

Classificação do teor das intimações recebidas do sistema Processo Judicial Eletrônico: uma análise comparativa dos algoritmos de Redes Neurais Recorrentes

Cleidiane de Souza Santos	Isabela Almeida Marinho da	Mônica Silva Rivas	Márcio Freire Cruz
Centro Universitário	Silveira	Centro Universitário	Centro Universitário
SENAI CIMATEC	Centro Universitário	SENAI CIMATEC	SENAI CIMATEC
Salvador, Brasil	SENAI CIMATEC	Salvador, Brasil	Salvador, Brasil
cleidianesouza@gmail.com	Salvador, Brasil	msrivas@gmail.com	m.cruz@ba.docente.senai.br
	iamsilveira2@gmail.com		

Resumo: Este estudo teve como objetivo desenvolver um modelo de Machine Learning para análise e classificação do gênero do movimento, com base no teor das intimações recebidas do sistema Processo Judicial Eletrônico (PJe), no âmbito da 1ª instância, com a finalidade de agilizar a atuação dos Promotores de Justiça. Para isso, foi explorada a eficácia de diferentes arquiteturas de Redes Neurais Recorrentes (RNNs) na classificação do teor das intimações judiciais recebidas pelo Ministério Público do Estado da Bahia (MPBA). Foram utilizadas três variantes de RNNs - RNN Simples, LSTM e BLSTM - para classificar os textos. Os modelos foram treinados e avaliados com base na acurácia de teste, precisão e revocação, utilizando a matriz de confusão. Os resultados da predição mostraram que os modelos LSTM e BLSTM apresentaram maior acurácia geral, enquanto os modelos RNNs Simples, embora menos precisos, demandaram menor tempo de treinamento. As métricas acurácia geral, precisão para a classificação do gênero do movimento "Ciência" e revocação para "Não Ciência" dos modelos escolhidos de cada arquitetura foram, respectivamente: RNN Simples - 86%, 76% e 83%; LSTM - 88%, 80% e 87%; BLSTM - 88%, 79% e 86%.

Palavras-chave: Redes Neurais Recorrentes, LSTM, BLSTM, Bidirecional, Classificação de Texto, Machine Learning, Processamento de Linguagem Natural, Intimações Judiciais, Ministério Público.

1 Introdução

No âmbito do Ministério Público do Estado da Bahia (MPBA), as intimações eletrônicas enviadas pelo sistema Processo Judicial Eletrônico (PJe) são recepcionadas pelo sistema Integrado de Dados, Estatística e Atuação (IDEA) para a efetivação da consulta eletrônica ao teor da intimação e a certificação nos autos da sua realização, como por exemplo, através de uma mera ciência. A referida consulta deve ser realizada em até dez dias corridos contados da data do envio da intimação, sob pena de considerar a intimação automaticamente realizada na data do término desse prazo.

Os movimentos de ciência são definidos apenas com base na análise do teor da intimação. Por sua vez, os demais movimentos exigem conhecimento do conteúdo integral do processo para análise, estudo e elaboração da peça.

Em 2023, o sistema IDEA recepcionou mais de 1.600 intimações/ dia do sistema PJe de 1ª instância. A análise das intimações judiciais é um desafio para os Promotores de Justiça devido ao volume exorbitante, à diversidade de conteúdo e à necessidade de resposta rápida e precisa.

O Processamento de Linguagem Natural

(PLN) é um campo de pesquisa que tem como objetivo investigar e propor métodos e sistemas de processamento computacional da linguagem humana [3], sendo uma de suas aplicações mais comuns a classificação de textos. A classificação do teor das intimações recebidas é um desafio por se tratar da linguagem jurídica, caracterizada por sua complexidade, vocabulário específico e estrutura formal.

Os modelos de Redes Neurais Recorrentes (RNNs) foram escolhidos para este estudo devido à sua capacidade de processar sequências de dados onde o contexto e a ordem temporal são essenciais para a compreensão. Três variantes de RNNs foram utilizadas para a classificação do teor das intimações: a RNN Simples, a Long Short-Term Memory (LSTM) e a Bidirecional LSTM (BLSTM).

O objetivo deste trabalho foi realizar uma triagem inicial classificando o gênero do movimento do teor das intimações judiciais em "Ciência" e "Não Ciência".

2 Trabalhos Relacionados/Referencial Teórico

Antes de explorar a aplicação das RNNs, foco deste estudo, foram utilizados como referencial os métodos tradicionais de classificação de textos: Regressão Logística e o Naive Bayes. Esses modelos fornecem uma base sólida para compreender como técnicas mais avançadas, como as RNNs, podem melhorar o resultado na classificação de textos.

Segundo [13], a Regressão Logística é uma técnica estatística usada para modelar a relação entre uma variável dependente binária e uma ou mais variáveis independentes. É amplamente utilizada para prever a probabilidade de ocorrência de um evento, sendo particularmente útil em situações onde a variável de resposta é categórica (0 ou 1, verdadeiro ou falso). De acordo com [14], Naive Bayes é um método de classificação baseado no Teorema de Bayes com a suposição de que todas as características são independentes umas das outras dado a classe. É um modelo

probabilístico que utiliza a probabilidade condicional para realizar a classificação.

