmo BM25 e pela técnica de reordenamento *Cross-Encoder*

Entretanto, apenas os resultados do Top 10 foram considerados para esta análise, uma vez que, conforme apresentado no gráfico da Figura 6.9, a quantidade de correspondências por produto na base WDC é consideravelmente menor. Além disso, observa-se que quase todos os produtos na base WDC apresentam descrições correspondentes.

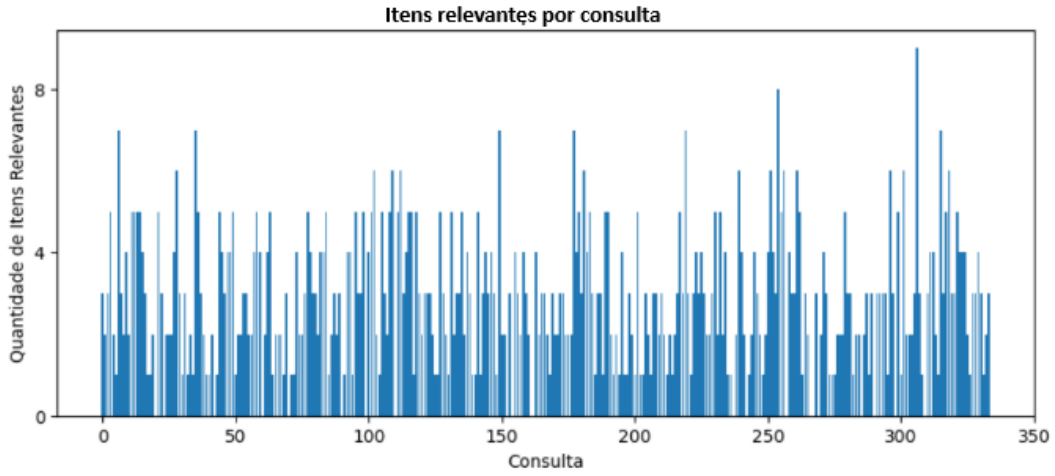

Figura 6.9 – Quantidade de itens relevantes por consulta realizada dos dados de testes

A Figura 6.10 apresenta as métricas de MRR@N e NDCG@N, comparando os resultados gerados pelo algoritmo BM25 e pelo MLPT. Conforme observado anteriormente, o MLPT conseguiu melhorar significativamente a qualidade dos resultados relevantes, alcançando cerca de 98% nas métricas de NDCG e MRR na primeira posição do ranking (Top 1).

Os títulos dos produtos nesta base de dados do WDC Products são, em média, mais longos e detalhados, o que pode contribuir para que o algoritmo BM25 apresente resultados mais relevantes em comparação aos produtos da base de dados das notas fiscais. Contudo, mesmo diante dessa diferença, conforme ilustrado pelos gráficos da Figura 6.11, a técnica de reordenamento *Cross-Encoder*, realizada pelo MLPT, ainda foi capaz de recuperar os itens mais relevantes nas primeiras posições do ranking. Por exemplo, na primeira posição do ranking (Top 1), há cerca de 300 itens relevantes encontrados

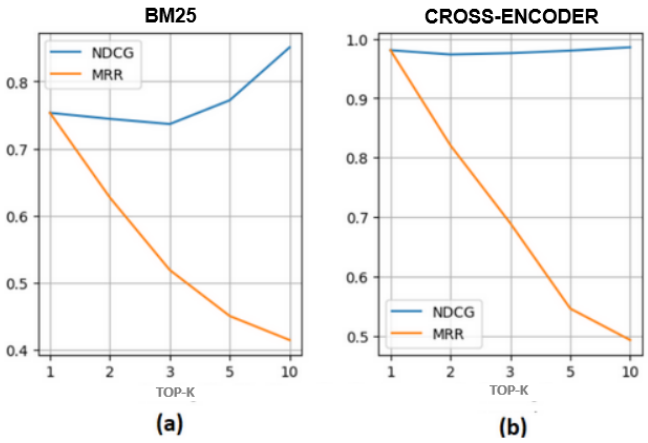

Figura 6.10 – Métricas de avaliação da qualidade dos rankings: (a) BM25 e (b) Cross-Encoder

pela técnica de *Cross-Encoder*, enquanto a técnica BM25 recuperou menos de 250 itens relevantes.

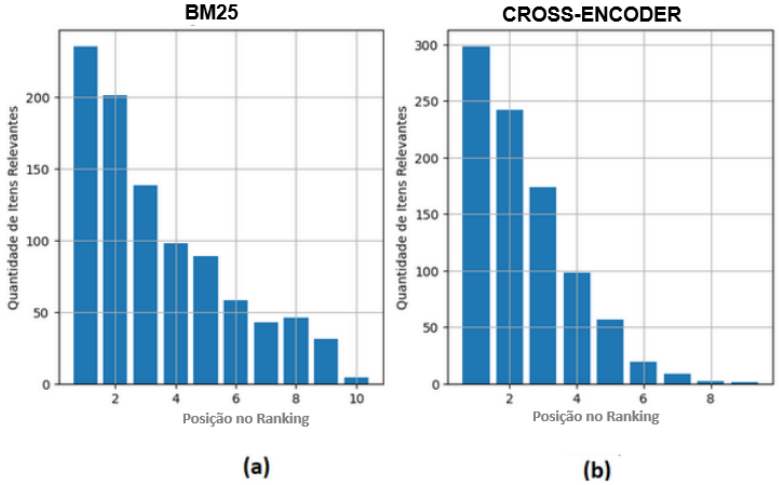

Figura 6.11 – Distribuição da quantidade de itens relevantes por posição nos resultados de busca em WDC Product: a) BM25; b) Cross-Encoder

Com a análise dos resultados apresentados nesta seção, será possível abordar a questão de pesquisa Q4 a seguir.

Questão de Pesquisa:

- **Q4:** Técnicas de aprendizagem de máquina supervisionada podem melhorar a relevância dos resultados de busca de produtos, quando usadas em conjunto com o algoritmo BM25 em um ambiente de RI?

1. Sim. A análise realizada a partir dos experimentos que envolveram os dados de notas

fiscais da categoria “Leite e Laticínios"(utilizando a estratégia *Hard Negative*) e os dados do WDC Products indica que os MLPTs podem efetivamente melhorar a relevância na busca de produtos correspondentes em um ambiente de RI. A solução proposta permitiu priorizar os itens mais relevantes no topo da lista de resultados.

6.6 AVALIAÇÃO DO STEPMATCH

Nas seções anteriores, foram avaliadas isoladamente as técnicas utilizadas nas etapas 2 e 3 do STEPMatch, respectivamente, referentes à Classificação de Correspondências de Produtos e à Busca de Produtos Correspondentes. Nesta seção, o STEPMatch será avaliado de forma integral, a partir de um cenário que reflete o uso típico em uma aplicação real.

6.6.1 Configuração do Cenário de Teste

Para simular essa aplicação real no processo de avaliação do STEPMatch, o seguinte cenário foi projetado:

1. **Escolha da Base de Dados:** A base de dados selecionada foi a de notas fiscais, por ser a que apresenta a maior quantidade e diversidade de produtos, além de conter complexidades adicionais, relacionadas às limitações dos caracteres nas descrições textuais dos produtos.

2. **Seleção do Modelo de Classificação de Correspondência:** O modelo utilizado nas etapas 2 e 3 do STEPMatch foi aquele que apresentou os melhores resultados nos experimentos com a base de dados de notas fiscais (seção 6.4.2). Assim, foi escolhido o modelo treinado com o MLPT BERTimbau, que alcançou um F1-score de 98,5%, conforme mostrado na Tabela 6.7.

3. **Seleção de Produtos:** Foram selecionados aleatoriamente 320 produtos do corpus de teste das notas fiscais, os quais pertencem à categoria Laticínios e Leites. A escolha dessa base de teste foi estratégica para evitar enviesar os resultados nas fases 2 e 3 do STEPMatch, assegurando que os dados utilizados não tenham sido previamente empregados no treinamento do modelo aplicado nessas fases. Além disso, essa base foi revisada manualmente, conforme descrito na seção 5.2.1.

4. **Inserção de Ruídos nos Dados de Teste:** Dentre os produtos selecionados, 150 foram manipulados com a inserção de ruídos nos códigos de GTIN, com o intuito de simular as inconsistências de dados comuns em cenários reais, causadas por erros de cadastro ou de catalogação. Os ruídos consistiram em:

    - Inserção de valores nulos nos campos de GTIN;

- Códigos inválidos (valores que não seguem o padrão esperado para um código de GTIN);
- Códigos de outros produtos.

A inclusão desses ruídos nos dados permite avaliar a eficiência do STEPMatch em identificar as correspondências corretas entre os produtos. O objetivo é analisar a capacidade da ferramenta, tanto em corrigir os produtos com ruídos inseridos, quanto em preservar corretamente os registros que não apresentavam erros.

5. **Definição do Mostruário de Produtos:** O mostruário de produtos utilizado na fase 3 do STEPMatch foi composto a partir de todas as descrições únicas de produtos presentes na base de dados de notas fiscais, totalizando cerca de 1 milhão de itens distintos. Esse cenário foi crucial para testar o STEPMatch em um ambiente real, caracterizado pela diversidade de descrições, assegurando que a abordagem fosse avaliada diante de diferentes tipos de produtos.

6. **Métricas de Avaliação:** Os resultados do cenário de teste foram analisados com base nas principais métricas de avaliação de correspondência de produtos, incluindo acurácia, precisão, recall e F1-score. A análise foi realizada, tanto para os produtos com códigos GTINs corretos, quanto para aqueles com ruídos, objetivando medir a eficácia do STEPMatch em diferentes situações. Além disso, os tempos de processamento das três etapas do STEPMatch também foram registrados e analisados, considerando a média dos tempos obtidos em 10 repetições, para assegurar a consistência e robustez da avaliação em termos de desempenho computacional.

