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

RICH ELTON CARVALHO RAMALHO

UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA E NLP PARA EXTRAÇÃO DE INFORMAÇÕES EM LICITAÇÕES DO DIÁRIO OFICIAL DO ESTADO DO ACRE

CAMPINA GRANDE - PB

2022

RICH ELTON CARVALHO RAMALHO

UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA E NLP PARA EXTRAÇÃO DE INFORMAÇÕES EM LICITAÇÕES DO DIÁRIO OFICIAL DO ESTADO DO ACRE

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.

Orientadores: Professor Dr. Cláudio de Souza Baptista e Professor Dr. Hugo Feitosa de Figueirêdo.

CAMPINA GRANDE - PB

2022

RICH ELTON CARVALHO RAMALHO

UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA E NLP PARA EXTRAÇÃO DE INFORMAÇÕES EM LICITAÇÕES DO DIÁRIO OFICIAL DO ESTADO DO ACRE

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

Professor Dr. Hugo Feitosa de Figueirêdo

Orientador – IFPB

Professor Dr. Maxwell Guimarães De Oliveira

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 02 de Setembro de 2022.

CAMPINA GRANDE - PB

ABSTRACT

Information Extraction Systems assist humans in searching for specific information in documents. However, most of these systems do not support documents in the Portable Document Format (PDF), which is widely used. In a PDF document, the text content is mixed with metadata or semi-structured data, which makes it difficult for Natural Language Processing (NLP) algorithms to extract the required information. The Court of Auditors of the State of Acre (TCE-AC) is the supervisory and controlling body of the use of public money and the budget and financial administration of the state of Acre, responsible for analyzing and judging the public accounts of the jurisdictions. Jurisdictions must publish information related to bids both in the TCE-AC bid management system and in the Official Gazette of the State of Acre (DOE), which uses the PDF format. It is the responsibility of the TCE-AC to verify that the bidding information is in both places, thus generating a lot of manual work. In this work, we present a PLN solution with the objective of extracting the DOE acts, automatically categorizing the acts as bidding or not, if so, advanced PLN techniques will be used to process and extract the entities and information from the bidding so that it is possible assist the TCE-AC to verify that the bid is also in the bid management system.

Utilizando Técnicas de Aprendizagem de Máquina e NLP para Extração de Informações em Licitações do Diário Oficial do Estado do Acre

Rich Elton Carvalho Ramalho
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil
rich.ramalho@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

Hugo Feitosa de Figueirêdo
Laboratório de Sistemas de Informação
Instituto Federal da Paraíba
Esperança, Paraíba, Brasil
hugoff@gmail.com

RESUMO

Sistemas de Extração de Informação auxiliam humanos na busca de informação específica em documentos. No entanto, a maioria destes sistemas não dão suporte a documentos no formato Portable Document Format (PDF), que é largamente utilizado. Em um documento PDF, o conteúdo do texto é misturado com metadados ou dados semi-estruturados, que dificultam os algoritmos de Processamento de Linguagem Natural (PLN) na extração da informação requerida. O Tribunal de Contas do Estado do Acre (TCE-AC) é o órgão fiscalizador e controlador do uso do dinheiro público e da administração orçamentária e financeira do estado do Acre, responsável por analisar e julgar as contas públicas dos jurisdicionados. Os jurisdicionados devem publicar informações relacionadas às licitações tanto no sistema de gerenciamento de licitações do TCE-AC como também no Diário Oficial do Estado do Acre (DOE), que usa o formato PDF. É de responsabilidade do TCE-AC verificar se as informações da licitação estão nos dois lugares, gerando assim, um grande trabalho manual. Neste trabalho, apresentamos uma solução de PLN com objetivo de extrair os atos do DOE, categorizar automaticamente os atos como licitação ou não, em caso afirmativo, serão utilizadas técnicas avançadas de PLN para processar e extrair as entidades e informações da licitação para que seja possível auxiliar o TCE-AC a verificar se a licitação encontra-se também no sistema de gerenciamento de licitações.

Palavras-chave

TCE, Aprendizado de Máquina, Processamento de Linguagem Natural, Licitação, Bert, Jurisdicionado, NER

1 INTRODUÇÃO

Os Tribunais de Contas dos Estados (TCE) exercem uma função vital no território brasileiro: fiscalizar as despesas e receitas dos estados e municípios. Apesar disso, sua atuação ainda pode parecer obscura para quem não é familiarizado com o órgão, ou para aqueles que não possuem tanto conhecimento nas áreas de direito ou administração. São órgãos públicos. Apesar de parecer estranho que um órgão do estado fiscalize o próprio estado, os TCEs são autônomos, ou seja, possuem independência financeira e administrativa. Ao contrário do que sugere o "Tribunal" no nome, o TCE não é um tribunal, e não pertence ao poder Judiciário, mas atua como um auxiliar do poder Legislativo no controle externo da Administração Pública. Cada estado brasileiro possui o seu próprio TCE, que trabalha de forma descentralizada, através de inspetorias regionais, exercendo o trabalho de fiscalização em menor escala [1].

O Tribunal de Contas do Estado do Acre (TCE-AC) é o órgão fiscalizador e controlador do uso do dinheiro público e da administração orçamentária e financeira do estado do Acre, responsável por analisar e julgar as contas públicas dos jurisdicionados. Os jurisdicionados devem publicar informações relacionadas às licitações tanto no sistema de gerenciamento de licitações do TCE-AC como também no Diário Oficial do Estado do Acre, que só permite a publicação de documentos no formato PDF.

