# Processamento de Linguagem Natural em Processos de Tomada de Contas Especial

Sabrina dos Passos Tortelli
sabrina.tortelli@edu.univali.br
Universidade do Vale do Itajaí – UNIVALI
São José, SC, Brasil

Alessandro Mueller
alessandro@univali.br
Universidade do Vale do Itajaí – UNIVALI
São José, SC, Brasil

## ABSTRACT

In recent years, due to the significant volume of produced text documents, challenges have arisen in the search and analysis of content, necessitating the development of techniques for the extraction of useful information. In the field of Law, where the majority of information is in legal texts, information extraction has become crucial for discovering knowledge in unstructured data. Named entity recognition, driven by the advancement of deep learning models, stands out as the main technique for this task. This project aimed to explore the possibility of expanding the number of explanatory variables beyond those available on the institutional website of Tribunal de Contas da União, using natural language processing techniques in the Special Accounting processes. The development of the proposal included web scraping for data collection, preprocessing of pieces, entity annotation, fine-tuning pre-trained models in the legal domain and named entity recognition task, in addition to extracting entities. From the selected texts, 388,201 records (tokens and/or phrases) were extracted, with 286,781 and 101,420 records from the Instruction and Judgment pieces, respectively, confirming the research hypothesis and demonstrating the feasibility of expanding variables using natural language processing.

## KEYWORDS

Natural Language Processing, Special Accounting Processes, Transfer Learning

## 1 INTRODUCÃO

O aumento na produção de documentos de texto fez surgir novos desafios em relação à busca e análise de seu conteúdo, tendo em vista que a leitura manual por seres humanos não é suficiente para processar rapidamente uma quantidade substancial de documentos. Assim, torna-se cada vez mais indispensável empregar técnicas automatizadas no processamento e recuperação de informações [1]. Essas técnicas de processamento estão contidas em uma subárea da inteligência artificial, denominada processamento de linguagem natural (PLN), que envolve o desenvolvimento de algoritmos e técnicas para permitir que os computadores compreendam, interpretem e gerem texto e discurso de forma semelhante a humanos. Algumas aplicações práticas do PLN incluem os chatbots e assistentes virtuais, a tradução automática, a análise de mídia social, a recuperação de informações (RI) e o processamento de texto, esse envolvendo a análise de sentimento, a classificação de texto, a sumarização automática e a extração de informações (EI) [2].

A extração de informação tem como objetivo identificar, estruturar e armazenar informações relevantes de um documento ou conjunto de documentos, sem, contudo, interpretar todo o documento, mas, sim, extraindo as partes relevantes e armazenando-as de maneira estruturada [3]. Uma das tarefas utilizadas para realizar a extração é conhecida como reconhecimento de entidades nomeadas (REN), que visa buscar entidades fundamentais em um texto e classificar em categorias predefinidas, como nomes de pessoas, locais, datas, números, nomes de empresas, entre outros. Um modelo REN recebe como entrada uma sequência de *tokens* que representam palavras e sinais de pontuação, e sua saída é uma sequência de *tags* de tipos pré-determinados que identificam as entidades encontradas no texto [3] [4].

Embora o PLN tenha avançado consideravelmente para atender às crescentes demandas, enfrenta desafios particulares em domínios específicos, como o Direito, devido às suas expressões peculiares. Um estudo realizado por Garcia [5], que analisou a jurisprudência do Tribunal de Contas da União (TCU), identificou limitações em um modelo preditivo com 70% de acurácia. O estudo utilizou dados estruturados de uma planilha disponibilizada na seção de jurisprudência do TCU. Diante disso, o autor propôs a incorporação de técnicas de PLN nos documentos completos do processo, visando obter variáveis explicativas adicionais às estruturadas presentes no site oficial. Diante da lacuna identificada no estudo de Garcia [5], este trabalho buscou verificar se é possível ampliar a quantidade de variáveis explicativas, além daquelas dispostas no site institucional do TCU, utilizando técnicas de processamento de linguagem natural nas peças de processos de Tomada de Contas Especial (TCE).

A motivação de usar processos TCE neste projeto se justifica pelo fato de serem processos públicos, com suas peças disponíveis *on-line*, apresentando potencial para produzir resultados de interesse público no futuro. A TCE, enquanto processo administrativo formalizado, é considerada uma medida extraordinária e tem como objetivo investigar a responsabilidade por danos à administração pública federal. Nesse contexto, são apurados os fatos, quantificado o dano, identificados os responsáveis e busca-se obter o ressarcimento correspondente [6].

Este trabalho está estruturado da seguinte forma: na seção 2 serão apresentados alguns trabalhos relacionados, a fim de contextualizar o projeto em relação ao estado da arte da área e na seção 3, mostra-se uma visão geral e global do funcionamento do sistema. Já na seção 4, são apresentados os procedimentos adotados para a construção e implementação do projeto, com a exposição das validações e dos resultados alcançados. Na seção 5, apresentam-se as conclusões e trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

A busca de trabalhos relacionados ao tema de interesse do estudo foi realizada com o fito de reconhecer o atual estado da arte e a compreensão sobre o assunto e técnicas utilizadas na pesquisa proposta. Três critérios principais para a busca foram adotados: ser

atual (produzido nos últimos cinco anos), abranger a extração de informações e ter necessariamente como tema principal o reconhecimento de entidades em textos legais. A pesquisa foi conduzida no portal de periódicos da Capes, o qual busca artigos em várias bases de dados.

A revisão destaca que, nos últimos cinco anos, as técnicas mais utilizadas e eficazes para o processamento de textos em domínio específico envolvem o aprendizado profundo na tarefa de REN. A maioria dos estudos apresenta metodologias para extração de entidades, comparando métricas obtidas com outras pesquisas relacionadas. Embora muitos estudos se baseiem em textos em inglês, há uma tendência crescente de pesquisas em outras línguas. No que se refere à língua portuguesa, existem diversos estudos de aprendizagem profunda que utilizam dados de domínio não específico e *corpora* variados, porém no domínio jurídico, a disponibilidade é limitada. Esta seção destaca dois estudos brasileiros de REN em textos jurídicos e um estudo chinês que combina ontologia e técnicas de extração de informações com resultados notáveis.