6.6.2 Resultados da Avaliação do STEPMatch

Esta seção apresenta as métricas de avaliação do STEPMatch, baseadas no ambiente de teste previamente descrito. Essas métricas foram calculadas a partir da entrada de um conjunto de produtos no STEPMatch, avaliando as correspondências de produtos geradas a partir do conjunto de teste. A Figura 6.12 ilustra a distribuição da quantidade de produtos por código GTIN utilizada na avaliação do STEPMatch. Essa representação revela uma diversidade significativa, mostrando tanto produtos com descrições variadas, como aqueles com descrições mais restritas, refletindo situações do mundo real.

A Tabela 6.17 apresenta a acurácia e as médias das métricas de precisão, *recall* e F1-score. A acurácia do STEPMatch foi de 98,11%, demonstrando uma excelente capacidade de identificar corretamente as correspondências entre produtos com base em suas descrições. Esses resultados refletem a eficiência do STEPMatch ao lidar com descrições de produtos em diferentes níveis de complexidade. No total, dos 320 produtos avaliados, apenas seis

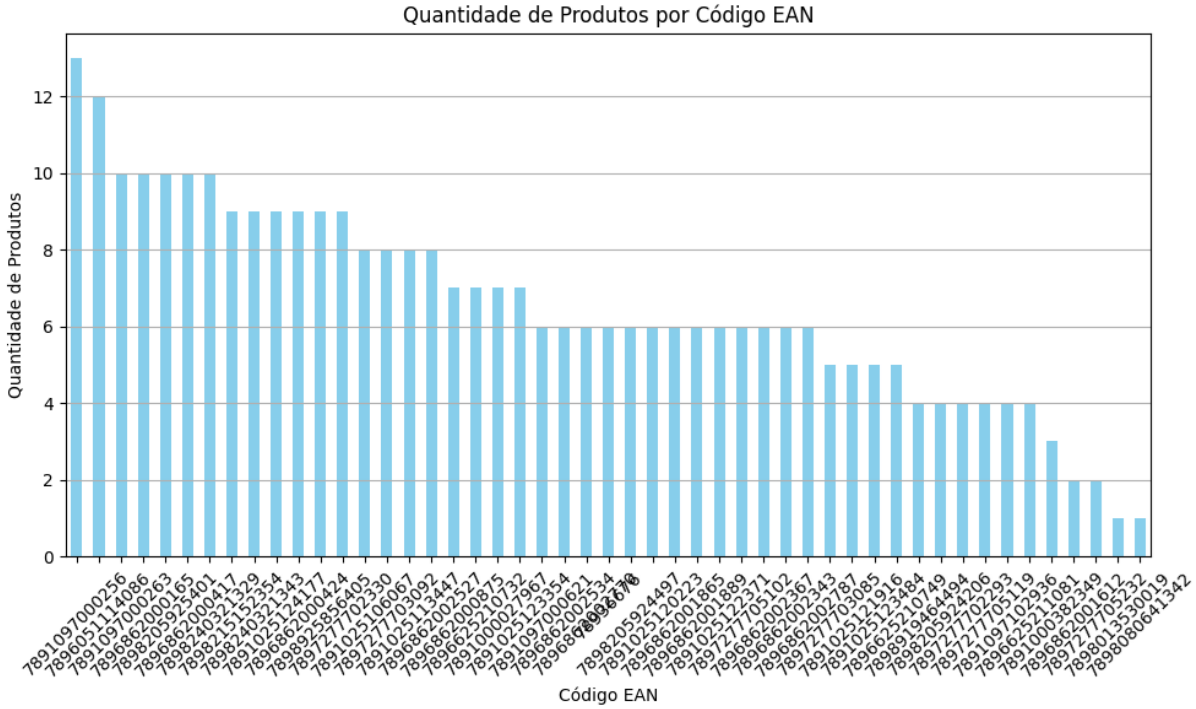

Figura 6.12 – Distribuição da Quantidade de Produtos por Código GTIN na Avaliação do STEPMatch.

não tiveram suas correspondências corretamente identificadas. Dentre os 150 produtos com erros introduzidos nos códigos GTIN, apenas três não tiveram suas correspondências corrigidas. Além disso, outros três produtos sem ruídos tiveram seus GTINs alterados incorretamente.

Tabela 6.17 – Resultados das Métricas de Avaliação do STEPMatch na Correspondência de Produtos

Média Ponderada			Acurácia
Precisão	Recall	F1-score	
99,71%	98,11%	98,83%	98,11%

Por fim, o STEPMatch também foi avaliado quanto ao tempo de processamento. Para isso, foi calculada a média de tempo de execução em cada etapa, juntamente com o desvio padrão, com base em 10 repetições para cada experimento. A avaliação considerou diferentes configurações da quantidade de produtos retornados na Fase 3 para a busca de correspondências. A Figura 6.13 ilustra o tempo de processamento em cada etapa, analisando os resultados para diferentes valores de Top-K, com $K \in 10, 20, 30, 40, 50$, obtidos pelo método de busca. É importante destacar que o tempo de indexação dos produtos na base de dados não foi incluído nesta avaliação.

Nas Fases 1 e 2, o método de busca não é utilizado, o que não impacta o tempo de processamento dessas etapas. Nessas fases, o tempo de execução varia de acordo com a quantidade de produtos e grupos a serem processados. Já a Fase 3 requer mais tempo de

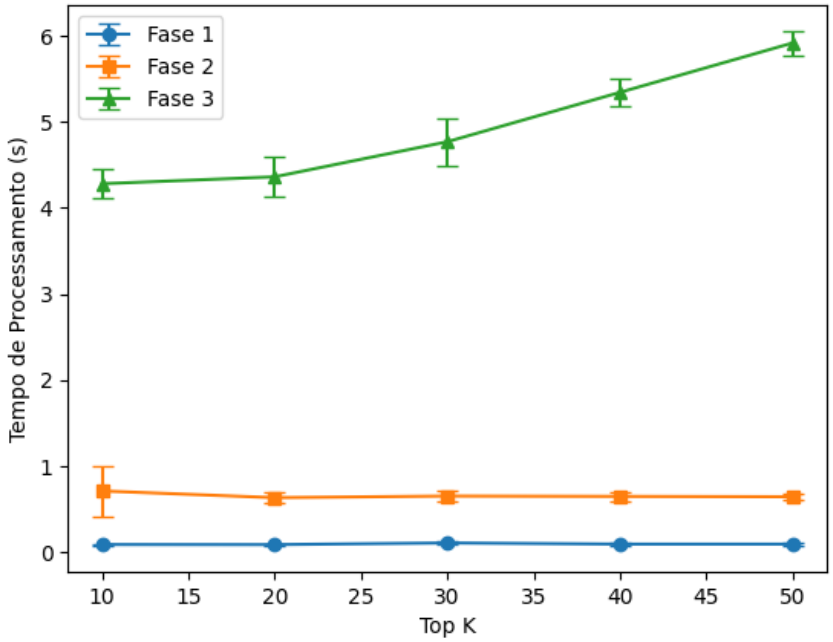

Figura 6.13 – Tempo de Processamento das Etapas do STEPMatch

processamento, devido às múltiplas operações envolvidas na busca de correspondência de um produto, incluindo o reordenamento feito pela técnica de *cross-encoder*. As métricas exibidas na Tabela 6.17, para $k = 10$, foram consistentes para outros valores de k, o que indica que, para esses dados de teste, $k = 10$ foi um valor adequado para o STEPMatch.

Esses resultados sugerem que o STEPMatch configura-se como uma abordagem promissora para a correspondência de produtos, apresentando um desempenho que pode ser explorado em cenários reais, como aqueles envolvendo a análise de produtos de notas fiscais. É importante ressaltar a aplicabilidade prática da metodologia proposta nesta tese, que já está sendo utilizada em uma aplicação em produção. A ferramenta Banco de Preços do Tribunal de Contas do Estado do Acre[8] utiliza técnicas de correspondência de produtos desenvolvidas neste trabalho para integrar dados de produtos comercializados no estado, oferecendo suporte à tomada de decisões em compras públicas.

6.7 AMEAÇAS À VALIDADE

Nesta seção, são discutidas ameaças à validade dos experimentos realizados, analisando prováveis fatores que podem interferir na confiabilidade e generalização dos resultados obtidos nos experimentos.

A primeira ameaça identificada refere-se à qualidade dos dados utilizados, principalmente os oriundos de notas fiscais. Em alguns casos, foram detectados erros no cadastro dos produtos, especificamente relacionados aos identificadores únicos (GTIN).

[8] <https://bancodeprecos.tceac.tc.br/banco-precos/#/>

Considerando que o GTIN foi utilizado como chave única para a criação de pares de produtos (correspondentes e não correspondentes), esses erros poderiam resultar na construção de corpora com anotações incorretas. Para mitigar esse risco, três abordagens foram implementadas: (1) Verificação manual conduzida por três pesquisadores em uma das bases de dados mais exploradas nos experimentos (Categoria de Leites e Laticínios); (2) Utilização de produtos extraídos das notas fiscais com maior volume de ocorrências em suas descrições, privilegiando os produtos mais comercializados; e (3) Replicação dos experimentos utilizando o *benchmark* WDC Products, um conjunto de dados externo e consolidado.

Outro aspecto relevante relacionado à qualidade dos dados está na separação adequada entre os conjuntos de treinamento, validação e teste. No experimento da seção 6.4.4, que avalia diferentes estratégias de criação de corpora utilizando o mesmo conjunto de dados de produtos, foi necessário implementar um mecanismo adicional para verificar e remover pares de descrições de produtos que estavam presentes no corpus de treinamento de uma estratégia e, ao mesmo tempo, no corpus de testes de outra estratégia. Esse procedimento foi fundamental para garantir que os modelos treinados em uma estratégia não fossem testados com dados previamente expostos na fase de treinamento, prevenindo assim o sobreajuste (*overfitting*) dos modelos e assegurando maior confiabilidade nos resultados obtidos.

