UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MATHEUS ALCANTARA DE SANTANA

UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA E NLP PARA O PROBLEMA DE PRODUCT MATCHING

CAMPINA GRANDE - PB

2022

MATHEUS ALCANTARA DE SANTANA

UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA E NLP PARA O PROBLEMA DE PRODUCT MATCHING

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Cláudio de Souza Baptista.

CAMPINA GRANDE - PB

2022

MATHEUS ALCANTARA DE SANTANA

UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA E NLP PARA O PROBLEMA DE PRODUCT MATCHING

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Cláudio de Souza Baptista

Orientador – UASC/CEEI/UFCG

Professora Dra. Joseana Macêdo Fechine

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 06 de abril de 2022.

CAMPINA GRANDE - PB

ABSTRACT

The e-commerce market grows every year, driven by technological advances that make the purchasing process more convenient and efficient. As a result, the number of sales grows, and the supply of products being sold over the internet also increases. Due to the large volume of offers, the difficulty in finding a specific product grows, as well as the consumer's ability to identify and group similar products in order to find the best deals. This occurs because, given two identical products, i.e. that have the same bar code, they are described in different ways. For this, there is a technique whose objective is to determine whether two products are equivalents, i.e., correspond to the same entity in the real world, using machine learning techniques, called product matching. In this work, several machine learning models were analyzed, including BERT, in order to choose the best model that will be used to identify products whose description does not match its bar code. The database used will be the product database of invoices issued in the state of Acre, Brazil, made available by an auditing agency of a federal state. At the end of the implementation, the model was able to satisfactorily classify invalid products.

Utilizando Técnicas de Aprendizagem de Máquina e NLP para o Problema de Product Matching

Matheus Alcantara de Santana
Laboratório de Sistemas de Informação
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil
matheus.santana@ccc.ufcg.edu.br

Cláudio de Souza Baptista
Laboratório de Sistemas de Informação
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil
baptista@computacao.ufcg.edu.br

RESUMO

O comércio eletrônico é um mercado que aumenta a cada ano, impulsionado pelos avanços tecnológicos que tornam mais cômodo e eficiente o processo de compra. Por consequência, o número de vendas cresce, aumentando também a oferta de produtos sendo comercializados na internet. Devido ao grande volume de ofertas, a dificuldade do consumidor de encontrar determinado produto cresce, bem como a sua capacidade de identificar e agrupar produtos iguais, com a finalidade de encontrar as melhores ofertas. Isso ocorre, pois, dados dois produtos iguais, ou seja, que possuem o mesmo código de barras, são descritos de formas diferentes. Para isso, existe uma técnica cujo objetivo é determinar se dois produtos são equivalentes, ou seja, correspondem à mesma entidade no mundo real, utilizando técnicas de aprendizagem de máquina, chamada product matching. No presente trabalho, foram analisados diversos modelos de aprendizagem de máquina, incluindo o BERT, com a finalidade de escolher o melhor modelo que será utilizado para identificar produtos os quais sua descrição não corresponde ao seu código de barras. A base de dados utilizada será a base de produtos de notas fiscais emitidas no Estado do Acre, disponibilizadas pelo Tribunal de Contas do Acre, TCE-AC. Ao final da implementação, o modelo foi capaz de classificar de maneira satisfatória os produtos inválidos.

Palavras-chave

product matching, machine learning, NLP, BERT.

1 INTRODUÇÃO

O aumento do número de vendas *online* cresce a cada ano. Segundo a Associação Brasileira de Comércio Eletrônico (ABComm), o comércio eletrônico teve um crescimento de 68% em 2020, atingindo assim a marca de R$126,3 bilhões [1]. Com o aumento da demanda, ocorre também um aumento na quantidade de fornecedores de um mesmo produto, gerando a necessidade de mecanismos que facilitem o agrupamento de um mesmo produto por parte do consumidor, de modo a encontrar as melhores condições de compra. Inclui-se aqui, o modelo de marketplace que vem sendo adotado por várias plataformas de e-commerce, no qual vários fornecedores ofertam seus produtos.

Entretanto, a tarefa de agrupar esses produtos idênticos não é uma tarefa trivial, pois os diferentes fornecedores de um mesmo produto não o representam da mesma forma. Além disso, pode haver inúmeros ruídos de naturezas diferentes nos dados dos produtos, como erros ortográficos, descrições incompletas, abreviadas ou genéricas, nome de marcas escritas de diferentes formas e informações irrelevantes. Mesmo existindo identificadores únicos, como EAN (European Article Number) e GTIN (Global Trade Item Number), eles podem estar incorretos ou ausentes, como mostra a figura 1. Ainda, pode ocorrer do código de barras e a descrição não corresponderem ao mesmo produto, como é o caso do último exemplo da figura 1, o qual o código de barras é válido, porém sua descrição está incorreta.