Uma das aplicações do Processamento de Linguagem Natural (PLN) é a extração de informações a partir de textos. Desde a difusão da Internet a partir dos anos 1990, e o aumento considerável de textos disponíveis nesse meio, os esforços de PLN passaram a se concentrar nas tarefas de extração, com o objetivo de estruturar a informação disponível nos textos, e assim facilitar o acesso a essas fontes [2]. Entretanto, a maioria dessas técnicas não dão suporte para documentos no formato PDF (Portable Document Format). Nesses documentos, o conteúdo do texto é misturado com metadados ou dados semi-estruturados, dificultando a extração das informações.

Atualmente existe uma enorme geração de dados (não-estruturados, semi-estruturados e estruturados) e isso está fazendo estudos em PLN e Aprendizado de Máquina convergirem cada vez mais. Esses estudos têm contribuído para a evolução de sistemas automáticos e soluções inteligentes na análise de textos e são utilizados em diversos projetos.

2 PROBLEMA E SOLUÇÃO

Os jurisdicionados são gestores de recursos arrecadados junto ao cidadão, os jurisdicionados - no âmbito municipal e estadual - são obrigados, por força de lei, a prestar contas da finalidade que dão a este dinheiro. O TCE-AC disponibiliza a estes administradores

serviços e sistemas eletrônicos que tornam mais ágil a prestação de contas. Como isso, acelera-se o trâmite processual, análise, instrução e julgamento. Em última instância, isso significa levar ao cidadão, de forma cada vez mais célere, informação qualificada para que ele tenha conhecimento sobre a gestão dos recursos que recolhe ao órgão ou ente público por meio dos impostos [3].

O TCE-AC possui um sistema de gerenciamento de licitações, onde esses jurisdicionados devem publicar as informações relacionadas à essas licitações. No entanto, visando atender o princípio de publicidade, não é somente nesse sistema que devem ser inseridas essas informações, pois também deve ser feita a publicação no Diário Oficial do Estado do Acre, tornando as informações de licitações públicas para toda a sociedade. É responsabilidade do TCE-AC verificar se o jurisdicionado realizou a publicação corretamente nos dois meios de publicações, gerando assim, um enorme trabalho manual.

Portanto, o objetivo desse trabalho é prover uma solução para auxiliar o TCE-AC no problema de verificação manual, na qual verifica se o jurisdicionado publicou a licitação no sistema de gerenciamento de licitações do TCE-AC e também no Diário Oficial do Estado do Acre. Como objetivos específicos tem-se a desenvolver um extrator de texto dos documentos PDF, com suporte às atividades de pré-processamento; induzir um modelo de aprendizagem para classificar se um texto refere-se a um edital de licitação ou não; e, por fim, extrair os elementos da publicação da licitação.

3 METODOLOGIA

Este projeto utilizou uma metodologia baseada no CRISP-DM (Cross Industry Standard Process for Data Mining), sendo esta metodologia bastante utilizada durante o ciclo de vida de um projeto de ciência de dados [4]. A metodologia CRISP-DM, contempla seis etapas, conforme pode ser constatado na figura 1, a saber: Compreensão do Negócio, Compreensão dos Dados, Preparação dos Dados, Modelagem, Avaliação e Implantação [5].

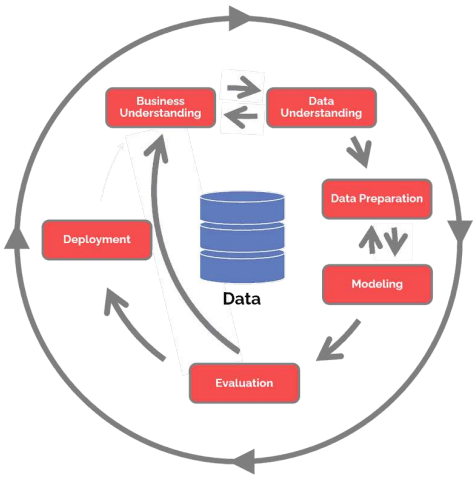

Figura 1: Etapas da metodologia CRISP-DM.

Fonte: (DATA SCIENCE PROCESS ALLIANCE, 2022) [5]

A etapa de Entendimento do Negócio (Business Understanding) foi contemplada na seção 2. A etapa de Compreensão dos Dados (Data Understanding), é onde foram coletadas as publicações do Diário Oficial do Estado do Acre e extraídos os textos dessas publicações separando por atos e entes, para análise e compreensão automática das informações a serem utilizadas na pesquisa. A etapa de Preparação dos Dados (Data Preparation) consistiu na implementação de técnicas de pré-processamento nos dados extraídos na etapa anterior, para então iniciar a rotulagem manual dos dados, gerando assim um corpus anotado. A etapa de Modelagem (Modeling) consistiu na escolha e implementação de vários modelos de aprendizado de máquina, utilizando-se de classificação supervisionada, com os devidos ajustes de hiperparâmetros. Também foi treinado um modelo para reconhecimento de entidades nomeadas nas licitações. A etapa Avaliação dos Modelos (Evaluation) foi feita a análise dos resultados dos modelos que foram gerados na etapa anterior, escolhendo o que apresentou o melhor desempenho. Por fim, a etapa Implementação da Ferramenta (Deployment), consiste na implementação dos modelos para conseguir extrair as licitações e suas informações de um Diário Oficial do Estado do Acre.