Outra ameaça analisada diz respeito ao risco de sobreajuste ou subajuste (*underfitting*) nos modelos de aprendizagem de máquina. Para mitigar esses riscos, foi adotada uma estratégia de *Bootstrapping* para estimar a incerteza dos modelos treinados. Adicionalmente, as épocas de treinamento foram ajustadas individualmente, com base na análise das curvas de aprendizado, garantindo um equilíbrio entre a capacidade do modelo de generalizar e a sua adequação aos dados de treinamento.

7 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as principais contribuições desta tese e delineia possíveis desdobramentos para o aperfeiçoamento da abordagem proposta, indicando caminhos para trabalhos futuros.

7.1 CONTRIBUIÇÕES

Esta tese apresentou uma abordagem para a tarefa de correspondência de produtos em descrições curtas, com foco nas notas fiscais eletrônicas emitidas no Brasil. O cenário tratado é caracterizado por descrições curtas, não estruturadas e, muitas vezes, inconsistentes, o que torna o problema de correspondência de produtos uma tarefa desafiadora. Nesse contexto, foi proposta a abordagem STEPMatch (*Short Text Product Matching*), que integra múltiplas técnicas para garantir a correspondência eficaz entre produtos em um ambiente como o de notas fiscais. O objetivo é integrar dados de produtos para apoiar os processos gerenciais que dependem de informações consistentes sobre os produtos.

Os resultados deste estudo demonstraram a eficácia do STEPMatch na identificação de correspondências entre produtos. Utilizando dados reais em um cenário de avaliação, o STEPMatch alcançou uma acurácia de 98,12%. Foram avaliados 320 produtos, dos quais 150 tiveram erros induzidos nas suas identificações, e apenas três itens não foram corretamente associados. Além disso, outros três produtos, que não continham erros induzidos, também não foram corretamente correspondidos, totalizando apenas seis erros de correspondência em todo o conjunto do teste.

Ao responder às questões de pesquisa, constatou-se que:

1. Os MLPTs treinados com dados com os dados de produtos demonstraram desempenho superior na tarefa de correspondência de produtos, lidando com as variações nas descrições de produtos, quando comparados com técnicas tradicionais;

2. As estratégias de Cross-Lingual Learning (CLL) foram eficazes em melhorar os resultados de classificação, mesmo com poucos dados anotados no idioma alvo. A técnica de CLL mostrou-se eficaz para a classificação de correspondências de produtos, especialmente em cenários com dados escassos. À medida que mais dados do idioma alvo foram incorporados, os resultados superaram os modelos de referência, demonstrando a capacidade do CLL em promover a generalização de modelos treinados em diferentes idiomas e domínios;

3. A geração de pares anotados de produtos, por meio de técnicas contrastivas com base em similaridades, resultou em modelos capazes de generalizar a aprendizagem e identificar corretamente novos produtos não vistos durante o treinamento, comprovando a eficácia da abordagem na construção de corpora de produtos;

4. A combinação do algoritmo BM25 - uma técnica tradicional de RI - com MLPTs treinados especificamente para a tarefa de correspondência de produtos melhorou significativamente a relevância dos resultados de busca.

Os experimentos realizados demonstraram que a abordagem proposta atinge uma alta taxa de acurácia na correspondência de produtos, especialmente em cenários onde as descrições de produtos são limitadas e sujeitas a variações contextuais. As técnicas aplicadas no reordenamento dos resultados de busca demonstraram ganhos substanciais em termos de precisão dos itens relevantes no topo da lista de resultados, superando as abordagens tradicionais utilizadas para esse tipo de problema.

Finalmente, esta pesquisa inovou no campo de correspondência de produtos ao propor mecanismos de incorporação de dados anotados de produtos de outros idiomas para treinar modelos de classificação de correspondência de produtos, experimentando diversas estratégias de CLL.

7.2 TRABALHOS FUTUROS

Embora o STEPMatch tenha se mostrado eficaz nos experimentos realizados, ainda há espaço para expandir o escopo da abordagem para outros domínios e categorias de produtos. Trabalhos futuros poderiam analisar a eficácia da abordagem em uma maior variedade de dados de produtos, incluindo diferentes tipos de categorias e fontes de informações, mesclando inclusive com dados do comércio eletrônico. Além disso, é fundamental avaliar a eficácia do STEPMatch em um cenário de fluxo contínuo de dados. Esse processo envolve a simulação de uma aplicação em condições reais, iniciando com uma carga inicial de dados e, em seguida, incorporando múltiplas recargas de dados. Dessa forma, busca-se verificar como o modelo se adapta ao recebimento constante de novos dados, avaliando o crescimento do mostruário de produtos e o desempenho do STEPMatch ao longo do tempo. Além disso, é importante investigar e propor abordagens para a definição do mostruário de produtos, a fim de evitar a indexação de todas as descrições distintas de produtos no sistema de RI, o que poderia melhorar a eficiência do método de busca implementado na Etapa 3.

No que se refere à Etapa 1 do STEPMatch, o método de *blocking* adotado foi o *Standard Blocking*. Futuras investigações devem explorar técnicas mais avançadas de

blocking (PAPADAKIS et al., 2021), como as abordagens probabilísticas e baseadas em aprendizado, com o objetivo de lidar melhor com grupos de produtos desbalanceados e aqueles que não possuem o código GTIN. A implementação de estratégias de *blocking* mais sofisticadas pode aumentar a eficiência na identificação de correspondências, especialmente em grandes volumes de dados com alta variabilidade de descrições.

Outro aspecto que merece atenção em trabalhos futuros é o aprimoramento das técnicas de correspondência de produtos para lidar com descrições ambíguas ou incompletas. O uso de atributos adicionais, como o código NCM, a unidade de medida e o preço dos produtos, pode fornecer um contexto mais robusto para aumentar a precisão do processo de correspondência. A técnica de voto majoritário, empregada para definir a descrição canônica dos grupos de produtos, também demanda uma análise cuidadosa, uma vez que definições inadequadas podem induzir erros e propagá-los nas verificações de correspondências realizadas nas etapas 2 e 3 do STEPMatch. Adicionalmente, a exploração de modelos generativos se apresenta como uma abordagem complementar promissora, seja para a geração de dados anotados, seja para o enriquecimento da base de dados com descrições mais detalhadas, potencialmente melhorando o desempenho geral do processo.

Trabalhos futuros poderiam explorar a aplicação de técnicas de CLL com uma análise mais aprofundada do impacto da diversidade dos conjuntos de dados no desempenho dos modelos treinados. Uma possível abordagem seria incorporar métricas como a Entropia de Shannon, o Coeficiente de Similaridade de Jaccard e a Divergência de Kullback-Leibler para analisar a diversidade do conjunto de dados utilizados, avaliando como a progressiva inclusão de amostras no idioma destino afeta a curva de aprendizado, especialmente o comportamento da curva de perda (Loss) ao longo do treinamento. Além disso, é relevante investigar o impacto do uso de outros idiomas no aprendizado cruzado, considerando variações linguísticas que podem influenciar a eficácia dos modelos em cenários reais.

No que tange à Etapa 3 do STEPMatch, métodos de busca que combinem abordagens léxicas e semânticas devem ser explorados para o desenvolvimento de mecanismos que permitam recuperar produtos semanticamente equivalentes. Isso é particularmente importante para identificar produtos que não são corretamente encontrados em buscas exclusivamente léxicas, como as realizadas pelo algoritmo BM25. A investigação de buscas em espaço vetorial, juntamente com o uso de estratégias de enriquecimento de consulta (*queries*) ou da própria base de dados, pode otimizar a recuperação de produtos relevantes, ampliando o alcance e a precisão das correspondências.

Por fim, planeja-se realizar uma comparação direta do STEPMatch com abordagens do estado da arte na área de Resolução de Entidades. Neste trabalho, foram realizadas diversas tentativas de utilizar ferramentas como Ditto (LI et al., 2020) e DeepMatch

(MUDGAL et al., 2018). No entanto, mesmo seguindo rigorosamente os manuais de instalação, surgiram erros recorrentes de importação de bibliotecas, mesmo após a instalação das versões exatas especificadas nos arquivos de configuração fornecidos (*environment.yml*). Embora esses problemas técnicos tenham impedido a comparação direta, os resultados obtidos pelo STEPMatch em um cenário real demonstram sua eficácia na correspondência de produtos. Pesquisas futuras podem se beneficiar da superação dessas dificuldades técnicas, permitindo uma ampliação das comparações das métricas de avaliação, tais como precisão, *recall* e acurácia, além do custo computacional e energético, validando adicionalmente a eficácia do STEPMatch em relação às soluções estabelecidas.

REFERÊNCIAS

AALST, W. M. P. van der. The data science revolution. In: **Unimagined Futures – ICT Opportunities and Challenges**. [S.l.]: Springer International Publishing, 2020. p. 5–19.

ADLOUNI, Y. E.; RODRíGUEZ, H.; MEKNASSI, M.; El Alaoui, S. O.; EN-NAHNAHI, N. A multi-approach to community question answering. **Expert Systems with Applications**, v. 137, p. 432–442, 2019. ISSN 0957-4174.

ALLEN, J. F. Natural language processing. In: _____. **Encyclopedia of Computer Science**. GBR: John Wiley and Sons Ltd., 2003. p. 1218–1222. ISBN 0470864125.

ALVES, A.; BAPTISTA, C. de S.; ANDRADE, D. O. S. de; OLIVEIRA, M. G. D.; OLIVEIRA, A. B. D. A spatiotemporal approach for social media sentiment analysis. **First Monday**, University of Illinois Libraries, jul 2021.

ARAÚJO, T. B. **Parallel blocking for entity resolution in the context of semi-structured data.** Tese (Doutorado) — Universidade Federal de Campina Grande, Campina Grande, Brazil, 2020.

AYO, F. E.; FOLORUNSO, O.; IBHARALU, F. T.; OSINUGA, I. A. Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. **Computer Science Review**, Elsevier BV, v. 38, p. 100311, nov 2020.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação - 2ed: Conceitos e Tecnologia das Máquinas de Busca**. Bookman Editora, 2013. ISBN 9788582600498. Disponível em: <https://books.google.com.br/books?id=YWk3AgAAQBAJ>.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. In: BENGIO, Y.; LECUN, Y. (Ed.). **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1409.0473>.