Código de barras	Descrição
SEM GTIN	SACO PLASTICO PARA LIXO 50 LITROS
07897664140202	DESINFETANTE MINUANO HERLBAL 500ML
NULL	TINTA HP 8100 PRETA BM CHEMICAL
7895110428980	CARTA IPECOL 114X162 CITRUS YELLOW

Figura 1: Exemplos de produtos com códigos de barras inválidos

Ainda, há órgãos de fiscalização dos gastos públicos, como os Tribunais de Contas dos Estados e o da União, que necessitam analisar diversos bens de consumo comprados pelos jurisdicionados, com a finalidade de garantir a execução orçamentária, bem como a transparência desses gastos, evitando problemas como sobrepreço, e primando pelo Princípio da Economicidade nas compras públicas. Tais Cortes de Contas necessitam, por exemplo, agrupar produtos iguais de modo a fazer análises que permitam identificar a média de preços desses produtos, para identificar nos termos de referência dos processos licitatórios dos jurisdicionados, aqueles que estão com preços praticados abusivamente.

O processo que classifica se dois produtos correspondem ao mesmo produto é conhecido como *product matching*. Segundo Paynes (2021) [2], *product matching* é o processo de unir o aprendizado de máquina e diferentes fontes de dados para fazer correspondência entre produtos. Usando técnicas de aprendizagem de máquina e processamento de linguagem natural, torna-se possível identificar padrões nos dados desses produtos e, dessa forma, parear dois produtos idênticos.

O presente trabalho teve como objetivo criar um modelo de aprendizagem de máquina capaz de classificar se duas descrições se referem à mesma entidade no mundo real. Para isso, foi feita a

análise entre alguns modelos de classificação supervisionada tradicionais e um modelo baseado em *transformers*. Depois da avaliação, foi escolhido o modelo que apresentou o melhor resultado para ser utilizado em uma aplicação real, no intuito de remover produtos inconsistentes, os quais suas descrições não correspondem ao seu código de barras.

O restante do trabalho está estruturado como segue. Na seção 2, foi feito um estudo do estado da arte na temática de *product matching*. Na seção 3 é abordada a metodologia utilizada neste trabalho, detalhando as etapas realizadas. Na seção 4, foi descrita a análise feita a partir das métricas obtidas com o treinamento dos modelos. A partir dessas análises, o melhor modelo foi implementado em uma aplicação real, detalhada na seção 5. Por último, na seção 6, são apresentadas as conclusões e sugestões para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Resolução de Entidade (Entity Resolution), também conhecida como Correspondência de Entidade (Entity Matching) ou ligação de registro (Record Linkage) é área de atuação que objetiva identificar entidades que representam os mesmos objetos no mundo real, mesmo que estejam descritos de formas distintas, possibilitando integração real dos dados em diversas aplicações. O problema da resolução de entidades tem sido amplamente estudado na literatura, principalmente pelo interesse comercial, tendo diversas abordagens já propostas [3–5].

Existem duas abordagens principais para tratar os problemas de correspondência [6]: não baseadas em aprendizagem de máquina, que realizam análises léxicas e contam com medidas de similaridades para compor a função de correspondência; as abordagens baseadas em aprendizagem, que fornecem diferentes tipos de medidas de similaridade, através de diversas características aprendidas pelos modelos de classificação, e que são responsáveis por identificar padrões e correspondências [7].

Problemas de resolução de entidades, geralmente, passam por duas etapas [8]: 1) Blocagem (Blocking) [9], que é a etapa responsável por minimizar o número de comparações necessárias; e 2) correspondência [6], que é a etapa que efetivamente decide se um determinado par de entidades representa a mesma entidade. Uma análise de diferentes modelos de classificação para a tarefa de pareamento (correspondência) de produtos foi realizada no trabalho de Jeremy Foxcroft et al. (2021) [10], comparando o desempenho de métodos de aprendizado de máquina tradicionais, como árvores de decisão, Random Forest e regressão logística, com o de métodos de estado da arte de aprendizagem profunda, como DeepMatcher, RoBERTa e DistilBERT. De forma similar, Kashif Shah et al. (2018) [11] analisaram modelos de rede neural rasa (Shallow Neural Network), como *fastText*, e redes siamesas, como certas variações do LSTM.

Ristoski et al. (2018) [12] usaram modelos de aprendizagem como florestas aleatórias para realizar o pareamento de produtos, em conjunto com CRF (Conditional Random Fields), além de CNN (Convolutional Neural Networks) para similaridade de imagens. O custo de obtenção de dados supervisionados e com pouco ruído pode ser alto. Logo, para minimizar o uso de dados supervisionados, os autores utilizaram de dados não estruturados, em formato Microdata, obtidos em sites de comércio eletrônico e, dessa forma, aumentaram o desempenho do modelo.

Para diminuir a quantidade de dados supervisionados, assim como o trabalho de Ristoski et al. (2018) [12], em Peeters et al. (2020) [13], foram utilizados dados de produtos disponíveis na web pública seguindo o formato schema.org para treinar os seus modelos de pareamento de produtos. Os modelos de aprendizagem profunda usados lidaram bem com os ruídos inerentes de dados obtidos na internet.

