Proposta recebida em Setembro 2024 e aceite para publicação em Novembro 2024.

Reconhecimento de Entidades Nomeadas e Vazamento de Dados em Textos Legislativos: Uma Reavaliação da Literatura

Named Entity Recognition and Data Leakage in Legislative Texts: A Literature Reassessment

Rafael Oleques Nunes
Universidade Federal do Rio Grande do Sul

André Susliz Spritzer
Universidade Federal do Rio Grande do Sul

Carla Maria Dal Sasso Freitas
Universidade Federal do Rio Grande do Sul

Dennis Giovani Balreira
Universidade Federal do Rio Grande do Sul

Resumo

Este trabalho trata do vazamento de dados no treinamento de modelos de Reconhecimento de Entidades Nomeadas (NER) em textos legislativos em português brasileiro, resultante de duplicatas e anotações inconsistentes, o que compromete a avaliação dos modelos. Após corrigir esse vazamento no *corpus* UlyssesNER-Br, foi realizado um novo *benchmark*, comparando os resultados com estudos anteriores em um cenário mais confiável. Também foi reavaliada uma abordagem semissupervisionada utilizando autoaprendizado e amostragem ativa. No entanto, ao reutilizar um *threshold* fixo, escolhido a partir de uma nuvem de valores antes da correção, os resultados foram insatisfatórios. Isso indica que um *threshold* dinâmico, que se adapte às características dos dados pós-correção, poderá proporcionar uma avaliação mais eficiente e precisa, indicando a necessidade de futuros estudos sobre a escolha de *thresholds*.

Palavras chave

vazamento de dados; reconhecimento de entidades nomeadas; textos legislativos; benchmark; autoaprendizado; português

Abstract

This work addresses data leakage in training Named Entity Recognition (NER) models in Brazilian Portuguese legislative texts, resulting from duplicates and inconsistent annotations, which compromise model evaluation. After correcting this leakage in the UlyssesNER-Br *corpus*, we conducted a new benchmark, comparing the results with previous studies in a more reliable setting. We also re-evaluated a semi-supervised approach using self-learning and active sampling. However, by reusing a fixed threshold, chosen from a cloud of values before the correction, the results were unsatisfactory. This indicates that a dynamic threshold, which adapts to the characteristics of the data post-correction, could provide a more efficient and accurate evaluation, highlighting the need for future studies on threshold selection.

Keywords

data leakage; named entity recognition; legislative texts; benchmark; self-learning; Portuguese

1. Introdução

O conceito de política democrática transcende o simples ato de votar. Ela exige vigilância contínua de uma sociedade civil ativa e de um público bem-informado, ambos responsáveis por controlar e vigiar os atos dos políticos. A transparência, definida como a disponibilização de informações ao público sobre as atividades e decisões das organizações, é um mecanismo essencial para garantir a responsabilização dos agentes públicos, permitindo que os cidadãos avaliem suas ações (Heald, 2006). Iniciativas de governo aberto ampliam essa transparência ao oferecer acesso a documentos e dados relacionados a atividades públicas e oficiais (Lathrop & Ruma, 2010). No entanto, o simples acesso à informação frequentemente não é suficiente para possibilitar uma fiscalização efetiva. Pesquisadores, jornalistas e cidadãos podem se deparar com dificuldades diante da vasta quantidade de projetos de lei, emendas e documentos redigidos em jargão técnico, tornando o processo de análise e compreensão das atividades políticas e legislativas uma tarefa particularmente desafiadora.

No contexto do Processamento de Linguagem Natural (PLN), o Reconhecimento de Entidades Nomeadas (NER, do inglês "*Named Entity Recognition*") desempenha um papel crucial. O NER envolve a identificação de entidades específicas em textos e sua classificação em categorias predefinidas, como pessoas, organizações

DOI: 10.21814/lm.16.2.450

e locais. Esse processo pode facilitar a compreensão de documentos ao destacar termos especializados e específicos do domínio, proporcionando uma visão geral mais clara dos textos, como demonstrado nos trabalhos de Sultanum et al. (2018) e Nunes et al. (2019). Além disso, o NER contribui indiretamente para a compreensão de textos ao possibilitar processos de enriquecimento, como a indexação de informações de dicionários, o que facilita a oferta de explicações contextuais e sinônimos. O NER também serve como um estágio inicial em diversas outras tarefas de PLN, como a construção de grafos de conhecimento específicos de domínio e a resolução de correferências (Kalamkar et al., 2022; Cohen, 2005).

Neste estudo investigamos o impacto do vazamento de dados nos resultados de modelos de NER, reavaliando o *benchmark* proposto na literatura e propondo a avaliação de novos modelos para comparação. Além disso, como NER pode ser aprimorado utilizando técnicas semissupervisionadas. Nossa abordagem consiste em uma estratégia de autoaprendizado para ajustar um modelo BERT projetado para NER, utilizando documentos legislativos escritos em português brasileiro como estudo de caso. Para o treinamento e a avaliação, utilizamos o UlyssesNER-Br (Albuquerque et al., 2022), um *corpus* de projetos de lei e consultas legislativas da Câmara dos Deputados do Brasil, explicitamente desenvolvido para NER.

Este artigo é uma versão revisada e ampliada do trabalho anterior (Nunes et al., 2024b). Nesta nova versão, adicionamos uma seção detalhada sobre o vazamento de dados encontrado no *corpus* UlyssesNER-Br, incluindo dados quantitativos e o processo de limpeza realizado. Também introduzimos o novo *corpus*, apresentando alterações tanto no nível de categorias quanto no de tipos. Esta nova versão já se encontra disponível no repositório oficial do UlyssesNER-Br. Além disso, incluímos a avaliação de novos modelos, a comparação entre as versões contaminadas e descontaminadas, o uso de validação cruzada nos experimentos e um novo o *benchmark* refeito agora com testes estatísticos, comparando os resultados com a literatura e analisando o impacto do vazamento de dados nos resultados anteriores.

Dessa forma, as principais contribuições deste artigo são: (i) a avaliação e identificação de diferentes tipos de vazamento de dados no *corpus* UlyssesNER-Br[1]; (ii) a avaliação de modelos de linguagem de compreensão baseados na arquitetura *Transformer*[2], tanto gerais quanto específicos de domínio, em português e multilíngue, resultando no estabelecimento de novos *benchmarks*; e (iii) uma discussão detalhada sobre o NER no contexto legislativo, contrastando os resultados encontrados com os disponíveis na literatura atual.

2. Trabalhos Relacionados

Esta seção explora a literatura sobre NER e o uso de dados não rotulados no processo de treinamento, especificamente em modelos de PLN. A Subseção 2.1 apresenta estudos relevantes sobre NER ao longo do tempo, com ênfase no uso do domínio jurídico em língua portuguesa. Já a Subseção 2.2 aborda o uso de dados não rotulados como uma forma de aumento de dados, concentrando-se especificamente em técnicas de autoaprendizado e aprendizado ativo.

2.1. Reconhecimento de Entidades Nomeadas

Há mais de uma década, Dozier et al. (2010) propuseram um dos sistemas de NER jurídico mais conhecidos, utilizando dados de tribunais dos Estados Unidos, consistindo principalmente em depoimentos, petições e jurisprudência. Os autores utilizaram três métodos para a tarefa de NER, que também podiam ser combinados em sistemas híbridos. Eles também introduziram cinco categorias de entidades, incluindo *jurisdição*, *tribunal*, *título*, *tipo de documento* e *juiz*. De forma semelhante, outros trabalhos exploraram NER no domínio jurídico para outras línguas, como o alemão (Darji et al., 2023; Glaser et al., 2018; Leitner et al., 2019), espanhol (Badji, 2018), grego (Angelidis et al., 2018) e romeno (Păiș et al., 2021). No que se refere ao português, (Santos & Guimarães, 2015) propuseram o primeiro sistema de NER utilizando a arquitetura CharWNN, que emprega uma rede neural perceptron de múltiplas camadas (Santos & Guimarães, 2015). A maioria dos trabalhos sobre NER em textos em português de domínio geral avalia seus modelos utilizando o *corpus* HAREM (Santos et al., 2006), que contém documentos de várias áreas.

No domínio jurídico em língua portuguesa, dois trabalhos recentes introduziram conjuntos de dados em português para NER em textos legais (Luz de Araujo et al., 2018; Albuquerque et al., 2022). Luz de Araujo et al.

[1] https://github.com/ulysses-camara/ulysses-ner-br/tree/main/PL-corpus_v2

[2] Código usado para treinar os modelos e realizar a divisão dos dados está disponível em https://github.com/RafaelOleques/NERLinguamatica2024

(2018) criaram o primeiro conjunto de dados para NER em textos jurídicos brasileiros, chamado LeNER-Br, reunindo 66 documentos legais de tribunais brasileiros, além de treinarem um modelo de rede neural de memória de longa e curta duração com campo aleatório condicional (BiLSTM-CRF) (Lample et al., 2016), resultando em um valor F1 total de cerca de 92% para a classificação de *tokens* e 86% para a classificação de entidades. Albuquerque e colegas (Albuquerque et al., 2022) propuseram, por sua vez, um *corpus* para NER chamado UlyssesNER-Br, composto por projetos de lei e consultas legislativas da Câmara dos Deputados, com 18 tipos de entidades distribuídos em sete categorias. Para validar o *corpus*, os autores implementaram modelos de Campo Aleatório Condicional (CRF, do inglês "*Conditional Random Field*") e Modelo de Markov Oculto (HMM, do inglês "*Hidden Markov Model*") Bird (2006), alcançando uma pontuação F1 de cerca de 80% na análise por categorias e 81% na análise por tipos.

De forma semelhante, três trabalhos recentes exploram contextos jurídicos específicos. Collovini et al. (2019) anotaram manualmente um conjunto de dados policiais usando textos de depoimentos, declarações e interrogatórios, com 916 entidades nomeadas da categoria "Pessoa", alcançando uma pontuação F1 de 89% utilizando BiLSTM-CRF-ELMo. Brito et al. (2023) desenvolveram o CDJUR-BR, um *corpus* do Judiciário brasileiro com entidades específicas do domínio, como *prova*, *pena*, *sentença* e *norma*. Eles obtiveram uma F1-Macro de 0,58 utilizando o modelo BERT (Devlin et al., 2018). Por fim, Correia et al. (2022) desenvolveram um *corpus* com entidades jurídicas de documentos do Supremo Tribunal Federal (STF), anotados por 76 estudantes de Direito. Utilizando BiLSTM-CRF, eles obtiveram um *F1-Weighted* de 93%.

Com base nos avanços em metodologias de ajuste fino e aprimoramentos de modelos, o trabalho proposto por Bonifacio et al. (2020) investigou o impacto do ajuste fino de modelos de linguagem em um grande *corpus* intradomínio de texto não rotulado para NER. Os resultados experimentais revelaram que o ajuste fino dos modelos nestes textos melhorou significativamente o desempenho de NER, particularmente para o modelo BERT, que alcançou resultados de ponta no *corpus* LeNER-Br de textos jurídicos brasileiros. Zanuz & Rigo (2022) introduziram os primeiros modelos BERT ajustados exclusivamente para o português brasileiro, para NER jurídico, alcançando novos resultados de ponta no conjunto de dados LeNER-Br.

2.2. Autoaprendizagem e Aprendizagem Ativa no Treinamento

O aumento de dados tem sido amplamente empregado em diversas tarefas de PLN para aprimorar o desempenho de modelos (Li et al., 2022; Feng et al., 2021; Anaby-Tavor et al., 2020). Um método alternativo para melhorar a qualidade de um *corpus* de treinamento é o uso de dados não rotulados. Em casos onde há uma quantidade substancial de dados não rotulados disponíveis, são utilizadas técnicas semi-supervisionadas (Li et al., 2022; Feng et al., 2021), como autoaprendizagem, aprendizagem ativa e suas variações. Essas técnicas têm demonstrado melhorar os resultados dos modelos em tarefas como classificação (Sha et al., 2022; Alves-Pinto et al., 2021; Mekala & Shang, 2020; Meng et al., 2020; Dong & de Melo, 2019; Dupre et al., 2019) e NER (Gao et al., 2021; Neto & Faleiros, 2021; Helwe & Elbassuoni, 2019; Clark et al., 2018; Tran et al., 2017; Chen et al., 2015).

A abordagem de autoaprendizagem envolve o uso de um *corpus* rotulado para treinar um modelo *professor*, que é utilizado para prever as classes dos dados não rotulados, os quais, por sua vez, são usados para treinar um modelo *aluno* (Dupre et al., 2019). Em algumas estratégias de autoaprendizagem, o modelo aluno pode servir como *professor* para a próxima iteração (Dupre et al., 2019). Embora isso represente uma abordagem tradicional de autoaprendizagem, métodos alternativos, como rótulos fracos, modelos *ensemble* e modificações na função de perda, também podem ser empregados. A aprendizagem ativa segue uma filosofia semelhante, mas incorpora métodos de consulta para selecionar instâncias de interesse para anotação manual. Essas instâncias anotadas são então usadas para treinar o modelo de forma iterativa.

3. *Corpus*

Esta seção apresenta uma visão geral sobre o *corpus* analisado no trabalho, bem como sobre o problema de vazamento de dados descoberto e sua adaptação.

3.1. UlyssesNER-Br

O UlyssesNER-Br (Albuquerque et al., 2022) é um *corpus* em português brasileiro que contém duas fontes de informação e está dividido em dois *corpora*, um para cada fonte de referência. O primeiro *corpus* contém 9.526 sentenças de 150 projetos de lei (*PL-corpus*) da Câmara dos Deputa-

dos do Brasil, enquanto o segundo possui 790 sentenças de solicitações de trabalho (*ST-corpus*).