BARBOSA, L. Learning representations of Web entities for entity resolution. **International Journal of Web Information Systems**, v. 15, n. 3, p. 346–358, aug 2019. ISSN 1744-0084.

BARLAUG, N.; GULLA, J. A. Neural networks for entity matching: A survey. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM New York, NY, v. 15, n. 3, p. 1–37, 2021.

BERNSTEIN, F.; KÖK, A. G.; XIE, L. Dynamic assortment customization with limited inventories. **Manufacturing & Service Operations Management**, INFORMS, v. 17, n. 4, p. 538–553, 2015.

BHASKARAN, J.; BHALLAMUDI, I. Good secretaries, bad truck drivers? occupational gender stereotypes in sentiment analysis. **arXiv preprint arXiv:1906.10256**, 2019.

BHATTACHARYA, I.; GETOOR, L. A latent dirichlet model for unsupervised entity resolution. In: SIAM. **Proceedings of the 2006 SIAM International Conference on Data Mining**. [S.l.], 2006. p. 47–58.

BILENKO, M.; MOONEY, R. J. Adaptive duplicate detection using learnable string similarity measures. In: **Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2003. (KDD '03), p. 39–48. ISBN 1581137370.

BINETTE, O.; STEORTS, R. C. (almost) all of entity resolution. **Science Advances**, American Association for the Advancement of Science (AAAS), v. 8, n. 12, mar 2022.

BIRJALI, M.; KASRI, M.; BENI-HSSANE, A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. **Knowledge-Based Systems**, Elsevier BV, v. 226, p. 107134, aug 2021.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, v. 5, 07 2016.

BOORUGU, R.; RAMESH, G. A survey on nlp based text summarization for summarizing product reviews. In: **2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)**. [S.l.: s.n.], 2020. p. 352–356.

BOUSDEKIS, A.; LEPENIOTI, K.; APOSTOLOU, D.; MENTZAS, G. A Review of Data-Driven Decision-Making Methods for Industry 4.0 Maintenance Applications. **Electronics**, v. 10, n. 7, 2021. ISSN 2079-9292.

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.; AMODEI, D. Language models are few-shot learners. In: **Proceedings of the 34th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546.

CAI, C.; LI, W.; HAN, H.; LIU, M. Risk scenario-based value estimation of bitcoin. **Procedia Computer Science**, Elsevier BV, v. 199, p. 1198–1204, 2022.

CHOI, J.; KALLUMADI, S.; MITRA, B.; AGICHTEIN, E.; JAVED, F. **Semantic Product Search for Matching Structured Product Catalogs in E-Commerce**. 2020.

CHOI, J. I.; KALLUMADI, S.; MITRA, B.; AGICHTEIN, E.; JAVED, F. **Semantic Product Search for Matching Structured Product Catalogs in E-Commerce**. 2020.

CHOUDHARY, A.; ARORA, A. Linguistic feature based learning model for fake news detection and classification. **Expert Systems with Applications**, Elsevier BV, v. 169, p. 114171, may 2021.

CHOWDHARY, K. R. Natural language processing. In: _____. **Fundamentals of Artificial Intelligence**. New Delhi: Springer India, 2020. p. 603–649. ISBN 978-81-322-3972-7.

CHOWDHURY, G. G. Natural language processing. **Annual review of information science and technology**, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003.

CHRISTEN, P. Febrl -: An open source data cleaning, deduplication and record linkage system with a graphical user interface. In: _____. New York, NY, USA: Association for Computing Machinery, 2008. p. 1065–1068. ISBN 9781605581934.

CHRISTEN, P. Febrl: A freely available record linkage system with a graphical user interface. In: **Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management - Volume 80**. AUS: Australian Computer Society, Inc., 2008. (HDKM '08), p. 17–25. ISBN 9781920682613.

CHRISTEN, P. Data matching systems. In: **Data Matching**. [S.l.]: Springer Berlin Heidelberg, 2012. p. 229–242.

CHRISTOPHIDES, V.; EFTHYMIOU, V.; PALPANAS, T.; PAPADAKIS, G.; STEFANIDIS, K. An overview of end-to-end entity resolution for big data. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 53, n. 6, p. 1–42, dec 2020.

CHRISTOPHIDES, V.; EFTHYMIOU, V.; STEFANIDIS, K. **Entity Resolution in the Web of Data**. [S.l.]: Springer International Publishing, 2015.

CICHY, R. M.; KAISER, D. Deep neural networks as scientific models. **Trends in Cognitive Sciences**, Elsevier BV, v. 23, n. 4, p. 305–317, apr 2019.

CONNEAU, A.; BAEVSKI, A.; COLLOBERT, R.; MOHAMED, A.; AULI, M. Unsupervised Cross-Lingual Representation Learning for Speech Recognition. In: **Proc. Interspeech 2021**. [S.l.: s.n.], 2021. p. 2426–2430.

CONNEAU, A.; KHANDELWAL, K.; GOYAL, N.; CHAUDHARY, V.; WENZEK, G.; GUZMÁN, F.; GRAVE, E.; OTT, M.; ZETTLEMOYER, L.; STOYANOV, V. Unsupervised cross-lingual representation learning at scale. In: JURAFSKY, D.; CHAI, J.; SCHLUTER, N.; TETREAULT, J. (Ed.). **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**. Online: Association for Computational Linguistics, 2020. p. 8440–8451. Disponível em: <https://aclanthology.org/2020.acl-main.747>.

DENG, L.; LIU, Y. **Deep learning in natural language processing**. [S.l.]: Springer, 2018.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for**

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186.

DOAN, A.; HALEVY, A.; IVES, Z. **Principles of data integration**. [S.l.]: Elsevier, 2012.

DONG, X. L.; HALEVY, A.; YU, C. Data integration with uncertainty. **The VLDB Journal**, Springer, v. 18, p. 469–500, 2009.

DUNN, H. L. Record linkage. **American Journal of Public Health and the Nations Health**, American Public Health Association, v. 36, n. 12, p. 1412–1416, 1946.

EBRAHEEM, M.; THIRUMURUGANATHAN, S.; JOTY, S. R.; OUZZANI, M.; TANG, N. Deeper - deep entity resolution. **CoRR**, abs/1710.00597, 2017. Disponível em: <http://arxiv.org/abs/1710.00597>.

EFTHYMIOU, V.; PAPADAKIS, G.; PAPASTEFANATOS, G.; STEFANIDIS, K.; PALPANAS, T. Parallel meta-blocking for scaling entity resolution over big heterogeneous data. **Information Systems**, Elsevier, v. 65, p. 137–157, 2017.

EISENSTEIN, J. **Natural language processing**. [S.l.]: MIT press, 2018.

ELMAGARMID, A. K.; IPEIROTIS, P. G.; VERYKIOS, V. S. Duplicate record detection: A survey. **IEEE Transactions on Knowledge and Data Engineering**, Institute of Electrical and Electronics Engineers (IEEE), v. 19, n. 1, p. 1–16, jan 2007.

EMBAR, V.; SISMAN, B.; WEI, H.; DONG, X. L.; FALOUTSOS, C.; GETOOR, L. Contrastive entity linkage: Mining variational attributes from large catalogs for entity linkage. In: **Automated Knowledge Base Construction**. [s.n.], 2020. Disponível em: <https://openreview.net/forum?id=fR44nF03Rb>.

ENAMOTO, L.; WEIGANG, L.; FILHO, G. P. R. Generic framework for multilingual short text categorization using convolutional neural network. **Multimedia Tools and Applications**, Springer, v. 80, p. 13475–13490, 2021.

FELLEGI, I. P.; SUNTER, A. B. A theory for record linkage. **Journal of the American Statistical Association**, Taylor & Francis, v. 64, n. 328, p. 1183–1210, 1969.

FEURER, M.; EGGENSPERGER, K.; FALKNER, S.; LINDAUER, M.; HUTTER, F. Auto-sklearn 2.0: hands-free automl via meta-learning. **J. Mach. Learn. Res.**, JMLR.org, v. 23, n. 1, jan 2022. ISSN 1532-4435.

FEURER, M.; KLEIN, A.; EGGENSPERGER, K.; SPRINGENBERG, J.; BLUM, M.; HUTTER, F. Efficient and robust automated machine learning. In: **Advances in Neural Information Processing Systems 28 (2015)**. [S.l.: s.n.], 2015. p. 2962–2970.

FIRMANI, D.; SAHA, B.; SRIVASTAVA, D. Online entity resolution using an oracle. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 9, n. 5, p. 384–395, 2016.

FIRMINO, A. A.; BAPTISTA, C. de S.; PAIVA, A. C. de. Using cross lingual learning for detecting hate speech in portuguese. In: STRAUSS, C.; KOTSIS, G.; TJOA, A. M.; KHALIL, I. (Ed.). **Database and Expert Systems Applications - 32nd International Conference, DEXA 2021, Virtual Event, September 27-30, 2021, Proceedings, Part II**. [S.l.]: Springer, 2021. (Lecture Notes in Computer Science, v. 12924), p. 170–175.

FOXCROFT, J.; CHEN, T.; PADMANABHAN, K.; KENG, B.; ANTONIE, L. Product matching lessons and recommendations from a real world application. In: **Canadian Conference on AI**. [S.l.: s.n.], 2021.

GHAHRAMANI, Z. Unsupervised learning. In: **Advanced Lectures on Machine Learning**. [S.l.]: Springer Berlin Heidelberg, 2004. p. 72–112.

GHAWI, R.; PFEFFER, J. Efficient hyperparameter tuning with grid search for text categorization using kNN approach with BM25 similarity. **Open Computer Science**, Walter de Gruyter GmbH, v. 9, n. 1, p. 160–180, jan 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: **Lecture Notes in Computer Science**. [S.l.]: Springer Berlin Heidelberg, 2005. p. 345–359.