Barbosa (2019) [7] utiliza diferentes representações de textos (embeddings e bag of words) para a tarefa de resolução de entidade 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, onde um classificador binário é aplicado para resolver o problema de resolução de entidade.

A principal contribuição desta pesquisa em relação ao estado da arte diz respeito à proposição de heurísticas para agrupamento de produtos, bem como o uso de redes neurais profundas, através do transformer BERT, que resultou no melhor desempenho dentre todos os classificadores utilizados. Por fim, o melhor modelo de indução obtido foi utilizado em uma base de dados real de notas fiscais eletrônicas do Estado do Acre, e os resultados obtidos mostraram-se satisfatórios.

3 METODOLOGIA

Este trabalho baseou-se na metodologia CRISP-DM, que pode ser traduzido como Processo Padrão Inter-Indústrias para Mineração de Dados, sendo esta metodologia bastante utilizada durante o ciclo de vida de um projeto de ciência de dados [14]. A metodologia CRISP, contempla seis etapas, conforme pode ser constatado na figura 2, a saber:

- *entendimento do problema*, em que se busca assimilar melhor o problema, buscando referências e aprendendo mais sobre as regras de negócio;
- *compreensão dos dados*, que visa entender as especificidades e identificar problemas na qualidade dos dados;
- *preparação dos dados*, cujo objetivo é selecionar os atributos a serem usados, removendo características indesejadas;
- *modelagem*, etapa cujo objetivo é testar os modelos de classificação, identificando os melhores parâmetros a serem usados no corpus em questão;
- *avaliação*, em que se busca analisar as métricas dos modelos anteriormente treinados e testados; e
- *implantação*, em que se escolhe o melhor modelo a partir das métricas obtidas anteriormente e implanta-o no conjunto de dados reais, nunca vistos pelo classificador.

A seguir, serão detalhadas cada uma destas etapas metodológicas.

3.1 Entendimento do problema

O *product matching* é a tarefa de decidir se, dados dois produtos quaisquer, os mesmos correspondem à mesma entidade no mundo real, [13]. Trata-se de uma tarefa não trivial, pelo fato de, na maioria das vezes, os produtos não serem registrados com o seu identificador único, sendo mais utilizado no Brasil o EAN-13, também conhecido como código de barras, podendo também serem registrados de forma errônea, como será visto em um exemplo posterior. Além disso, não há uma padronização quando se trata das descrições

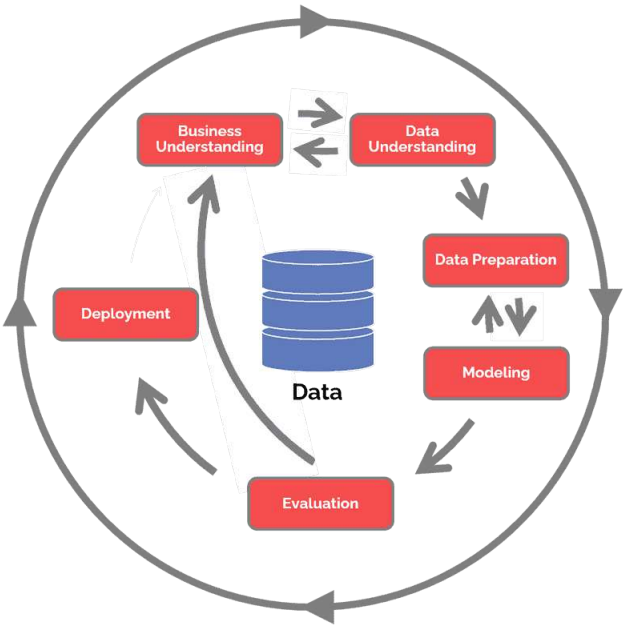

Figura 2: Etapas da metodologia CRISP-DM

Fonte: (DATA SCIENCE PROCESS ALLIANCE, c2022) [15]

dos produtos, fazendo com que essa tarefa de classificação torne-se ainda mais complexa.

Vários estudos tentam resolver esse problema usando métodos de aprendizagem de máquina, usando técnicas de processamento de linguagem natural e, em alguns casos, utilizando aprendizagem profunda.

No presente estudo, foi utilizado um banco de dados do Tribunal de Contas do Acre (TCE-AC) contendo produtos de notas fiscais emitidas no estado do Acre. Um problema comum em dados de notas fiscais está relacionado a erros de cadastros de produtos com os seus respectivos códigos de barras, dificultando as análises e agrupamentos que poderiam ser feitas a partir desses dados. Após o *deployment* do modelo de indução final, pretende-se conseguir classificar com precisão se o identificador único de determinado produto está correto ou não.