As RNNs surgiram da necessidade de armazenar estados ocultos para processar dados sequenciais, onde a ordem importa, visando sanar a limitação das redes neurais tradicionais que tratam cada entrada de forma independente. Esta abordagem foi influenciada pelos estudos de [12] e [5], que introduziram arquiteturas capazes de capturar dependências temporais em dados sequenciais. Anteriormente, o modelo de Hopfield, descrito em [11], já havia estabelecido a base para redes neurais com capacidades de memória, embora em um contexto diferente, enfatizando a importância dos estados estáveis e da autoassociação em redes neurais.

RNN Simples é a forma mais básica de RNN, em que a propagação de informações da rede se dá através das conexões entre as unidades que formam um ciclo, que permite a persistência da informação ao longo da sequência, utilizando estados ocultos que são atualizados a cada passo de tempo. No entanto, sua capacidade de capturar dependências de longo prazo é limitada pela dificuldade de lembrar dessas informações devido ao desvanecimento de gradientes, conforme discutido por [2].

LSTM, variação das RNNs descrita por [10], é projetada para capturar dependências de longo prazo em dados sequenciais. Sua arquitetura contém células de memória com mecanismos de portas — entrada, esquecimento e saída — que regulam o fluxo de informações, permitindo adicionar ou remover conteúdo de forma controlada. Isso evita os problemas de explosão e desaparecimento de gradientes, tornando-a eficaz para manter informações relevantes por longos períodos.

Rede Neural Recorrente Bidirecional (BRNN), introduzida por [17], consiste em duas camadas separadas, onde uma camada processa a sequência de entrada na direção temporal normal, enquanto a outra camada processa a sequência na direção inversa. As saídas dessas duas camadas são então combinadas, geralmente concatenadas, para

formar a saída final em cada ponto da sequência. Isso a torna capaz de considerar informações contextuais tanto do passado quanto do futuro em uma sequência de dados. A Bidirecional LSTM (BLSTM), RNN que combina a estrutura bidirecional com células LSTM, visa melhorar as tarefas de processamento sequencial, especialmente em problemas onde o contexto completo de uma sequência é crucial, conforme discutido por [8].

Alguns estudos já exploraram a aplicação de RNN para a classificação de textos: [1] explorou o uso de RNN para a classificação de fake news em língua portuguesa; [6] aplicou técnicas de redes neurais para a classificação de textos, com foco em análise de sentimentos; [15] investigou a mineração de textos para a classificação de processos judiciais trabalhistas.

3 Metodologia

O processo Knowledge Discovery in Databases (KDD) foi utilizado para guiar a metodologia.

3.1 Coleta de Dados

Para viabilizar a pesquisa, inicialmente, foi obtida a autorização da administração superior do MPBA para acessar os dados do sistema IDEA. Posteriormente, procedeu-se à extração dos dados necessários para a construção do dataset, com a consulta direta ao banco de dados SQL Server, onde foram obtidos os registros de recebimento das intimações, incluindo a identificação do processo, a data de recebimento, a identificação do aviso, o nome do arquivo HTML do teor da intimação, a identificação do movimento e a descrição desse movimento. As intimações referem-se à processos sem sigilo, do ano de 2023, recebidas do sistema PJe de 1ª instância. O resultado da consulta retornou 593.988 registros. Para complementar os dados, foi necessário obter os arquivos HTML armazenados no servidor de arquivos. Com base nos nomes dos arquivos HTML foi realizada a recuperação de 593.907 arquivos, totalizando o tamanho de 16,1 GB.

3.2 Preparação dos Dados

Em posse das duas bases de dados, contendo os dados das intimações e os arquivos HTML, foi possível unir as informações em um dataset único. O primeiro passo consistiu em identificar quais registros das intimações tinham o nome correspondente a um arquivo HTML. Após a exclusão dos registros sem arquivo referenciado e dos registros que referenciavam o mesmo arquivo, restaram 593.831 registros. Em seguida, os arquivos HTML foram lidos e o texto extraído foi armazenado no dataset. Durante a leitura, observou-se que parte dos arquivos estavam corrompidos não sendo possível extrair o texto. Estas ocorrências foram excluídas, restando 588.368 registros. Por fim, foi criado o atributo indicativo do gênero do movimento, a partir da descrição do movimento. O valor 1 foi atribuído para os registros com a descrição iniciada por "ATOS FINALÍSTICOS > Ciência", e os demais receberam valor 0.

3.3 Análise Exploratória

O conjunto de dados base era composto de oito atributos, sendo cinco do tipo inteiro e três do tipo objeto. Os atributos com relevância para este estudo são: DsMovimento, campo objeto, que indica a especificação do movimento a que se refere a intimação; StCiencia, campo inteiro, que indica o gênero do movimento, onde 0 significa "Não Ciência" e 1 significa "Ciência"; e Conteudo_HTML, campo objeto, contendo o teor da intimação.