HAMBARDE, K. A.; PROENçA, H. Information retrieval: Recent advances and beyond. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 11, p. 76581–76604, 2023. ISSN 2169-3536.

HAN, J.; PEI, J.; TONG, H. (Ed.). **Data Mining: Concepts and Techniques**. Fourth edition. [S.l.]: Morgan Kaufmann, 2023. 752 p. ISBN 978-0-12-811760-6.

HEIJDEN, N. van der; YANNAKOUDAKIS, H.; MISHRA, P.; SHUTOVA, E. Multilingual and cross-lingual document classification: A meta-learning approach. In: **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**. Online: Association for Computational Linguistics, 2021. p. 1966–1976.

HERBERT, D.; KANG, B. H. Intelligent conversation system using multiple classification ripple down rules and conversational context. **Expert Systems with Applications**, v. 112, p. 342–352, 2018. ISSN 0957-4174.

H.GOMAA, W.; FAHMY, A. A. A survey of text similarity approaches. **International Journal of Computer Applications**, Foundation of Computer Science, v. 68, n. 13, p. 13–18, apr 2013.

IBA, H.; NOMAN, N. (Ed.). **Deep Neural Evolution**. [S.l.]: Springer Singapore, 2020.

ITO, S.; FUJIMAKI, R. Large-scale price optimization via network flow. **Advances in Neural Information Processing Systems**, v. 29, 2016.

JACOB, E. **Introduction to Natural Language Processing**. [S.l.]: MIT press, 2019.

JAIN, A.; PATEL, H.; NAGALAPATTI, L.; GUPTA, N.; MEHTA, S.; GUTTULA, S.; MUJUMDAR, S.; AFZAL, S.; MITTAL, R. S.; MUNIGALA, V. Overview and importance of data quality for machine learning tasks. In: **Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.]: ACM, 2020.

JOVANOVIC, J.; BAGHERI, E. Electronic commerce meets the semantic web. **It Professional**, IEEE, v. 18, n. 4, p. 56–65, 2016.

JäRVELIN, K.; KEKäLäINEN, J. Cumulated gain-based evaluation of IR techniques. **ACM Transactions on Information Systems**, Association for Computing Machinery (ACM), v. 20, n. 4, p. 422–446, oct 2002.

KIECKBUSCH, D. S.; FILHO, P. G.; OLIVEIRA, V. D.; WEIGANG, L. Scan-nf: A cnn-based system for the classification of electronic invoices through short-text product description. In: **WEBIST**. [S.l.: s.n.], 2021. p. 501–508.

KIM, S.; MIN, J.; CHO, M. Transformatcher: Match-to-match attention for semantic correspondence. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022. p. 8697–8707.

KONDA, P. V. **Magellan: Toward building entity matching management systems**. [S.l.]: The University of Wisconsin-Madison, 2018.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. et al. Supervised machine learning: A review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, Amsterdam, v. 160, n. 1, p. 3–24, 2007.

KUMAR, G.; JAIN, S.; SINGH, U. P. Stock market forecasting using computational intelligence: A survey. **Archives of Computational Methods in Engineering**, Springer Science and Business Media LLC, v. 28, n. 3, p. 1069–1101, feb 2020.

KöPCKE, H.; THOR, A.; RAHM, E. Evaluation of entity resolution approaches on real-world match problems. **Proceedings of the VLDB Endowment**, Association for Computing Machinery (ACM), v. 3, n. 1-2, p. 484–493, sep 2010.

LADANAVAR, S.; KAMBLE, R.; GOUDAR, R.; KALIWAL, R.; RATHOD, V.; DESHPANDE, S.; GM, D.; KULKARNI, A. Enhancing user query comprehension and contextual relevance with a semantic search engine using bert and elasticsearch. **EAI Endorsed Transactions on Internet of Things**, v. 10, 08 2024.

LAI, P.; YE, F.; FU, Y.; CHEN, Z.; WU, Y.; WANG, Y.; CHANG, V. Cognlg: Cognitive graph for kg-to-text generation. **Expert Syst. J. Knowl. Eng.**, v. 41, n. 1, 2024. Disponível em: <https://doi.org/10.1111/exsy.13461>.

LAMPLE, G.; BALLESTEROS, M.; SUBRAMANIAN, S.; KAWAKAMI, K.; DYER, C. Neural architectures for named entity recognition. In: KNIGHT, K.; NENKOVA, A.; RAMBOW, O. (Ed.). **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. San Diego, California: Association for Computational Linguistics, 2016. p. 260–270.

LAN, Z.; CHEN, M.; GOODMAN, S.; GIMPEL, K.; SHARMA, P.; SORICUT, R. **ALBERT: A Lite BERT for Self-supervised Learning of Language Representations**. 2020.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Springer Science and Business Media LLC, v. 521, n. 7553, p. 436–444, may 2015.

LI, H. **Learning to Rank for Information Retrieval and Natural Language Processing**. [S.l.]: Springer International Publishing, 2015.

LI, Q.; ZHANG, H.; HONG, X. Knowledge structure of technology licensing based on co-keywords network: A review and future directions. **International Review of Economics & Finance**, Elsevier BV, v. 66, p. 154–165, mar 2020.

LI, Q.; ZHAO, S.; ZHAO, S.; WEN, J. Logistic regression matching pursuit algorithm for text classification. **Knowledge-Based Systems**, Elsevier BV, p. 110761, jul 2023.

LI, Y.; LI, J.; SUHARA, Y.; DOAN, A.; TAN, W.-C. Deep entity matching with pre-trained language models. **Proceedings of the VLDB Endowment**, Association for Computing Machinery (ACM), v. 14, n. 1, p. 50–60, sep 2020.

LILLIS, D. On the evaluation of data fusion for information retrieval. In: **Forum for Information Retrieval Evaluation**. [S.l.]: ACM, 2020.

LIMA, R. R. de; FERNANDES, A. M. R.; BOMBASAR, J. R.; SILVA, B. A. da; CROCKER, P.; LEITHARDT, V. R. Q. An empirical comparison of portuguese and multilingual BERT models for auto-classification of NCM codes in international trade. **Big Data and Cognitive Computing**, MDPI AG, v. 6, n. 1, p. 8, jan 2022.

LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L.; STOYANOV, V. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. 2019.

LU, J.; LIN, C.; WANG, W.; LI, C.; WANG, H. String similarity measures and joins with synonyms. In: **Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data**. [S.l.]: ACM, 2013.

LUCENA, L. F.; FILHO, T. de Menezes e S.; RêGO, T. G. do; MALHEIROS, Y. Automatic recognition of units of measurement in product descriptions from tax invoices using neural networks. In: **Computational Processing of the Portuguese Language: 15th International Conference, PROPOR 2022, Fortaleza, Brazil, March 21–23, 2022, Proceedings**. Berlin, Heidelberg: Springer-Verlag, 2022. p. 156–165. ISBN 978-3-030-98304-8.

LUDERMIR, T. B. Inteligência artificial e aprendizado de máquina: estado atual e tendências. **Estudos Avançados**, FapUNIFESP (SciELO), v. 35, n. 101, p. 85–94, apr 2021.

ŁUKASIK, S.; MICHAŁOWSKI, A.; KOWALSKI, P. A.; GANDOMI, A. H. Text-based product matching with incomplete and inconsistent items descriptions. In: PASZYNSKI, M.; KRANZLMÜLLER, D.; KRZHIZHANOVSKAYA, V. V.; DONGARRA, J. J.; SLOOT, P. M. A. (Ed.). **Computational Science – ICCS 2021**. Cham: Springer International Publishing, 2021. p. 92–103. ISBN 978-3-030-77964-1.

LV, Y.; ZHAI, C. Adaptive term frequency normalization for bm25. In: **Proceedings of the 20th ACM International Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2011. (CIKM '11), p. 1985–1988. ISBN 9781450307178. Disponível em: <https://doi.org/10.1145/2063576.2063871>.

LV, Y.; ZHAI, C. Lower-bounding term frequency normalization. In: **Proceedings of the 20th ACM International Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2011. (CIKM '11), p. 7–16. ISBN 9781450307178. Disponível em: <https://doi.org/10.1145/2063576.2063584>.

LV, Y.; ZHAI, C. When documents are very long, bm25 fails! In: **Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2011. (SIGIR '11), p. 1103–1104. ISBN 9781450307574. Disponível em: <https://doi.org/10.1145/2009916.2010070>.

LV, Y.; ZHAI, C. A log-logistic model-based interpretation of tf normalization of bm25. In: BAEZA-YATES, R.; VRIES, A. P. de; ZARAGOZA, H.; CAMBAZOGLU, B. B.; MURDOCK, V.; LEMPEL, R.; SILVESTRI, F. (Ed.). **Advances in Information Retrieval**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 244–255. ISBN 978-3-642-28997-2.

MANNING, C. D. **An introduction to information retrieval**. [S.l.]: Cambridge university press, 2009.

MANNING, C. D. Human language understanding & reasoning. **Daedalus**, MIT Press, v. 151, n. 2, p. 127–138, 2022.

MIKOLOV, T.; CORRADO, G.; CHEN, K.; DEAN, J. Efficient estimation of word representations in vector space. In: . [S.l.: s.n.], 2013. p. 1–12.

MIN, B.; ROSS, H.; SULEM, E.; VEYSEH, A. P. B.; NGUYEN, T. H.; SAINZ, O.; AGIRRE, E.; HEINTZ, I.; ROTH, D. Recent advances in natural language processing via large pre-trained language models: A survey. **ACM Computing Surveys**, Association for Computing Machinery (ACM), jun 2023.

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. ISBN 0070428077.

MOŻDŻONEK, M.; WRÓBLEWSKA, A.; TKACHUK, S.; ŁUKASIK, S. Multilingual transformers for product matching–experiments and a new benchmark in polish. In: IEEE. **2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.], 2022. p. 1–8.