O trabalho de Heck et al. [4] - trabalho 3.1 - visou extrair e reconhecer entidades nomeadas em textos jurídicos em português brasileiro. Utilizando modelos de redes neurais LSTM (*Long Short-Term Memory*) e o pré-treinado BERT (*Bidirectional Encoder Representations from Transformers*), especialmente o BERTimbau do trabalho de Souza et al. [7], o estudo treinou um modelo para a tarefa REN, identificando entidades como Pessoa, Tempo, Local, Organização, Legislação e Jurisprudência. O conjunto de dados usado para treinamento foi o LeNER-Br, composto por 70 textos jurídicos manualmente anotados, que foi comparado com o modelo *baseline* apresentado no artigo de Luz de Araújo (2018, apud Heck et al. [4]). O modelo mais eficaz foi a rede BERT-CRF *large* (BERT com uma camada de campos aleatórios condicionais), superando o *baseline* com uma precisão de 90,16%, *recall* de 91,86% e *F1-score* de 91,00%.

Fernandes et al. [8] - trabalho 3.2 - propuseram uma metodologia para extrair valor das decisões dos tribunais brasileiros, envolvendo a anotação de decisões judiciais, criação de modelos de aprendizado profundo e visualização das informações extraídas. Três *corpora* foram construídos a partir de dados públicos do Tribunal de Justiça do Estado do Rio de Janeiro, anotados com entidades relacionadas a reivindicações comuns em processos envolvendo seguradoras, categorias jurídicas de decisões de primeira instância e categorias modificadas ou mantidas pela segunda instância. Cinco modelos, incluindo Bi-LSTM (redes de memória bidirecional de longo prazo), CRF, Bi-LSTM-CRF (rede bidirecional de memória de curto prazo longo e campos aleatórios condicionais), Bi-LSTM-CE (representação distribuída de palavras em incorporações de palavras e da representação de nível de caractere de palavras em incorporações de caracteres) e Bi-LSTM-CE-CRF (representação distribuída de palavras em incorporações de palavras e da representação de nível de caractere de palavras em incorporações de caracteres e campos aleatórios condicionais), foram aplicados aos *corpora* para a extração de informações. Os resultados foram promissores, permitindo a criação de visualizações interativas das informações agregadas extraídas, facilitando o acompanhamento da jurisprudência do Tribunal de Justiça do Estado do Rio de Janeiro por advogados, juízes e juristas.

Ren et al. [9] - trabalho 3.3 - apresentaram um método que combina ontologia e aprendizado profundo para extrair fatos jurídicos de textos chineses. Utilizando o conjunto de dados CAIL2021_IE, composto por 500 textos legais chineses sobre roubo, o estudo propõe uma arquitetura dividida em modelagem de conhecimento, pré-processamento, classificação de parágrafos e extração de fatos. O módulo de modelagem de conhecimento emprega a ontologia *Chinese Legal Text Ontology* (CLTO) para guiar a classificação de parágrafos e extração de fatos. O pré-processamento inclui limpeza de dados, verificação de parágrafos e normalização do texto. A classificação de parágrafos utiliza um método baseado em regras, enquanto a extração de fatos inclui extratores baseados em regras e aprendizado profundo, este último incorporando o modelo pré-treinado BERT. O extrator baseado em regras é preciso para fatos básicos, mas limitado para casos complexos, enquanto o extrator de aprendizado profundo alcança uma precisão média de 90,41%, *recall* de 92,49% e *F1-score* de 91,43%. Os resultados indicam que o método proposto é eficaz na extração de fatos jurídicos de textos chineses.

| **Critérios** | **Trabalho 3.1** | **Trabalho 3.2** | **Trabalho 3.3** | **Este Trabalho** |
|---|---|---|---|---|
| **Contexto** | Legislações de diversas cortes | Processos envolvendo seguradoras | Processos sobre roubos | Processos de Tomada de Contas Especial |
| ***Corpora*** | Utilizada de outro estudo | Construída | Utilizada de outro estudo | Construída |
| **Idioma** | Português | Português | Chinês | Português |
| **Tarefa** | REN | REN | REN | REN |
| **Tipo de Aprendizado** | Aprendizado Profundo | Aprendizado Profundo | Aprendizado Profundo e Baseado em Regras | Aprendizado Profundo |
| **Modelo de Aprendizado** | LSTM e BERT (BERTimbau) | CRF, Bi-LSTM, Bi-LSTM-CRF, Bi-LSTM-CE e Bi-LSTM-CE-CRF | BERT, Bi-LSTM e CRF | BERT (BERTimbau) |
| **Métricas** | Precisão, *Recall* e *F1-score* geral e por entidade | Precisão, *Recall* e *F1-score* geral | Precisão, *Recall* e *F1-score* geral e por entidade | Precisão, *Recall* e *F1-score* geral e por entidade |

**Tabela 1: Análise comparativa dos trabalhos relacionados**

As características utilizadas na avaliação e comparação dos estudos foram definidas com base nos objetivos deste trabalho. Os critérios e a comparação dos estudos são apresentados na Tabela 1. Observa-se que dois estudos (trabalho 3.1 e o 3.3) sobre a utilização de PLN em textos jurídicos empregam bases anotadas prontas para treinar ou ajustar o modelo, principalmente quando o objetivo que se quer atingir é a comparação de modelos. Porém, realizar a anotação de entidades, gerando um *corpus* específico para a base é melhor para a obtenção de métricas robustas na tarefa de reconhecimento

de entidades nomeadas, mas, evidentemente, essa tarefa de anotação requer tempo e trabalho. Uma semelhança entre o primeiro e o terceiro estudo é o uso de modelos pré-treinados para realizar a tarefa REN e em todos os casos, também é empregado o uso de redes de memória bidirecional de longo prazo, que podem ser combinadas ou não com CRF.

## 3 VISÃO GERAL DO SISTEMA

Na visão geral do sistema (Figura 1) observa-se que sua operação inicia através da tarefa de *web scraping*, que realiza uma busca no site do TCU com o objetivo de obter dados de processos TCE, capturando dados não estruturados (peças dos processos). Uma vez coletados, esses dados passam por uma etapa de pré-processamento, com a limpeza e normalização, para, em seguida, ser realizada a anotação das entidades em parte dessas peças.