Nessa etapa, foi definido que seria utilizada a linguagem Python, tendo em vista o seu vasto conjunto de bibliotecas para ciência de dados, aprendizagem de máquina e aprendizagem profunda. Dessa forma, foi escolhido o ambiente Jupyter[1], utilizando a interface web, JupyterLab[2], bem como bibliotecas bastante utilizadas, como pandas[3], NumPy[4], scikit-learn[5] e TensorFlow[6].

[1]https://jupyter.org/
[2]https://github.com/jupyterlab/jupyterlab
[3]https://pandas.pydata.org/
[4]https://numpy.org/
[5]https://scikit-learn.org/
[6]https://www.tensorflow.org/

3.2 Compreensão dos dados

A tabela de produtos utilizada continha diversas informações sobre várias mercadorias de notas fiscais emitidas no estado do Acre, como a descrição do produto e o código de barras. Ainda, continha informações sobre quantidade, preço, CNPJ do fabricante, códigos tributários, como CEST, CFOP e NCM, entre outros. Após uma análise dos registros, tendo em vista o objetivo final da pesquisa, foram selecionados os seguintes atributos:

- *Descrição*: título do produto, que contém as características principais da entidade, como o modelo, a marca e especificidades dos produtos.
- *EAN*: código de barras, utilizado com identificador único do produto.
- *NCM*: classificação fiscal, utilizado para agrupar produtos semelhantes em classes.

Após a definição das informações a serem utilizadas, foi feita uma inspeção no conjunto de dados visando inspecionar qualidade dos mesmos. Foi detectado, por exemplo, que o código de barras de alguns produtos não seguiam o formato EAN, impossibilitando o uso dos mesmos no treinamento dos modelos. Também foram detectados, nessa etapa, outros problemas como NCMs inválidos, caracteres especiais, e descrições de produtos repetidas. Além disso, notou-se que algumas medidas como quilogramas, metros, litros, entre outras, eram abreviadas de formas diferentes, o que poderia dificultar a classificação desses produtos. Por exemplo, a unidade de medida litro se encontrava representada de diversas formas nos dados, como “L”, “LT”, “LTS”, “LITRO”, “LITROS”. Todos os problemas identificados foram tratados na etapa seguinte de preparação de dados.

3.3 Preparação dos dados

Na fase de preparação dos dados, foram utilizadas técnicas de pré-processamento de texto visando remover as características que poderiam prejudicar na etapa de treinamento dos modelos. Para isso, foram realizadas as seguintes etapas, como mostradas na figura 3:

(1) *Remoção de EANs inválidos*: O formato EAN possui regras específicas de construção, tornando possível validar se o código de barras é válido ou não. Utilizando a biblioteca Python, barcodenumber[7], foram removidas as entidades que não seguiam esse formato. Após a remoção, restaram cerca de 230 mil produtos.
(2) *Remoção de descrições repetidas*: a partir de uma análise, observaram-se vários produtos com a mesma descrição. Após a contagem dessas descrições, foram retiradas as descrições repetidas dos dados a serem utilizados pelos modelos, resultando em cerca de 90 mil produtos.
(3) *Remoção de NCMs inválidos*: produtos que contêm NCMs que são compostos unicamente por zeros são removidos.
(4) *Padronização das medidas*: utilizando-se de expressões regulares (regex), as unidades de medida mais recorrentes foram substituídas por uma representação padrão: o nome da unidade de medida escrita por extenso. Por exemplo,

[7]https://pypi.org/project/barcodenumber/

Figura 3: Etapas do pré-processamento dos dados.

as diferentes representações de metro, M, MT, MTS, METRO, foram substituídas pela unidade METROS, e assim por diante.

(5) *Remoção de caracteres especiais*: foram removidos caracteres especiais que apresentavam certa recorrência nas descrições e que não tinham importância significativa para a diferenciação de palavras, como "(", ")", "-", ";", entre outros.

No conjunto de dados, torna-se necessário encontrar uma descrição que represente um determinado código de barras, para que os modelos de classificação possam comparar essas duas descrições e definir se correspondem à mesma entidade. Desse modo, foi estabelecida uma descrição canônica para cada EAN, sendo essa uma representação textual da entidade referenciada pelo código de barras. A descrição canônica é o termo que mais aparece na descrição de um dado produto.

Dessa maneira, foi feita uma tabela auxiliar, que continha uma coluna com a quantidade de vezes que aquela mesma descrição apareceu associada àquele código das barras. Para cada código de barras, foi selecionada uma descrição, que correspondia à descrição canônica dos produtos com aquele EAN. Como último passo, foi feito uma junção (*join*), unindo todas as descrições com suas respectivas descrições canônicas, utilizando o identificador único.

