vol. 10, 2022
DOI: 10.19092/reed.v9.573

TÉCNICAS DE APRENDIZADO DE MÁQUINAS APLICADAS À CLASSIFICAÇÃO DE DECISÕES JUDICIAIS

Dimmy Magalhães[1]
Aurora Pozo[2]
Sidnei Machado[3]

RESUMO

A análise de processos judiciais é uma tarefa que demanda grande trabalho de recursos humanos e muitas horas de trabalho de juízes e assessores para classificar os casos, analisá-los e produzir decisões conforme a jurisprudência e precedentes judiciais. A grande parte desse trabalho é manual e repetitivo e, por isso, classificar o texto e extrair a semântica desse corpus é uma etapa de apoio relevante a esse processo. O objetivo desta pesquisa é desenvolver uma metodologia capaz de gerar automaticamente classificações de textos jurídicos, utilizando técnicas de processamento de linguagem natural. A aplicação foi realizada em 430.000 acórdãos do Tribunal Regional do Trabalho do Paraná, produzidos nos anos de 2006 a 2018. Utilizamos técnicas de geração de representação de palavras para extração de dados. Em seguida, adotamos técnicas de agrupamento para aproximar semanticamente as decisões judiciais semelhantes pelo seu conteúdo. Esses grupos são usados para criar rótulos artificiais para cada documento. Por último, aplicamos técnicas de classificação para produzir modelos capazes de captar a semântica do texto. Os resultados obtidos são muito promissores na captura do contexto semântico dos textos jurídicos e revela que essa metodologia pode ser apropriada como suporte para o processo de classificação e decisão pelo Poder Judiciário no Brasil.

PALAVRAS-CHAVE: aprendizado de máquina; classificação; agrupamento; direito; análise de decisão judicial.

[1] Universidade Federal do Paraná, ORCID
[2] Universidade Federal do Paraná, ORCID
[3] Universidade Federal do Paraná, ORCID

vol. 10, 2022
DOI: 10.19092/reed.v9.573

MACHINE LEARNING TECHNIQUES APPLIED TO THE CLASSIFICATION OF JUDICIAL DECISIONS

Dimmy Magalhães
Aurora Pozo
Sidnei Machado

ABSTRACT

The analysis of judicial processes is an expensive task, requiring a long time of judges and advisors, either to make decisions or to classify according to the current jurisprudence. However, this process is repetitive and extracting the semantics of this corpus can be a step to support this process. The purpose of this research is to develop a methodology table of automatically generating classifications of legal documents, making use of techniques of natural language processing. Firstly, we collected 430,000 Brazilian labor court judgments from 2006 to 2018. Secondly, we propose the use of word embedding techniques for data representation. Thirdly, we use clustering techniques to semantically group the similar judicial decisions. Fourth, the clusters are used to create artificial labels for each document. Finally, we use classification techniques to produce models able to capture the semantics of judicial text. Results show a promise towards capturing the semantic context of legal texts and thus, this methodology may be used as support for the Brazilian decision-making process.

KEYWORDS: machine learning; classification; clustering; law; judicial decision analysis.

vol. 10, 2022
DOI: 10.19092/reed.v9.573

1. INTRODUÇÃO

O Poder Judiciário brasileiro tem cerca de 77,1 milhões de processos judiciais em tramitação (Justiça em Números, 2020), este número era de 78,6 milhões no ano de 2018 (Justiça em Números, 2019). Esta redução de quase 2% representa um avanço sem precedentes em toda a série histórica, segundo o Conselho Nacional de Justiça. Ainda assim, o Poder Judiciário possui uma taxa de congestionamento de 68,5% demonstrando que existe uma crescente busca pelos serviços judiciários e um inevitável acúmulo de processos. A produtividade na área jurídica cresceu 14,1%, resultado da melhora na capacitação dos servidores do Poder Judiciário, mas também fruto da implantação de ferramentas computacionais capazes de agilizar o trâmite, tais como Processo Judicial Eletrônico (PJe) e a política de digitalização de processos judiciais.

Existe uma preocupação constante do Poder Judiciário com a resolução do conflito e a baixa desse acervo, provocando inúmeras iniciativas, como os mutirões de baixa e campanhas de conciliação judicial. Esta preocupação se reflete nas esferas estaduais, nas quais é permanente a necessidade de inovação das metodologias no trato de processos judiciais. A resolução do conflito é uma área de alta complexidade, pois exige a mobilização do conhecimento técnico-jurídico altamente sofisticado para análise e construção argumentativa pelo profissional do direito, cuja base para a tomada de decisão é uma articulação entre fatos (hipótese fática) e fundamentos jurídicos (argumentação jurídica).

Dado o progressivo crescimento do número de processos judiciais, derivado de uma maior demanda por acesso à justiça, o Poder Judiciário não possui recursos suficientes para sanar o deficit de baixa de processos judiciais com o modelo de governança atual que pratica. Nesse cenário, o uso das técnicas de inteligência artificial (IA) é ferramenta central para contribuir para amenizar esse quadro de acúmulo do acervo processual. O processamento de linguagem natural (PLN) é uma ferramenta útil na otimização da análise de processos judiciais, pois o uso do processamento de texto, com técnicas de agrupamento, sumarização, extração de informações podem auxiliar o profissional do direito no processo de construção do processo o decisório.