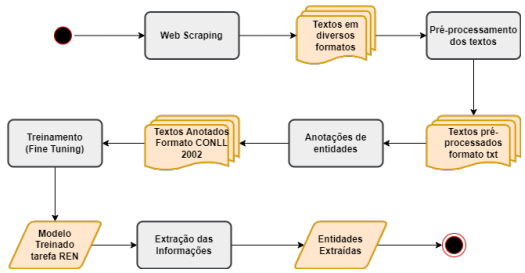


**Figura 1: Visão Geral do Sistema**

Com a anotação das entidades realizada nos textos, pode ser feito o ajuste fino de um modelo pré-treinado selecionado. Na etapa seguinte, de extração de informações, o modelo pré-treinado, previamente ajustado na etapa anterior, é empregado para realizar o reconhecimento das entidades presentes no restante dos textos jurídicos, permitindo obter um conjunto estruturado e organizado.

## 4 DESENVOLVIMENTO E RESULTADOS

Nesta seção são apresentadas cada etapa da construção e implementação do sistema, além da exposição dos resultados alcançados, possibilitando a compreensão do impacto e da eficácia do projeto.

### 4.1 *Web Scraping*

Inicialmente, realizou-se uma análise manual no site do TCU para verificar a disponibilidade de informações sobre os processos de Tomada de Contas Especial. Utilizando uma ferramenta de pesquisa do site, foram baixados dados básicos sobre os processos, filtrando-os para identificar os encerrados. A coleta foi realizada durante o período de férias do TCU, em janeiro de 2023, no período noturno, a uma taxa de rastreamento de 1 solicitação a cada 10 segundos, para evitar impactos na rede do Tribunal.

Por meio de *web scraping*, implementado em Python, utilizando as bibliotecas Selenium Chrome Webdriver e Pandas, foram geradas URLs para cada processo, permitindo a captura de informações específicas. Foram capturadas, além das peças, informações estruturadas disponíveis na página, como número, ano, status e tipo do processo, unidade técnica responsável, unidade responsável por agir, relator, responsáveis, assunto, unidades jurisdicionadas, *links* das peças e movimentações para trabalhos futuros.

Diante dos dados coletados, foi realizada uma seleção para delineação do escopo a ser trabalhado. Sendo assim, utilizou-se dados relativos aos anos de 2016 a 2022, com um total de 4713 processos encerrados e um total de 8180 peças, sendo 3652 Instruções e 4528 Acórdãos. Após a seleção, os documentos foram nomeados de acordo com o número do processo e o tipo de peça, visando facilitar sua posterior localização e leitura.

### 4.2 Pré-processamento dos textos

No pré-processamento dos textos todas as peças foram padronizadas em uma mesma extensão de arquivos, sendo escolhida a extensão PDF, por ter uma quantidade maior de peças baixadas nessa mesma extensão. O pré-processamento dos textos foi realizado utilizando a linguagem Python, onde as peças passaram por uma limpeza, que eliminou ruídos e informações desnecessárias, como cabeçalhos, rodapés, números de página e palavras-chave específicas do domínio. Em seguida, foi realizada a normalização, que visou padronizar o formato dos textos, incluindo a remoção de caracteres especiais, acentos, espaços em excesso e quebras de linha. Por fim, os textos foram transformados e salvos no formato TXT.

### 4.3 Anotação de entidades

Nesta etapa foi realizada uma seleção aleatória para a base de anotação de entidades, resultando em um total de 80 documentos (40 Instruções e 40 Acórdãos). Essa quantidade foi escolhida, pois no estudo de trabalhos relacionados observou-se que a quantidade de documentos usados para a anotação não é muito expressiva, sendo, muitas vezes, utilizada bases menores que este trabalho.

A autora, embora sem especialização na área jurídica, desempenhou a tarefa de anotação das entidades. Para esse propósito, utilizou a ferramenta INCEpTION [10], plataforma de anotação de fácil instalação e modular, adaptável a uma variedade de domínios de pesquisa e disponibilizada como software de código aberto.

Ao final da anotação, as entidades foram exportadas no padrão IOB2, o qual é o padrão adotado no modelo *deep learning* selecionado. Nesse formato, a marcação 'B' é utilizada para indicar o início de uma entidade, 'I' para palavras subsequentes dentro da mesma entidade, e 'O' para palavras que não constituem uma entidade. O arquivo exportado tem o formato CoNLL 2002, em que cada palavra é apresentada em uma linha distinta, seguido de um espaço e sua anotação correspondente. Este método de organização de dados é exemplificado na Figura 2.

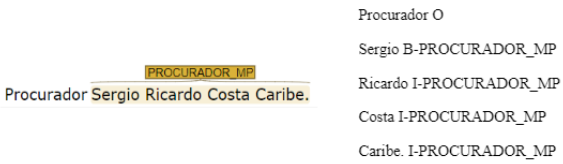


**Figura 2: Exemplo de exportação do padrão IOB2 (à direita) da frase (à esquerda)**

Foram definidas 17 entidades, cada tipo é explicado a seguir, juntamente com o quantitativo de *tokens* por *tags* que foram anotadas:

- AREA (425): Área afetada (saúde, educação, infraestrutura, assistência social, entre outras);
- JULGAMENTO_CONTAS (135): Como as contas são julgadas (regulares, regulares com ressalva ou irregulares);
- MINISTRO (1281): Nome dos ministros relatores citados no processo ou participantes das sessões de decisão;
- MOTIVO_TCE (357): Motivo de instauração da TCE (motivo da irregularidade como omissão no dever de prestar contas, dano ao erário, desvio de dinheiro ou outro);
- PROCESSO (58): Número do processo TCE;
- PROCESSO_VINCULADO (1): Número do processo vinculado ao processo principal;
- PROCURADOR_MP (256): Procurador do Ministério Público que participa das sessões do pleno;
- PROPOSTA_ENCAMINHAMENTO (177): Proposta de encaminhamento que o técnico ou os ministros fazem de acordo com o andamento processual;
- RECOMENDACAO_PLENARIO (176): Recomendação feita ao plenário (arquivar, prosseguir com o processo ou outra);
- RESPONSAVEL (1988): Nome do responsável que responde pelo processo;
- RESPONSAVEL_CARGO (325): Cargo do responsável no órgão público ou empresa que recebe o dinheiro público;
- SANSAO (231): Sansão que o responsável recebe caso for considerado culpado;
- TIPO_INSTRUCAO (196): Tipo de Instrução a que o documento se refere (mérito, arquivamento, citação);
- UN_INSTAURADORA (297): Unidade ou órgão público que instaurou o processo de TCE (quem forneceu o dinheiro público e sofreu o prejuízo);
- UN_JURISDICIONADA (363): Unidade ou órgão público que recebeu o dinheiro;
- VALOR_DANO_ATUALIZADO (99): Valor do dano atualizado com juros de mora;
- VALOR_DANO_INDICIO (3): Valor inicial do dano, quando o processo é instaurado;