Para tarefa de anotação dos dados, foram rotulados com $match = 1$, os dados que continham o mesmo código de barras, com sua determinada descrição canônica. Fazia-se necessário a presença produtos de descrições discrepantes, ou seja, que não correspondiam a mesma entidade, $match = 0$. Logo, esses registros foram gerados escolhendo aleatoriamente uma nova descrição canônica, pertencente a outro produto, e, dessa forma, garantindo que a descrição canônica escolhida não pertencia ao produto. Além disso, com o propósito de manter uma similaridade entre as descrições, assegurou-se que o NCM do produto e da descrição canônica fossem iguais. Na figura 4, observa-se um exemplo de como os produtos com descrições não correspondentes foram gerados. A partir do produto da figura 4, o qual têm-se descrições de dois produtos equivalentes, foi escolhida aleatoriamente outra descrição canônica, pertencente a um produto diferente, gerando assim um registro onde as descrições não se referem à mesma entidade no mundo real.

Ainda, para melhorar a qualidade dos dados a serem utilizados para treinar os modelos, foi formulada uma heurística que consistia em descartar registros que não seguissem a seguinte regra: dada a descrição e descrição canônica de determinado produto, só será

Descrição	PAPEL A4 ALCALINO BRANCO MED 210X297 MM
Descrição canônica	PAPEL A4 75 G EXECUTIVE OFFICE C 500FL 210x297MM
Match	1

Descrição	PAPEL A4 ALCALINO BRANCO MED 210X297 MM
Descrição canônica	PAPEL CASCA DE OVO TAM A4 MED 210x290mm
Match	0

Figura 4: Exemplo de geração de dados não equivalentes.

válido o registro no qual ambas as descrições tiverem pelo menos uma das duas primeiras palavras em comum. Um exemplo desta heurística é apresentado na figura 5, em que o primeiro registro se encaixa na heurística, pois as duas primeiras palavras das descrições são iguais, diferente do segundo registro, em que as duas primeiras palavras não são idênticas. Com a heurística, foram removidos cerca de 9.000 mercadorias.

Descrição	Descrição canônica
PILHA ALCALINA AA C 2UND LR6 RAYOVAC	PILHA ALCALINA RAYOVAC 2UN
TROFEU FUTEBOL 118 CM	BOLA VOLEIBOL PRO 6.0 ADULTO PENALTY

Figura 5: Exemplo do uso da heurística nas descrições de produtos

Alguns dos modelos utilizados necessitam de *features* para realizar o processamento e classificação dos dados. Para isso, foram gerados algumas métricas oriundas de processamento de linguagem natural, como distâncias entre palavras (Levenshtein, Damerau–Levenshtein, Hamming), além de métricas de similaridade entre palavras da biblioteca de Python, *fuzzwuzy*. Esse conjunto de métricas foi provido como entrada nesses modelos.

3.4 Modelagem

Os modelos de classificação escolhidos para serem analisados podem ser divididos em dois tipos: modelos de classificação tradicionais e modelos utilizando aprendizagem profunda. Os algoritmos tradicionais utilizados foram: *Random Forest*, XGBoost, *Naive Bayes* e SVM. O BERTimbau[8], um modelo BERT pré-treinado utilizando palavras da língua portuguesa [16], foi o modelo de aprendizagem profunda selecionado, pois é um bom representante do estado da arte.

[8]https://github.com/neuralmind-ai/portuguese-bert

Para testar todos os modelos, foi utilizada validação cruzada - *k-fold cross validation* - com 5 subdivisões, ou seja $k = 5$, utilizando a classe ShuffleSplit da biblioteca scikit-learn. A divisão treino e teste utilizada na validação cruzada foi de 70% dos registros para treino e 30% para teste.

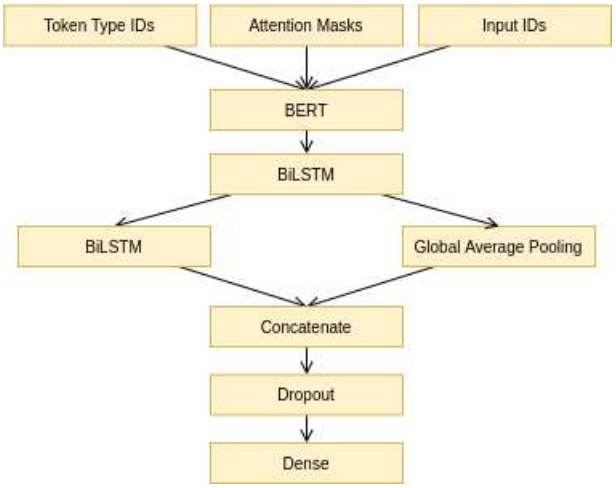

Figura 6: Camadas do modelo de aprendizagem profunda utilizado

Na figura 6, pode-se observar a estrutura do modelo de aprendizagem profunda utilizado. Foi utilizada a biblioteca Keras[9] para a maioria das camadas observadas. A camada do BERT, entretanto, foi importada da biblioteca transformers[10], especificamente o modelo BERTimbau, uma versão pré-treinada com palavras em português brasileiro. Os hiperparâmetros utilizados na camada do BERT foram os padrões providos pela classe *TFBertModel*, utilizando 4 épocas no seu treinamento, valor este dentro dos limites recomendados pela literatura. Nas outras camadas, foram utilizados também os valores padrões em seus parâmetros.