MOZDZONEK, M.; WROBLEWSKA, A.; TKACHUK, S.; LUKASIK, S. Multilingual Transformers for Product Matching – Experiments and a New Benchmark in Polish. In: **2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.]: IEEE, 2022. p. 1–8. ISBN 978-1-6654-6710-0.

MRABET, M. A. E.; MAKKAOUI, K. E.; FAIZE, A. Supervised machine learning: A survey. In: **2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet)**. [S.l.: s.n.], 2021. p. 1–10.

MUDGAL, S.; LI, H.; REKATSINAS, T.; DOAN, A.; PARK, Y.; KRISHNAN, G.; DEEP, R.; ARCAUTE, E.; RAGHAVENDRA, V. Deep Learning for Entity Matching. In: **Proceedings of the 2018 International Conference on Management of Data**. New York, NY, USA: ACM, 2018. p. 19–34. ISBN 9781450347037.

NANDWANI, P.; VERMA, R. A review on sentiment analysis and emotion detection from text. **Social Network Analysis and Mining**, Springer Science and Business Media LLC, v. 11, n. 1, aug 2021.

NARARATWONG, R.; KERTKEIDKACHORN, N.; ICHISE, R. Knowledge graph visualization: challenges, framework, and implementation. In: IEEE. **2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)**. [S.l.], 2020. p. 174–178.

NASCIMENTO, D. C.; PIRES, C. E. S.; MESTRE, D. G. Exploiting block co-occurrence to control block sizes for entity resolution. **Knowledge and Information Systems**, Springer Science and Business Media LLC, v. 62, n. 1, p. 359–400, mar 2019.

NAVARRO, G. A guided tour to approximate string matching. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 33, n. 1, p. 31–88, mar. 2001. ISSN 0360-0300.

OLIVEIRA, A. B. D.; BAPTISTA, C. d. S.; FIRMINO, A. A.; PAIVA, A. C. D. A large language model approach to detect hate speech in political discourse using multiple language corpora. In: **Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing**. New York, NY, USA: Association for Computing Machinery, 2024. (SAC '24), p. 1461–1468. ISBN 9798400702433.

OLIVEIRA, A. B. de; BAPTISTA, C. de S.; FIRMINO, A. A.; PAIVA, A. C. de. Using multilingual approach in cross-lingual transfer learning to improve hate speech detection. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). **Proceedings of the 25th International Conference on Enterprise Information Systems, ICEIS 2023, Volume 1, Prague, Czech Republic, April 24-26, 2023**. [S.l.]: SCITEPRESS, 2023. p. 374–384.

O'REILLY, T. What is web 2.0: Design patterns and business models for the next generation of software. **International Journal of Digital Economics**, p. 17–37, 2007.

P., U.; GOVINDAN, V.; Madhu Kumar, S. Enhanced sparse representation classifier for text classification. **Expert Systems with Applications**, v. 129, p. 260–272, 2019. ISSN 0957-4174.

PAGANELLI, M.; BUONO, F. D.; BARALDI, A.; GUERRA, F. Analyzing how bert performs entity matching. **Proc. VLDB Endow.**, VLDB Endowment, v. 15, n. 8, p. 1726–1738, apr 2022. ISSN 2150-8097.

PAJILA, P. B.; SUDHA, K.; SELVI, D. K.; KUMAR, V. N.; GAYATHRI, S.; SUBRAMANIAN, R. S. A survey on natural language processing and its applications.

In: **2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)**. [S.l.: s.n.], 2023. p. 996–1001.

PAN, M.; PEI, Q.; LIU, Y.; LI, T.; HUANG, E. A.; WANG, J.; HUANG, J. X. Sprf: A semantic pseudo-relevance feedback enhancement for information retrieval via conceptnet. **Knowledge-Based Systems**, v. 274, p. 110602, 2023. ISSN 0950-7051.

PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, n. 10, p. 1345–1359, 2010.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. **Foundations and Trends® in Information Retrieval**, v. 2, n. 1–2, p. 1–135, 2008. ISSN 1554-0669.

PAPADAKIS, G.; NEJDL, W. Efficient entity resolution methods for heterogeneous information spaces. In: **2011 IEEE 27th International Conference on Data Engineering Workshops**. [S.l.]: IEEE, 2011.

PAPADAKIS, G.; SKOUTAS, D.; THANOS, E.; PALPANAS, T. Blocking and Filtering Techniques for Entity Resolution. **ACM Computing Surveys**, v. 53, n. 2, p. 1–42, mar 2021. ISSN 0360-0300.

PATIL, R.; BOIT, S.; GUDIVADA, V.; NANDIGAM, J. A survey of text representation and embedding techniques in nlp. **IEEE Access**, v. 11, p. 36120–36146, 2023.

PEETERS, R.; BIZER, C. Supervised contrastive learning for product matching. In: **Companion Proceedings of the Web Conference 2022**. New York, NY, USA: Association for Computing Machinery, 2022. (WWW '22), p. 248–251. ISBN 9781450391306.

PEETERS, R.; BIZER, C.; GLAVAŠ, G. Intermediate training of bert for product matching. **small**, v. 745, n. 722, p. 2–112, 2020.

PEETERS, R.; DER, R. C.; BIZER, C. WDC products: A multi-dimensional entity matching benchmark. In: TANCA, L.; LUO, Q.; POLESE, G.; CARUCCIO, L.; ORIOL, X.; FIRMANI, D. (Ed.). **Proceedings 27th International Conference on Extending Database Technology, EDBT 2024, Paestum, Italy, March 25 - March 28**. [S.l.]: OpenProceedings.org, 2024. p. 22–33.

PEETERS, R.; PRIMPELI, A.; WICHTLHUBER, B.; BIZER, C. Using schema.org annotations for training and maintaining product matchers. In: **Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics**. [S.l.]: ACM, 2020.

PENNINGTON, J.; SOCHER, R.; MANNING, C. GloVe: Global vectors for word representation. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Disponível em: <https://aclanthology.org/D14-1162>.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. In: WALKER, M.; JI, H.; STENT, A. (Ed.). **Proceedings of the 2018 Conference of the**

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 2227–2237. Disponível em: <https://aclanthology.org/N18-1202>.

PETROVSKI, P.; BRYL, V.; BIZER, C. Integrating product data from websites offering microdata markup. In: **Proceedings of the 23rd International Conference on World Wide Web**. [S.l.]: ACM, 2014.

PIKULIAK, M.; ŠIMKO, M.; BIELIKOVA, M. Cross-lingual learning for text processing: A survey. **Expert Systems with Applications**, Elsevier, v. 165, p. 113765, 2021.

PRIMPELI, A.; PEETERS, R.; BIZER, C. The WDC training dataset and gold standard for large-scale product matching. In: **Companion Proceedings of The 2019 World Wide Web Conference**. [S.l.]: ACM, 2019.

PROVOST, F.; FAWCETT, T. Data science and its relationship to big data and data-driven decision making. **Big Data**, v. 1, n. 1, p. 51–59, 2013.

QIN, T.; LIU, T.-Y.; XU, J.; LI, H. Letor: A benchmark collection for research on learning to rank for information retrieval. **Inf. Retr.**, v. 13, p. 346–374, 08 2010.

QIU, M.; YANG, L.; JI, F.; ZHAO, W.; ZHOU, W.; HUANG, J.; CHEN, H.; CROFT, W. B.; LIN, W. Transfer learning for context-aware question matching in information-seeking conversations in e-commerce. **arXiv preprint arXiv:1806.05434**, 2018.

RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I. et al. Language models are unsupervised multitask learners. **OpenAI blog**, v. 1, n. 8, p. 9, 2019.

RATERIA, S.; SINGH, S. Transparent, low resource, and context-aware information retrieval from a closed domain knowledge base. **IEEE Access**, v. 12, p. 44233–44243, 03 2024.

RISTOSKI, P.; PETROVSKI, P.; MIKA, P.; PAULHEIM, H. A machine learning approach for product matching and categorization. **Semantic web**, IOS Press, v. 9, n. 5, p. 707–728, 2018.

ROBERTSON, S.; ZARAGOZA, H. The probabilistic relevance framework: Bm25 and beyond. **Foundations and Trends® in Information Retrieval**, v. 3, n. 4, p. 333–389, 2009. ISSN 1554-0669. Disponível em: <http://dx.doi.org/10.1561/1500000019>.

ROBERTSON, S.; ZARAGOZA, H.; TAYLOR, M. Simple bm25 extension to multiple weighted fields. In: **Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2004. (CIKM '04), p. 42–49. ISBN 1581138741.

RODRIGUES, J. a.; GOMES, L.; SILVA, J. a.; BRANCO, A.; SANTOS, R.; CARDOSO, H. L.; OSóRIO, T. Advancing neural encoding of portuguese with transformer albertina pt-*. In: **Progress in Artificial Intelligence: 22nd EPIA Conference on Artificial Intelligence, EPIA 2023, Faial Island, Azores, September 5–8, 2023,**

Proceedings, Part I. Berlin, Heidelberg: Springer-Verlag, 2023. p. 441–453. ISBN 978-3-031-49007-1.

ROMUALDO, A.; REAL, L.; CASELI, H. Measuring brazilian portuguese product titles similarity using embeddings. In: **Anais do Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)**. [S.l.: s.n.], 2021. p. 121–132.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM J. Res. Dev.**, v. 44, p. 206–227, 1959.

SANH, V.; DEBUT, L.; CHAUMOND, J.; WOLF, T. **DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter**. 2020.

SANTANA, M. A. de; BAPTISTA, C. de S.; ALVES, A. L. F.; FIRMINO, A. A.; JANUÁRIO, G. da S.; CALDERA, R. W. da S. Using machine learning and NLP for the product matching problem. In: **Intelligent Sustainable Systems**. [S.l.]: Springer Nature Singapore, 2023. p. 439–448.

SAQUETE, E.; TOMáS, D.; MOREDA, P.; MARTíNEZ-BARCO, P.; PALOMAR, M. Fighting post-truth using natural language processing: A review and open challenges. **Expert Systems with Applications**, v. 141, p. 112943, 2020. ISSN 0957-4174.