## 4.4 Treinamento (*Fine Tuning*)

Utilizando a técnica de Transferência de Aprendizagem foi realizada a tarefa de treinamento *Fine Tuning* (ou, ajuste fino) de um modelo pré-treinado selecionado. O modelo escolhido para essa tarefa foi o BERTimbau, treinado com *corpus* em português e com base nas arquiteturas BERT, CRF e Bi-LSTM para reconhecimento de entidades nomeadas. A biblioteca Hugging Face[1] disponibiliza os modelos *large* e *base*, sendo o primeiro treinado com 335 milhões e o segundo com 110 milhões de parâmetros, o que permitiu a esses modelos capturar informações linguísticas gerais.

Para a obtenção dos melhores resultados, foram realizadas três etapas (Figura 3). Na primeira foi realizada um *Fine Tuning* para obtenção de um modelo de linguagem especializado no domínio jurídico dos processos TCEs, na segunda, um *Fine Tuning* realizando a tarefa REN e, na terceira, um novo *Fine Tuning* da tarefa REN com uma base de anotações ampliada.

[1]https://huggingface.co/

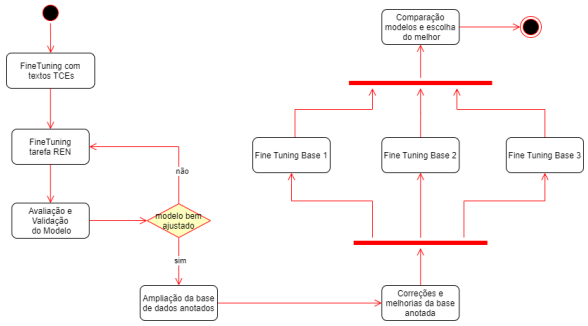


**Figura 3: Diagrama de Atividades do treinamento**

*4.4.1 Fine Tuning para obter um modelo de linguagem especializado no domínio jurídico dos processos TCEs.* Dado o estilo de redação e o vocabulário específico do domínio jurídico, o ajuste fino do modelo BERTimbau foi feito com os textos originais de processos TCEs, a fim de obter um modelo de linguagem natural e um tokenizador especializados para o domínio jurídico alvo do trabalho. O treinamento foi realizado utilizando IDE VsCode, ambiente virtual Anaconda, Python versão 3.11.5 com notebook Jupyter e bibliotecas necessárias (Datasets e Transformers da plataforma Hugging Face, NLKT, Pandas, Scikit-learn) rodando dentro do sistema operacional Linux Ubuntu para Windows-WSL (Subsistema do Windows para Linux). Além disso foi necessária a instalação da plataforma CUDA (*Compute Unified Device Architecture*) para acesso à placa de vídeo NVIDIA no Ubuntu. Foi possível realizar o treinamento apenas com o modelo *base* devido às limitações da VRAM da máquina local (configurações mostradas na Figura 4).


**Figura 4: Configuração da máquina local**

Esse treinamento foi realizado com 1000 arquivos, os quais passaram por um pré-processamento dos textos, utilizando o tokenizador do modelo pré-treinado para ajustar à entrada de dados que o modelo de treinamento aceita. O conjunto de dados foi dividido em treinamento (80%) e validação (20%), com 73900 e 18475 textos em cada *dataset*, respectivamente. Essa divisão dos dados em 80/20 foi utilizada por ser uma prática comum na comunidade de aprendizado de máquina e em trabalhos relacionados nos modelos de processamento de linguagem natural, a qual depende também do tamanho do conjunto de dados e da complexidade do modelo.

A tarefa de treinamento utilizada foi a de modelagem de linguagem mascarada (MLM), em que substitui-se aleatoriamente alguns *tokens* por [MASK] e o modelo realiza a previsão da palavra mascarada.

Na Tabela 2 pode-se visualizar o valor de inicialização de cada hiperparâmetro do modelo de treinamento e sua influência no treinamento.

**per_device_batch_size** = 8 # *Especifica o número de exemplos a serem processados por dispositivo GPU por etapa de treinamento*
**gradient_accumulation_steps** = 1 # *Define o número de etapas de treinamento antes da otimização*
**learning_rate** = 2e-5 # *Taxa de aprendizado inicial para o otimizador AdamW*
**num_train_epochs** = 5 # *Número total de épocas de treinamento*
**weight_decay** = 0.01 # *Parâmetro de decaimento de peso para regularização L2, evitando sobreajuste ao penalizar pesos maiores*
**save_total_limit** = 2 # *Número máximo de arquivos de modelo a serem salvos*
**logging_steps** = 100 # *Frequência de registro das métricas de treinamento*
**eval_steps** = logging_steps # *Frequência com que o modelo é avaliado em termos de etapas de treinamento*
**evaluation_strategy** = 'steps' # *A estratégia de avaliação define quando o modelo deve ser avaliado, nesse caso, o modelo é avaliado a cada eval_steps*
**logging_strategy** = 'steps' # *A estratégia de registro define quando os registros de treinamento são criados, nesse caso, o registro acontece a cada logging_steps*
**save_strategy** = 'steps' # *A estratégia de salvamento define quando os checkpoints do modelo são salvos, nesse caso, um checkpoint é salvo a cada save_steps*
**save_steps** = logging_steps # *Frequência com que os checkpoints do modelo são salvos, baseada no número de etapas de treinamento*
**load_best_model_at_end** = True # *O modelo com a melhor avaliação será carregado no final do treinamento*
**fp16** = True # *Habilita o uso de ponto flutuante de 16 bits (FP16) para treinamento, acelerando-o e reduzindo o uso de memória*
**metric_for_best_model** = 'eval_loss' # *Obter o melhor modelo de acordo com a métrica*
**greater_is_better** = False