Ambos os *corpora* do UlyssesNER-Br possuem dois *sub-corpora* com diferentes níveis de entidades: *categoria* e *tipo*. As categorias englobam cinco entidades tradicionais (Albuquerque et al., 2022): "PESSOA", "DATA", "ORGANIZAÇÃO", "EVENTO" e "LOCALIZAÇÃO". Além disso, incluem "FUNDAMENTO" e "PRODUTODELEI" como referências a entidades legislativas. Os tipos, por sua vez, são especializações das categorias, como, por exemplo, "PRODUTOsistema", "PRODUTOprograma" e "PRODUTOoutros" como particularizações da categoria "PRODUTODELEI".

Infelizmente, o *ST-corpus* com solicitações de trabalho não está disponível publicamente, pois se trata de informações internas da Câmara dos Deputados[3], que os autores do UlyssesNER-Br não foram autorizados a compartilhar. Portanto, utilizamos apenas o *PL-corpus* com as informações dos projetos de lei neste estudo.

Em nosso trabalho, utilizamos apenas as entidades de categoria para o processo de autoaprendizagem, uma vez que os autores apontaram que os resultados com categorias e tipos não apresentaram diferenças significativas. Dessa forma, as categorias mostraram-se uma solução mais simples e robusta para o aprendizado do modelo (Albuquerque et al., 2022). A Tabela 1 indica o número de exemplos de cada categoria no *corpus* disponibilizado pelos autores, conforme mostrado na coluna *#Original*. Ressaltamos que a frequência calculada nas tabelas refere-se à ocorrência da entidade completa, e não à frequência de *tokens*.

3.2. Vazamento de dados

Vazamento de dados é um problema crítico em aprendizado de máquina e PLN (Balloccu et al., 2024), no qual os dados do conjunto de teste são inadvertidamente acessados pelo modelo durante o treinamento, resultando em uma superestimação dos resultados e comprometendo a avaliação (Balloccu et al., 2024). Em *corpora* legislativos, o risco de vazamento é particularmente elevado devido à recorrência de termos específicos e padrões linguísticos semelhantes entre os diferentes conjuntos, como é possível ver na Tabela 2.

Identificação e tratamento de instâncias repetidas. Encontramos 70 sentenças com duplicatas e a mesma anotação, e 4 com duplicatas, mas com anotações divergentes. A Tabela 3 apresenta exemplos de ambos os tipos de repetições.

A maioria das instâncias com anotações divergentes foi identificada como casos de anotação parcial, onde as anotações eram essencialmente idênticas, mas com trechos faltantes. No entanto, encontramos um caso de anotação ambígua, em que a mesma entidade nomeada foi marcada como pertencente a classes diferentes em sentenças duplicadas, conforme ilustrado na Tabela 3.

Embora este estudo se concentre apenas no nível de categorias, também analisamos o *sub-corpus* de tipos. Neste *sub-corpus*, identificamos um maior número de sentenças duplicadas, com 512 ocorrências de anotações iguais e 3 com anotações divergentes, conforme apresentado na Tabela 4. Nesse caso, as anotações divergentes eram essencialmente iguais, mas apresentavam partes faltantes.

Além disso, identificamos sobreposições entre os conjuntos originalmente disponibilizados para treino, validação e teste, como mostrado na Tabela 5. Essa tabela revela duplicatas dentro de cada conjunto, o que pode levar ao reforço de padrões específicos. Observamos também sobreposições entre diferentes conjuntos, o que pode resultar em uma superestimação dos resultados, uma vez que o modelo pode ser treinado com dados que estavam presentes em mais de um conjunto.

Após a identificação de instâncias repetidas, consultamos um especialista para revisar as anotações, garantindo maior precisão ao conjunto de dados. Nas instâncias em que a anotação original foi considerada inadequada, seja por anotação parcial ou ambígua, o especialista atribuiu uma nova anotação mais adequada ao contexto. Já para os casos em que a anotação era essencialmente igual entre as ocorrências repetidas, mantivemos apenas a primeira ocorrência no *corpus*. Na Subseção 3.3, apresentamos as estatísticas do novo conjunto gerado e uma descrição atualizada do *corpus*.

3.3. *Corpus* atualizado

Após a implementação das estratégias de mitigação de vazamento de dados e a padronização das anotações, realizamos uma análise estatística detalhada do *corpus* atualizado. Nesta subseção, discutimos a nova divisão dos dados, que foi feita tanto para a abordagem de *holdout* quanto para a validação cruzada.

[3] https://github.com/Convenio-Camara-dos-Deputados/ulyssesner-br-propor/tree/main/Corpora

	Entidades			Sentenças		
Classe	**#Original**	**#Filtrado**	**Δ**	**#Original**	**#Filtrado**	**Δ**
DATA	603	427	-176	522	346	-176
EVENTO	23	23	0	21	21	0
FUNDAMENTO	721	716	-5	522	519	-3
LOCAL	615	607	-8	325	319	-6
ORGANIZACAO	610	598	-12	469	460	-9
PESSOA	861	847	-14	545	536	-9
PRODUTODELEI	330	319	-11	277	267	-10
Soma	3.763	3.537	-226	2.681	2.468	-213

Tabela 1: Número de instâncias e número de sentenças que possuem cada classe no *corpus* UlyssesNER-Br no nível de categoria, antes e após a filtragem proposta neste trabalho.

Padrão	**Categorias**	**Tipos**
sala das sessões , em de de [ANO]	88	79
projeto de lei nº , de [ANO] (do sr .	42	44

Tabela 2: Exemplos de padrões identificados no *PL-corpus*. A tabela apresenta as frases padrão, seguidas pela quantidade de repetições nos *sub-corpora* de categorias e tipos.

Tipo	**Elemento**	**Exemplos**
Duplicada	Sentença:	câmara dos deputados projeto de lei nº , de 2019 (do sr .
	Anotação:	B-ORGANIZACAO I-ORGANIZACAO I-ORGANIZACAO O O O O O O B-DATA O O O O
Parcial	Sentença:	sala das sessões , em de agosto de 2019 .
	Anotação 1:	O O O O O O B-DATA I-DATA I-DATA O
	Anotação 2:	O O O O O O O O O O
Ambígua	Sentença:	recentemente , foi publicada a lei nº 13.819 , de 2019 6 , que instituiu a política nacional de prevenção da automutilação e do suicídio , e trouxe diversas inovações ao ordenamento jurídico , no contexto da prevenção desse agravo
	Anotação 1:	O O O O O B-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO O O O O O B-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO O O O O O O O O O O O O O O O O O O O
	Anotação 2:	O O O O O B-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO I-FUNDAMENTO O O O O O B-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI I-PRODUTODELEI O O O O O O O O O O O O O O O O O O O

Tabela 3: Exemplos de anotações duplicadas, parciais e ambíguas encontradas no *sub-corpus* de categorias.

Tipo	Elemento	Exemplos
Duplicada	Sentença:	sala das sessões , em de de 2019 .
	Anotação:	O O O O O O O B-DATA O
Parcial	Sentença:	câmara dos deputados projeto de lei nº , de 2011 (do sr .
	Anotação 1:	O O O O O O O O O B-DATA O O O O
	Anotação 2:	B-ORGgovernamental I-ORGgovernamental I-ORGgovernamental O O O O O O B-DATA O O O O

Tabela 4: Exemplos de anotações duplicadas e parciais encontradas no *sub-corpus* de tipos.

Conjuntos	#Instâncias
Treino e Validação	
#Sentenças[1]	13
#Treino[2]	95
#Validação[2]	30
Treino e Teste	
#Sentenças[1]	22
#Treino[2]	128
#Teste[2]	33
Validação e Teste	
#Sentenças[1]	5
#Validação[2]	21
#Teste[2]	10
Treino, Validação e Teste	
#Sentenças[1]	5
#Treino[2]	85
#Validação[2]	21
#Teste[2]	10

Tabela 5: Número de instâncias duplicadas com entidades nomeadas entre os conjuntos originais do UlyssesNER-Br. Ambos os *sub-corpus* possuem os mesmos valores. [1]Cada sentença é contada somente uma vez, independente do número de cópias. [2]Cada sentença é contadas N vezes, sendo N o número de cópias que ela possui.

Distribuição das entidades no *corpus*. As Tabelas 1 e 6 mostram a distribuição das entidades em ambos os *sub-corpora*. Na coluna *#Original* estão o número de instâncias originais de cada classe; na coluna *#Filtrado*, o número de instâncias após a filtragem para remoção de sentenças duplicadas; e na coluna Δ, a diferença entre os dois conjuntos. É relevante observar que, embora o *sub-corpus* de tipos contenha entidades especializadas em comparação com o *sub-corpus* de categorias, não foi realizada a anotação aninhada entre os níveis, o que resultou em discrepâncias nas estatísticas entre eles.

Observamos uma redução no número de sentenças, com uma diminuição de 319 instâncias no nível de categorias e de 787 instâncias no nível de tipos. As Tabelas 1 e 6 fornecem detalhes sobre o impacto dessa redução nas sentenças de cada classe.

A análise das tabelas revela que a filtragem resultou em uma redução na quantidade de *tokens* e sentenças no *corpus*. Embora a distribuição das entidades não tenha mostrado variações significativas na frequência das classes —com as classes majoritárias continuando a dominar e as minoritárias permanecendo menos frequentes—, a filtragem reduziu o número total de sentenças disponíveis para análise. No entanto, a remoção de duplicatas pode ter contribuído para uma melhoria na qualidade geral dos dados. A comparação entre os *sub-corpus* de categorias e tipos revela diferenças na distribuição que podem impactar o treinamento e a avaliação dos modelos. Assim, é fundamental considerar essas diferenças e seus potenciais efeitos na performance dos modelos, além de avaliar a necessidade de técnicas adicionais para o balanceamento de dados.

4. Metodologia

Nesta seção, detalhamos a metodologia utilizada neste trabalho. Descrevemos o *corpus* não rotulado empregado no processo de autoaprendizagem, a divisão do *corpus* para o treinamento dos modelos, os modelos utilizados, os critérios de avaliação e o pipeline de autoaprendizagem implementado.

4.1. Ementas da Câmara dos Deputados do Brasil

As ementas dos projetos de lei são obtidas por meio da API da Câmara dos Deputados[4] abrangendo de 1991 a 2022. Essas ementas são, então, segmentadas em frases usando a expressão regular “`. (?=[A-Za-z])`” para identificar pontos

[4] https://dadosabertos.camara.leg.br/swagger/api.html

	Entidades			Sentenças		
Classe	**#Original**	**#Filtrado**	**Δ**	**#Original**	**#Filtrado**	**Δ**
DATA						
DATA	629	383	-246	541	309	-232
EVENTO						
EVENTO	27	18	-9	25	17	-8
FUNDAMENTO						
FUNDapelido	191	157	-34	176	145	-31
FUNDlei	529	442	-87	463	386	-77
FUNDprojetodelei	18	13	-5	14	11	-3
LOCAL						
LOCALconcreto	509	418	-91	285	232	-53
LOCALvirtual	62	45	-17	47	36	-11
ORGANIZACAO						
ORGgovernamental	460	387	-73	375	312	-63
ORGnaogovernamental	132	104	-28	105	83	-22
ORGpartido	31	27	-4	23	21	-2
PESSOA						
PESSOAcargo	304	260	-44	259	220	-39
PESSOAgrupocargo	161	140	-21	119	106	-13
PESSOAindividual	401	337	-64	368	310	-58
PRODUTODELEI						
PRODUTOprograma	64	53	-11	56	47	-9
PRODUTOsistema	17	15	-2	16	14	-2
PRODUTOoutros	257	207	-50	217	175	-42
Soma	3.792	3.006	-786	3.089	2.424	-665

Tabela 6: Classes e instâncias no *corpus* UlyssesNER-Br no nível de tipos, antes e após a filtragem proposta neste trabalho.

seguidos por letras, dividindo o texto em frases. Escolhemos essa expressão regular porque o domínio do texto legislativo inclui construções como "`Art. 123`", onde o ponto faz parte do nome do artigo e não indica o fim de uma frase.

Esse processo produziu 428.573 frases, com uma contagem média de palavras de 32,52 e um desvio padrão de 42,64. Ao obter frases, excluímos as que já estavam presentes no *corpus* UlyssesNER-Br, certificando-nos de incluir apenas frases distintas para evitar a inserção de contaminação ou *overfitting*. Todas essas frases não tinham informações NER.

4.2. Divisão do *corpus*

Para treinar e validar nossa abordagem, utilizamos um método de divisão de validação cruzada com cinco partições, inspirado no artigo original do UlyssesNER-Br (Albuquerque et al., 2022). A principal distinção é que usamos divisão estratificada, uma modificação influenciada pela abordagem de Sechidis et al. (2011). Também introduzimos uma etapa adicional de pré-processamento que gera uma lista equivalente ao número de entidades distintas possíveis. Dentro dessa lista, cada posição é marcada com um valor 1, se a entidade correspondente estiver presente na sentença, e com 0, caso contrário. Essa modificação nos permitiu estratificar a divisão com base na presença de cada entidade.

É importante destacar a relevância desse passo de estratificação, especialmente devido ao desequilíbrio substancial de classes no *corpus*. Esse desequilíbrio é evidente ao examinar exemplos de classes minoritárias e majoritárias, como "Eventos", com apenas 21 instâncias, em contraste com "Pessoa", que possui 847 instâncias no *sub-corpus* de categorias.

4.3. Modelos

Descrevemos brevemente os modelos *Transformers* mais proeminentes para a língua portuguesa. Utilizamos os modelo com as arquiteturas BERT, DeBERTa e RoBERTa para a tarefa de NER e o modelo SBERT para a amostragem ativa.

BERTimbau é um modelo BERT pré-treinado e ajustado para o português brasileiro (Souza et al., 2020). Até onde sabemos, o BERTimbau é o estado da arte em Reconhecimento de Entidades Nomeadas, similaridade textual de sentenças e reconhecimento de inferências textuais em português brasileiro. Este modelo possui duas versões principais, *large* e *base*. No texto, quando não referenciado explicitamente, o modelo referido é o BERTimbau-*large*.