Para os modelos tradicionais, suas implementações foram importadas da biblioteca scikit-learn, exceto o modelo XGBoost, importado da biblioteca xgboost[11]. Os valores dos parâmetros utilizados nos modelos estão detalhados na tabela 1, os mesmos valores padrões definidos pelas implementações utilizadas.

4 AVALIAÇÃO DE RESULTADOS

As métricas utilizadas para a análise dos modelos, medindo, dessa forma, sua eficácia diante o problema, foram: acurácia, precisão, *recall* e F1-score. Essas métricas são baseadas nos dados obtidos da matriz de confusão: verdadeiros positivos (TP), falsos positivos (FP), verdadeiros negativos (TN) e falsos negativos (FN).

A tabela 2 apresenta a média dos resultados dos modelos avaliados. Em todas as métricas utilizadas na análise, o BERT + BiLSTM supera todos os modelos de classificação. Tudo indica que, por se tratar de um modelo com um pré-treinamento e com uma abordagem bem mais sofisticada ao lidar com as descrições passadas para o modelo, o BERT + BiLSTM conseguiu melhor classificar os dados. Na figura 7, é possível observar o gráfico das curvas ROC de todos os modelos, em que é possível observar novamente a superioridade do BERT + BiLSTM, que apresenta uma curva com maiores valores no eixo y, que representa a taxa de verdadeiros positivos.

[9]https://keras.io/
[10]https://huggingface.co/docs/transformers/
[11]https://xgboost.ai/

Random Forest		XGBoost	
Parâmetro	Valor	Parâmetro	Valor
bootstrap	true	max_depth	6
min_samples_leaf	1	learning_rate	0.3
max_features	auto	n_estimators	100
min_samples_split	2	colsample_bytree	1
n_estimators	100	subsample	1
SVM		Naive Bayes	
Parâmetro	Valor	Parâmetro	Valor
C	1.0	var_smothing	10^{-9}
gamma	scale	-	-
kernel	rbf	-	-

Tabela 1: Parâmetros utilizados nos modelos tradicionais

Modelo	Acurácia	Precisão	Recall	F1-score	AUC
BERT + BiLSTM	**92,04%**	**92,35%**	**91,59%**	**91,96%**	**92,05%**
Random Forest	89,58%	91,20%	87,47%	89,30%	89,60%
XGBoost	89,47%	90,89%	87,60%	89,21%	89,46%
SVM	88,93%	91,52%	85,67%	88,50%	88,92%
Naive Bayes	87,80%	89,98%	84,91%	87,37%	87,81%

Tabela 2: Métricas dos modelos treinados com dados com heurística

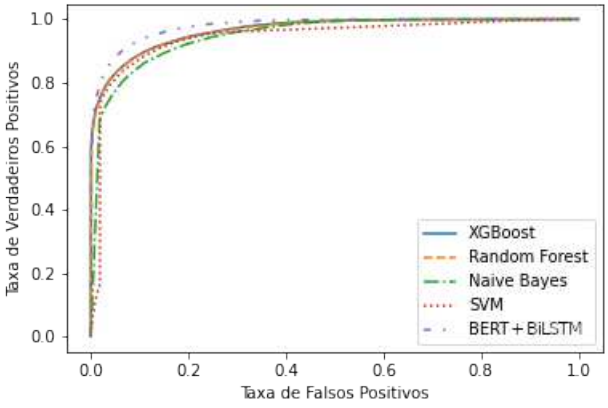

Figura 7: Curva ROC dos modelos treinados com dados com heurística

Como teste de ablação, os modelos foram treinados sem a heurística definida anteriormente e aplicada nos dados, com o intuito de verificar a sensibilidade dos modelos mediante os dados incorretos. Dessa forma, pode-se observar na tabela 4 as métricas dos modelos

Código de barras	Descrição do produto	Válido?
7895110428980	PAPEL A4 210X297 OFFICE EXECUTIVE JANDAIA	Sim
7895110428980	PAPEL A4 210X297 MILIMETROS 75 GRAMAS RS C/ 500FLS	Sim
7895110428980	PAPEL A4 OFFICE C/ 500 FLS	Sim
7895110428980	COPOS DESCARTAVEIS	Não
7895110428980	TONER SAMSUNG ML 1666/SCX3201/3206	Não

Tabela 3: Exemplos da classificação realizada pelo BERT nos dados do TCE-AC

Modelo	Acurácia	Precisão	Recall	F1-score	AUC
BERT + BiLSTM	**90,26%**	**92,16%**	**87,92%**	**89,96%**	**90,25%**
Random Forest	87,65%	90,16%	84,32%	87,14%	87,62%
XGBoost	87,54%	90,03%	84,22%	87,03%	87,51%
SVM	86,80%	90,49%	82,03%	86,05%	86,77%
Naive Bayes	85,51%	87,98%	82,01%	84,89%	85,49%