SCHULTE, J. P.; GIUNTINI, F. T.; NOBRE, R. A.; NASCIMENTO, K. C. do; MENEGUETTE, R. I.; LI, W.; GONÇALVES, V. P.; Rocha Filho, G. P. ELINAC: Autoencoder Approach for Electronic Invoices Data Clustering. **Applied Sciences**, v. 12, n. 6, p. 3008, mar 2022. ISSN 2076-3417.

SCHUTH, A.; SIETSMA, F.; WHITESON, S.; RIJKE, M. de. Optimizing base rankers using clicks. In: **Lecture Notes in Computer Science**. [S.l.]: Springer International Publishing, 2014. p. 75–87.

SENNRICH, R.; HADDOW, B.; BIRCH, A. Neural machine translation of rare words with subword units. In: ERK, K.; SMITH, N. A. (Ed.). **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 1715–1725.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. In: CERRI, R.; PRATI, R. C. (Ed.). **Intelligent Systems**. Cham: Springer International Publishing, 2020. p. 403–417. ISBN 978-3-030-61377-8.

STAPPEN, L.; BRUNN, F.; SCHULLER, B. **Cross-lingual Zero- and Few-shot Hate Speech Detection Utilising Frozen Transformer Language Models and AXEL**. 2020.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. [S.l.]: MIT press, 2018.

TAN, C.; SUN, F.; KONG, T.; ZHANG, W.; YANG, C.; LIU, C. A survey on deep transfer learning. In: **Artificial Neural Networks and Machine Learning – ICANN 2018**. [S.l.]: Springer International Publishing, 2018. p. 270–279.

TRACZ, J.; WÓJCIK, P. I.; JASINSKA-KOBUS, K.; BELLUZZO, R.; MROCZKOWSKI, R.; GAWLIK, I. BERT-based similarity learning for product matching. **Proceedings of Workshop on Natural Language Processing in E-Commerce**, p. 66–75, 2020.

TROTMAN, A.; PUURULA, A.; BURGESS, B. Improvements to bm25 and language models examined. In: **Proceedings of the 19th Australasian Document Computing Symposium**. New York, NY, USA: Association for Computing Machinery, 2014. (ADCS '14), p. 58–65. ISBN 9781450330008. Disponível em: <https://doi.org/10.1145/2682862.2682863>.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. **Mind**, Oxford University Press (OUP), LIX, n. 236, p. 433–460, oct 1950.

VANDIC, D.; FRASINCAR, F.; KAYMAK, U.; RIEZEBOS, M. Scalable entity resolution for Web product descriptions. **Information Fusion**, v. 53, p. 103–111, jan 2020. ISSN 15662535.

VASHISHTHA, S.; SUSAN, S. Fuzzy rule based unsupervised sentiment analysis from social media posts. **Expert Systems with Applications**, v. 138, p. 112834, 2019. ISSN 0957-4174.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

VIEIRA, H. S.; SILVA, A. S. da; CALADO, P.; MOURA, E. S. de. A distantly supervised approach for recognizing product mentions in user-generated content. **Journal of Intelligent Information Systems**, Springer Science and Business Media LLC, v. 59, n. 3, p. 543–566, may 2022.

VIJAYARANI, S.; ILAMATHI, M. J.; NITHYA, M. et al. Preprocessing techniques for text mining-an overview. **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2015.

VOORHEES, E.; HARMAN, D. K. **TREC**: Experiment and evaluation in information retrieval (digital libraries and electronic publishing). [S.l.]: The MIT Press, 2005. 472 p. ISBN 9780262220736.

WAHAB, M. H. H.; ALI, N. H.; HAMID, N. A. W. A.; SUBRAMANIAM, S. K.; LATIP, R.; OTHMAN, M. A review on optimization-based automatic text summarization approach. **IEEE Access**, v. 12, p. 4892–4909, 2024. Disponível em: <https://doi.org/10.1109/ACCESS.2023.3348075>.

WANG, A.; SINGH, A.; MICHAEL, J.; HILL, F.; LEVY, O.; BOWMAN, S. R. **GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding**. [S.l.]: arXiv, 2018.

WANG, J.; PAN, M.; HE, T.; HUANG, X.; WANG, X.; TU, X. A pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval. **Information Processing and Management**, v. 57, n. 6, p. 102342, 2020. ISSN 0306-4573.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big Data**, Springer Science and Business Media LLC, v. 3, n. 1, may 2016.

WILKE, M.; RAHM, E. Towards multi-modal entity resolution for product matching. In: **GvDB**. [S.l.: s.n.], 2021.

XU, P.; LU, J. Towards a unified framework for string similarity joins. **Proc. VLDB Endow.**, VLDB Endowment, v. 12, n. 11, p. 1289–1302, jul 2019. ISSN 2150-8097.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. **Proceedings of the IEEE**, IEEE, v. 109, n. 1, p. 43–76, 2020.

APÊNDICE A – CRAWLER DE PRODUTOS POR NCM

```
1
2  from selenium import webdriver
3  from selenium.webdriver.common.by import By
4  import pandas as pd
5
6  driver = webdriver.Chrome()
7
8  total_list = []
9
10 page = 0
11 nomes = [page]
12 #examples
13 categoria = "04012010-leite-uht-ultra-high-temperature"
14 url = "https://cosmos.bluesoft.com.br/ncms/"+categoria+"/products?page={}"
15
16 while len(nomes) != 0:
17     page += 1
18     driver.get(url.format(page))
19
20     nomes = driver.find_elements(by=By.CLASS_NAME, value='description a')
21     eans = driver.find_elements(by=By.CLASS_NAME, value='barcode a')
22
23     for nome, ean in zip(nomes, eans):
24         total_list.append([nome.text, ean.text])
25
26 pd.DataFrame(total_list, columns=['nome', 'ean'])
```

APÊNDICE B – PRÉ-PROCESSAMENTO DOS DADOS DE PRODUTOS

```
1
2  def validate_barcode(self, barcode: str) -> bool:
3      return barcodenumber.check_code('EAN13', barcode)
4
5  def str_stem(self, s: str) -> str:
6   if isinstance(s, str):
7    s = s.upper()
8    # Removendo os ';'
9    s = re.sub(r'( *)\;( *)', ' ', s)
10
11   # Trocar virgula entre os decimais por ponto e eliminar os espacos
12   s = re.sub(r'([0-9]*)( *)\,( *)([0-9]*)', r'\1.\4', s)
13
14   regex_start = r'(([0-9]+)|(\/))(\s?)'
15   regex_end = r'(\s|$|\/|\.|\))'
16
17   s = re.sub(regex_start + r'(M|MTS?|METROS?)' + regex_end, r'\1 METROS ', s)
18   s = re.sub(regex_start + r'(A|AMPERES)' + regex_end, r'\1 METROS ', s)
19   s = re.sub(regex_start + r'(L|LITROS?|LTS?)' + regex_end, r'\1 LITROS ', s)
20   s = re.sub(regex_start + r'(GR?|GRAMAS?)' + regex_end, r'\1 GRAMAS ', s)
21   s = re.sub(regex_start + r'(MM)' + regex_end, r'\1 MILIMETROS ', s)
22   s = re.sub(regex_start + r'(UG|MCG)' + regex_end, r'\1 MICROGRAMAS ', s)
23   s = re.sub(regex_start + r'(ML)' + regex_end, r'\1 MILILITROS ', s)
24   s = re.sub(regex_start + r'(MG)' + regex_end, r'\1 MILIGRAMAS ', s)
25   s = re.sub(regex_start + r'(UND?|UN|UNID?|UNIDADES?)' + regex_end, r'\1 UNIDADES
26   s = re.sub(regex_start + r'(KG|KILOS?)' + regex_end, r'\1 KILOGRAMAS ', s)
27   s = re.sub(regex_start + r'(W|WATTS?)' + regex_end, r'\1 WATTS ', s)
28   s = re.sub(regex_start + r'(V|VOLTS?)' + regex_end, r'\1 WATTS ', s)
29
30   s = re.sub(r'\sC\/\s', r' COM ', s)
31   s = re.sub(r'\sC\/', r' COM ', s)
32   return s
33
34 def normalize_string(self, string: str) -> str:
35     result = string.upper()
```

```
36      result = self.str_stem(string)
37      result = re.sub(r'\(', r' ', result)
38      result = re.sub(r'\)', r' ', result)
39      result = re.sub(r'\-', r' ', result)
40      result = re.sub(r'\+', r' ', result)
41      result = re.sub(r'\;', r' ', result)
42      result = re.sub(r'\'', r' ', result)
43      result = re.sub(r'\*', r' ', result)
44      result = re.sub(r'\]', r' ', result)
45      result = re.sub(r'\[', r' ', result)
46      result = re.sub(r'\:', r' ', result)
47      result = re.sub(r'\/', r' ', result)
48      result = re.sub(r'\~', r' ', result)
49      result = re.sub(r'\#', r' ', result)
50      result = re.sub(r'^\s*\.', r' ', result)
51
52      result = re.sub(r'(\D|^)\.(\d|$)', r'\1 \2', result)
53      result = re.sub(r'(\d|^)\.(\D|$)', r'\1 \2', result)
54      result = re.sub(r'(\D|^)\.(\D|$)', r'\1 \2', result)
55
56      result = result.split(' ')
57      result = list(map(unidecode, result))
58      result = list(filter(lambda word: word != '', result))
59      return ' '.join(result)
60
61  def remove_unicode_characters(self, string):
62      result = string
63      result = re.sub(r'\u0001', ' ', result)
64      result = re.sub(r'\u0002', ' ', result)
65      return result
66
67  def clean_data(self, df_produtos: DataFrame):
68      cleaned_dataframe = df_produtos.copy()
69
70      # Validando os codigos de barras
71      valid_col = cleaned_dataframe.apply(lambda row: self.validate_barcode(row['ean']
72      cleaned_dataframe = cleaned_dataframe[valid_col]
73
74      # Limpando os NCMs que contem somente 0
```

```
75      cleaned_dataframe.drop(cleaned_dataframe.loc[cleaned_dataframe['ncm'].str.contai
76
77      cleaned_dataframe = cleaned_dataframe.astype({'id': int, 'descricao': str})
78
79      # Normalizando as descricoes
80      cleaned_description_col = cleaned_dataframe.apply(lambda row: self.normalize_str
81      cleaned_dataframe.loc[:,'descricao'] = cleaned_description_col
82
83      # Removendo caracteres Unicode
84      cleaned_description_col = cleaned_dataframe.apply(lambda row: self.remove_unicod
85      cleaned_dataframe.loc[:,'descricao'] = cleaned_description_col
86
87      # Removendo strings vazias
88      cleaned_dataframe = cleaned_dataframe.loc[cleaned_dataframe.descricao.str.strip(
89
90      return cleaned_dataframe
91
92   # Heuristica pra analisar se as descricoes contem pelo menos 2 palavras identicas
93  def isSameProduct(self, descricao, descricao_canonica) -> bool:
94          descricao1_set = set(descricao.split(' '))
95  descricao2_set = set(descricao_canonica.split(' '))
96  tokens_intersecao = descricao1_set.intersection(descricao2_set)
97  return len(tokens_intersecao)>=2
```