XLM-RoBERTa (XLM-R) (Conneau et al., 2020) é um modelo multilíngue que tem a capacidade de processar 100 idiomas diferentes sem a necessidade de incorporações específicas para cada língua. Ele combina abordagens do RoBERTa (Liu et al., 2019) dentro do framework do XLM (Conneau et al., 2020), permitindo que o modelo identifique o idioma a partir dos *tokens* de entrada. O foco do XLM-R está na modelagem de linguagem mascarada para sentenças monolíngues, excluindo a técnica de modelagem de linguagem de tradução utilizada no XLM.

Albertina PT-BR (Rodrigues et al., 2023) é uma versão do modelo de linguagem Albertina projetada para o português brasileiro. Baseado na arquitetura DeBERTa (He et al., 2021), este modelo possui uma licença permissiva e utiliza um *corpus* multilíngue com 36 bilhões de *tokens*, ampliando sua capacidade de entendimento e geração de texto em português.

BERTikal é um modelo BERT ajustado para a linguagem jurídica brasileira (Polo et al., 2021). A base utilizada foi o BERTimbau base cased (Souza et al., 2020), empregando um objetivo de Modelagem de Linguagem Mascarada (MLM). Os documentos utilizados para o treinamento do BERTikal incluíram publicações e petições de tribunais brasileiros, além de documentos legais mais extensos, principalmente oriundos do Tribunal de Justiça de São Paulo (TJSP).

Jurisbert e **BERTimbaulaw** são desenvolvidos para o português brasileiro, como proposto por Viegas et al. (2023). Os modelos utilizaram informações resumidas coletadas de tribunais brasileiros por meio de web scraping. Enquanto o Jurisbert foi treinado do zero, após o pré-treinamento do BERT base (Devlin et al., 2018), o BERTimbaulaw aproveitou o pré-treinamento do BERTimbau base (Souza et al., 2020).

LegalBERT-pt (Silva et al., 2021) é um modelo BERT pré-treinado com *corpora* jurídicos brasileiros. Os autores realizaram o pré-treinamento de dois tipos de modelos: LegalBert-pt SC e LegalBert-pt FP. O modelo SC foi elaborado a partir do treinamento de um modelo BERT do zero, com a mesma configuração do BERTimbau base. Por outro lado, o modelo FP pré-treinou um BERTimbau-Base utilizando *corpora* específicos do domínio jurídico. Os *corpora* utilizados no estudo englobam diversos casos judiciais brasileiros e, para o modelo SC, artigos da Wikipédia em português, totalizando 1.500.000 documentos legais e um vocabulário de 36.345 palavras.

RoBERTaLexPT (Garcia et al., 2024) é um modelo baseado no RoBERTa (Liu et al., 2019), que foi aprimorado com o *corpus* LegalPT (Garcia et al., 2024), especializando-o para a linguagem jurídica portuguesa. O *corpus* utilizado na fase de pré-treinamento abrange uma variedade de conjuntos de dados, predominantemente dados jurídicos, mas também incluindo informações gerais, em português brasileiro e português europeu.

Legal-XLM-RoBERTa e **Legal-portuguese-RoBERTa** são modelos adaptados a partir do pré-treinamento do XLM-R (Conneau et al., 2020) para o domínio jurídico (Niklaus et al., 2024). Ambos os modelos foram treinados utilizando o *corpus* Multilegalpile (Niklaus et al., 2024), sendo que o primeiro foi treinado com todo o *corpus*, enquanto o segundo focou especificamente na porção em português.

SBERT (Reimers & Gurevych, 2019) é uma modificação dos modelos BERT que utiliza redes siamesas e *triplets* para obter *embeddings* contextuais referentes a sentenças inteiras. Para gerar *embeddings* para textos em português, utilizamos a versão multilíngue do SBERT (Reimers & Gurevych, 2020), que está disponível no Hugging Face Hub[5] na técnica de amostragem ativa.

4.4. Avaliação de modelos

Com a atualização do *corpus*, realizamos uma nova avaliação dos modelos contextuais em português, multilíngues e específicos do domínio legal, previamente feita por Garcia et al. (2024). Embora o estudo original apresente vários hiperparâmetros utilizados, os valores finais de alguns que passaram por otimização, como *batch size* e *learning rate*, não foram informados. Por isso, optamos por utilizar os mesmos hiperparâmetros

[5] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1

do nosso trabalho anterior (Nunes et al., 2024b), que, em grande parte, estão alinhados com os de Garcia et al. (2024).

Entretanto, nossa avaliação difere da realizada anteriormente, pois, para evitar que resultados otimistas ou pessimistas sejam gerados aleatoriamente devido à abordagem *holdout*, optamos pela validação cruzada estratificada com 5 partições. Esse tipo de análise permite obter uma avaliação mais justa do modelo por meio da média das partições, além de proporcionar uma noção maior de robustez com base na análise do desvio padrão. Por fim, utilizamos as médias das partições em cada métrica para testes estatísticos (Gomes et al., 2021; Nunes et al., 2024a), a fim de determinar se as diferenças encontradas entre os modelos são significativas.

4.5. Autoaprendizado

A Figura 1 ilustra o processo de autoaprendizagem. Seguimos esse *pipeline* em cada iteração da validação cruzada e divisão de amostras. O *pipeline* começa com o treinamento do primeiro classificador usando os dados de treinamento (1 a 4). Uma vez que nosso *corpus* de ementas contém uma grande quantidade de dados, a técnica de amostragem é usada para otimizar o tempo de treinamento ao longo das iterações de autoaprendizagem (5 a 6).

Inspirados pela amostragem dinâmica de Sha et al. (2022), nossa técnica de amostragem começa com uma amostragem aleatória do *corpus* de ementas, produzindo $N\%$ dos dados não rotulados com um mínimo de 2.000 amostras. Em seguida, usamos a amostragem baseada em diversidade (Tran et al., 2017; Chen et al., 2015) aplicando similaridade de cosseno aos dados amostrados em relação aos dados de treinamento. Posteriormente, obtemos as amostras mais dissimilares em um total de $K\%$ dos dados amostrados, com um mínimo de 1.000, que são usados no modelo para prever os *labels* de NER para cada sentença. Os *embeddings* usados para calcular a similaridade de cosseno são gerados pelo SBERT, devido à sua capacidade de reconhecer características importantes de sentenças.

Após a etapa de amostragem ativa, aplicamos o classificador NER a cada sentença (7 a 9.a). Para determinar quais sentenças devem ser adicionadas aos dados de treinamento, medimos a confiança média da predição das entidades previstas (Gao et al., 2021) (10). Se a confiança média for igual ou superior a um limite, a sentença é utilizada nos dados de treinamento (11.a a 12.a) e removida do *corpus* de ementas (11.b a 12.b); caso contrário, ela é mantida no *corpus* de ementas e não é usada para o treinamento. Posteriormente, o *pipeline* é reiniciado usando o novo conjunto de treinamento para treinar um novo modelo e repetir todo o processo.

O processo é interrompido quando uma condição de parada antecipada é encontrada com base no F1 geral. Descrevemos os hiperparâmetros na Seção 5.3. Também implementamos um critério de parada antecipada no qual nenhum dado foi adicionado ao treinamento ou se o conjunto não rotulado ficou vazio (ou seja, todos os dados disponíveis foram utilizados). Ocasionalmente, devido à abordagem de amostragem aleatória, é possível que os dados selecionados aleatoriamente não contenham exemplos adequados para o treinamento, resultando em nenhuma adição ao conjunto de treinamento. Nesses casos, implementamos um critério de espera que permite um máximo de W novas amostragens antes de encerrar o processo de autoaprendizagem. Cada uma dessas novas amostragens usa sementes aleatórias diferentes para gerar conjuntos distintos, visando resolver possíveis problemas com a seleção aleatória inicial.

5. Avaliação experimental

Nesta seção, apresentamos uma avaliação experimental da abordagem proposta. Descrevemos o *setup* do experimento, incluindo o hardware utilizado e o ambiente de desenvolvimento. Também detalhamos os hiperparâmetros do nosso modelo, o treinamento com autoaprendizagem e as métricas usadas para a avaliação.

5.1. Setup

Utilizamos um computador com uma GPU Nvidia GeForce RTX 4090 e 64 GB de RAM para o treinamento e avaliação dos modelos, bem como para a obtenção dos dados da API da BCoD[6]. Optamos pela linguagem de programação Python 3.7.6 devido à sua vasta gama de bibliotecas para aprendizado de máquina e processamento de linguagem natural.

5.2. Hiperparâmetros do modelo

Calibramos o modelo com base em estudos anteriores (Zanuz & Rigo, 2022; Bonifacio et al., 2020). Utilizamos os modelos descritos na Seção 4.3, treinados com o *PL-corpus* para a tarefa de Reconhecimento de Entidades Nomeadas. Para

[6] https://dadosabertos.camara.leg.br/swagger/api.html

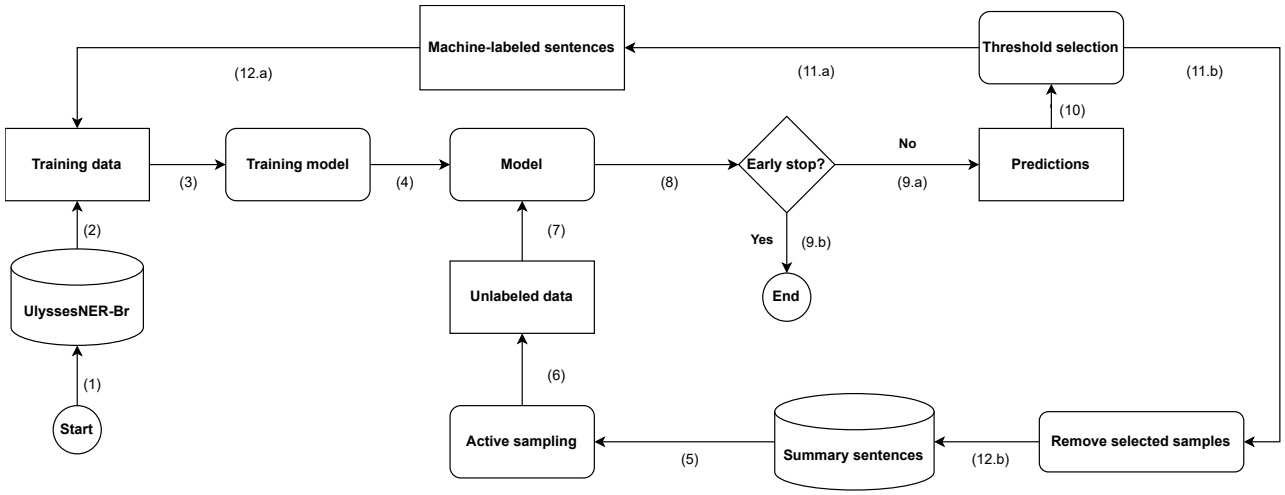


Figura 1: *Pipeline* da técnica de autoaprendizagem (Nunes et al., 2024b).

o treinamento dos modelos, utilizamos a API de treino da Huggingface[7], configurando o comprimento máximo de 512 *tokens* por sentença, com a aplicação de *padding* e truncamento conforme necessário.

Os hiperparâmetros utilizados para construir o modelo foram: *evaluation_strategy = epochs*, *save_total_limit = 5*, *learning_rate = 2e-05*, *weight_decay = 0.01*, *otimizador = Adam com betas = (0.9, 0.999) e epsilon = 1e-08*. Os demais parâmetros não especificados seguiram os valores padrão do modelo. Também configuramos o *save_strategy* para *epoch* e usamos o F1 micro como a métrica escolhida para o melhor modelo.

5.3. Hiperparâmetros do autoaprendizado

Para a amostragem ativa, utilizamos $N = 0.05$ para a porcentagem de amostras aleatórias, $K = 0.6$ para a porcentagem de amostras dissimilares e 42 como a primeira *seed* para a amostragem aleatória. Definimos $W = 5$ como o número máximo de novas amostragens aleatórias para aumentar a quantidade de dados de treinamento. Também aplicamos uma paciência igual a 3 para aguardar um aumento no F1 geral durante as iterações de autoaprendizado. Usamos 0,99 como *threshold* de confiança médio para as previsões seguindo o valor encontrado por Nunes et al. (2024b).

5.4. Métricas

Utilizamos a biblioteca seqeval[8] para calcular as métricas. Um aspecto interessante dessa biblioteca é que ela computa os resultados com base na sequência de tags para cada entidade (começando com um "B-TAG" seguido de "I-TAG"); para uma sentença completa, é essencial reconhecer claramente uma entidade, em vez de apenas um *token* específico.

Calculamos as seguintes métricas: F1-Score, precisão, revocação e acurácia (somente para o caso geral). Optamos pelo F1-Score como a principal métrica para nossas análises, pois desejamos equilibrar a previsão correta da classe positiva e a precisão dessa classe.

5.5. Métodos de avaliação de desempenho

A escolha da técnica de validação é fundamental para garantir uma avaliação precisa do desempenho do modelo. Para comparar as diferentes versões dos *sub-corpora*, utilizamos a Validação Cruzada Repetida Estratificada com K-partições, que proporciona uma análise mais robusta e reduz a variação nos resultados. Já a Validação Cruzada Estratificada com K-partições e Retenção foi aplicada no processo de autoaprendizagem, permitindo ajustar o modelo de forma eficaz e selecionar o *threshold* ideal com base no desempenho das diferentes partições.

Validação Cruzada Repetida Estratificada com K-partições (*Repeated Stratified K-fold Cross Validation*). Para avaliar o impacto da descontaminação em *sub-corpus*, essa técnica é fundamental para garantir resultados mais confiáveis. A validação cruzada estratificada assegura a distribuição equilibrada das classes em cada partição, enquanto a repetição do processo reduz a dependência de divisões aleatórias, resultando em uma média e desvio padrão mais representativos do desempenho do modelo (Kim, 2009; Machado et al., 2015).