Das 588.368 instâncias do conjunto de dados, nove apresentaram registros nulos no atributo Conteudo_HTML, sendo necessária a eliminação imediata destas. Dessa forma, a coleção de dados foi redimensionalizada para 588.359 instâncias.

Analisando o atributo DsMovimento, foram identificados 164 movimentos distintos e altamente desbalanceados. Cerca de 90% dos dados estavam concentrados em apenas dez movimentos, sendo as duas maiores classes: "ATOS FINALÍSTICOS > Manifestação (920198)", movimento de "Não Ciência",

com 312.992 (53,2%) instâncias e a classe "ATOS FINALÍSTICOS > Ciência > Outras ciências (920197)", movimento de "Ciência", com 73.531 (12,5%) instâncias.

O atributo StCiencia também estava desbalanceado, sendo 372.036 (63,2%) instâncias da classe "Não Ciência" e 216.323 (36,8%) instâncias da classe "Ciência".

O atributo Conteudo_HTML foi analisado quanto ao número de palavras nas intimações, com um máximo de 88.508 palavras e um mínimo de zero, onde a média foi de 292 palavras e a mediana de 122. Observou-se que 75% das intimações tinham até 276 palavras e apenas 13% excediam 500 palavras, sendo que textos com zero palavra continham apenas quebras de linha. Intimações do movimento "Ciência" continham mais palavras, com mediana de 235, comparadas às "Não Ciência", cuja mediana era de 103; no entanto, a intimação com o maior número de palavras, 88.508, pertencia ao movimento "Não Ciência".

Analisando ainda o atributo Conteudo_HTML, foram identificadas 55.116 (9,37%) ocorrências de intimações que possuíam o mesmo teor de outras intimações, porém com classificações divergentes.

3.4 Recorte da Amostra

Os textos que resultaram em conteúdo vazio ou apresentaram textos repetidos com indicativos de classes divergentes entre si, foram removidos do dataset. Posteriormente, devido às limitações de recursos computacionais para treinamento do modelo, foram eliminados da base os textos com mais de 500 palavras, que representavam apenas 13% do total, conforme Subseção 3.3. Também foram eliminados os textos compostos por apenas uma palavra, por não terem conteúdo expressivo. O dataset foi então reduzido para 459.686 registros, dos quais 306.416 foram classificados como não ciência e 153.270 como ciência.

Para assegurar um conjunto de dados balanceado, todas as amostras classificadas como ciência foram mantidas, enquanto a amostra de não ciência foi ajustada para corresponder à mesma quantidade, resultando em um total de 306.540 registros para a construção dos modelos. Optou-se pela exclusão de parte da amostra da classe de maior ocorrência devido ao volume significativo dos dados e às limitações de recursos citadas anteriormente. O recorte da amostra levou em consideração a distribuição proporcional de movimentos em relação ao dataset original, para evitar a eliminação aleatória de registros do mesmo movimento.

3.5 Pré-processamento

3.5.1 Anonimização, Sumarização e Eliminação de parágrafos irrelevantes

Durante o pré-processamento, foi realizada uma tentativa de anonimização dos dados, a fim de não expor nomes de réus e vítimas. Inicialmente, foram utilizadas as bibliotecas NLTK e SpaCy, porém muitos nomes não foram identificados como entidades. Em seguida, foi realizado um teste com o modelo "pierreaguillou/ner-bert-large-cased-pt-lenerbr"[9], especializado no domínio jurídico brasileiro, que obteve um desempenho superior ao das bibliotecas. No entanto, a anonimização da amostra de 24 mil registros consumiu cerca de 55 horas, tornando impraticável a utilização contínua de um equipamento dedicado para anonimizar o dataset completo. Paralelamente, foram realizadas tentativas de sumarizar os textos longos com modelos como o gpt-3.5-turbo da OpenAI¹, mas para não expor os dados aos modelos, a tarefa de anonimização era um pré-requisito. Sem a possibilidade de anonimização em um ambiente seguro, optou-se por realizar o processamento localmente, dispensando o uso do Google Colaboratory² para treinar os modelos em ambiente de GPU. Diante da impossibilidade da sumarização, foi implementado um script utilizando a biblioteca BeautifulSoup para eliminar

¹<https://openai.com/api/>

²<https://colab.research.google.com/>

partes irrelevantes dos textos, removendo parágrafos com determinados padrões de formatação. No entanto, a alta variabilidade dos documentos resultou na exclusão de conteúdos importantes para a compreensão do texto, levando à descontinuidade desta estratégia.