Tabela 4: Métricas dos modelos treinados com dados sem heurística

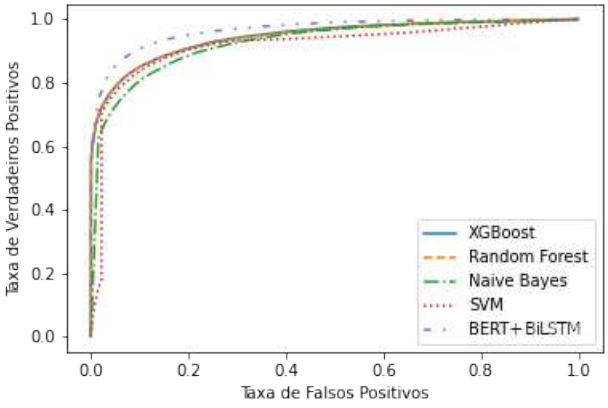

Figura 8: Curva ROC dos modelos treinados com dados sem heurística

sem a heurística. A curva ROC desses dados, mostrada na figura 8, mostra um padrão semelhante ao observado na curva dos dados com heurística.

O teste de ablação permitiu observar que o uso da heurística proposta resulta em melhores resultados. Os dados removidos pela heurística correspondem a cerca de 10% dos dados. Tratam-se de uma parcela razoável dos dados que, em sua grande maioria, eram compostos por produtos inconsistentes e, dessa forma, gerariam dados inválidos.

5 IMPLEMENTAÇÃO: ESTUDO DE CASO EM NOTAS FISCAIS ELETRÔNICAS DO ESTADO DO ACRE

Uma vez verificado que o modelo BERT + BiLSTM apresentou os melhores resultados dentre os modelos testados, o mesmo foi aplicado em todo o conjunto de dados dos produtos presentes em notas fiscais, relativas às vendas realizadas no Estado do Acre. Esses dados estão presentes no banco de dados do TCE-AC (Tribunal de Contas do Acre). O uso do modelo nos dados visa identificar produtos cuja descrição não corresponde ao seu código de barras, com a finalidade de remover esses dados inconsistentes.

Foram utilizados os mesmos procedimentos observados no preparo dos dados para identificar a descrição canônica do produto, sendo a descrição essa a que mais se repete dado um código de barras. Dadas ambas descrições, o modelo foi executado, resultando em um parâmetro que indicava se o registro é válido ou não. Foram classificados 235.060 produtos, dos quais 18.344 foram classificados como inválidos, isto é, a sua descrição não corresponde ao EAN associado.

Na tabela 3, pode-se observar um exemplo de classificação feita pelo modelo, que corresponde à coluna "Válido?". O modelo identificou produtos cuja descrição correspondia com o seu devido identificador, bem como produtos que não correspondia e que, portanto, podem ser excluídos dos dados.

Ainda, foi feita uma anotação manual, inserindo uma nova coluna que indicava se a descrição correspondia a descrição canônica. Os dados foram escolhidos aleatoriamente sendo divididos em 100 que foram anotados pelo modelo como válidos e 100 como inválidos. Dos 100 produtos cuja descrição foi anotada como válida, 74 foram corretamente classificadas, em 19 casos, os produtos diferiam, e em 7 não foi possível determinar. Dados cuja anotação foi equivalente à inválida, foram detectados 63 classificados corretamente, 25 classificados de forma incorreta, 12 casos os quais não foi possível identificar se os produtos correspondiam ou não.

6 CONCLUSÃO

O presente trabalho procurou encontrar o melhor modelo para a tarefa de correspondência de produtos, *product matching*, realizando uma análise entre diversos modelos, incluindo o modelo pré-treinado baseado em *transformers*, o BERT, em conjunto com o BiLSTM. Esse modelo foi eleito dentre os demais para ser aplicado em um cenário real, utilizando-o para remover dados inválidos no banco de notas fiscais do TCE-AC, e, por consequência, colaborar para análises futuras feitas com esses dados. Os resultados mostraram-se satisfatórios, pois o modelo escolhido conseguiu remover vários produtos inconsistentes.

No entanto, com teste de ablação, observou-se que o modelo se mostrou sensível a erros de anotação. Apesar dos processos realizados no intuito de remover dados inconsistentes durante a fase de preparação dos dados, os registros ainda apresentaram produtos anotados cujos códigos de barras e descrições não correspondiam ao mesmo produto.

Tendo em vista trabalhos futuros, pretende-se melhorar o pré-processamento dos dados, visando minimizar os itens inconsistentes. Por exemplo, atributos como o preço podem ser usados para auxiliar no processo de extração desses produtos. Ainda, podem ser utilizado métodos de otimização de hiperparâmetros, melhorando ainda mais as métricas do modelo.