3.1 Compreensão dos Dados

3.1.1 Download dos Diários Oficiais do Estado do Acre.

Para a pesquisa foi feito o download de alguns diários oficiais do estado do Acre. Escolhemos os diários dos meses de agosto, setembro, outubro e novembro do ano de 2021. Foi construído um crawler, escrito em Python [6] com uso das bibliotecas BeautifulSoup[1] e requests[2]. Esse crawler recebe o mês e ano, acessa a página do Diário Oficial do Estado do Acre [7], como pode ser vista na figura 2, realiza o filtro pelos parâmetros e por fim realiza download de todas as publicações daquele mês e ano, salvando os PDFs, sendo o nome a data de publicação do DOE (e.g., "dou_ac_02_08_21", um diário publicado no dia 2 de agosto de 2021). No final foram baixados um total de 83 diários do estado do Acre, sendo 22 diários do mês de agosto, 22 diários de setembro, 19 diários de outubro e 20 diários de novembro, todos do ano de 2021.

3.1.2 Extração dos Dados para o Formato JSON.

Após baixar todos os diários oficiais dos meses de agosto, setembro, outubro e novembro de 2021, a etapa seguinte seria estruturar as informações dos PDFs para que fosse possível trabalhar com os dados na pesquisa. A figura 3 é um exemplo de um Diário Oficial do Estado do Acre do dia 2 de agosto de 2021.

Foi construído um extrator[3], escrito em Python e com auxílio da biblioteca PyMuPDF[4] para leitura dos arquivos no formato PDF. Na figura 4 podemos ver o funcionamento do extrator, em que recebe um ou mais PDFs do DOE, um leitor vai ler esses PDFs e enviar para o extrator, que por sua vez vai extrair o sumário do arquivo, depois vai capturar os textos de todas as páginas.

Após a extração do texto é realizado um pré-processamento no texto para separar os atos por títulos e subtítulos, obtidos na primeira fase do algoritmo, por fim será retornado um arquivo JSON com todos os atos daquele diário. O formato JSON foi o escolhido

[1] https://www.crummy.com/software/BeautifulSoup/bs4/doc/index.html
[2] https://github.com/psf/requests
[3] https://github.com/richecr/tcc
[4] https://github.com/pymupdf/PyMuPDF

Figura 2: Página do Diário Oficial do Estado do Acre.

Fonte: (DIÁRIO OFICIAL DO ESTADO DO ACRE, 2022) [7]

Figura 3: Diário Oficial do Estado do Acre do mês de agosto de 2021.

Fonte: (DIÁRIO OFICIAL DO ESTADO DO ACRE, 2022) [7]

para estruturar esses dados, pois é um formato leve de troca de informações entre sistemas, muito simples de ler, com uma maior velocidade na execução e transporte de dados quando comparado com o formato XML, por exemplo, também são arquivos com tamanhos reduzidos e grandes empresas utilizam esse formato (e.g., Google, Facebook, Twitter, entre outras) [8]. Na figura 5 pode-se observar um exemplo de uma saída do algoritmo. Os atos separados por entes.

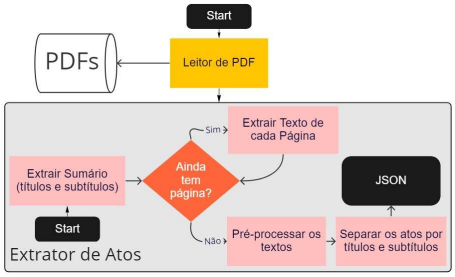

Figura 4: Diagrama do extrator do texto do PDF do Diário Oficial do Estado do Acre.

Fonte: Autoria Própria

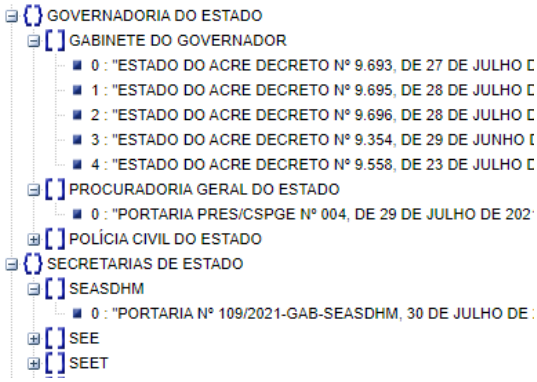

Figura 5: Exemplo de Saída do Extrator de Atos.

Fonte: Autoria Própria

3.2 Preparação dos Dados

3.2.1 Rotulagem dos dados.

Com os atos extraídos e já estruturados em um formato de fácil leitura e manipulação. Então, começamos a parte de rotulagem desses atos. Mas isso era um problema, pois era preciso saber identificar quando era uma licitação e em caso afirmativo tivemos que rotular algumas informações, são elas: identificação da licitação, objeto, ente e órgão, esses últimos dois apenas quando estavam presentes na licitação. O ente conseguimos obter pelo módulo do extrator, mas o órgão nem sempre está presente na licitação e só por meio de análise mais profunda do texto é capaz de saber qual é, mas essa informação não é algo importante para identificar se a licitação está publicada no sistema de gerenciamento do TCE-AC, licon [9].