3.5.2 Limpeza dos Dados

Foram realizadas algumas etapas para preparar o teor da intimação para a sua classificação: eliminação das URLs, endereços de e-mail e dígitos através do uso de expressões regulares, conversão das letras para minúsculo, remoção de pontuações e acentos, remoção de stopwords utilizando o modelo `pt_core_news_lg` do SpaCy. Foi criado o atributo `Conteudo_sem_stop_word` para armazenar o resultado desse pré-processamento. Complementando esta etapa, foi realizada a análise da frequência das palavras nos textos. As palavras com maior frequência foram: "bahia", "ba", "salvador", "santos" e "silva". Nesta nova tentativa, os locais e nomes/sobrenomes mais frequentes foram substituídos pelos termos [Local] e [Nome], minimizando a interferência no vocabulário de treinamento. Para representar este novo tratamento, foi criado o atributo `Conteudo_sem_stop_word_v2`. Após a limpeza dos dados, cinco registros do arquivo de amostra ficaram com o texto pré-processado vazio, totalizando agora uma amostra de 306.535 registros para a construção dos modelos de classificação.

Além da amostra para a construção do modelo de classificação, preparou-se outra para a predição dos modelos de melhor desempenho, composta por 588.368 registros, os mesmos da Subseção 3.2. Após a limpeza dos dados no atributo `Conteudo_sem_stop_word`, 94 registros vazios foram eliminados e o dataset para a predição foi reduzido para 588.274 registros.

3.6 Desenvolvimento do Modelo

O desenvolvimento foi realizado na linguagem de programação Python (versão 3.7.11). Para a preparação e treinamento dos modelos foram utilizadas as bibliotecas Scikit-Learn (versão 1.4.2) e TensorFlow (versão 2.16.1). O ambiente de desenvolvimento foi configurado no Anaconda (versão 2.5.2) e o código foi executado no Jupyter Notebook (versão 7.0.8), em PCs Windows sem GPU.

Como baseline para a classificação do teor das intimações, os modelos clássicos de Regressão Logística e Naive Bayes foram empregados. Tais modelos foram escolhidos pela simplicidade e eficiência computacional. Na sequência, foram implementados os três modelos de RNNs: RNN Simples, LSTM e BLSTM.

O dataset utilizado para a construção dos modelos foi o de 306.535 amostras, citado na Subsubseção 3.5.2. Os dados foram divididos em conjuntos de treinamento, validação e teste, utilizando 70%, 15% e 15%, respectivamente. Foram utilizadas dez épocas com a parada antecipada baseada na perda de validação e a paciência igual a cinco. Optou-se por limitar o treinamento a dez épocas baseado em experimentos preliminares, onde o modelo alcançou um ponto de estabilização na redução da perda próximo a esse número. Além disso, foram consideradas as limitações de tempo de treinamento e recursos computacionais.

Os textos foram tokenizados resultando em um vocabulário de 78.832 palavras únicas no conjunto de treinamento. A camada de embedding padrão do TensorFlow foi utilizada para converter as sequências de tokens em representações densas de baixa dimensão.

Em todos os modelos, os pesos das camadas da rede foram inicializados usando o método GlorotUniform, como descrito por [7], enquanto os pesos recorrentes foram inicializados utilizando o método Orthogonal, ambos com a mesma semente de aleatoriedade. A função de perda utilizada foi a BinaryCrossentropy, por se tratar de apenas duas classes, e o otimizador foi o Adam, devido à sua eficácia em ajustar

Tabela 1: Configurações treinadas para cada arquitetura

Modelo	Camadas	Unidades	Função de Ativação	Otimizador	Dimensões de Saída	Tamanho do Vocabulário	Tamanho do Lote
RNN Simples	1 RNN	16, 32 e 64	Sigmoid e ReLU	Adam com taxas de aprendizado de 1e-3 e 1e-4	64 e 256	2.000 e 5.000	32 e 64
LSTM	1 e 2 LSTM Dropout de 0.1 e 0.2 (apenas nos modelos de 2 camadas)	1ª: 16, 32 e 64 2ª: 8, 16 e 32	Sigmoid e ReLU	Adam com taxas de aprendizado de 1e-3 e 1e-4	128, 256 e 300	1.000, 2.000, 3.000 e 5.000	32 e 64
BLSTM	1 e 2 LSTM Dropout de 0.2 e 0.5 (apenas nos modelos de 2 camadas)	1ª: 64 e 128 2ª: 32 e 64	Sigmoid e ReLU	Adam com taxas de aprendizado de 1e-3 e 1e-4	64 e 256	2.000 e 5.000	32 e 64
	1 Densa	32 e 64					

Fonte: Dados da pesquisa.

adaptativamente a taxa de aprendizado para cada parâmetro, especialmente em dados esparsos, conforme recomendado por [16].

Preliminarmente, foram realizados experimentos utilizando uma amostra reduzida de 24 mil registros, visando identificar os melhores parâmetros e configurações para o desempenho dos modelos. A escolha de alguns valores de hiperparâmetros, como as unidades e as dimensões de saída, levou em consideração a capacidade de processamento disponível e o tempo de treinamento. Foram exploradas também variações nos atributos: Conteúdo_HTML, Conteúdo_sem_stop_word e Conteúdo_sem_stop_word_v2.