[7] https://huggingface.co/docs/transformers/en/main_classes/trainer

[8] https://github.com/chakki-works/seqeval

Essa abordagem proporciona uma avaliação mais robusta, permitindo comparações diretas mais precisas entre diferentes versões do *corpus* (Kim, 2009). A repetição da validação também facilita a realização de testes estatísticos mais sólidos, favorecendo uma análise objetiva dos efeitos da descontaminação, com menor influência de variações aleatórias.

Validação Cruzada Estratificada com K–partições e Retenção (*Holdout Stratified K-Fold Cross-Validation*). Seguindo a abordagem previamente descrita, realizamos um *benchmark* ajustando o modelo BERT para a tarefa de NER utilizando um *corpus* legislativo em português (Albuquerque et al., 2022). O treinamento começou com o ajuste fino inicial do modelo através de uma validação cruzada estratificada de cinco partições, a fim de determinar o valor ideal do *threshold* de autoaprendizagem, conforme descrito na Subseção 5.3. Em seguida, aplicamos o *threshold* em cada partição, conforme mencionado na Seção 4.5.

É importante destacar que o autoaprendizado gera métricas para os classificadores em cada iteração. Em vez de depender apenas do classificador final do *pipeline*, adotamos uma abordagem mais robusta, selecionando as métricas do classificador que apresentou o melhor F1-Score geral. Dentro de cada partição, o *threshold* para o desempenho ideal foi determinado com base no F1-Score mais alto. Para estabelecer o melhor F1-Score para o modelo final, selecionamos o melhor F1-Score em torno das 5 partições, usando a média e o desvio padrão para identificar o *threshold* que consistentemente produziu resultados superiores. Utilizamos esse *threshold* durante a fase de treinamento *holdout*.

5.6. Testes estatísticos

Para ter uma melhor comparação dos modelos, aplicamos testes estatísticos. Para isso, foi utilizada a média do F1-Score de cada classe como um vetor para representar a distribuição do resultado do modelo, seguindo a metodologia utilizada por Gomes et al. (2021) e Nunes et al. (2024a). Com isso, aplicamos o teste de Shapiro-Wilk (Shapiro & Wilk, 1965) para verificar a normalidade, e identificamos que as distribuições não são normais em ambos os *sub-corpora*, tanto com contaminação quanto sem.

Diante da não normalidade dos dados, utilizamos o teste de Friedman (Friedman, 1937) em combinação com o teste *post-hoc* de Nemenyi (Nemenyi, 1963) para comparar os modelos dentro da mesma versão de um *sub-corpus* específico, avaliando simultaneamente múltiplos modelos. O teste de Wilcoxon Signed-Rank (Wilcoxon, 1945), por sua vez, foi aplicado para comparar o desempenho de cada modelo entre as versões contaminada e descontaminada de um mesmo *sub-corpus*, realizando uma comparação direta entre as versões de um único modelo.

6. Resultados e discussão

Nesta seção, analisamos o desempenho dos modelos empregados em nossos experimentos com o *corpus* atualizado, seguindo o nível de detalhamento utilizado em trabalhos anteriores (Albuquerque et al., 2022, 2023; Garcia et al., 2024; Nunes et al., 2024b), focando nas métricas macro. Incluímos testes estatísticos para identificar possíveis diferenças significativas entre os resultados e comparamos nossos achados com a literatura existente. Além disso, consideramos o impacto potencial do vazamento de dados, que pode influenciar a interpretação dos resultados obtidos. Por fim, apresentamos os resultados da aplicação de autoaprendizagem no *sub-corpus* de categorias. Devido à importância e ao custo computacional da validação utilizada para avaliar a contaminação e descontaminação dos *sub-corpora*, limitamos a avaliação à versão descontaminada (Nunes et al., 2024b), pois não encontramos diferença significativa entre as versões.

6.1. Análise dos resultados do *sub-corpus* de categorias

Analisando a Tabela 7 com os experimentos sobre o *corpus* descontaminado, identificamos uma variação significativa no F1-Score entre o melhor e o pior modelo, com médias que vão de 69,59% até 83,42%. Essa variação indica que, apesar de alguns modelos terem desempenho destacado, a eficácia em NER pode ser altamente variável.

Ao examinarmos os modelos com melhor desempenho, não encontramos um padrão claro que indique que modelos com pré-treinamento continuado no contexto legal superem aqueles com pré-treinamento específico para o português ou multilíngues. Os dois modelos que compartilham a arquitetura XLM-RoBERTa (Conneau et al., 2019) apresentam médias 81,99% (XLM-R-base, desvio padrão 2,41) e 82,51% (RoBERTaLexPT, desvio padrão 1,83), englobando tanto modelos em português com dados legais quanto modelos multilíngues. Em relação a esta arquitetura, vemos uma tendência de o resultado melhorar com dados mais próximos aos da tarefa, tanto com uma maior média quanto com um menor desvio padrão, seja pelo idioma ou seja pelo domínio.

Modelo	Precisão	Revocação	F1
BERTimbau-*large*	80,71 ± 3,22	86,38 ± 2,45	83,42 ± 2,50
RoBERTaLexPT	79,33 ± 2,51	86,00 ± 1,93	82,51 ± 1,83
BERTimbau-*base*	79,70 ± 2,31	85,50 ± 1,95	82,49 ± 1,92
LegalBert-pt_FP	79,34 ± 3,37	85,60 ± 2,13	82,33 ± 2,58
XLM-R-*base*	78,59 ± 3,40	85,76 ± 1,85	81,99 ± 2,41
BERTimbaulaw	78,20 ± 2,47	85,39 ± 2,33	81,62 ± 2,10
Jurisbert	75,21 ± 2,90	82,10 ± 2,47	78,48 ± 2,34
LegalBert-pt_SC	69,45 ± 3,38	76,95 ± 3,25	72,98 ± 3,00
BERTikal	66,34 ± 3,36	73,25 ± 2,87	69,59 ± 2,78

Tabela 7: Resultados da avaliação dos modelos no ***sub-corpus* descontaminado de categorias**.

Modelo	Precisão	Revocação	F1
BERTimbau-*large*	81,57 ± 2,31	87,06 ± 2,25	84,21 ± 2,06
BERTimbau-*base*	81,01 ± 2,29	86,10 ± 2,12	83,46 ± 1,89
LegalBert-pt_FP	80,31 ± 2,42	85,92 ± 2,42	83,00 ± 2,07
XLM-R-*base*	79,32 ± 2,14	85,56 ± 2,52	82,31 ± 2,10
RoBERTaLexPT	78,84 ± 3,11	85,93 ± 2,44	82,21 ± 2,47
BERTimbaulaw	78,93 ± 2,34	85,30 ± 2,62	81,97 ± 2,17
Jurisbert	76,56 ± 2,82	82,78 ± 2,33	79,52 ± 2,16
LegalBert-pt_SC	70,69 ± 2,79	77,57 ± 2,71	73,93 ± 2,15
BERTikal	69,15 ± 2,74	74,77 ± 2,67	71,83 ± 2,46

Tabela 8: Resultados da avaliação dos modelos no ***sub-corpus* contaminado de categorias**.

Em contraste, dois modelos apresentam médias em torno de 70%: BERTikal (média 69,59%) e LegalBert-pt SC (média 72,98%) com os maiores desvios padrões (2,78 e 3,00, respectivamente). Neste grupo observamos somente modelos BERT com pré-treino em textos legais, demonstrando que nem sempre o modelo com o pré-treino dentro do domínio dos textos da tarefa possui o melhor desempenho.

O modelo BERTimbau, que já havia demonstrado resultados de estado da arte para NER em português e no domínio legal (Souza et al., 2020; Zanuz & Rigo, 2022; Nunes et al., 2024b; Garcia et al., 2024), exibe uma média de desempenho alta em suas versões *base* e *large* com, respectivamente, 82,49% e 83,42%. É interessante destacar que o modelo *base* do BERTimbau alcançou resultados comparáveis à versão *large* com uma diferença de F1-Score de 0,93, além de um menor desvio padrão, apresentando uma tendência de maior robustez para o aprendizado desta tarefa. O que nos leva a sugerir que modelos menores ainda são competitivos para a tarefa de NER em detrimento de suas versões maiores.

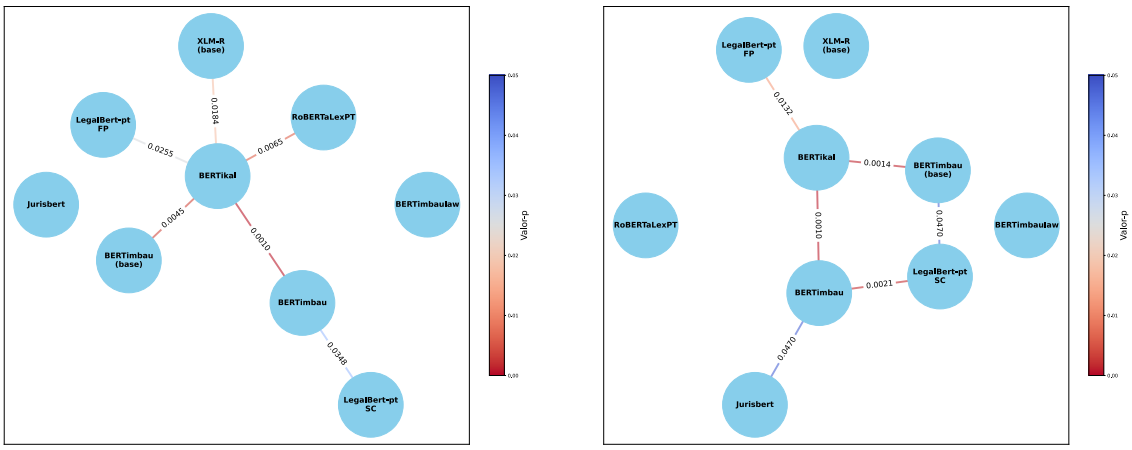
Ainda que o modelo BERTimbau *large* tenha obtido a maior média, observamos que ao levar em conta o *trade-off* entre média e desvio padrão, o modelo RoBERTaLexPT se sobressai, apresentando uma média próxima com diferença de 0,91 e menor desvio padrão com diferença de 0,67. Essa maior estabilidade no desvio padrão sugere que o modelo RoBERTaLexPT pode tender a ser mais adequado ao contexto legal, onde a consistência nos resultados é fundamental.

Em relação à comparação do desempenho dos modelos aplicados à versão descontaminada do *sub-corpus* de categorias, a Figura 2a mostra que, com o teste de Friedman seguido do *post-hoc* de Nemenyi, o modelo com maior diferença estatística em comparação aos demais é o BERTikal, que apresenta a menor média de F1-Score entre os modelos analisados (ver Tabela 7). Além disso, o BERTimbau, que obteve a maior média, demonstrou uma diferença significativa não apenas em relação ao BERTikal, mas também ao LegalBert-pt_sc, que ficou com o segundo pior F1-Score. Esses resultados indicam uma tendência favorável ao BERTimbau como o modelo preferido.

No entanto, é importante destacar que, embora o BERTimbau tenha mostrado um desempenho superior nos testes, sendo significativamente melhor que dois modelos (BERTikal e LegalBert-pt_SC), o valor-p em relação ao LegalBert-pt_SC é ligeiramente mais elevado, o que sugere uma diferença com menor confiança, apesar de ainda ser significativa. Mesmo assim, o BERTimbau e o RoBERTaLexPT apresen-

Modelo	F1 (Contaminado)	F1 (Descontaminado)	Δ F1	Valor-p
BERTimbau-*large*	84,21 ± 2,06	83,42 ± 2,50	-0,79	0,578125
RoBERTaLexPT	82,21 ± 2,47	82,51 ± 1,83	+0,30	0,687500
BERTimbau-*base*	83,46 ± 1,89	82,49 ± 1,92	-0,97	0,578125
LegalBert-pt_FP	83,00 ± 2,07	82,33 ± 2,58	-0,67	0,937500
XLM-R-*base*	82,31 ± 2,10	81,99 ± 2,41	-0,32	0,296875
BERTimbaulaw	81,97 ± 2,17	81,62 ± 2,10	-0,65	0,578125
JurisBert	79,52 ± 2,16	78,48 ± 2,34	-1,04	0,578125
LegalBert-pt_SC	73,93 ± 2,15	72,98 ± 3,00	-0,95	0,687500
BERTikal	71,83 ± 2,46	69,59 ± 2,78	-2,24	0,115851

Tabela 9: Resultados da avaliação dos modelos no ***sub-corpus* de categorias** com e sem contaminação, incluindo a diferença na métrica F1 e o valor-p obtido pelo teste de Wilcoxon Signed-Rank.



(a) Versão descontaminada. **(b)** Versão contaminada.

Figura 2: Gráficos de rede que ilustram a comparação entre os modelos, utilizando o ***sub-corpus* de categorias**. As arestas conectam os modelos que apresentam diferenças significativas, sendo o valor-p mapeado para cor da aresta. Nodos desconectados não tem diferenças significativas para os demais. As figuras com os resultados completos estão disponíveis no Apêndice A.

tam desempenhos bastante semelhantes, com diferenças estatísticas próximas. Como discutido, o RoBERTaLexPT se destaca pela maior estabilidade, apresentando uma média de F1-Score similar à do BERTimbau, mas com um desvio padrão menor, o que sugere maior consistência nos resultados. Esses fatores indicam que a escolha entre os dois modelos pode depender de aspectos como variabilidade dos resultados, estabilidade e o tamanho do modelo. O RoBERTaLexPT pode ser mais adequado em cenários que exigem um modelo mais leve e estável. Por outro lado, o BERTimbau, devido ao seu treinamento com uma maior diversidade de dados e ao seu uso mais amplo, oferece uma capacidade de generalização superior, tornando-o uma escolha robusta, especialmente em tarefas com dados variados ou quando há uma quantidade limitada de dados disponíveis.