Para auxiliar nessa anotação dos dados foi usado uma planilha de controle criada pelo TCE-AC (figura 6), na qual um colaborador é responsável por ler os diários oficiais e procurar pelas licitações que aparecem pela primeira vez no diário e então esse colaborador adiciona essas informações (identificação, ente, órgão, objeto, mês, número do diário e página) nessa planilha de controle. Essa planilha facilitou o trabalho para rotular os dados, pois era apenas filtrar na planilha pelos meses dos dados que extraímos e assim íamos

olhando a planilha e localizando aquela licitação em nossos dados e então era feito a anotação das informações.

Ente	Órgão	Identificação	Objeto	Data da Publicação	Nº do Diário	Nº da Pá
PREFEITURA MUNICIPAL DE XAPURI	PREFEITURA MUNICIPAL DE XAPURI	PREGÃO PRESENCIAL SRP Nº 053/2021	Contratação de Empresa fornecedora de Madeira de primeira qualidade e Madeira branca.	29/11/2021	13.173	205
PREFEITURA MUNICIPAL DE XAPURI	PREFEITURA MUNICIPAL DE XAPURI	PREGÃO PRESENCIAL SRP Nº 054/2021	contratação da empresa fornecedora de Cesta Básica (Gênero Alimentícios não Perecíveis).	29/11/2021	13.173	205
PREFEITURA MUNICIPAL DE XAPURI	PREFEITURA MUNICIPAL DE XAPURI	PREGÃO PRESENCIAL SRP Nº 055/2021	contratação da empresa fornecedora de Equipamentos e Material Permanente.	29/11/2021	13.173	205
SECRETARIA DE ESTADO DE PLANEJAMENTO E GESTÃO - SEPLAG	DEPARTAMENTO DE ESTRADAS DE RODAGEM, INFRAESTRUTURA HIDROVIÁRIA E AEROPORTUÁRIA DO ACRE – DERACRE	CONCORRÊNCIA Nº 031/2021 - DERACRE	Contratação de Pessoa Jurídica para Pavimentação de Vias Urbanas em Senador Guiomard.	30/11/2021	13.174	22
SECRETARIA DE ESTADO DE PLANEJAMENTO E GESTÃO - SEPLAG	SECRETARIA DE ESTADO DE SAÚDE - SESACRE	PREGÃO ELETRÔNICO SRP Nº 374/2021 – SESACRE	Aquisição de material médico-hospitalar Curativos Especiais, para atender as demandas da SESACRE.	30/11/2021	13.174	23
SECRETARIA DE ESTADO DE PLANEJAMENTO E GESTÃO - SEPLAG	DEPARTAMENTO DE ESTRADAS DE RODAGEM, INFRAESTRUTURA HIDROVIÁRIA E AEROPORTUÁRIA DO ACRE – DERACRE	PREGÃO PRESENCIAL SRP Nº 119/2021 – DERACRE	Contratação de empresa para fornecimento de Hospedagem e café da manhã nos municípios de Xapuri, Assis Brasil, Epitaciolândia e/ou Brasiléia.	30/11/2021	13.174	23

Figura 6: Planilha de Controle utilizada para anotação.

Fonte: Autoria Própria

Usamos a ferramenta Doccano [10] para facilitar na rotulagem desses dados. O doccano é uma ferramenta de anotação de texto de código aberto para humanos. Ele fornece recursos de anotação para classificação de texto, rotulagem de sequência e tarefas de sequência a sequência. Só precisamos fazer upload dos nossos dados no formato aceito pela ferramenta (JSON) e ele gera toda uma interface para anotação dos dados de forma mais intuitiva, como pode ser observada na figura 7.

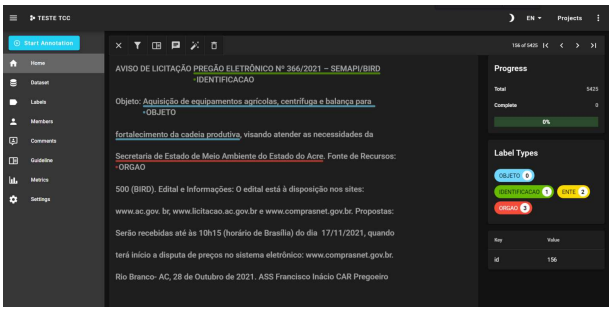

Figura 7: Exemplo de uma licitação no doccano.

Fonte: Autoria Própria

Para essa anotação com auxílio da planilha, foi criado um código em Python para que com o número da identificação da licitação retorne a posição que esse ato se encontra nos dados do doccano. Com isso, pela interface web conseguimos buscar pela posição e assim realizar a anotação das informações (identificação, ente, órgão e objeto) no ato da licitação. Com isso facilitando um trabalho que poderia ter sido bem complicado e com possíveis erros de anotações, já que fizemos o uso da planilha, o risco de cometer erros é mínimo, já que é isso que o TCE-AC utiliza para verificar se a licitação já está no licon.

Por fim, teremos um arquivo JSON contendo todos os atos de licitações com suas informações anotadas, como pode ser visto na figura 8, na qual o ato com id igual a 3483 é uma licitação e o com id igual 3484 não é. Após isso, fizemos um pré processamento desses dados, verificamos os atos que possuem anotações e criamos um novo arquivo JSON com o ato e com a informação se é ou não uma licitação, como pode ser visto na figura 9, na qual os atos com ids iguais a 3482 e 3483 são licitações e os com id iguais a 3481, 3484 e 3485 não são licitações. Casos em que o ato tem alguma anotação quer dizer que é uma licitação. Esses dois arquivos foram usados para treinamento do modelo de Reconhecimento de Entidades Nomeadas (NER) e para os modelos de classificação de textos, respectivamente.