**Tabela 2: Parâmetros utilizados no treinamento**

| Step | Training Loss | Validation Loss |
|---|---|---|
| 100 | 1.8829 | 1.5158 |
| 600 | 1.3422 | 1.2273 |
| 1200 | 1.2010 | 1.1429 |

**Tabela 3: Resultados do treinamento do modelo em steps**

O processo de treinamento finalizou no *step* 1200, devido à técnica *Early Stopping* (Parada Precoce) utilizada, na qual o treinamento para assim que o erro no conjunto de validação começa a subir. A cada 100 *steps* o modelo avalia o erro no conjunto de validação, sendo cada atualização ocorrendo após o processamento de um lote de 1000 exemplos (*batch size* = 1000). A quantidade de *steps* depende principalmente dos parâmetros relacionados ao tamanho do conjunto de treinamento e do tamanho do lote. Quanto maior o lote, é necessário mais aceleração computacional para o paralelismo da GPU (Unidade de Processamento Gráfico), e a generalização torna-se mais deficiente. Por outro lado, quanto menor o lote, o aprendizado é iniciado antes de o modelo ver todos os dados, tornando-o mais rápido e permitindo uma generalização mais eficaz. Ainda, quanto maior o conjunto de treinamento, mais *steps* são necessários para passar por todos os exemplos.

Além desses parâmetros, o otimizador utilizado (AdamW) também pode influenciar o comportamento dos *steps* durante o treinamento. Na Tabela 3 são apresentados os valores das perdas do

**gradient_accumulation_steps** = 2
**save_total_limit** = 3
**logging_steps** = 290
**metric_for_best_model** = 'eval_f1'
**greater_is_better** = True

**Tabela 4: Hiperparâmetros de treinamento para a tarefa REN**

treinamento e da validação do primeiro, último e *step* intermediário. O resumo do treinamento mostra que a função de perda, representada pelos valores de *Training Loss* e *Validation Loss*, está diminuindo ao longo do treinamento, significando que o modelo está aprendendo a partir dos dados e melhorando sua capacidade de generalização. Há uma redução substancial dos valores entre o *step* 100 e 1200, indicando uma melhoria contínua na capacidade do modelo de se ajustar aos dados. Além disso, a diferença entre *Training Loss* e *Validation Loss* é relativamente pequena, sinalizando que o modelo está generalizando bem para dados não vistos (conjunto de validação). Por esses motivos, a eficiência do treinamento pode ser considerada positiva.

*4.4.2 Fine Tuning da tarefa REN.* Nesta etapa foi realizada a especialização do modelo para a tarefa de reconhecimento de entidade nomeada no domínio jurídico dos processos TCEs, em que o modelo deve prever um rótulo para cada *token*, classificando as entidades encontradas no texto. Utilizou-se o modelo treinado na etapa anterior e os 80 arquivos anotados na fase de anotação de entidades. Foi necessário realizar um pré-processamento para ajustar e preparar os dados anotados no formato da entrada de dados do modelo. Os dados foram distribuídos em dois *datasets* de treinamento e validação, distribuídos com 6700 e 1676 textos em cada *dataset*, respectivamente.

As métricas mais utilizadas para quantificar o desempenho desse tipo de modelo incluem a precisão, o *recall* e a *F1-score*. Então foi utilizada a métrica sequencial (comumente usada para avaliar resultados no conjunto de dados CoNLL) por meio da biblioteca Datasets do Hugging Face. Foram calculadas as métricas gerais do modelo e para categoria de *tags*. Além disso, os hiperparâmetros da Tabela 4 foram alterados em relação ao primeiro treinamento. Outros parâmetros foram testados para verificar se seria possível alcançar um melhor desempenho, conforme a lista abaixo:

- per_device_size: iniciou com 4, alterado para 8 (melhor) e por fim 16 (não chegou ao final do treinamento).
- learning_rate: iniciou com 2e-5, alterado para 1e-4 (melhor) e por fim 3e-4 (métricas apresentaram oscilação, indicando que os hiperparâmetros estavam mal ajustados).
- num_train_epochs: iniciou com 3, alterado para 5 (melhor) e por fim 8.

Além disso, foram avaliados o desempenho de quatro modelos alterando seus hiperparâmetros para validar e escolher o melhor modelo:

(1) Modelo 1 - *Batch Size* de 670 e 3 épocas
(2) Modelo 2 - *Batch Size* de 1340 e 3 épocas
(3) Modelo 3 - *Batch Size* de 670 e 5 épocas
(4) Modelo 4 - *Batch Size* de 670 e 5 épocas - treinado com o modelo BERTimbau *base*, sem ajuste fino dos textos TCEs.

| Métrica | Modelo 1 | Modelo 2 | Modelo 3 | Modelo 4 |
|---|---|---|---|---|
| **Perda** | 0.1026 | 0.1086 | 0.1142 | 0.1082 |
| **Precisão** | 0.7099 | 0.7004 | 0.7666 | 0.7503 |
| **Recall** | 0.7746 | 0.7660 | 0.8251 | 0.8288 |
| **F1-score** | 0.7409 | 0.7318 | **0.7948** | 0.7876 |
| **Acurácia** | 0.9745 | 0.9740 | 0.9774 | 0.9763 |

**Tabela 5: Resultados das métricas de validação para cada modelo**

A Tabela 5 exibe um resumo dos resultados dos modelos avaliados (valores com até quatro casas após a vírgula).

O *F1-score* é uma métrica muito utilizada para tarefas de REN, pois equilibra a precisão e o *recall*. Já a acurácia que mede a proporção de exemplos classificados corretamente, não pode ser considerada no estudo, pois pode trazer resultados enganosos, pelo fato de haver um desequilíbrio de classes. Então, de acordo com quatro modelos apresentados, foi escolhido o Modelo 3 que obteve o maior *F1-score* (0.7948). Pode-se verificar, também, que o Modelo 4, o qual não utilizou o modelo treinado com os dados jurídicos de TCEs, teve um resultado bem próximo do Modelo 3, podendo indicar que o treinamento com a base específica não seja tão necessário quanto se acreditava.