Para a RNN Simples, foram treinadas 17 variações para o atributo Conteúdo_sem_stop_word, 16 para o Conteúdo_HTML e 16 para o Conteúdo_sem_stop_word_v2. Para a LSTM, foram treinadas 39 variações para o atributo Conteúdo_sem_stop_word, 16 para o Conteúdo_HTML e oito para o Conteúdo_sem_stop_word_v2. Já para a BLSTM, foram treinadas 37 variações para o atributo Conteúdo_sem_stop_word, oito para o Conteúdo_HTML e quatro para o Conteúdo_sem_stop_word_v2. Os modelos que utilizaram o atributo Conteúdo_sem_stop_word alcançaram os melhores resultados, enquanto que o atributo Conteúdo_sem_stop_word_v2 mostrou-se menos eficaz, levando à

decisão de não utilizá-lo no dataset maior.

Quanto aos hiperparâmetros, os modelos foram treinados com as configurações da Tabela 1.

Após o treinamento com o dataset de 24 mil registros, foram identificados os modelos que apresentaram os melhores resultados em termos de acurácia de teste e tempo de treinamento. Esses modelos foram então utilizados para um novo treinamento com o dataset maior de 306.535 registros. No caso da RNN Simples, foram treinados 26 modelos, sendo oito com o atributo Conteúdo_HTML e 18 com o atributo Conteúdo_sem_stop_word. Para a arquitetura LSTM, 19 modelos foram treinados, sendo sete com o atributo Conteúdo_HTML e 12 com o atributo Conteúdo_sem_stop_word. Para a arquitetura BLSTM, dez modelos foram treinados, sendo um com o atributo Conteúdo_HTML e nove com o atributo Conteúdo_sem_stop_word.

3.7 Avaliação do Modelo

A acurácia de teste foi a principal métrica escolhida para a avaliação dos modelos treinados, por sua simplicidade e adequação ao problema de classificação em questão, onde a classificação correta de cada instância era importante. O tempo de treinamento dos modelos, diante das limitações de recursos

computacionais disponíveis, foi levado em consideração na escolha do modelo.

A matriz de confusão também foi utilizada na análise, o que permitiu a visualização das previsões corretas e incorretas em cada classe, identificando possíveis desequilíbrios e erros na classificação.

Para complementar, após a seleção do melhor modelo, foi realizada a predição do conjunto de dados completo, composto por 588.274 registros. Na análise da predição, também foi considerada a métrica de precisão, pois é importante minimizar os falsos positivos, já que os movimentos do gênero "Não Ciência" exigem uma análise além do teor da intimação. A revocação também foi uma métrica importante para assegurar que o modelo identificasse corretamente os casos de "Não Ciência".

4 Resultados e Discussão

A análise comparativa entre os algoritmos de RNNs para classificação do teor das intimações mostrou que, embora a RNN Simples tenha apresentado um tempo de execução menor, os modelos LSTM e BLSTM superaram seu desempenho em termos de acurácia de teste. Para as três arquiteturas, o atributo `Conteudo_sem_stop_word_v2`, com maior nível de limpeza nos dados, teve a acurácia de teste inferior ao do atributo `Conteudo_sem_stop_word`. Também foi constatado que a acurácia de teste dos modelos que utilizaram a função de ativação ReLU foi inferior aos que utilizaram a função de ativação Sigmoid, comprovando que para as classificações binárias o mais indicado é a função de ativação Sigmoid, como descrito por [4]. Os resultados também indicaram que a combinação dos hiperparâmetros influenciou no tempo de treinamento dos modelos.

Dentre os modelos RNN Simples treinados com o dataset de 24 mil registros, os resultados variaram de 83,94% a 50,72% na acurácia de teste. Arredondando a acurácia de teste para números inteiros, os resultados mostraram que dois modelos atingiram 84%; oito atingiram 83%; 14 atingiram 82%; sete atin-

giram 81%; oito atingiram 80%; três atingiram 79%; dois atingiram 78% e 77%; um atingiu 76%, 74% e 51%. Este último resultado foi obtido em uma das configurações que utilizou a função de ativação ReLU. Em relação ao tempo de treinamento, os resultados variaram entre 101 e 1.913 segundos. Em média, o tempo de treinamento ficou em 446 segundos, com 15 modelos acima da média e 34 abaixo da média.

Foram realizados testes no dataset de 306.535 registros com as configurações que obtiveram os melhores resultados. O mesmo modelo performou melhor nos dois datasets. Os resultados variaram de 88,25% a 84,68% na acurácia de teste. Arredondando a acurácia de teste para números inteiros, os resultados mostraram que oito modelos atingiram 88%, nove atingiram 87%, cinco atingiram 86% e quatro atingiram 85%. O tempo de treinamento variou entre 2.128 e 13.690 segundos. Em média, o tempo de treinamento ficou em 6.417 segundos, com 11 modelos acima da média e 15 abaixo da média.

Os dados do modelo escolhido para a RNN Simples foram detalhados no número 1 da Tabela 2.