APÊNDICE C – IMPLEMENTACÃO DO STEPMATCH - ALGORITMO 1: IDENTIFICADOR DE CORRESPONDÊNCIAS

```
1
2  # %% [markdown]
3  # Carregar o csv
4  # Selecionar o modelo de treinamento desejado (padrao = melhor modelo)
5  # Agrupar os produtos por ean
6  # Criar coluna com quantidade de vezes que a descricao aparece
7  # Ordenar grupo por quantidade de vezes que a descricao aparece
8  # Percorre os grupos verificando se a descricao que mais aparece da match com os de
9  # <img src="imgs/processo.png" alt="drawing" width="1200"/>
10
11 import torch
12 import warnings
13 import pandas as pd
14 from time import sleep
15 from typing import Tuple
16 from elasticsearch import Elasticsearch, helpers
17 from sklearn.metrics.pairwise import cosine_similarity
18 from sklearn.feature_extraction.text import TfidfVectorizer
19 from transformers import AutoTokenizer, AutoModelForSequenceClassification, logging
20
21 logging.set_verbosity_error()
22 warnings.simplefilter(action="ignore", category=pd.errors.SettingWithCopyWarning)
23
24 # Variables
25 csv = "./bases/leite_normal/test_sujo.csv"
26 bert = "neuralmind/bert-base-portuguese-cased"
27 model_path = "./modelos/leite_normal/pytorch_model.bin"
28
29 # Titulo principal com base na similaridade do cosseno
30 similaridade = False
31
32 # Config Model
33 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
34 max_length = 180
```

```
nclasses = 2

# Elastic Search
host = "http://localhost:9200"
user = "elastic"
password = "pass"
index_elastic = "index_elastic"
size = 50
es = Elasticsearch(host, basic_auth=(user, password))
es.indices.delete(index=index_elastic)

# Functions
model = AutoModelForSequenceClassification.from_pretrained(
    bert, num_labels=nclasses
).to(device)
model.load_state_dict(torch.load(model_path))
tokenizer = AutoTokenizer.from_pretrained(bert, do_lower_case=False)

es = Elasticsearch(host, basic_auth=(user, password))
if not es.ping():
    raise ValueError("Conexao com Elasticsearch falhou")
if es.indices.exists(index=index):
    raise RuntimeError(f"O indice '{index}' ja existe")
else:
    es.indices.create(index=index)

def Modelo(title1: list, title2: list, proba: bool = False):
    inputs = tokenizer(
        title1,
        title2,
        return_tensors="pt",
        padding=True,
        truncation=True,
        max_length=max_length,
    ).to(device)
    output = model(**inputs)
    list_labels = torch.argmax(output.logits, 1).tolist()
    list_proba = torch.softmax(output.logits, 1).tolist()
    list_proba_match = [proba[1] for proba in list_proba]
```

```
74      if proba:
75          return [list_labels, list_proba_match]
76      return list_labels
77
78  def SearchNewEan(text: str):
79      """ Retorna o titulo e o ean do Elastic Search """
80      resp = es.search(index=index, query={"match": {"title": text}}, size=size)
81      lista = [i["_source"] for i in resp["hits"]["hits"]]
82
83      if not lista:
84          return False
85
86      df = pd.DataFrame(lista, columns=["ean", "title"])
87      df["title_main"] = text
88
89      [list_labels, list_proba_match] = Modelo(
90          df["title_main"].tolist(), df["title"].tolist(), proba=True
91      )
92
93      df["predit"] = list_labels
94      df["proba"] = list_proba_match
95      df = df.sort_values("proba", ascending=False)
96
97      if df.iloc[0]["predit"] == 1:
98          return [df.iloc[0]["title"], df.iloc[0]["ean"]]
99      else:
100         return False
101
102 def GenerateActions(records: list, index_name: str):
103     for record in records:
104         yield {
105             "_index": index_name,
106             "_source": record,
107         }
108
109 def MainTitle(textos):
110     """
111     Descobre o titulo principal de uma lista com base na similaridade do cosseno
112     """
```

```
113     # Vetorizacao TF-IDF
114     vectorizer = TfidfVectorizer()
115     tfidf_matrix = vectorizer.fit_transform(textos)
116 
117     # Calculo da similaridade de cosseno
118     cosine_similarities = cosine_similarity(tfidf_matrix)
119 
120     # Conversao para DataFrame para melhor visualizacao
121     df_similaridade = pd.DataFrame(cosine_similarities, index=textos, columns=[texto
122 
123     # Mostra a matriz de similaridade
124     ordenados = (
125         df_similaridade.mean()
126         .reset_index(name="proba")
127         .sort_values("proba", ascending=False)
128     )
129 
130     return ordenados.iloc[0]["level_0"]
131 
132 # # ETAPA 1
133 def ETAPA_1(data: pd.DataFrame) -> pd.DataFrame:
134     df = pd.DataFrame()
135     for ean in data["ean"].unique():
136         group = data[data["ean"] == ean]
137         group["vezes"] = group["title"].map(group.value_counts("title"))
138         group = group.drop_duplicates(subset=["title"])
139         group = group.sort_values(by="vezes", ascending=False)
140         df = pd.concat([df, group])
141     # df = df.reset_index(drop=True)
142     return df
143 
144 # ETAPA 2
145 def ETAPA_2(data: pd.DataFrame) -> pd.DataFrame:
146     df = pd.DataFrame()
147     for ean in data["ean"].unique():
148         group = data[data["ean"] == ean]
149         if similaridade:
150             main_title = MainTitle(group["title"].to_list())
151         else:
```

```
152             main_title = group.iloc[0]["title"]
153         group["main_title"] = main_title
154         list_labels = Modelo(group["main_title"].tolist(), group["title"].tolist())
155         group["match"] = list_labels
156         df = pd.concat([df, group])
157 
158     df = df[["ean", "title", "match"]]
159     # df = df.reset_index(drop=True)
160     return df
161 
162 # ETAPA 3
163 def ETAPA_3(data: pd.DataFrame) -> Tuple[pd.DataFrame, pd.DataFrame, pd.DataFrame]:
164     matchs = data[data["match"] == 1]
165     matchs = matchs[["ean", "title"]]
166     records_match = matchs.to_dict(orient="records")
167     helpers.bulk(es, GenerateActions(records_match, index))
168     sleep(1)
169 
170     no_matchs = data[data["match"] == 0]
171     relocated = pd.DataFrame()
172     unallocated = pd.DataFrame()
173 
174     for i, row in no_matchs.iterrows():
175         new = SearchNewEan(row["title"])
176         if new:
177             row["new_ean"] = new[1]
178             row["title_comparacao"] = new[0]
179             new_row = pd.DataFrame(
180                 [row[["new_ean", "title_comparacao", "ean", "title"]]]
181             ).rename(columns={"ean": "ean_antigo", "new_ean": "ean"})
182             relocated = pd.concat([relocated, new_row])
183         else:
184             unallocated = pd.concat(
185                 [unallocated, pd.DataFrame([row[["ean", "title"]]])]
186             )
187 
188     records_relocated = relocated.to_dict(orient="records")
189     helpers.bulk(es, GenerateActions(records_relocated, index))
190 
```

```
    # matchs = matchs.reset_index(drop=True)
    # relocated = relocated.reset_index(drop=True)
    # unallocated = unallocated.reset_index(drop=True)

    return matchs, relocated, unallocated

# Run
teste_sujo = pd.read_csv(csv)
entrada = teste_sujo.rename(columns={
    "title_correto":"title",
    "ean_errado":"ean"
})
entrada = entrada[["title", "ean"]]

df_etapa1 = ETAPA_1(entrada)
df_etapa2 = ETAPA_2(df_etapa1)
matchs, relocated, unallocated = ETAPA_3(df_etapa2)

sujos = teste_sujo[teste_sujo["ean_correto"] != teste_sujo["ean_errado"]]

# Produtos que foram realocados corretamente
c = pd.DataFrame()
for i, r in relocated.iterrows():
    try:
        s = sujos.loc[i]
        if list(s[["ean_correto", "ean_errado"]]) == list(r[["ean", "ean_antigo"]]):
            c = pd.concat([c, pd.DataFrame([r])])
    except:
        pass

nao_realocados = unallocated[unallocated.index.isin(sujos.index.difference(relocate
movidos_match = matchs[matchs.index.isin(sujos.index.difference(relocated.index))]
realocados_mais = relocated.loc[relocated.index.difference(sujos.index)]
print(f"Dos {len(sujos)} produtos sujos, foram realocados: {len(c)}")
print(f"Dos {len(sujos)} produtos sujos, nao foram realocados: {len(nao_realocados)
print(f"Dos {len(sujos)} produtos sujos, foram considerados match: {len(movidos_mat
print(f"Produtos realocados que nao estavam sujos: {len(realocados_mais)}")
```