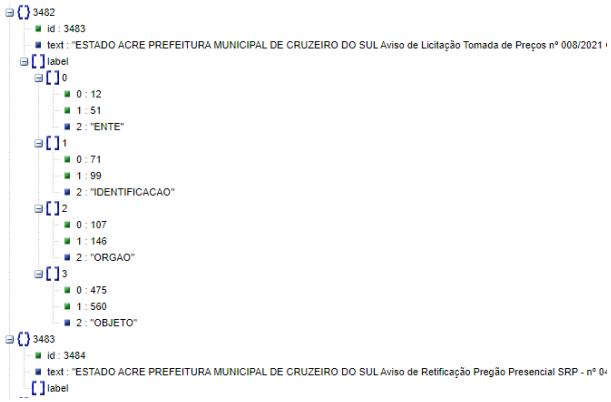

Figura 8: Exemplo de saída do doccano após anotação dos dados.

Fonte: Autoria Própria

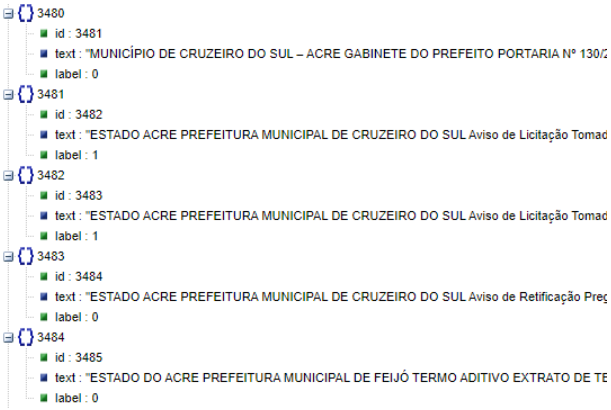

Figura 9: Exemplo de saída da anotação dos dados.

Fonte: Autoria Própria

Ao final do processo de rotulagem dos dados, foram obtidos uma base de dados com um total de 6.499 atos, porém o conjunto de dados ficou bastante desbalanceado, sendo 5639 não licitações e 860 licitações, conforme pode ser visto na figura 10.

3.2.2 Pré-processamento dos dados.

Os dados coletados foram submetidos a algumas técnicas de pré-processamento, que são descritas nesta subseção. A primeira técnica diz respeito à capitalização, isto é, a normalização do texto em uma só tipografia, caixa alta ou baixa.

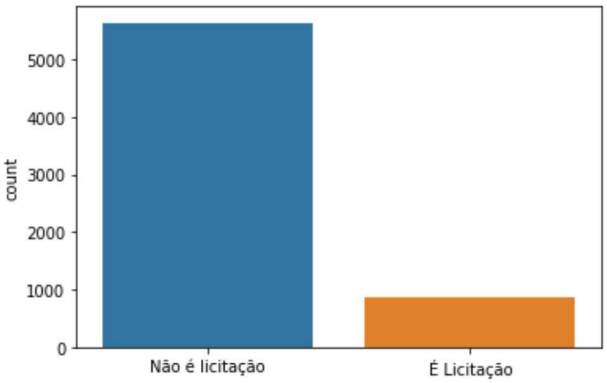

Figura 10: Distribuição dos atos rotulados.

Fonte: Autoria Própria

Outra técnica de pré-processamento utilizada diz respeito ao balanceamento do corpus para evitar possível enviesamento nos modelos de classificação a serem utilizados. Utilizou-se da técnica estatística de subamostragem, com o auxílio do algoritmo RandomUnderSampler da biblioteca imblearn para remover instâncias da classe negativa de forma aleatória [11]. Dessa forma, o conjunto de treino balanceado resultou em 602 instâncias para cada classe. Por fim, foram aplicadas as técnicas de pré-processamento: remoção de stopwords e tokenização, fazendo uso da biblioteca nltk [12] e do algoritmo TfidfVectorizer da biblioteca sklearn [13], respectivamente.

3.3 Modelagem

3.3.1 Criação dos modelos para classificação e ajuste de hiperparâmetros.

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 escolhidos foram: XGBoost, LogisticRegression, DecisionTreeClassifier e RandomForest. O BERTimbau, um modelo BERT pré-treinado utilizando palavras da língua portuguesa [14], foi o modelo de aprendizagem profunda selecionado. Também foi usado um modelo da biblioteca Spacy [15] para treinar um modelo para reconhecimento de entidade nomeada, NER [16]. Esse modelo NER treinado foi usado para conseguir extrair as informações (identificação, ente, órgão e objeto) de uma licitação do diário.

O conjunto de dados foi dividido em treino e teste, usando o algoritmo train_test_split da biblioteca sklearn. Dessa forma, 30% dos dados foram destinados ao conjunto de testes, enquanto que dos 70% destinados para treinamento, sendo que 10% deste conjunto de treinamento foram utilizados durante o treinamento como conjunto de validação. Para reduzir as chances dos modelos sofrerem de overfitting, foi utilizado o método de validação cruzada k-folds com o parâmetro k = 10 através do algoritmo StratifiedKFold disponibilizado pelo sklearn.

Para um melhor ajuste dos hiperparâmetros dos modelos criados durante o processo de treinamento, foi utilizado o algoritmo GridSearchCV[5] do sklearn. As métricas utilizadas para avaliação dos modelos foram: acurácia, f1-score, precisão e recall. Conforme as tabelas 1, 2, 3 e 4 foram testadas diferentes combinações de hiperparâmetros a fim de encontrar o modelo que melhor maximizou as métricas de acurácia e f1-score [17]. Todo o código utilizado para o treinamento dos modelos podem ser consultados através do google colaboratory[6].