Por fim, as análises sugerem que a performance dos modelos de NER não depende unicamente do tamanho ou do tipo de pré-treinamento. A relação entre o alinhamento com o domínio e a arquitetura é crucial, e modelos menores podem, em alguns casos, apresentar menor variância e resultados estáveis. Essa discussão ressalta a necessidade de uma avaliação cuidadosa ao selecionar modelos para tarefas de NER, onde a interação entre arquitetura, pré-treinamento e alinhamento ao domínio são fatores importantes.

Comparando a versão contaminada e descontaminada. Comparando os resultados das versões contaminada e descontaminada do *sub-corpus*, apresentados na Tabela 9, observa-se que a maioria dos modelos, exceto o RoBERTaLexPT, apresentou uma redução de aproximadamente 1% no F1-Score. Contudo, todos os mode-

los exibiram valores de p acima de 0,05, indicando que não houve diferença significativa no desempenho entre as versões com e sem contaminação. Os resultados com as demais métricas detalhadas de cada versão do *corpus* podem sem encontradas nas Tabelas 7 e 8.

A ausência de significância estatística na diferença dos resultados dos modelos entre as versões contaminada e descontaminada do *sub-corpus* pode ser atribuída a duas hipóteses: (i) as instâncias removidas não impactavam significativamente o desempenho dos modelos ou (ii) a eliminação do vazamento de instâncias duplicadas, aliada à correção de anotações ambíguas e parciais, gerou um equilíbrio entre a diminuição do F1-Score pela remoção das duplicatas e o aumento do F1-Score pela melhoria na qualidade das anotações, resultando em uma performance praticamente inalterada.

Neste estudo, não avaliamos os resultados em diferentes cenários com variações no tipo de contaminação corrigida. No entanto, esse é um experimento que pode ser conduzido no futuro para compreender melhor o impacto de cada nível de contaminação.

Embora não tenha sido encontrada diferença significativa entre os modelos nas versões contaminada e descontaminada do *sub-corpus*, foi possível observar variações significativas entre os modelos dentro de cada versão, quando comparados entre si. A Figura 2b apresenta uma topologia distinta da versão descontaminada (Figura 2a), evidenciada pela redução de conexões entre os nós e pelo aumento de nós desconectados. Por exemplo, o modelo BERTikal possui três arestas na versão contaminada, enquanto apresenta cinco na versão descontaminada.

Além disso, diferenças estatísticas entre os modelos também variam conforme a versão do *sub-corpus*. Na versão contaminada, os modelos XLM-R (base) e RoBERTaLexPT não apresentam diferenças significativas entre os demais modelos, mas passam a apresentar na versão descontaminada. Da mesma forma, o BERTimbau (*large*) apresenta diferença significativa em relação ao Jurisbert, enquanto o BERTimbau (base) ganha diferença significativa em relação ao LegalBert-pt SC.

Assim, observamos que, apesar da ausência de diferença entre as versões dos modelos, a relação entre os modelos muda. Ou seja, as versões contaminada e descontaminada do *sub-corpus* fornecem diferentes perspectivas sobre o desempenho relativo dos modelos, podendo alterar quais modelos se destacam em cada cenário.

6.2. Análise dos resultados do *sub-corpus* de tipos

Analisando os resultados, encontramos alguns pontos importantes, especialmente quando comparada ao nível de categorias. A Tabela 7 demonstra que, embora o desempenho geral do *sub-corpus* descontaminado tenha caído em relação às categorias, isso é esperado devido à maior complexidade do *sub-corpus* de tipos, que inclui 16 classes especializadas, enquanto o *sub-corpus* de categorias tem apenas 7 classes mais amplas. Essa maior granularidade torna a tarefa de classificação mais desafiadora, justificando a queda nos valores de F1-Score.

Os resultados do *sub-corpus* de tipos revelam uma discrepância semelhante à observada na Subseção 6.1, com o pior modelo alcançando um F1-Score de 63,36% e o melhor, 78,59%, apresentando uma diferença de 15,23 pontos percentuais. Essa variação reforça a disparidade na eficácia dos diferentes modelos de NER, evidenciando que a escolha do modelo impacta significativamente o desempenho na tarefa.

Sobre o melhor modelo, não é possível determinar qual se destaca de forma clara com base apenas nas médias. Dos 9 modelos avaliados, 7 apresentaram médias acima de 70%, com a maior sendo 78,59% e outros cinco modelos variando entre 76,31% e 79,79%. Dessa forma, no nível de tipos, também não é possível concluir que modelos específicos para o domínio legislativo ou jurídico apresentam desempenho superior, nem que modelos desenvolvidos exclusivamente para o português superam os multilíngues em termos de resultados.

Em relação ao desvio padrão, o BERTimbau (versão *large*) novamente se destaca por apresentar o maior desvio padrão entre os melhores modelos, assim como observado no nível de categorias. Neste caso, ele alcança o maior desvio padrão dentre todos os modelos avaliados. Por outro lado, o XLM-R, excetuando o BERTikal, que apresentou o pior desempenho geral, registrou o menor desvio padrão, consolidando-se como uma alternativa sólida para a tarefa, ao aliar resultados competitivos à menor variância. Essa arquitetura tem se mostrado competitiva com o BERTimbau no domínio legislativo, conforme também observado no nível de categorias (veja a Seção 6.1), especialmente em sua versão adaptada para textos legislativos em português (RoBERTaLexPT).

Modelo	Precisão	Revocação	F1
BERTimbau	75,71 ± 4,82	81,75 ± 5,16	78,59 ± 4,83
XLM-R-base	73,25 ± 2,99	80,73 ± 2,18	76,79 ± 2,43
BERTimbau-base	73,34 ± 3,18	80,26 ± 2,44	76,63 ± 2,71
LegalBert-pt_FP	73,05 ± 3,50	80,52 ± 2,83	76,59 ± 3,03
RoBERTaLexPT	73,38 ± 3,62	80,15 ± 2,83	76,59 ± 3,03
BERTimbaulaw	72,69 ± 3,01	80,34 ± 2,33	76,31 ± 2,53
Jurisbert	70,52 ± 3,71	78,11 ± 3,30	74,09 ± 3,30
LegalBert-pt_SC	64,94 ± 3,70	72,41 ± 3,60	68,43 ± 3,23
BERTikal	61,13 ± 1,97	65,79 ± 2,19	63,36 ± 1,78

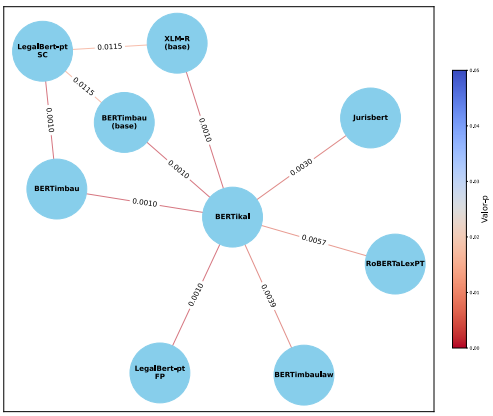
Tabela 10: Resultados da avaliação dos modelos no ***sub-corpus*** **descontaminado de tipos**.

Modelo	Precisão	Revocação	F1
BERTimbau-base	81,03 ± 03,37	86,57 ± 02,41	83,70 ± 02,82
RoBERTaLexPT	80,82 ± 02,95	85,96 ± 02,25	83,30 ± 02,48
BERTimbaulaw	80,38 ± 02,38	86,09 ± 01,83	83,13 ± 01,92
LegalBert-pt_FP	80,01 ± 03,11	86,07 ± 02,21	82,92 ± 02,50
XLM-R-base	80,09 ± 03,08	85,95 ± 02,41	82,91 ± 02,65
Jurisbert	78,47 ± 02,98	84,81 ± 02,59	81,51 ± 02,65
LegalBert-pt_SC	75,76 ± 03,80	82,09 ± 03,18	78,78 ± 03,31
BERTimbau	76.12 ± 07.86	75,03 ± 07,34	74.64 ± 07.59
BERTikal	68,73 ± 02,02	74,65 ± 02,07	71,56 ± 01,85

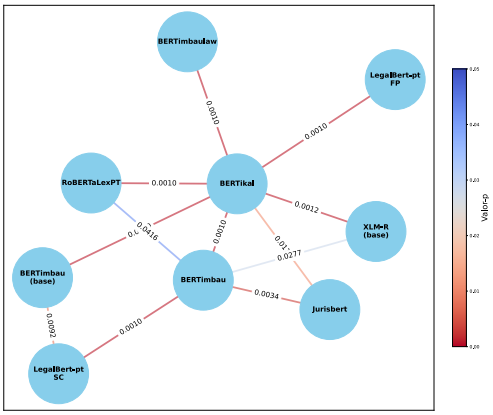
Tabela 11: Resultados da avaliação dos modelos no ***sub-corpus*** **contaminado de tipos**.

Modelo	F1 (Contaminado)	F1 (Descontaminado)	Δ F1	Valor-p
BERTimbau	74.64 ± 07.59	78,59 ± 04,83	+03.36	0,000031
XLM-R-base	82,91 ± 02,65	76,79 ± 02,43	-06,12	0,000031
BERTimbau-base	83,70 ± 02,82	76,63 ± 02,71	-07,07	0,000031
LegalBert-pt-FP	82,92 ± 02,50	76,59 ± 03,03	-06,33	0,000031
RoBERTaLexPT	83,30 ± 02,48	76,59 ± 03,03	-06,71	0,000031
BERTimbau-law	83,13 ± 01,92	76,31 ± 02,53	-06,82	0,000031
Jurisbert	81,51 ± 02,65	74,09 ± 03,30	-07,42	0,000031
LegalBert-pt-SC	78,78 ± 03,31	68,43 ± 03,23	-10,35	0,000031
BERTikal	71,56 ± 01,85	63,36 ± 01,78	-08,20	0,001474

Tabela 12: Resultados da avaliação dos modelos no ***sub-corpus*** **de tipos** com e sem contaminação, incluindo a diferença na métrica F1 e o valor-p obtido pelo teste de Wilcoxon Signed-Rank.



(a) Versão descontaminada.



(b) Versão contaminada.

Figura 3: Gráficos de rede que ilustram a comparação entre os modelos, utilizando o ***sub-corpus* de tipos**. As arestas conectam os modelos que apresentam diferenças significativas, sendo o valor-p mapeado para cor da aresta. Nodos desconectados não tem diferenças significativas para os demais. As figuras com os resultados completos estão disponíveis no Apêndice A.

Em relação à comparação do desempenho dos modelos aplicados à versão descontaminada do *subcorpus* de tipos, a Figura 3a apresenta as diferenças significativas entre os modelos, conforme os resultados do teste de Friedman seguido do *post-hoc* de Nemenyi. Observa-se que o modelo BERTikal, como esperado, exibe diferenças significativas em relação a todos os modelos com F1-Score acima de 70%, exceto o LegalBert-pt-SC, cujo desempenho é mais próximo ao seu. Além disso, os modelos BERTimbau (em ambas as versões) e XLM-R também apresentam diferenças significativas em relação ao LegalBert-pt-SC, que possui o segundo pior desempenho. Esses resultados sugerem uma tendência para a escolha dos modelos BERTimbau e XLM-R, dado que se destacam com o maior número de diferenças estatísticas e, simultaneamente, apresentam os melhores desempenhos gerais.

Comparando a versão contaminada e descontaminada. A Tabela 12 mostra a diferença dos F1-Scores entre a versão contaminada e descontaminada do *sub-corpus* com os modelos avaliados (demais completas podem ser encontradas nas Tabelas 10 e 11). Em relação ao já analisado no *sub-corpus* de categorias, chama a atenção dois pontos discrepantes: maiores diferenças e significância estatística na comparação de todos os modelos.

Tanto as diferenças quanto a significância estatística podem ser atribuídas à natureza dos *sub-corpus*. O nível de tipos apresenta uma maior complexidade de classificação devido ao maior número de classes e às especificações mais detalhadas em comparação ao nível de categorias (veja as Tabelas 1 e 6). Além disso, classes derivadas de uma mesma categoria podem compartilhar similaridades semânticas, dificultando a distinção entre elas pelos modelos. Outro fator relevante é o menor número de exemplos disponíveis no *sub-corpus* de tipos em relação ao de categorias, além de um menor número médio de exemplos por classe, o que acentua o desafio de treinamento para os modelos.

O BERTimbau (versão *large*), apesar de alcançar a maior média no *sub-corpus* descontaminado, apresentou uma média de 70,03% com desvio padrão de 7,34 no *sub-corpus* contaminado, evidenciando um desempenho inconsistente entre as duas versões do *sub-corpus*. Esse elevado desvio padrão indica uma alta variabilidade nos resultados ao longo das execuções, o que reflete um comportamento instável em relação aos demais modelos avaliados. Essa variação pode ser associada à presença de vazamento de informações no *sub-corpus* contaminado, que tende a mascarar a real capacidade de generalização do modelo.

Esse comportamento impactou diretamente a análise estatística entre os modelos no *sub-corpus* contaminado (veja Figura 3b). Enquanto no *sub-corpus* descontaminado o BERTimbau apresentou diferenças estatisticamente significativas apenas com os dois piores modelos, no contaminado ele mostrou diferenças significativas com cinco modelos, incluindo o RoBERTaLexPT e o XLM-R, que estão entre os melhores e destacados previamente. Curiosamente, uma exceção foi sua própria versão base, o BERTimbau-base, que obteve a melhor média no *sub-corpus* contaminado. Esses resultados reforçam o impacto negativo do vazamento de informações, não apenas no desempenho médio, mas também na estabilidade observada, destacando a importância de *sub-corpus* descontaminados para uma análise mais confiável e robusta.