*4.4.3 Fine Tuning da tarefa REN com a base de anotações ampliada.* A anotação de entidades em textos é uma tarefa que requer muita atenção, tempo e conhecimento do domínio para realizar anotações adequadas a fim de obter um modelo de treinamento satisfatório. Os modelos treinados com a base de 80 arquivos anotados não tiveram um desempenho ótimo como pode-se observar nos resultados apresentados, além disso não existe na literatura nenhum trabalho de anotação em textos de TCEs e seu processamento de linguagem natural para poder fazer uma comparação dos modelos produzidos neste trabalho com outros.

Por esses motivos, foi introduzida a técnica *Human in the Loop* (ou humano no ciclo), a qual foi adaptada para esse trabalho, já que a técnica original consiste em um *pipeline*, em que há várias iterações e em cada iteração do modelo o humano faz correções ou acrescenta melhorias, gerando um *feedback*, que posteriormente é utilizado para refinar o modelo (Yu *et al.*, 2015 apud Araújo [11]). A adaptação dessa técnica teve o objetivo de minimizar o esforço de anotação manual, em que é feita uma extração de dados com o modelo selecionado da etapa anterior para realizar uma ampliação da base de dados. Com isso, há um aumento da velocidade do trabalho de anotação manual, pois são realizadas apenas melhorias e correções na base anotada de forma automatizada.

Assim, foi feita uma nova seleção aleatória de 40 Instruções e 40 Acórdãos a fim de extrair as entidades dos textos utilizado o *pipeline* com o classificador treinado na etapa anterior. As entidades da saída do *pipeline* são apresentadas na Figura 5. Ainda foi necessário realizar um pós-processamento para retornar com os arquivos anotados à ferramenta INCEpTION para melhorias. Foi verificado que na primeira anotação, várias entidades não foram reconhecidas até a palavra final, pois não tinham sido considerados símbolos após a palavra ou qualquer outra letra e número que tivesse unida à entidade, porém na exportação dos textos a ferramenta considera o espaço vazio entre as palavras para realizar a tokenização. Além disso, na anotação inicial, uma palavra foi anotada com uma ou mais *tags* e não sendo considerado a segunda anotação. Também houve problemas com a falta de reconhecimento de números dos processos, processos vinculados e valores de dano, então buscou-se realizar a correção colocando a palavra TC como o início da entidade junto com os números de processos e a palavra R$ para os valores. Todos esses problemas foram corrigidos na base original e na nova base.

| Tags | Base 1 | Base 2 | Base 3 |
|---|---|---|---|
| AREA | 283 | 515 | 567 |
| JULGAMENTO_CONTAS | 135 | 418 | 352 |
| MINISTRO | 1281 | 3189 | 3228 |
| MOTIVO_TCE | 317 | 532 | 647 |
| PROCESSO | 177 | 468 | 428 |
| PROCESSO_VINCULADO | 1 | 1 | 1 |
| PROCURADOR_MP | 358 | 960 | 914 |
| PROPOSTA_ENCAMINHAMENTO | 240 | 297 | 377 |
| RECOMENDACAO_PLENARIO | 176 | 295 | 321 |
| RESPONSAVEL | 1450 | 2703 | 2904 |
| RESPONSAVEL_CARGO | 640 | 1007 | 1200 |
| SANSAO | 142 | 426 | 463 |
| TIPO_INSTRUCAO | 123 | 353 | 410 |
| UN_INSTAURADORA | 298 | 908 | 776 |
| UN_JURISDICIONADA | 548 | 815 | 989 |
| VALOR_DANO_ATUALIZADO | 90 | 123 | 159 |
| VALOR_DANO_INDICIO | 59 | 50 | 86 |

**Tabela 6: Quantidade de *tokens* por *tags* das bases anotadas**


```
[{'entity': 'B-RESPONSAVEL', 'score': 0.998569, 'index': 5, 'word': 'Am',
'start': 14, 'end': 16},
{'entity': 'B-RESPONSAVEL', 'score': 0.9956447, 'index': 6, 'word': '##pla',
'start': 16, 'end': 19},
{'entity': 'I-RESPONSAVEL', 'score': 0.9841859, 'index': 7, 'word': '##com',
'start': 19, 'end': 22},
{'entity': 'I-RESPONSAVEL', 'score': 0.99914885, 'index': 8, 'word': 'Ind',
'start': 23, 'end': 26},
{'entity': 'I-RESPONSAVEL', 'score': 0.99907863, 'index': 9, 'word': '##ustria',
'start': 26, 'end': 32}]
```


**Figura 5: Exemplo da saída do pipeline**

Ao final desta anotação, que foi mais rápida que toda a anotação inicial, os arquivos foram exportados, pré-processados e separados em três bases, para ser realizada comparações entre os modelos a serem gerados:

(1) Base com os 80 arquivos iniciais anotados manualmente revisados e corrigidos;
(2) Base com 80 arquivos anotados manualmente revisados e corrigidos e 80 arquivos anotados na etapa automatizada;
(3) Base com 80 arquivos anotados manualmente e 80 arquivos anotados na etapa automatizada, todos sendo revisados e corrigidos;

A intenção foi verificar o comportamento dos modelos em relação à quantidade de arquivos e à revisão e correção das anotações. Especificamente, pretendeu-se avaliar se ajustes nesses elementos poderiam otimizar a eficácia do modelo resultando em melhorias significativas nas métricas de desempenho do modelo.