XGBoost	
Hiperparâmetro	**Valores (o melhor valor em negrito)**
min_child_weight	**1** e 5
colsample_bytree	**0.6** e 1.0
subsample	**0.6** e 0.8
max_depth	**3** e 4

Tabela 1: Hiperparâmetros para o modelo XGBoost.

LogisticRegression	
Hiperparâmetro	**Valores (o melhor valor em negrito)**
C	0.0001, 0.001, 0.01,0.1, 1, 10 e **100**
penalty	l1 e **l2**
solver	sag, saga e **lbfgs**

Tabela 2: Hiperparâmetros para o modelo LogisticRegression.

DecisionTreeClassifier	
Hiperparâmetro	**Valores (o melhor valor em negrito)**
criterion	**gini** e entropy
max_depth	4, **5**, 6,7,8,9,10,11,12,15,20,30,40,50,70,90,120 e 150

Tabela 3: Hiperparâmetros para o modelo DecisionTreeClassifier.

RandomForest	
Hiperparâmetro	**Valores (o melhor valor em negrito)**
n_estimators	200 e **500**
max_features	**auto** e log2
max_depth	**4** e 8
criterion	**gini** e entropy

Tabela 4: Hiperparâmetros para o modelo RandomForest.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn-model-selection-gridsearchcv
[6]https://colab.research.google.com/drive/12iWo8cQOlC2Gy-1XppCmbxBopM-JKr0D?usp=sharing

3.3.2 Modelo BERTimbau.

O BERT é uma técnica de aprendizado de máquina baseada em transformador para pré-treinamento de processamento de linguagem natural (NLP) desenvolvida pelo Google. No ano de 2020, a NeuralMind, startup focada em soluções de análise de texto e imagens usando inteligência artificial, decidiu lançar o BERT treinado para o idioma português [18] e disponibilizou de forma gratuita em seu GitHub[7].

A biblioteca Hugging Face disponibiliza uma API para podemos usar esse modelo BERTimbau de uma forma bem simples de usar, por isso escolhemos ela para realizar o treinamento do nosso modelo. Para treinamento, foram utilizados tensores pytorch [19] durante 4 épocas utilizando o mesmo conjunto de treinamento dos modelos do sklearn. O código utilizado pode ser acessado através do google colaboratory[8].

3.4 Avaliação do Modelos

Nesta subseção são apresentados os resultados dos modelos de classificação utilizados e também é feita uma análise para a escolha do melhor modelo para o problema de classificação proposto. Os modelos treinados e testados foram divididos em duas subseções. Primeiramente, são abordados os modelos extraídos a partir do Sklearn e XGBoost, modelos de classificação tradicionais. Por fim, o modelo BERTimbau, baseado em rede neural profunda, será avaliado.

3.4.1 Modelos do Sklearn e XGBoost.

Após encontrar os melhores hiperparâmetros para cada um dos modelos escolhidos, esses modelos foram treinados e avaliados com o conjunto de testes, calculando as seguintes métricas: acurácia, f1-score, recall e precisão. Os resultados podem ser observados nas tabelas 5, 6, 7 e 8.

XGBoost	
Métrica	**Valor**
Acurácia	0.9851
F1-Score	0.9689
Recall	0.9897
Precisão	0.9505

Tabela 5: Métricas do modelo XGBoost.

LogisticRegression	
Métrica	**Valor**
Acurácia	0.9800
F1-Score	0.9584
Recall	0.9802
Precisão	0.9392

Tabela 6: Métricas do modelo LogisticRegression.

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

[8]https://colab.research.google.com/drive/12iWo8cQOlC2Gy-1XppCmbxBopM-JKr0D?usp=sharing

DecisionTreeClassifier	
Métrica	**Valor**
Acurácia	0.9769
F1-Score	0.9528
Recall	0.9834
Precisão	0.9274

Tabela 7: Métricas do modelo DecisionTreeClassifier.

RandomForest	
Métrica	**Valor**
Acurácia	0.9589
F1-Score	0.9198
Recall	0.9697
Precisão	0.8836

Tabela 8: Métricas do modelo RandomForest

Com base nas métricas dos modelos obtidas, podemos ver que todos eles tiveram bons resultados para o nosso problema de classificação de licitações. O XGBoost foi o modelo que obteve as melhores métricas, mas vale ressaltar que ele tem um processo de treinamento lento comparado aos demais modelos testados.

3.4.2 BERTimbau.

Mesmo tendo obtido bons resultados com os modelos do sklearn e o XGBoost, quando treinado o modelo pré-treinado do BERTimbau conseguimos obter resultados ainda melhores. Todas as métricas foram superadas, conforme a tabela 9 e a matriz de confusão da figura 11, assim sendo o melhor modelo para a construção do sistema de classificação de licitações do DOE.

Para verificação de overfitting no modelo, foi construído o gráfico da figura 12, representando o comportamento do erro durante o treinamento com os dados de treino e validação. Podemos concluir que o modelo não sofre de overfitting, pois o erro com os dados de validação vai diminuindo à medida que o de treinamento também diminui, ou seja, o modelo está generalizando bem.

BERTimbau	
Métrica	**Valor**
Acurácia	0.9867
F1-Score	0.9867
Recall	0.9867
Precisão	0.9869

Tabela 9: Métricas do modelo BERTimbau.