Modelo	Garcia et al. (2024)	Contaminado	Descontaminado	Δ C	Δ D
BERTimbau-base	86,39	83,46 ± 1,89	82,49 ± 1,92	-02,96	-03,90
BERTimbau	87,77	84,21 ± 2,06	83,42 ± 2,50	-03,56	-04,35
BERTikal	79,21	71,83 ± 2,46	69,59 ± 2,78	-07,38	-09,62
JurisBert	81,67	79,52 ± 2,16	78,48 ± 2,34	-02,15	-03,19
BERTimbaulaw	87,11	81,97 ± 2,17	81,62 ± 2,10	-05,14	-05,49
RoBERTaLexPT	88,56	82,21 ± 2,47	82,51 ± 1,83	-06,35	-06,07

Tabela 13: Comparação do F1-Score macro para cada modelo no **nível de categorias** entre os experimentos de Garcia et al. (2024) e os nossos experimentos. Δ *C* faz referência à diferença do resultado de (Garcia et al., 2024) e os resultados do *sub-corpus* contaminado, enquanto Δ *D* faz referência à diferença com os resultados do *sub-corpus* descontaminado.

6.3. Comparando com a literatura

Conforme discutido na Seção 3.2, a presença de instâncias duplicadas pode provocar *overfitting* e inflar os resultados das métricas de avaliação. Esse comportamento foi observado quando comparamos nossos resultados com os da literatura (Albuquerque et al., 2023; Nunes et al., 2024b; Garcia et al., 2024).

Albuquerque et al. (2023) e Nunes et al. (2024b) avaliaram o *sub-corpus* de categorias utilizando técnicas de avaliação baseadas em cálculos de média e desvio padrão, visando obter uma análise mais robusta dos resultados. Ambos os estudos realizaram cinco execuções dos modelos para calcular essas métricas, com a principal diferença sendo que Albuquerque et al. (2023) utilizou divisões aleatórias dos dados, enquanto Nunes et al. (2024b) adotou validação cruzada com cinco partições estratificadas.

Em Albuquerque et al. (2023), foi utilizado um modelo[9] baseado no BERTimbau base, que teve seu pré-treinamento continuado com o *corpus* LeNER-Br (Luz de Araujo et al., 2018), seguido de *finetuning* no mesmo *corpus*. Já Nunes et al. (2024b) empregou o BERTimbau base, sem pré-treinamento adicional em *corpus* legal. Ambos os estudos utilizaram uma camada linear para classificar as entidades nomeadas.

Nos dois estudos, os F1-Scores macro ficaram próximos de 83%, sendo 83,17 ± 1,15 em Albuquerque et al. (2023) e 83,53 ± 2,56 em Nunes et al. (2024b). Apesar de pequenas diferenças nas médias e nos desvios padrão, os resultados demonstram uma proximidade significativa.

Garcia et al. (2024) também avaliou o BERTimbau base, mas utilizando a estratégia de *holdout* com divisão aleatória dos dados, conforme a separação disponibilizada no artigo do *corpus* (Albuquerque et al., 2022). Nesse caso, foi obtido um F1-Score macro de 86,39%, aproximadamente 3% superior aos resultados reportados por Albuquerque et al. (2023) e Nunes et al. (2024b). Essa diferença pode ser atribuída à execução única do modelo, que pode não refletir com precisão o desempenho real, ou à seleção aleatória de um conjunto de teste mais fácil, o que pode ter proporcionado um resultado otimista.

Comparando os resultados com os obtidos neste trabalho, observa-se uma pequena diferença entre eles (Albuquerque et al., 2023; Nunes et al., 2024b), no qual o resultado obtido neste trabalho com o retreinamento do modelo contamiado obteve média e desvio padrão entre os dois anteriores. Entretanto, encontramos um desempenho aproximadamente 3% inferior quando comparado ao estudo que utilizou apenas uma execução (Garcia et al., 2024). Além disto, refizemos o *finetuning* dos demais modelos testados por Garcia et al. (2024). A única exceção foi o Albertina e versões maiores dos modelos XLM-R que por limitações de hardware não puderam ser executadas.

Os resultados dos experimentos realizados no *sub-corpus* de categorias, comparando os valores reportados por Garcia et al. (2024) com os obtidos com o *corpus* atualizado, podem ser vistos na Tabela 13. Além do decréscimo já mencionado para o BERTimbau base, observa-se que os demais modelos também apresentaram quedas de desempenho, com o Jurisbert mostrando o menor decréscimo (-2,15) e o BERTikal o maior (-7,38).

Ao compararmos com os resultado sobre o *sub-corpus* descontaminado, encontramos uma maior diferença com os trabalhos anteriores. Primeiro, temos um decréscimo de cerca de 1% em relação aos trabalhos de Albuquerque et al. (2023) e Nunes et al. (2024b). Segundo, as variações comparadas com os testes usando *holdout* de Garcia et al. (2024) se tornam maiores, vari-

[9] https://huggingface.co/pierreguillou/ner-bert-base-cased-pt-lenerbr

Modelo	Garcia et al. (2024)	Contaminado	Descontaminado	Δ C	Δ D
BERTimbau	84,74	74.64 ± 07.59	78,59 ± 04,83	-10,10	-06,15
BERTimbau-base	83,83	83,70 ± 02,82	76,63 ± 02,71	-00,13	-07,20
BERTikal	75,70	71,56 ± 01,85	63,36 ± 01,78	-04,14	-12,34
JurisBert	77,97	81,51 ± 02,65	74,09 ± 03,30	+03,54	-03,88
BERTimbaulaw	84,42	83,13 ± 01,92	76,31 ± 02,53	-01,29	-08,11
RoBERTaLexPT	86,03	83,30 ± 02,48	76,59 ± 03,03	-02,73	-09,44

Tabela 14: Comparação do F1-Score macro para cada modelo no **nível de tipos** entre os experimentos de Garcia et al. (2024) e os nossos experimentos. Δ *C* faz referência a diferência do resultado de (Garcia et al., 2024) e os resultados do *sub-corpus* contaminado, enquanto Δ *D* faz referência a diferência com os resultados do *sub-corpus* descontaminado.

ando entre 3,19% e 9,62%. Os decréscimos apresentados em ambas as versões do *sub-corpus* reinteram a importância de múltiplas execuções de modelos para se obter uma faixa de resultados mais confiável.

Embora os modelos BERTimbau e RoBERTaLexPT tenham alcançado as maiores médias, os testes estatísticos (ver Subseção 6.1) indicam que não há uma diferença significativa entre eles e os demais modelos testados por Garcia et al. (2024), com poucas exceções, como o modelo BERTikal. No estudo original, os autores não realizaram testes estatísticos; contudo, dada a proximidade dos resultados, é provável que não haja uma diferença significativa entre o RoBERTaLex (que apresentou o melhor desempenho) e os demais modelos. Essa situação é semelhante à observada em nosso trabalho, reforçando a ideia de que o desempenho entre os modelos é bastante próximo, excetuando-se *outliers* como o BERTikal.

O mesmo se aplica ao *sub-corpus* de tipos, onde identificamos apenas experimentos utilizando modelos *transformers* no trabalho de Garcia et al. (2024). A comparação entre os resultados desse estudo e os nossos pode ser vista na Tabela 14. Nela, observamos uma diferença menos acentuada entre os resultados obtidos com o *sub-corpus* original e aqueles com o *sub-corpus* contaminado em nosso experimentos, sendo o BERTimbau base o modelo com o menor decréscimo de desempenho (-00,13) e o BERTikal apresentando o maior decréscimo (-04,14), ademais houve um acréscimo no resultado do JurisBert de +03,54 pontos percentuais. Este acréscimo reforça a importância da utilização de métodos que testam o modelo repetidamente, como a validação cruzada, visto que métodos como *holdout* podem não somente proporcionar resultados maiores que o esperado, mas também menores, tal como visto no exemplo do JurisBert.

Ao compararmos os resultados de (Garcia et al., 2024) nas versões contaminada e descontaminada do *sub-corpus* no nível de tipos, observamos uma variação significativamente maior do que a registrada no nível de categorias (veja Tabela 13). Uma possível explicação para essa maior variação é o menor número de instâncias no nível de tipos (3.006 entidades em 2.424 sentenças) em comparação ao nível de categorias (3.537 entidades em 2.468 sentenças). Além disso, o nível de tipos possui um número maior de classes (16) em relação ao nível de categorias (7), o que torna a tarefa de classificação mais complexa. Essa combinação de menor quantidade de dados e maior número de classes amplifica o impacto da eliminação de vazamento, resultando em maior variação nos resultados.

6.4. Utilização de autoaprendizagem

A seleção do valor do *threshold* é um aspecto crítico de nossa abordagem, pois desempenha um papel fundamental na determinação do desempenho geral do sistema de NER. Na fase de validação cruzada, nossa avaliação revelou que o valor do *threshold* de 0,99 demonstrou consistentemente desempenho superior, resultando em um F1-Score na faixa de 86,70 ± 2,28 em todos as partições. É importante destacar que *thresholds* de 0,95 e 0,975 apresentam resultados semelhantes, conforme mostrado na Tabela 15. Assim, para escolher o *threshold* entre eles, selecionamos aquele que apresenta maior ganho na maioria das entidades. Baseamos essa escolha no melhor F1-Score obtido na validação cruzada, conforme detalhado na Seção 5.5, o que provou ser a decisão mais eficaz nos cinco folds. Consequentemente, os resultados apresentados nesta seção abrangem as métricas finais obtidas com o *threshold* definido em 0,99.

A Tabela 15 mostra o impacto do aprendizado automático nos resultados finais. Esta tabela apresenta o F1-Score para as classes de entidades. Nossa abordagem foi capaz de alcançar resultados significativamente superiores para a maioria das classes, como pode ser observado na entidade "LOCAL", que teve um aumento de 8% em seu F1-Score. De forma semelhante, "ORGANIZACAO" apresentou um aumento de 6% e uma redução no desvio padrão, assim como "PRODUTODELEI", que teve um incremento de 5% e também uma redução no desvio padrão. É interessante destacar a entidade "EVENTO", que não pôde ser prevista com os dados originais, apresentando um F1-Score de 0,0 ± 0,0, e conseguimos alcançar 58,10 ± 34,16. As entidades "DATA", "FUNDAMENTO" e "PESSOA" não apresentaram impactos significativos, registrando apenas pequenos aumentos na média ou reduções no desvio padrão.

No artigo original do UlyssesNER-Br (Albuquerque et al., 2022), os autores utilizaram o *corpus* para treinar um modelo de HMM e um modelo de CRF. Além disso, também utilizaram a arquitetura BiLSTM-CRF e a arquitetura Glove para comparar com os resultados obtidos no trabalho de Luz de Araujo et al. (2018) com o *corpus* LeNER-Br. Nesse sentido, a Tabela 16 demonstra os resultados superiores, nos quais apenas o uso de um modelo BERT ajustado para o português (Souza et al., 2020) conseguiu aumentar o F1-Score em 6,64%. No entanto, a introdução do aprendizado automático se mostrou um fator importante no aumento do F1-Score em 9,81%.

Para realizar uma análise mais aprofundada dos resultados, realizamos um treinamento final com os dados utilizados na validação cruzada e os testamos em um conjunto de dados previamente não utilizado. A abordagem de validação cruzada com divisão serviu a um duplo propósito: ela não apenas ajudou no ajuste do hiperparâmetro do *threshold*, mas também ofereceu uma maneira mais abrangente de validar os resultados com um número predefinido de divisões. Além disso, essa abordagem possibilitou uma análise detalhada dos resultados no conjunto de teste, com atenção especial ao impacto e às consequências de cada categoria.

A curva de aprendizado para cada entidade, conforme ilustrado na Figura 4, mostra o impacto significativo do autoaprendizado nas classes, resultando em um aumento considerável na métrica em comparação com o resultado padrão na iteração zero. No entanto, vale ressaltar que algumas oscilações foram observadas em iterações específicas, possivelmente devido à introdução de

Threshold	DATA	EVENTO	FUNDAMENTO	LOCAL	ORGANIZACAO	PESSOA	PRODUTODELEI	F1 macro
0,9	94,49 ± 03,13	48,89 ± 33,41	88,01 ± 02,12	85,54 ± 03,50	82,56 ± 05,95	83,98 ± 04,06	75,11 ± 05,72	85,02 ± 02,45
0,925	94,62 ± 02,65	54,53 ± 32,68	89,34 ± 01,61	85,10 ± 04,34	83,16 ± 04,49	84,40 ± 03,55	71,36 ± 04,14	85,12 ± 02,31
0,95	**95,08 ± 01,89**	49,05 ± 33,86	89,99 ± 02,45	87,49 ± 05,03	**84,82 ± 02,57**	85,55 ± 05,24	76,01 ± 07,18	**86,56 ± 1,99**
0,975	95,08 ± 03,41	50,48 ± 34,02	**90,50 ± 02,54**	85,11 ± 04,25	85,30 ± 05,75	85,75 ± 04,83	75,83 ± 05,94	**86,48 ± 02,91**
0,99	94,77 ± 02,65	**58,10 ± 34,16**	88,60 ± 02,29	**86,46 ± 03,73**	84,89 ± 05,77	**87,48 ± 02,79**	**75,42 ± 04,47**	**86,70 ± 02,28**
0,9975	93,35 ± 02,66	03,64 ± 07,27	88,91 ± 02,24	80,78 ± 02,89	79,12 ± 04,33	87,91 ± 03,32	72,81 ± 05,65	84,16 ± 02,28
0,999	93,10 ± 02,21	20,00 ± 24,49	88,37 ± 01,65	80,87 ± 04,06	79,73 ± 02,94	87,22 ± 03,02	70,74 ± 09,06	83,87 ± 02,36
Standard	94,25 ± 02,69	00,00 ± 00,00	88,59 ± 03,86	78,96 ± 03,90	78,33 ± 04,44	87,77 ± 03,19	70,44 ± 07,40	83,53 ± 02,56

Tabela 15: Resultados da validação cruzada para cada *threshold* com autoaprendizagem e o resultado sem autoaprendizagem.