Dentre os modelos LSTM treinados com o dataset de 24 mil registros, os resultados variaram de 84,86% a 50% na acurácia de teste. Arredondando a acurácia de teste para números inteiros, os resultados mostraram que três modelos atingiram 85%, 25 atingiram 84%, 26 atingiram 83%, quatro atingiram 82%, um atingiu 81% e quatro atingiram 50%. Este último resultado foi obtido em uma das configurações que utilizou a função de ativação ReLU. Em relação ao tempo de treinamento, os resultados variaram entre 279 e 2.327 segundos. Em média, o tempo de treinamento ficou em 693 segundos, com 22 modelos acima da média e 41 abaixo da média.

Foram realizados testes no dataset de 306.535 registros com as configurações que obtiveram os melhores resultados. O modelo de melhor acurácia de teste no dataset menor não foi o que obteve a maior acurácia de teste no dataset maior, podendo ser atribuído ao fato de que modelos diferentes respondem de maneira distinta à variação no tamanho dos da-

Tabela 2: Resultado do modelo escolhido para cada arquitetura

Nº	Configuração do Modelo	Dataset com 24 mil amostras				Dataset com 306.535 amostras			
		Acurácia Teste (%)	Perda Teste	Acurácia Validação (%)	Tempo (s)	Acurácia Teste (%)	Perda Teste	Acurácia Validação (%)	Tempo (s)
1	RNN Simples c/ 1 camada RNN de 16 units, activation=sigmoid, Adam(1e-3), output_dim=256, vocab_size=5000, batch_size=64, atributo=Conteudo sem stop word	83,00	0,4689	84,67	155	88,25	0,3093	87,99	2.128
2	LSTM c/ 1 camada LSTM de 32 units, activation=sigmoid, Adam(1e-3), output_dim=256, vocab_size=5000, batch_size= 64, atributo=Conteudo sem stop word	84,56	0,4249	85,83	447	88,91	0,3017	88,58	5.290
3	BLSTM c/ 1 camada LSTM de 32 units, 1 Dense de 32 units, activation=sigmoid, Adam(1e-3), output_dim=256 vocab_size=5000, batch_size=64 atributo=Conteudo sem stop word	85,44	0,3909	84,11	526	88,70	0,3026	88,98	6.750

Fonte: Dados da pesquisa.

dos, onde maiores volumes de dados podem beneficiar mais configurações que têm melhor capacidade de generalização. Os resultados variaram de 89,06% a 86,64% na acurácia de teste. Arredondando a acurácia de teste para números inteiros, os resultados mostraram que cinco modelos atingiram 89%, 12 atingiram 88% e dois atingiram 87%. O tempo de treinamento variou entre 5.290 e 29.480 segundos. Em média, o tempo de treinamento ficou em 13.617 segundos, com oito modelos acima da média e 11 abaixo da média.

Os dados do modelo escolhido para a LSTM foram detalhados no número 2 da Tabela 2.

Dentre os modelos BLSTM treinados com o dataset de 24 mil registros, os resultados variaram de 85,83% a 80,42% na acurácia de teste. Arredondando a acurácia de teste para números inteiros, os resultados mostraram que cinco modelos atingiram 86%; 25 atingiram 85%; 12 atingiram 84%; três atingiram 83% e 82%; um atingiu 80%. Este último resultado foi obtido em uma das configurações que utilizou a função de ativação ReLU. Em relação ao tempo de treinamento, os resultados variaram entre 218 e 8.191 segundos. Em média, o tempo de treinamento ficou em 1.811 segundos, com 17 modelos acima da média e 32 abaixo da média.

Foram realizados testes no dataset de 306.535 registros com as configurações que obtiveram os me-

lhores resultados. O mesmo modelo performou melhor nos dois datasets. Os resultados variaram de 89,04% a 88,39% na acurácia de teste. Arredondando a acurácia para números inteiros, os resultados mostraram que nove modelos atingiram 89% e apenas um atingiu 88%. O tempo de treinamento variou entre 6.750 e 42.504 segundos. Em média, o tempo de treinamento ficou em 20.961 segundos, com 5 modelos acima da média e 5 abaixo da média.