3.5 Reconhecimento de Entidades Nomeadas nos Atos de Licitações

Após a construção e escolha do melhor modelo para classificação de um ato do DOE em licitação ou não, podemos partir para a segunda etapa, a construção de um modelo para reconhecer as partes importantes de uma licitação, são elas: “IDENTIFICACAO”,

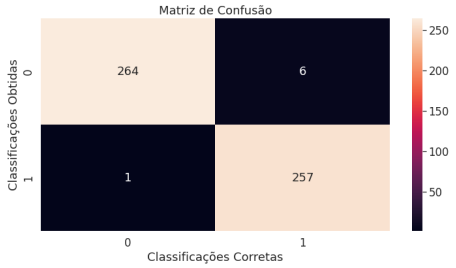

Figura 11: Matriz de Confusão.

Fonte: Autoria Própria

Figura 12: Gráfico do erro.

Fonte: Autoria Própria

"ENTE", "ORGAO" e "OBJETO". O código usado para treinamento desse modelo pode ser encontrado no google colab[9].

Para isso usamos uma técnica conhecida como Reconhecimento de Entidade Nomeada (NER), é uma técnica de NLP que consiste em identificar e categorizar informações-chave (entidades) em textos [20]. O Spacy é uma biblioteca para processamento avançado de linguagem natural em Python e Cython. Ele foi desenvolvido com base nas pesquisas mais recentes e foi projetado desde o primeiro dia para ser usado em produtos reais [15].

Antes do treinamento do modelo era preciso realizar alguns passos de pré-processamento nos dados, pois a interface de entrada de treinamento do modelo NER do Spacy era diferente de como nossos dados estavam estruturados. Depois dessa transformação, foi necessário uma outra etapa, pois o modelo NER do Spacy não aceita sobreposição de entidades nas mesmas posições dos textos. Por exemplo, o texto "... TOMADA DE PREÇOS Nº. 004/2021 – CPL/PMBJ..." é a identificação da licitação, mas a sigla "PMBJ" também é a ente e isso quebra o treinamento do modelo. A solução encontrada foi duplicar esses textos com suas anotações. Os atos que tinham essas sobreposições eram duplicados, contendo apenas a anotação que não sobrepõe. No exemplo acima, a nossa solução foi criar um texto com anotação da "ENTE" e outro com o mesmo texto só que anotando apenas a "IDENTIFICACAO". Essa sobreposição só ocorre nos tipos de "ENTE", "ORGAO" e "IDENTIFICACAO", pois em alguns atos essas informações estão contidas dentro da identificação do ato. Após essas etapas de pré-processamento, obtemos um total de 1.053 atos, ou seja, 193 atos tiveram que ser duplicados, visto que inicialmente tínhamos 860 atos de licitações.

Na etapa de treinamento do modelo, foi feita uma divisão dos dados em treino e teste, igual nos modelos de classificação discutidos acima. Separamos 953 atos para treinamento e 100 atos para teste para validação do modelo treinado. Usamos um modelo em branco do NER em português do brasil, ou seja, um modelo sem ter nenhuma entidade nomeada. O modelo foi treinado usando 100 épocas, com o valor de dropout rate (taxa de abandono) igual a 0.3, esse atributo significa a dificuldade do modelo memorizar os dados, assim evitando que ele apenas memorize os dados de treino e não seja bom com dados não vistos. Por fim, após treinar o modelo, foi utilizado os dados de testes para verificar se o modelo realmente generalizou bem. Foi usado os 100 atos de dados de testes, entre eles tem 84 anotações do tipo de "IDENTIFICACAO", 84 anotações de "OBJETO", 64 anotações de "ORGAO" e 59 anotações de "ENTE", essa distribuição pode ser observada pela figura 13, lembrando que em cada ato pode ter mais de um tipo anotado.

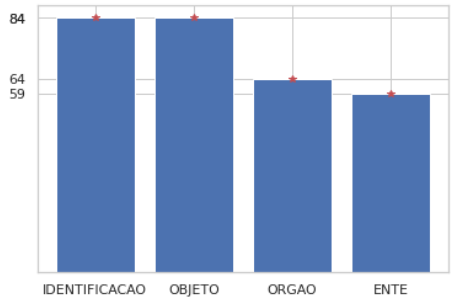

Figura 13: Distribuição dos dados de testes.

Fonte: Autoria Própria

Pela figura 14, pode-se ver os acertos que o modelo obteve por cada tipo de anotação nos dados de testes. Os tipos de "IDENTIFICACAO" e "ENTE" foram os que tiveram mais precisão, os dois com uma acurácia de 96%, seguido pelo tipo de "OBJETO" com uma acurácia de 80% e o tipo de "ORGAO" com 67%, o pior resultado entre os quatro.

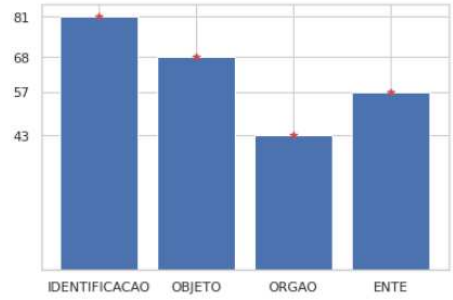

Figura 14: Acertos da predição do modelo.