Modelo	Acurácia	Precisão	Revocação	F1-Score
HMM	93,07 ± 00,78	60,45 ± 02,18	30,82 ± 01,81	40,74 ± 01,83
CRF	97,27 ± 00,77	83,42 ± 00,91	70,40 ± 01,54	76,28 ± 01,12
BiLSTM-CRF + Glove	97,66 ± 00,47	80,48 ± 02,69	73,63 ± 02,65	76,89 ± 02,49
BERTimbau	98,30 ± 00,32	80,17 ± 03,67	87,63 ± 01,13	83,53 ± 02,56
BERTimbau + Autoaprendizagem	**98,45 ± 00,24**	**85,37 ± 02,91**	**89,02 ± 01,45**	**86,70 ± 02,28**

Tabela 16: Resultados originais e nossos resultados com BERT e *self-learning* utilizando o *threshold* de 0,99.

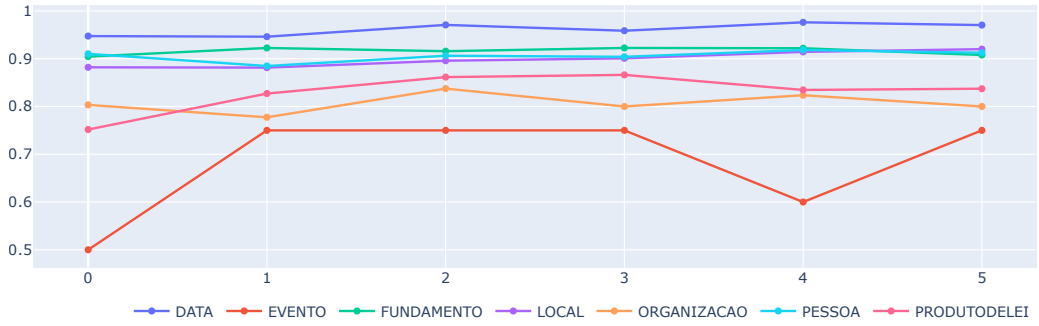


Figura 4: F1-Score para cada entidade ao longo das iterações.

exemplos mal anotados no conjunto de treinamento. Apesar disso, a tendência geral demonstra a robustez do uso do autoaprendizado e destaca a influência do *threshold* escolhido na filtragem de uma parte substancial dos dados ruidosos.

Da mesma forma, a Figura 5a ilustra a curva de aprendizado para o número acumulado de sentenças adicionadas ao longo das iterações, com foco no F1-Score geral. Na quarta iteração, alcançamos nosso maior F1-Score de 90%, destacando o impacto positivo do aumento do *corpus* original. Esse resultado é bastante promissor em comparação com o F1-Score de 87,28% obtido usando apenas o modelo BERT na iteração zero.

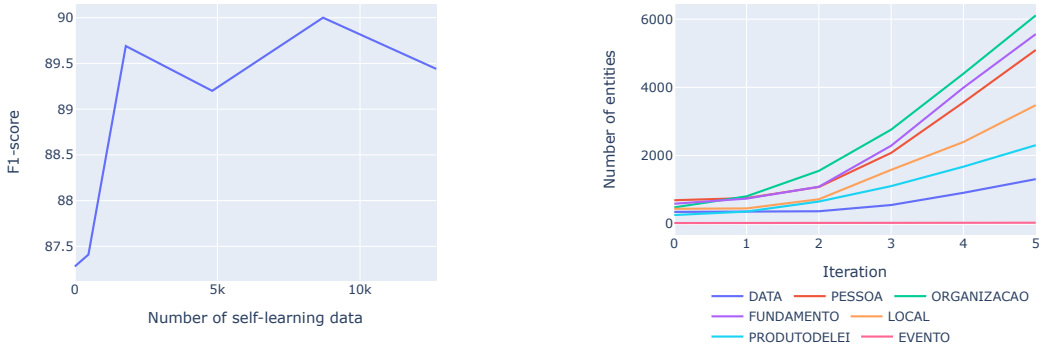
A Figura 5b mostra o aumento no número de exemplos para cada entidade durante as iterações. Vale ressaltar que tanto as classes com mais quanto as com menos dados apresentaram um aumento considerável no número de exemplos. Mesmo assim, as classes "DATA" e "EVENTO" tiveram o menor aumento. Acreditamos que esse fato ocorreu porque as datas possuem formatos específicos, facilitando a filtragem de ruído, e "EVENTO", por ser a classe minoritária, aumentou ligeiramente ao longo das iterações justamente devido ao seu pequeno número de dados.

7. Conclusão

Neste trabalho, abordamos de maneira abrangente o problema do vazamento de dados no treinamento de modelos de Reconhecimento de Entidades Nomeadas (NER) em textos legislativos em português brasileiro. Identificamos como duplicatas e anotações inconsistentes comprometem a avaliação dos modelos, levando a resultados superestimados. Através de um processo cuidadoso de correção, removemos duplicatas e padronizamos as anotações, resultando em um *corpus* de maior qualidade, o que nos permitiu realizar um novo *benchmark*.

Essas melhorias no tratamento dos dados proporcionaram uma avaliação mais precisa dos modelos e destacaram a importância de uma abordagem rigorosa no pré-processamento de *corpora* para tarefas de Processamento de Linguagem Natural (PLN). A pesquisa ressalta que o vazamento de dados é um aspecto relevante que merece atenção para assegurar a validade dos resultados em contextos especializados.

Ademais, apresentamos um método de NER utilizando autoaprendizagem e amostragem ativa, aplicando-o ao texto legislativo em português do *corpus* UlyssesNER-BR. Nossos resultados mostram que o BERTimbau com autoaprendizagem obteve uma F1-score média geral de 86,70 ± 2,28 na validação cruzada e um resultado final de 90%, demonstrando um desempenho robusto no reconhecimento de entidades em comparação ao uso apenas do BERTimbau e aos *benchmarks* anteriores. Esse achado destaca a eficácia do BERTimbau com autoaprendizagem para o Reconhecimento de Entidades Nomeadas no domínio legal/legislativo, evidenciando seu potencial para tarefas de análise de texto jurídico.



(a) Curva de aprendizado mostrando a relação entre o número cumulativo de dados adicionados em cada iteração e seu respectivo F1-Score.

(b) Número cumulativo de exemplos de cada entidade adicionados em cada iteração.

Figura 5: Impacto da autoaprendizagem na fase de treinamento.

Apesar dos resultados positivos, nosso estudo em autoaprendizagem apresenta algumas limitações. Realizamos os experimentos apenas no nível de categoria de entidade, sem avaliar o comportamento do modelo no nível de tipo, nem o impacto da contaminação na autoaprendizagem. Como trabalho futuro, planejamos explorar o nível de tipo e as correlações entre os níveis, além de investigar o efeito da contaminação. Também pretendemos adotar uma abordagem de ensemble do nosso modelo com o BERTimbau afinado com o *corpus* LeNER-Br[10] utilizando entidades equivalentes entre os *corpora* no ensemble. Em relação ao ajuste fino dos modelos, planejamos realizar experimentos com as versões BERT-CRF e BERT-LSTM-CRF do BERTimbau, disponíveis no repositório oficial[11]. Também pretendemos realizar experimentos com outros modelos recentes baseados em BERT para o português, como o Albertina (Rodrigues et al., 2023). Nossos resultados também destacam a importância de um conjunto de dados diverso e representativo para o ajuste fino de modelos em domínios específicos. Pesquisas futuras devem focar na ampliação dos dados de treinamento, na curadoria de novos dados com especialistas para torná-los disponíveis para uso geral e na exploração de outras técnicas de pré-treinamento e ajuste fino para melhorar o desempenho de modelos de NER no domínio legislativo.

[10] https://huggingface.co/Luciano/bertimbau-large-lener_br

[11] https://github.com/neuralmind-ai/portuguese-bert/tree/master/ner_evaluation

Agradecimentos

Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Também agradecemos o apoio financeiro da agência de fomento brasileira CNPq. Alguns experimentos neste trabalho utilizaram a infraestrutura PCAD, http://gppd-hpc.inf.ufrgs.br, no INF/UFRGS.

Referências

Albuquerque, Hidelberg O., Rosimeire Costa, Gabriel Silvestre, Ellen Souza, Nádia F. F. da Silva, Douglas Vitório, Gyovana Moriyama, Lucas Martins, Luiza Soezima, Augusto Nunes, Felipe Siqueira, João P. Tarrega, Joao V. Beinotti, Marcio Dias, Matheus Silva, Miguel Gardini, Vinicius Silva, André C. P. L. F. de Carvalho & Adriano L. I. Oliveira. 2022. UlyssesNER-Br: a corpus of brazilian legislative documents for named entity recognition. Em *Computational Processing of the Portuguese Language (PROPOR)*, 3–14. 10.1007/978-3-030-98305-5_1

Albuquerque, Hidelberg O., Ellen Souza, Adriano L. I. Oliveira, David Macêdo, Cleber Zanchettin, Douglas Vitório, Nádia F. F. da Silva & André C. P. L. F. de Carvalho. 2023. On the assessment of deep learning models for named entity recognition of brazilian legal documents. Em *EPIA Conference on Artificial Intelligence*, 93–104. 10.1007/978-3-031-49011-8_8

Alves-Pinto, Ana, Christoph Demus, Michael Spranger, Dirk Labudde & Eleanor Hobley. 2021. Iterative named entity recognition with conditional random fields. *Applied Sciences* 12(1). 330. 10.3390/app12010330

Anaby-Tavor, Ateret, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper & Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! Em *AAAI Conference on Artificial Intelligence*, vol. 34 05, 7383–7390. 10.1609/aaai.v34i05.6233

Angelidis, Iosif, Ilias Chalkidis & Manolis Koubarakis. 2018. Named entity recognition, linking and generation for greek legislation. Em *International Conference on Legal Knowledge and Information Systems (JURIX)*, 1–10. 10.3233/978-1-61499-935-5-1

Badji, Ines. 2018. *Legal entity extraction with NER systems*: Universidad Politécnica de Madrid. Tese de Doutoramento.

Balloccu, Simone, Patrícia Schmidtová, Mateusz Lango & Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. Em *18th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 67–93.

Bird, Steven. 2006. NLTK: the natural language toolkit. Em *21st International Conference on Computational Linguistics (COLING) and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, 69–72. 10.3115/1225403.1225421

Bonifacio, Luiz Henrique, Paulo Arantes Vilela, Gustavo Rocha Lobato & Eraldo Rezende Fernandes. 2020. A study on the impact of intradomain finetuning of deep language models for legal named entity recognition in Portuguese. Em *Intelligent Systems*, 648–662. 10.1007/978-3-030-61377-8_46

Brito, Maurício, Vládia Pinheiro, Vasco Furtado, João Araújo Monteiro Neto, Francisco das Chagas Jucá Bomfim, André Câmara Ferreira da Costa & Raquel Silveira. 2023. CDJUR-BR: Uma coleção dourada do judiciário Brasileiro com entidades nomeadas refinadas. Em *14th Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)*, 177–186. 10.5753/stil.2023.234217

Chen, Yukun, Thomas A. Lasko, Qiaozhu Mei, Joshua C. Denny & Hua Xu. 2015. A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics* 58. 11–18. 10.1016/j.jbi.2015.09.010

Clark, Kevin, Minh-Thang Luong, Christopher D. Manning & Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. arXiv [cs.CL]. 10.48550/arXiv.1809.08370

Cohen, Aaron M. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics* 6(1). 57–71. 10.1093/bib/6.1.57

Collovini, Sandra, Joaquim Francisco Santos Neto, Bernardo Scapini Consoli, Juliano Terra, Renata Vieira, Paulo Quaresma, Marlo Souza, Daniela Barreiro Claro & Rafael Glauber. 2019. IberLEF 2019 Portuguese named entity recognition and relation extraction tasks. Em *Iberian Languages Evaluation Forum (IberLEF)*, 390–410.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer & Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. Em *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer & Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. arXiv [cs.CL]. 10.48550/arXiv.1911.02116

Correia, Fernando A., Alexandre A.A. Almeida, José Luiz Nunes, Kaline G. Santos, Ivar A. Hartmann, Felipe A. Silva & Hélio Lopes. 2022. Fine-grained legal entity annotation: A case study on the Brazilian supreme court. *Information Processing & Management* 59(1). 102794. 10.1016/j.ipm.2021.102794

Darji, Harshil, Jelena Mitrović & Michael Granitzer. 2023. German BERT model for legal named entity recognition. arXiv [cs.CL/cs.LG]. 10.48550/arXiv.2303.05388

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv [cs.CL]. arXiv.1810.04805

Dong, Xin Luna & Gerard de Melo. 2019. A robust self-learning framework for cross-lingual

text classification. Em *Conference on Empirical Methods (EMNLP) in Natural Language Processing and International Joint Conference on Natural Language Processing (IJCNLP)*, 6306–6310. 10.18653/v1/D19-1658

Dozier, Christopher, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni & Ramdev Wudali. 2010. *Named entity recognition and resolution in legal text* 27–43. Springer Berlin Heidelberg. 10.1007/978-3-642-12837-0_2

Dupre, Robert, Jiri Fajtl, Vasileios Argyriou & Paolo Remagnino. 2019. Improving dataset volumes and model accuracy with semi-supervised iterative self-learning. *IEEE Transactions on Image Processing* 29. 4337–4348. 10.1109/tip.2019.2913986

Feng, Steven Y., Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura & Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. arXiv [cs.CL]. 10.48550/arXiv.2105.03075

Friedman, Milton. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32(200). 675–701. 10.2307/2279372

Gao, Shang, Olivera Kotevska, Alexandre Sorokine & J Blair Christian. 2021. A pre-training and self-training approach for biomedical named entity recognition. *PLOS ONE* 16(2). e0246310. 10.1371/journal.pone.0246310

Garcia, Eduardo AS, Nadia FF Silva, Felipe Siqueira, Hidelberg O Albuquerque, Juliana RS Gomes, Ellen Souza & Eliomar A Lima. 2024. RoBERTaLexPT: A legal RoBERTa model pretrained with deduplication for Portuguese. Em *16th International Conference on Computational Processing of Portuguese (PROPOR)*, 374–383.