Na Tabela 6 são apresentados os quantitativos de *tokens* por *tag* por base de dados anotadas.

| Métrica | Valor |
|---|---|
| **Step Global** | 4600 |
| **Perda** | 1.1541 |
| **Época** | 0.64 |

**Tabela 7: Resultado treinamento com 2000 arquivos**

| Métrica | Modelo 3 - T1 (Treino 1) | Modelo 1 - T2 (Treino 2) | Modelo 2 - T2 (Treino 2) | Modelo 3 - T2 (Treino 2) |
|---|---|---|---|---|
| **Perda** | 0.1142 | 0.0281 | 0.1086 | 0.0149 |
| **Precisão** | 0.7666 | 0.9704 | 0.7004 | 0.9832 |
| **Recall** | 0.8251 | 0.9732 | 0.7660 | 0.9745 |
| **F1-score** | 0.7948 | 0.9718 | 0.7318 | **0.9788** |
| **Acurácia** | 0.9774 | 0.9955 | 0.9740 | 0.9974 |

**Tabela 8: Resultados da validação de cada modelo**

Para tentar melhorar as métricas dos modelos também foi realizado um novo *Fine Tuning* para obtenção de um modelo de linguagem especializado no domínio jurídico dos processos TCEs com 2000 arquivos (Tabela 7), utilizando os mesmos parâmetros do treinamento do modelo com 1000 arquivos, com a quantidade de textos de 146056 e 36515 para treinamento e validação, respectivamente. Esse modelo com 2000 arquivos foi utilizado no novo treinamento da tarefa REN das três bases de dados. Em todo o pré-processamento, treinamento e validação foram utilizados os mesmos parâmetros do melhor modelo da tarefa REN escolhido anteriormente. A base com 160 arquivos ficou dividido com 14268 e 3567 textos por treinamento e validação, respectivamente.

Como pode-se observar na Tabela 8, o Modelo 3 - T2 foi o que trouxe as melhores métricas, principalmente de *F1-score*. Em relação ao Modelo 3 - T1, o Modelo 3 - T2 apresentou um ganho significativo com as correções e melhorias realizadas no *dataset*, além disso se aproximou muito das métricas do Modelo 1 - T2 que apresenta menos arquivos. Pode-se perceber, também, que o Modelo 2 - T2 apresentou as piores métricas dos três modelos e também em relação ao Modelo 3 - T1, o que pode significar que as primeiras anotações realizadas na base foram ruins e isso influenciou a extração utilizada no Modelo 2 - T2.

Essas mesmas conclusões podem ser observadas na Figura 6, onde apresenta-se o desempenho dos modelos por tipo de entidade, sendo que ao lado de cada barra é mostrada a quantidade de *tokens* de cada entidade, utilizada por cada modelo no *dataset* de validação. Apesar de o Modelo 1 - T2 apresentar menos *tokens* por entidade que os outros modelos, ele poderia ser utilizado para a extração final por ter sido muito próximo do Modelo 3 - T2. Para a extração das entidades na próxima etapa, o Modelo 3 - T2 foi escolhido por apresentar os melhores resultados na tarefa REN.

## 5 EXTRAÇÃO DAS ENTIDADES NA BASE COMPLETA

Depois de todas as tarefas de treinamento dos modelos, foi feita a extração das entidades em toda a base de arquivos. O *pipeline* desenvolvido para esta tarefa é muito semelhante ao feito para a extração

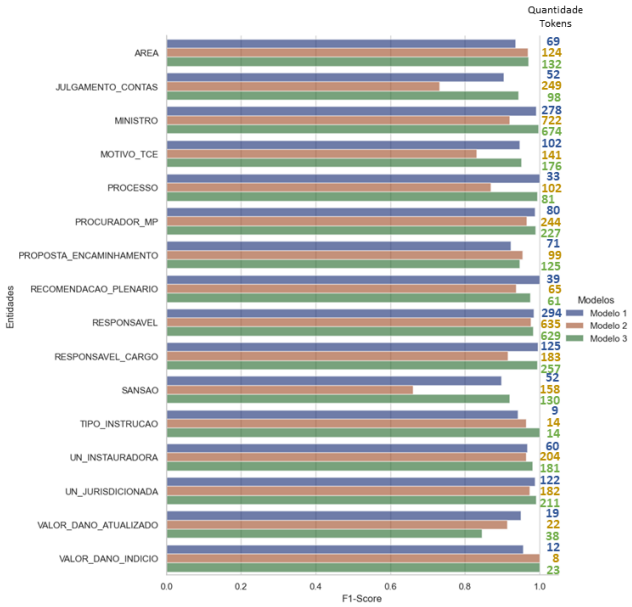


**Figura 6: Desempenho dos modelos por tipo de entidades**

| Tags | Total Registros |
|---|---|
| RESPONSAVEL | 96537 |
| MINISTRO | 64618 |
| MOTIVO_TCE | 58967 |
| RESPONSAVEL_CARGO | 34647 |
| UN_INSTAURADORA | 25411 |
| AREA | 23478 |
| SANSAO | 14925 |
| PROCURADOR_MP | 11515 |
| UN_JURISDICIONADA | 11347 |
| PROCESSO | 10577 |
| PROPOSTA_ENCAMINHAMENTO | 8838 |
| JULGAMENTO_CONTAS | 8074 |
| RECOMENDACAO_PLENARIO | 7133 |
| VALOR_DANO_INDICIO | 5444 |
| TIPO_INSTRUCAO | 3717 |
| VALOR_DANO_ATUALIZADO | 2973 |

**Tabela 9: Registros extraídos por *tags***

realizada anteriormente no aumento da base de anotada. Os resultados revelaram a extração de informações de 4612 processos dos 4713 selecionados. Essa discrepância pode ser atribuída à metodologia de seleção dos processos. Do conjunto de peças analisadas, o extrator retirou 388201 registros (*tokens* e/ou frases), sendo 286781 e 101420 registros das peças de Instrução e Acórdãos, respectivamente. A distribuição desses registros por *tags* é apresentada na Tabela 9.

Em uma análise geral das extrações é revelado que muitas palavras ou frases não são relevantes, enquanto aquelas que têm relevância exigem um pós-processamento para resumi-las e transformá-las em informações úteis. Nota-se a ausência de extração da *tag* do tipo 'PROCESSO_VINCULADO', isso é devido à falta de mais exemplos anotados nos processos, sendo assim, poderia ser feita

a exclusão dessa *tag* das anotações para reduzir os custos de processamento. A utilização dos *tokens* extraídos do tipo de entidade 'VALOR_DANO_ATUALIZADO' revelou-se limitada, resultando em apenas 17 valores considerados corretos/bem formatados. Além disso, os *tokens* da *tag* 'PROCESSO' também não podem ser aproveitados, uma vez que não houve extração de nenhum número de processo completo.