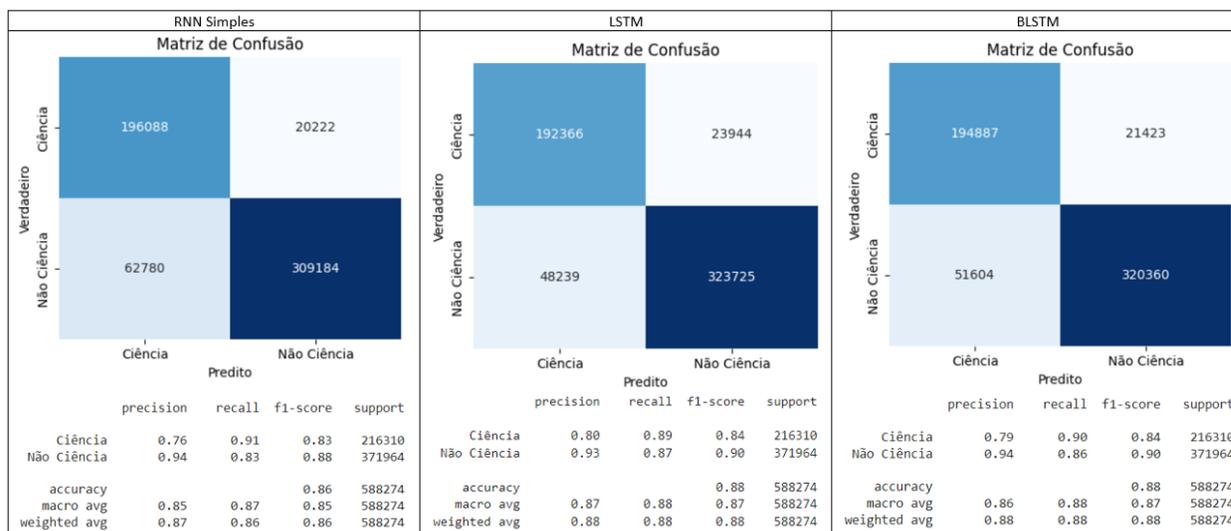
Os dados do modelo escolhido para a BLSTM foram detalhados no número 3 da Tabela 2.

Em relação ao treinamento dos modelos clássicos, a regressão logística apresentou uma acurácia de teste competitiva em comparação com as RNNs, enquanto o Naive Bayes apresentou um desempenho inferior. Com o dataset de 306.535 registros e atributo Conteudo_HTML, o modelo de Regressão Logística alcançou acurácia de teste de 87%, em seis segundos de treinamento. Já o modelo Naive Bayes obteve uma acurácia de teste de 77%, com tempo de treinamento de 39 segundos. A remoção de stopwords reduziu ligeiramente a acurácia de teste em ambos os modelos, que alcançaram 84% e 76% respectivamente, e tempo de treinamento de dois e 24 segundos.

4.1 Resultados da Predição

A predição foi realizada com os modelos escolhidos de cada variante (RNN Simples, LSTM,

Figura 1: Matriz de confusão do modelo escolhido para cada arquitetura



Fonte: Dados da pesquisa.

BLSTM) no dataset completo, com 588.274 registros, conforme indicado na Figura 1.

No contexto deste artigo, se uma intimação do gênero do movimento "Não Ciência" for erroneamente classificada como "Ciência" pelo modelo, induzirá a resposta apenas pelo teor da intimação, ao passo que é necessário analisar o conteúdo integral do processo. Portanto, é importante minimizar falsos positivos (classificar "Não Ciência" como "Ciência"), tornando a precisão para "Ciência" e a revocação para "Não Ciência" métricas relevantes para o negócio.

Entre os modelos, o LSTM teve a melhor precisão para "Ciência" e melhor revocação para "Não Ciência", porém o BLSTM apresentou um resultado muito próximo. Já o modelo RNN Simples teve os menores valores das métricas: precisão para "Ciência", revocação para "Não Ciência" e acurácia geral.

O modelo clássico que apresentou melhor resultado na predição foi Regressão Logística, com acurácia geral de 84%, precisão de 74% para "Ciência" e revocação de 82% para "Não Ciência", utilizando o atributo Conteúdo_HTML. Esse também foi o melhor modelo clássico no treinamento.

5 Conclusão

Este estudo comparou o desempenho de diferentes algoritmos de Redes Neurais Recorrentes na classificação do gênero do movimento do teor das intimações recebidas do sistema PJe, em "Ciência" e "Não Ciência", visando a realização de uma análise preliminar que auxilie o trabalho dos Promotores de Justiça.

Considerando os resultados dos experimentos, o modelo que utilizou arquiteturas LSTM mostrou-se mais eficaz, alcançando uma acurácia geral de 88%, precisão de 80% para a classificação do gênero do movimento "Ciência" e revocação de 87% para "Não Ciência". A arquitetura Bidirecional obteve resultados muito próximos, entretanto demandou mais tempo de treinamento, sendo a acurácia geral de 88%, precisão de 79% para "Ciência" e revocação de 86% para "Não Ciência". Por outro lado, a RNN Simples apresentou menor acurácia geral, com 86%, menor precisão para a classificação do gênero do movimento "Ciência", atingindo 76%, e menor revocação para "Não Ciência", com 83%. Em contrapartida, ela demandou menor tempo de treinamento e uso de recursos computacionais em comparação aos demais modelos. A decisão entre utilizar RNNs Simples ou

os modelos LSTMs deve considerar o equilíbrio entre a métrica desejada e os recursos computacionais disponíveis.

A principal dificuldade no desenvolvimento deste estudo foi a limitação de recursos computacionais, especialmente a falta de GPUs, que restringiu a capacidade de explorar modelos mais complexos. Além disso, a sensibilidade dos dados tratados impôs restrições ao uso de ferramentas externas.