Fonte: Autoria Própria

[9]https://colab.research.google.com/drive/1lLbpiSQbguaY0Z6ZJcKFu9OM7XPJy4uV?usp=sharing

4 CONCLUSÃO

Este trabalho procurou encontrar um método para extrair os atos de um DOE-AC, que são no formato PDF, encontrar um modelo inteligente capaz de classificar esses atos como licitação ou não e por fim conseguir extrair algumas informações dessas licitações para que seja possível verificar se a licitação do DOE também se encontra no sistema de gerenciamento de licitações do TCE-AC, Licon, com intuito de auxiliar o TCE-AC em verificar se o jurisdicionado fez a publicação correta nos dois meios. Para isso, foi feita uma análise entre algumas bibliotecas para extração de textos dos PDFs, para conseguir estruturar o PDF em um arquivo JSON. Depois, foram analisados diversos modelos de classificação, onde o BERTimbau, modelo pré-treinado, obteve os melhores resultados, sendo eleito o mais apto a realizar a tarefa objetivada. E por fim, utilizamos a técnica NER para extração das entidades específicas da licitação e assim permitindo que se possa verificar se a licitação foi publicada no licon.

AGRADECIMENTOS

Agradeço aos meus pais, Pedro Nolascio e Luzia Rosiene, por todo o apoio, incentivo e sacrifício, graças a eles que isso se tornou possível. As minhas duas irmãs, Brenda e Ellen, também ao meu sobrinho, Adam Ramalho, e a todos aos meus familiares, em especial à Rosilda Carvalho, por terem de alguma forma me dado forças e apoio para continuar. Aos meus amigos: Lucas Andrade, Igor Silveira, Matheus Santana, Yuri Souza, Vinicius Barbosa, José Davi, Levi Gomes, Samuel Vasconcelos, Cindy Evelyn e Luma Silveira por terem contribuído com ensinamentos acadêmicos e/ou pessoais. Agradeço também a todos os meus professores que contribuíram para a minha formação. Por fim, quero agradecer ao meu avô, Dijalma Carvalho Leite (in memorian), por ter ajudado a minha família e por ter sido uma das pessoas mais importantes da minha vida.

REFERÊNCIAS

[1] DOUGLAS, A. *O que faz o Tribunal de Contas do Estado*. 2020. Disponível em: <https://portal.unicap.br/-/o-que-faz-o-tribunal-de-contas-do-estado->. Acesso em: 9 de jul. de 2022. 1

[2] NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, v. 18, n. 5, p. 544–551, 09 2011. ISSN 1067-5027. Disponível em: <https://doi.org/10.1136/amiajnl-2011-000464>. Acesso em: 9 de jul. de 2022. 1

[3] TRIBUNAL de Contas do Estado do Parana. 2022. Disponível em: <https://www1.tce.pr.gov.br/conteudo/jurisdicionados/292958/area/251>. Acesso em: 20 de abr. de 2022. 2

[4] VASCAONCELLOS, P. Crisp-dm, semma e kdd: conheça as melhores técnicas para exploração de dados. 2017. Disponível em: <https://paulovasconcellos.com.br/crisp-dm-semma-e-kdd-conheÃğa-as-melhores-tÃĺcnicas-para-exploraÃğÃčo-de-dados-560d294547d2>. Acesso em: 2 de maio de 2022. 2

[5] A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, v. 181, p. 526–534, 2021. ISSN 1877-0509. CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921002416>. 2

[6] ROSSUM, G. V.; DRAKE, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697. 2

[7] DIáRIO Oficial do Estado do Acre. 2022. Disponível em: <http://diario.ac.gov.br/>. Acesso em: 20 de abr. de 2022. 2, 3

[8] NURSEITOV, N. et al. Comparison of json and xml data interchange formats: a case study. *Caine*, v. 9, p. 157–162, 2009. 3

[9] LICON. 2011. Disponível em: <http://sistemas.tce.ac.gov.br/licon/>. Acesso em: 5 de maio de 2022. 3

[10] NAKAYAMA, H. et al. *doccano: Text Annotation Tool for Human*. 2018. Software available from https://github.com/doccano/doccano. Disponível em: <https://github.com/doccano/doccano>. 4

[11] LEMAîTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, v. 18, n. 17, p. 1–5, 2017. Disponível em: <http://jmlr.org/papers/v18/16-365.html>. 5

[12] BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: "O'Reilly Media, Inc.", 2009. 5

[13] PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. 5

[14] SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. In: ____. [S.l.: s.n.], 2020. p. 403–417. ISBN 978-3-030-61376-1. 5

[15] MATTHEW, H. et al. spacy: Industrial-strength natural language processing in python. 2020. 5, 7

[16] O que é o NER (Reconhecimento de Entidade Nomeada) no Serviço Cognitivo do Azure para Linguagem? 2022. Disponível em: <https://docs.microsoft.com/pt-br/azure/cognitive-services/language-service/named-entity-recognition/overview>. 5

[17] CALZOLARI, N. et al. (Ed.). *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), 2016. Disponível em: <https://aclanthology.org/L16-1000>. 5

[18] SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*, 2019. Disponível em: <http://arxiv.org/abs/1909.10649>. 6

[19] PASZKE, A. et al. Automatic differentiation in pytorch. In: *NIPS-W*. [S.l.: s.n.], 2017. 6

[20] MOON, S. et al. Automated construction specification review with named entity recognition using natural language processing. *Journal of Construction Engineering and Management*, American Society of Civil Engineers, v. 147, n. 1, p. 04020147, 2021. 7