## 6 CONSIDERAÇÕES FINAIS

Realizar o processamento de linguagem natural não é uma tarefa trivial, porém, a cada ano, melhoram-se as técnicas e abordagens. A análise dos resultados dos dados de treinamento e validação de todos os modelos gerados neste trabalho sugere que a quantidade de dados de treinamento é muito importante para o desempenho de um modelo de *Fine Tuning* de domínio específico. Um conjunto de dados maior e mais variado pode melhorar a capacidade do modelo de generalizar a partir de seus exemplos e, assim, performar melhor em dados não vistos anteriormente. Isso é evidenciado ao observar as métricas de desempenho (precisão, *recall* e *F1-score*) que tendem a ser mais estáveis e elevadas quando o modelo é treinado com mais dados. Devido às limitações da máquina local, somente pôde ser feito o treinamento utilizando o modelo BERTimbau *base*. Trabalhos relacionados trazem resultados melhores com a utilização do modelo *large*, e, no caso deste trabalho, pode ser que fossem obtidas melhores métricas com esse modelo.

A qualidade da anotação é ainda mais crítica do que a quantidade em tarefas REN, caso as anotações fossem realizadas por especialistas jurídicos, a agilidade na anotação e os resultados poderiam ser melhores. Modelos podem ser treinados com uma grande quantidade de dados, mas se os dados não forem bem anotados, o modelo irá aprender a partir de exemplos incorretos, levando a um desempenho ruim. Por outro lado, um conjunto menor de dados de alta qualidade, com anotações precisas, pode resultar em um modelo mais eficaz e preciso, que aprende as características corretas e é capaz de fazer predições acuradas, mesmo que seja com base em menos exemplos. Entretanto, a quantidade de *tokens* anotados para cada tipo de entidade também tem forte influência no treinamento e extração final. Então o ideal é buscar um equilíbrio entre a quantidade e a qualidade dos dados de treinamento. Um volume suficiente de dados bem anotados é o cenário ideal para treinar um modelo robusto. No entanto, se os recursos forem limitados, priorizar a qualidade da anotação pode ser mais benéfico do que simplesmente aumentar a quantidade de dados.

Diante dos resultados alcançados e das análises realizadas, pode-se confirmar a hipótese de pesquisa levantada quando da proposição deste trabalho de que é possível ampliar a quantidade de variáveis explicativas nos processos de Tomada de Contas Especial usando técnicas de processamento de linguagem natural, revelando o potencial do processamento de linguagem natural para a extração de dados diversos daqueles disponíveis no site institucional do TCU.

### 6.1 Trabalhos Futuros

Considerando os desdobramentos do projeto, como prospectivas para trabalhos subsequentes, destaca-se a possibilidade de realizar uma análise exploratória abrangente das entidades extraídas e a aplicação de estatística descritiva para aprimorar os resultados. Com isso, propõe-se disponibilizar os resultados por meio de uma aplicação web para proporcionar uma visualização acessível à comunidade, permitindo uma compreensão mais clara do uso dos recursos públicos. Isso pode resultar em maior transparência e capacidade dos cidadãos de exigir responsabilidade dos agentes políticos.

Além disso, destaca-se a possibilidade de explorar a estatística inferencial para a previsão de valores futuros. O trabalho também pode servir como base para pesquisas comparativas entre diferentes modelos ou a construção de um modelo de aprendizagem profunda a partir do zero, utilizando as fundamentações e a base de dados estabelecidas neste estudo. Ademais, foram identificadas áreas passíveis de aprimoramento, como a necessidade de uma anotação mais abrangente, incluindo partes do discurso e relações entre entidades.

Outras investigações podem ser conduzidas para avaliar a aplicabilidade dos métodos utilizados em diferentes *datasets*. Além do mais, seria pertinente examinar se o modelo 4, oriundo do primeiro treinamento, poderia ser empregado de maneira aceitável em outras iterações, permitindo assim a economia de várias etapas, incluindo treinamento com base específica e ajuste fino nos textos. Sugere-se também a exploração de combinações entre modelos de transformadores e outras arquiteturas de *deep learning* para melhorar o reconhecimento dessas anotações adicionais, contribuindo para uma análise mais refinada dos dados.

## REFERÊNCIAS

[1] Carlos Andre Reis Pinheiro. *Inteligência analítica*. Ciencia Moderna, 2008.
[2] PI Neves, DA Corrêa, and MC Cavalcanti. Uma análise sobre abordagens e ferramentas para extração de informação. *Revista Militar de Ciência e Tecnologia*, (3o):32–58, 2020.
[3] Robinson Santos Castro. Extração de informação: conceitos, plataformas e sistemas. 2013.
[4] Amabyle Rabeche Heck et al. Processamento de linguagem natural aplicado a reconhecimento de entidades nomeadas em textos legais em português brasileiro. Master's thesis, Florianópolis, SC., 2022.
[5] Gilson Garcia. Tribunais de contas, controle preventivo, controle social e jurimetria: um estudo sobre as representações para suspensão de licitações. *Revista Controle - Doutrina e Artigos*, 19:160–193, 01 2021. doi: 10.32586/rcda.v19i1.650.
[6] Tribunal de Contas da União/PLENÁRIO. Instrução normativa nº 71, de 28 de novembro de 2012. *Diário Oficial da União*, 2012.
[7] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Bertimbau: pretrained bert models for brazilian portuguese. In *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9*, pages 403–417. Springer, 2020.
[8] William Paulo Ducca Fernandes, Isabella Zalcberg Frajhof, Guilherme da Franca Couto Fernandes de Almeida, Ariane Moraes Bueno Rodrigues, Simone Diniz Junqueira Barbosa, Carlos Nelson Konder, Rafael Barbosa Nasser, Gustavo Robichez de Carvalho, and Hélio Côrtes Vieira Lopes. Extracting value from brazilian court decisions. *Information systems (Oxford)*, 106:101965, 2022. ISSN 0306-4379.
[9] Yong Ren, Jinfeng Han, Yingcheng Lin, Xiujiu Mei, and Ling Zhang. An ontology-based and deep learning-driven method for extracting legal facts from chinese legal texts. *Electronics*, 11(12):1821, 2022.
[10] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *proceedings of the 27th international conference on computational linguistics: system demonstrations*, pages 5–9, 2018.
[11] Natanael da Silva Araújo. Reconhecimento de entidades nomeadas em textos de boletins de ocorrências. 2019.

## A RECURSOS *ON-LINE*

O *dataset* e métodos aplicados no estudo podem ser encontrados no repositório do *Github* da autora.