Como trabalho futuro, a aplicação de modelos mais avançados, como Transformers, poderá ser explorada para melhorar os resultados, o que demandaria a criação de um ambiente seguro e com recursos de GPU. Pretende-se também, realizar a classificação multiclases dos movimentos do gênero "Ciência" que o Promotor de Justiça deve registrar no sistema IDEA.

Referências

- [1] Jairo Luciano Dias Alves. Redes neurais recorrentes aplicadas à classificação de fake news em língua portuguesa. Master's thesis, Universidade Federal Fluminense, 2020. URL <https://app.uff.br/riuff/handle/1/13195>.
- [2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.
- [3] H. M. Caseli and M. G. V. Nunes, editors. *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. BPLN, 2 edition, 2024. ISBN 978-65-00-95750-1. URL <https://brasileiraspln.com/livro-pln/2a-edicao/>.
- [4] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.
- [5] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.
- [6] Ernanny Figueiredo. Classificação de sentimentos em textos de e-commerce utilizando redes neurais artificiais. Master's thesis, Universidade Estadual do Oeste do Paraná, 2022. URL https://tede.unioeste.br/bitstream/tede/5924/5/Ernanny_Figueiredo2022.pdf.
- [7] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In Yee Whye Teh and Mike Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL <https://proceedings.mlr.press/v9/glorot10a.html>.
- [8] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4, 2005. doi: 10.1109/IJCNN.2005.1556215.
- [9] Pierre Guillou. Ner bert large cased pt lener br. <https://huggingface.co/pierreguillou/ner-bert-large-cased-pt-lenerbr>, 2021. Acessado em 21 de fevereiro de 2024.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.
- [11] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.
- [12] M. I. Jordan. Serial order: A parallel distributed processing approach. Technical Report ICS Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.
- [13] David G. Kleinbaum and Mitchel Klein. *Logistic Regression: A Self-Learning Text*. Springer, 2010.

- [14] Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.
- [15] Ana Carolina Pereira Rocha. Mineração de textos para classificação de processos judiciais trabalhistas. Master's thesis, Universidade de Brasília, 2019. URL <http://www.realp.unb.br/jspui/handle/10482/37933>.
- [16] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.
- [17] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

**CENTRO UNIVERSITÁRIO SENAI CIMATEC
ESPECIALIZAÇÃO EM DATA SCIENCE & ANALYTICS****ATA DE APRESENTAÇÃO DE PROJETO FINAL DE CURSO**

Ata de apresentação do Projeto Final de Curso “**Classificação do teor das intimações recebidas do sistema Processo Judicial Eletrônico: uma análise comparativa dos algoritmos de Redes Neurais Recorrentes**”, submetido pelo aluno **Mônica Silva Rivas** como parte dos requisitos para obtenção do Certificado de **Especialista em Data Science & Analytics** pelo Centro Universitário SENAI CIMATEC, às 14h30 do dia 09 de agosto de 2024. Reuniu-se remotamente pela plataforma Microsoft Teams, a Banca Examinadora designada pelo Prof. Dr. Márcio Freire Cruz - Orientador, constituída pelo Prof. Dr. Márcio Freire Cruz e pelo Prof. Msc. Braian Varjão Gama Bispo. A coordenação deu início aos trabalhos com as devidas orientações, e a exposição foi realizada pelo estudante dentro do prazo de tempo estabelecido. Ao final da apresentação a banca reuniu-se atribuindo a seguinte nota: 9,6 (Nove pontos e seis décimos).

A banca de avaliadores decidiu pela:**(X) Aprovação do trabalho**

Caberá ao aluno apresentar em no máximo em 30 (trinta) dias a contar da data de assinatura desta Ata, uma cópia do trabalho em PDF, constando as considerações pontuadas pela banca. A Ata de Apresentação do Projeto Final de Curso deve ser digitalizada e inserida na última página do artigo.

() Reprovação do trabalho

O aluno terá que se matricular novamente no TCC – Trabalho de Conclusão de Curso e ser submetido a uma banca avaliadora no semestre seguinte.

As ações consequentes ao status de Aprovação deverão obedecer ao prazo proposto acima sob pena do parecer final ser modificado para o status de Reprovado automaticamente e sem possibilidade de recurso.

Para constar, lavrou-se a presente ata que vai assinada por todos os membros da Banca. Por estarem cientes de suas obrigações estão de acordo com os termos desse documento:

Salvador, 09 de agosto de 2024

Assinado eletronicamente por:
Marcio Freire
CPF: ***.235.375-**
Data: 16/09/2024 16:55:01 -03:00

Electronically signed by:
Braian Varjão Gama Bispo
CPF: ***.291.145-**
Date: 9/18/2024 6:26:03 PM -03:00

Márcio Freire Cruz
Professor Orientador

Braian Varjão Gama Bispo
Membro da banca

Assinado eletronicamente por:
Patrícia Freitas Tourinho
CPF: ***.733.265-**
Data: 16/09/2024 16:33:13 -03:00

Patrícia Freitas Tourinho
Coordenadora do Pós-Graduação Lato Sensu