Glaser, Ingo, Bernhard Waltl & Florian Matthes. 2018. Named entity recognition, extraction, and linking in german legal contracts. Em *IRIS: Internationales Rechtsinformatik Symposium*, 325–334

Gomes, Diogo da Silva Magalhães, Fábio Corrêa Cordeiro, Bernardo Scapini Consoli, Nikolas Lacerda Santos, Viviane Pereira Moreira, Renata Vieira, Silvia Moraes & Alexandre Gonçalves Evsukoff. 2021. Portuguese word embeddings for the oil and gas industry: Development and evaluation. *Computers in Industry* 124. 103347. 10.1016/j.compind.2020.103347

He, Pengcheng, Xiaodong Liu, Jianfeng Gao & Weizhu Chen. 2021. DeBERTa: decoding-enhanced BERT with disentangled attention. Em *International Conference on Learning Representations (ICLR)*,

Heald, David Albert. 2006. Varieties of transparency. Em Christopher Hood & David Heald (eds.), *Transparency: The Key to Better Governance?*, 25–43. Oxford University Press

Helwe, Chadi & Shady Elbassuoni. 2019. Arabic named entity recognition via deep co-learning. *Artificial Intelligence Review* 52(1). 197–215. 10.1007/s10462-019-09688-6

Kalamkar, Prathamesh, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn & Vivek Raghavan. 2022. Named entity recognition in Indian court judgments. arXiv [cs.CL/cs.AI]. 10.48550/arXiv.2211.03442

Kim, Ji-Hyun. 2009. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis* 53(11). 3735–3745. 10.1016/j.csda.2009.04.009

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami & Chris Dyer. 2016. Neural architectures for named entity recognition. arXiv [cs.CL]. 10.48550/arXiv.1603.01360

Lathrop, Daniel & Laurel Ruma. 2010. *Open government: Collaboration, transparency, and participation in practice*. O'Reilly

Leitner, Elena, Georg Rehm & Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. Em *15th International Conference on Semantic Systems (SEMANTiCS)*, 272–287. 10.1007/978-3-030-33220-4_20

Li, Bohan, Yutai Hou & Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open* 3. 71–90. 10.1016/j.aiopen.2022.03.001

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer & Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv [cs.CL]. 10.48550/arXiv.1907.11692

Luz de Araujo, Pedro H., Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus

Stauffer, Samuel Couto & Paulo Bermejo. 2018. LeNER-Br: a dataset for named entity recognition in Brazilian legal text. Em *Computational Processing of the Portuguese Language (PROPOR)*, 313–323. 10.1007/978-3-319-99722-3_32

Machado, Gustavo, Mariana Recamonde Mendoza & Luis Gustavo Corbellini. 2015. What variables are important in predicting bovine viral diarrhea virus? a random forest approach. *Veterinary research* 46. 85. 10.1186/s13567-015-0219-7

Mekala, Dheeraj & Jingbo Shang. 2020. Contextualized weak supervision for text classification. Em *58th Meeting of the Association for Computational Linguistics (ACL)*, 323–333. 10.18653/v1/2020.acl-main.30

Meng, Yu, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang & Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. arXiv [cs.CL/cs.LG]. 10.48550/arXiv.2010.07245

Nemenyi, Peter Bjorn. 1963. *Distribution-free multiple comparisons*: Princeton University. Tese de Doutoramento

Neto, José Reinaldo C. S. A. V. S. & Thiago de Paulo Faleiros. 2021. Deep active-self learning applied to named entity recognition. Em *10th Brazilian Conference on Intelligent Systems (BRACIS)*, 405–418. 10.1007/978-3-030-91699-2_28

Niklaus, Joel, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis & Daniel Ho. 2024. MultiLegalPile: A 689GB multilingual legal corpus. Em *62nd Meeting of the Association for Computational Linguistics (ACL)*, 15077–15094. 10.18653/v1/2024.acl-long.805

Nunes, Rafael O., João E. Soares, Henrique D. P. dos Santos & Renata Vieira. 2019. MeSHx-notes: web-system for clinical notes. Em *1st International Workshop on Artificial Intelligence in Health (AIH)*, 5–12. 10.1007/978-3-030-12738-1_1

Nunes, Rafael O, Andre S Spritzer, Dennis G Balreira, Carla MDS Freitas & Joel L Carbonera. 2024a. An evaluation of large language models for geological named entity recognition. Em *36th International Conference on Tools with Artificial Intelligence (ICTAI)*,

Nunes, Rafael Oleques, Dennis Giovani Balreira, André Suslik Spritzer & Carla Maria Dal Sasso Freitas. 2024b. A named entity recognition approach for Portuguese legislative texts using self-learning. Em *16th International Conference on Computational Processing of Portuguese (PROPOR)*, 290–300.

Păiș, Vasile, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi & Alexandru Ianov. 2021. Named entity recognition in the Romanian legal domain. Em *Natural Legal Language Processing Workshop*, 9–18. 10.18653/v1/2021.nllp-1.2

Polo, Felipe Maia, Gabriel Caiaffa Floriano Mendonça, Kauê Capellato J Parreira, Lucka Gianvechio, Peterson Cordeiro, Jonathan Batista Ferreira, Leticia Maria Paz de Lima, Antônio Carlos do Amaral Maia & Renato Vicente. 2021. LegalNLP–natural language processing methods for the Brazilian legal language. arXiv [cs.CL/cs.LG]. 10.48550/arXiv.2110.15709

Reimers, Nils & Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. Em *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. 10.18653/v1/D19-1410

Reimers, Nils & Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4512–4525. 10.18653/v1/2020.emnlp-main.365

Rodrigues, João, Luís Gomes, João Silva, António Branco, Rodrigo Santos, Henrique Lopes Cardoso & Tomás Osório. 2023. Advancing neural encoding of Portuguese with transformer Albertina PT-*. Em *EPIA Conference on Artificial Intelligence*, 441–453. 10.1007/978-3-031-49008-8_35

Santos, Cícero Nogueira dos & Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. Em *5th Named Entity Workshop*, 25–33. 10.18653/v1/W15-3904

Santos, Diana, Nuno Seco, Nuno Cardoso & Rui Vilela. 2006. HAREM: An advanced NER evaluation contest for Portuguese. Em *5th International Conference on Language Resources and Evaluation (LREC)*,

Sechidis, Konstantinos, Grigorios Tsoumakas & Ioannis Vlahavas. 2011. On the stratification of multi-label data. Em *European Conference in Machine Learning and*

Knowledge Discovery in Databases, 145–158. 10.1007/978-3-642-23808-6_10

Sha, Lele, Yuheng Li, Dragan Gasevic & Guanliang Chen. 2022. Bigger data or fairer data? augmenting BERT via active sampling for educational text classification. Em *29th International Conference on Computational Linguistics (COLING)*, 1275–1285.

Shapiro, Samuel Sanford & Martin B Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52(3/4). 591–611. 10.2307/2333709

Silva, Nádia, Marília Silva, Fabíola Pereira, João Tarrega, João Beinotti, Márcio Fonseca, Francisco Andrade & André Carvalho. 2021. Evaluating topic models in Portuguese political comments about bills from Brazil's chamber of deputies. Em *10th Brazilian Conference on Intelligent Systems (BRACIS)*, 104–120. 10.1007/978-3-030-91699-2_8

Souza, Fábio, Rodrigo Nogueira & Roberto Lotufo. 2020. BERTimbau: pretrained BERT models for Brazilian Portuguese. Em *9th Brazilian Conference on Intelligent Systems (BRACIS)*, 403–417. 10.1007/978-3-030-61377-8_28

Sultanum, Nicole, Devin Singh, Michael Brudno & Fanny Chevalier. 2018. Doccurate: A curation-based approach for clinical text visualization. *IEEE Transactions on Visualization and Computer Graphics* 25(1). 142–151. 10.1109/TVCG.2018.2864905

Tran, Van Cuong, Ngoc Thanh Nguyen, Hamido Fujita, Dinh Tuyen Hoang & Dosam Hwang. 2017. A combination of active learning and self-learning for named entity recognition on Twitter using conditional random fields. *Knowledge-Based Systems* 132. 179–187. 10.1016/j.knosys.2017.06.023

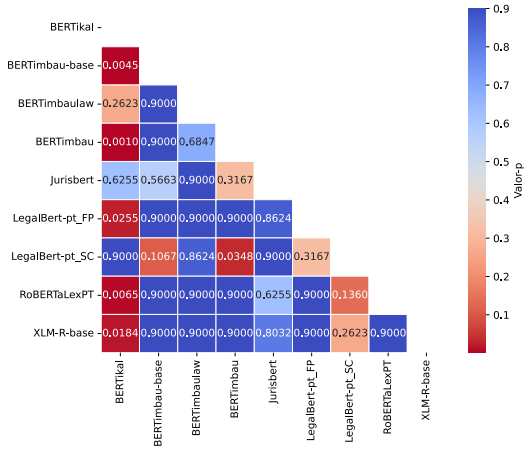
Viegas, Charles F. O., Bruno C. Costa & Renato P. Ishii. 2023. JurisBERT: a new approach that converts a classification corpus into an STS one. Em *International Conference on Computational Science and Its Applications*, 349–365. 10.1007/978-3-031-36805-9_24

Wilcoxon, Frank. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 1(6). 80–83. 10.2307/3001968

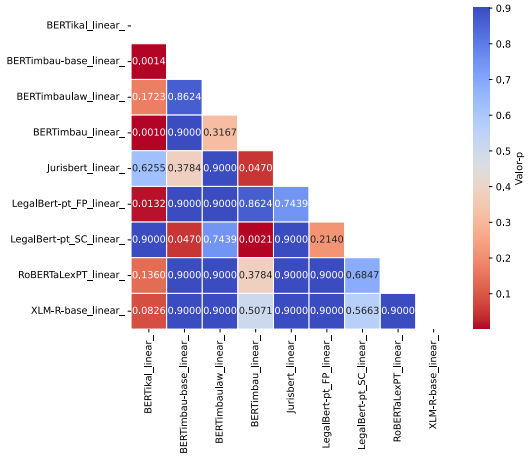
Zanuz, Luciano & Sandro José Rigo. 2022. Fostering judiciary applications with new fine-tuned models for legal named entity recognition in Portuguese. Em *15th International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 219–229. 10.1007/978-3-030-98305-5_21

A. Resultados dos testes estatísticos

Neste apêndice, apresentamos os resultados completos dos testes estatísticos discutidos na Seção 6. Enquanto na seção principal os testes foram apresentados de forma resumida, com abstrações visuais, as Figuras 6a, 6b, 7a e 7b exibem diretamente os valores-p, oferecendo uma visão mais detalhada e precisa dos resultados. A diagonal principal foi omitida, pois os valores de um modelo comparado consigo mesmo são sempre iguais. Da mesma forma, os valores acima da diagonal principal foram omitidos, uma vez que são redundantes, sendo espelhados dos valores abaixo da diagonal, o que contribui para uma apresentação mais clara e objetiva dos dados.

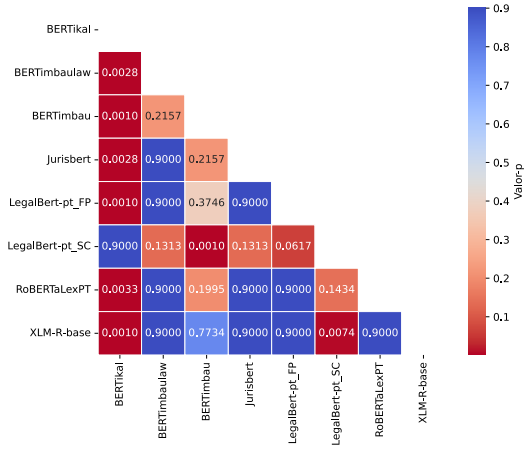


(a) Versão descontaminada.

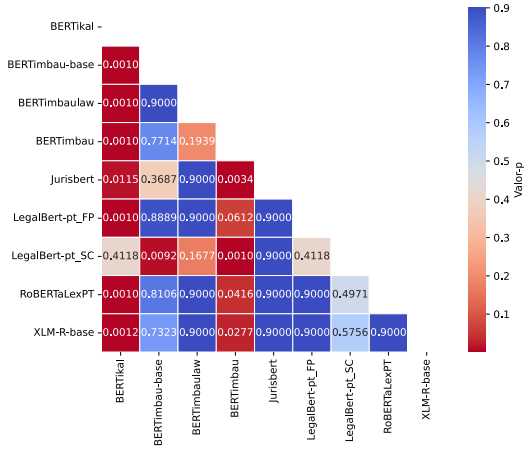


(b) Versão contaminada.

Figura 6: Mapas de calor com os resultados do teste de Friedman com o teste de Nemenyi como *post-hoc* para o ***sub-corpus* de categorias**. O modelo BERTimbau referido na figura corresponde ao modelo *large*.



(a) Versão descontaminada.



(b) Versão contaminada.

Figura 7: Mapas de calor com os resultados do teste de Friedman com o teste de Nemenyi como *post-hoc* para o ***sub-corpus* de tipos**. O modelo BERTimbau referido na figura corresponde ao modelo *large*.