AGRADECIMENTOS

Agradeço a Deus, por ter me mantido firme no meu propósito. Ao Professor Doutor Cláudio de Souza Baptista, meu orientador, pelas oportunidades e pela confiança durante todo esse tempo de convivência. Aos meus colegas de universidade, os quais agradeço por toda ajuda e apoio durante toda a graduação, em especial aqueles do Laboratório de Sistemas de Informação. Aos meus pais, por todo o amor, pois entregaram sempre tudo que tinham para me ver bem, como também toda a minha família, avôs e tios, por todo o suporte que sempre me ofereceram. Agradeço à minha esposa, por sempre estar ao meu lado e por ter me confiado a honra de ser o seu marido e, por fim, à minha filha, Maria Catarina, por me fazer o pai mais feliz do mundo.

REFERÊNCIAS

[1] CÉSAR, C. O crescimento dos marketplaces em 2021. 26 abr. 2021. Disponível em: <https://abcomm.org/noticias/o-crescimento-dos-marketplaces-em-2021/>. Acesso em: 13 de jan. de 2022. 1

[2] PAYNES, M. *5 Best Use Cases For Product Matching In Ecommerce How You Can Implement Each One.* 2021. Disponível em: <https://www.width.ai/post/product-matching-in-ecommerce>. Acesso em: 02 de fev. de 2022. 1

[3] CHRISTEN, P. Febrl -: An open source data cleaning, deduplication and record linkage system with a graphical user interface. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2008. (KDD '08), p. 1065–1068. ISBN 9781605581934. Disponível em: <https://doi.org/10.1145/1401890.1402020>. 2

[4] FIRMANI, D.; SAHA, B.; SRIVASTAVA, D. Online entity resolution using an oracle. *Proc. VLDB Endow.*, VLDB Endowment, v. 9, n. 5, p. 384–395, jan 2016. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/2876473.2876474>.

[5] BARLAUG, N.; GULLA, J. A. Neural networks for entity matching: A survey. *ACM Trans. Knowl. Discov. Data*, Association for Computing Machinery, New York, NY, USA, v. 15, n. 3, apr 2021. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/3442200>. 2

[6] KöPCKE, H.; THOR, A.; RAHM, E. Evaluation of entity resolution approaches on real-world match problems. VLDB Endowment, v. 3, n. 1–2, p. 484–493, sep 2010. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/1920841.1920904>. 2

[7] BARBOSA, L. Learning representations of web entities for entity resolution. In: . International Journal of Web Information Systems, 2019. v. 15 No. 3, p. 346–358. Disponível em: <https://doi.org/10.1108/IJWIS-07-2018-0059>. 2

[8] CHRISTOPHIDES, V. et al. Entity resolution in the web of data. In: *Proceedings of the 23rd International Conference on World Wide Web*. New York, NY, USA: Association for Computing Machinery, 2014. (WWW '14 Companion), p. 203–204. ISBN 9781450327459. Disponível em: <https://doi.org/10.1145/2567948.2577263>. 2

[9] EFTHYMIOU, V. et al. Matching web tables with knowledge base entities: From entity lookups to entity embeddings. In: D'AMATO, C. et al. (Ed.). *The Semantic Web – ISWC 2017*. Cham: Springer International Publishing, 2017. p. 260–277. ISBN 978-3-319-68288-4. 2

[10] FOXCROFT, J. et al. Product matching lessons and recommendations from a real world application. *Proceedings of the Canadian Conference on Artificial Intelligence*, Canadian Artificial Intelligence Association (CAIAC), 6 2021. Https://caiac.pubpub.org/pub/klikfzaf. Disponível em: <https://caiac.pubpub.org/pub/klikfzaf>. 2

[11] SHAH, K.; KOPRU, S.; RUVINI, J.-D. Neural network based extreme classification and similarity models for product matching. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. New Orleans - Louisiana: Association for Computational Linguistics, 2018. p. 8–15. Disponível em: <https://aclanthology.org/N18-3002>. 2

[12] RISTOSKI, P. et al. A machine learning approach for product matching and categorization. *Semantic Web*, v. 9, p. 707–728, 2018. 2

[13] PEETERS, R. et al. Using schema.org annotations for training and maintaining product matchers. In: . New York, NY, USA: Association for Computing Machinery, 2020. (WIMS 2020), p. 195–204. ISBN 9781450375429. Disponível em: <https://doi.org/10.1145/3405962.3405964>. 2

[14] CHAPMAN, P. et al. *CRISP-DM 1.0 Step-by-step data mining guide*. [S.l.], 2000. Disponível em: <https://maestria-datamining-2010.googlecode.com/svn-history/r282/trunk/dmct-teorica/tp1/CRISPWP-0800.pdf>. 2

[15] WHAT is CRISP DM? *Data Science Process Alliance*, c2022. Disponível em: <https://www.datascience-pm.com/crisp-dm-2/>. Acesso em: 01 de mar. de 2022. 3

[16] 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. 4