Neste trabalho, propõe-se a análise experimental de técnicas de aprendizado de máquina aplicadas ao processamento de linguagem natural no domínio do Direito a fim de agrupar e classificar decisões judiciais. Para este experimento, foram utilizadas 430 mil decisões judiciais produzidas pelo Tribunal Regional do Trabalho entre 2006 e 2018. Em uma primeira etapa, foram criados subconjuntos com 5, 10, 20 e 50 mil decisões judiciais selecionadas aleatoriamente do conjunto total. Nesses textos foram aplicadas técnicas de pré-processamento como em (Kim, 2014). Em seguida foram gerados modelos de *word embeddings* (representação de palavras) utilizando o *framework BERT* (Devlin *et al*., 2019). Cada texto foi representado por um vetor a partir da média dos vetores de cada palavra do texto, em seguida estes vetores foram agrupados utilizando o algoritmo *K-Means* (MacQueen *et al*., 1967). Por último, para cada modelo de agrupamento foi realizada uma classificação utilizando como entrada os vetores agrupados e como saída os seus rótulos. Para esta tarefa foram executados os seguintes algoritmos de classificação: *Multi-layer Perceptron* (MLP), *Random Forest* (RF), *Support Vector Machine* (SVM).

O BERT foi selecionado devido a sua alta capacidade de representar contexto, alcançando expressivos resultados em tarefas de processamento de linguagem natural, tais como: GLUE (Wang *et al*., 2018) (82.1%); SQuAD v1.1 (93.2%) e SQuAD v2.0 (86.3%) (Rajpurkar *et a*l., 2016); SWAG (86.3%) (Zellers *et al*., 2018).

Como contribuição deste trabalho, pode-se listar a: (i) vasta varredura de diferentes técnicas de PLN; (ii) criação de modelo de agrupamento capaz de direcionar trabalhos da catalogação de decisões judiciais; (iii) criação de modelo de classificação de decisões judiciais, propondo uma abordagem não supervisionada para o processo.

O restante do artigo está organizado em quatro seções. Os trabalhos relacionados são discutidos na seção 2. A seção 3 apresenta uma visão geral sobre as técnicas de NLP, *clusterização* (agrupamento) e classificação. Na seção 4, é apresentada a metodologia dos experimentos. A seção 5 descreve os resultados e discussões. Na seção 6 são descritas as vantagens, limitações e futuras direções de pesquisa.

2. TRABALHOS RELACIONADOS

Embora a pesquisa em processamento de linguagem natural esteja avançada em muitos aspectos e idiomas, existem poucos trabalhos publicados relacionados à análise semântica de processos judiciais brasileiros, ainda que vários projetos possam ser identificados em muitos tribunais. Os investimentos na área de IA para o domínio legal têm ganhado relevância nos últimos anos, visto que o próprio Conselho Nacional de Justiça (CNJ) implantou o Centro de Inteligência Artificial para promover inovação e celeridade nos serviços do Sistema de Justiça. Segundo o Justiça em Números de 2018, o Poder Judiciário brasileiro incrementou em 25% o orçamento anual aplicado à tecnologia da informação.

Em seu trabalho, Sewald *et al*. (Junior *et al*., 2012) propõem a modelagem do conhecimento legal brasileiro baseado na Engenharia do Conhecimento a fim de conceituar, inferir necessidades e definir soluções sob a ótica do conhecimento legal. Em Júnior (2001), os autores buscam utilizar técnicas de raciocínio baseado em casos (RBC) para inferir similaridade entre fatos. Em 2018 o Projeto Victor (Maia Filho & Junquilho, 2018) usou técnicas de análise de frequência para inferir semântica em processos judiciais eletrônicos e, mesmo ainda não sendo um projeto finalizado, tem orientado as pesquisas e objetivos na área de processamento automático de documentos judiciais. O projeto Sinapse propõe a otimização de processo de reconhecimento de padrões em textos judiciais, auxiliando em tarefas repetitivas do trâmite de processos judiciais.

Polo (2021) propõe modelos de linguagem pré-treinados (Phraser, Word2Vec, Doc2Vec, FastText e BERT) para textos jurídicos brasileiros, a ferramenta está disponível publicamente[4] e é uma importante contribuição para modelos de análise de texto jurídico. O conjunto de textos de treinamento conta com documentos de quase todos os tribunais do país, com destaque para o Tribunal de Justiça de São Paulo. Os autores unificaram o uso de diferentes ferramentas de processamento de linguagem natural para análise de textos jurídicos pela indústria brasileira, governo e academia, fornecendo as ferramentas necessárias e material acessível.

[4] Disponível em: https://github.com/felipemaiapolo/legalnlp

No contexto de representação de textos para o português do Brasil, destaca-se o trabalho de Souza *et al*. (2020) que treinou o modelo BERT para o português, usando dados do brWaC. O modelo apresenta um relevante ganho de contextualidade, já que, usando técnicas de *transfer learning*, consegue gerar representações acuradas para o português. Ainda no contexto do idioma português, destaca-se o trabalho de Hartmann *et al*. (2017) que apresentam um modelo de linguagens pré-treinadas para o português do Brasil, semelhante ao que é encontrado no Word2Vec, FastText, Glove, entre outros.

No contexto internacional, McCarty propõe o uso de modelos formais de análise, que chama de *quasi-logical form* (QLF) para extração de características semânticas de documentos legais permitindo assim sua interpretação (McCarty, 2007). Fersini apresenta o projeto JUMAS que utiliza *Automatic Speech Recognition*, reconhecimento de emoções e recuperação de informação em domínio legal (Fersini *et al*., 2010). Ainda no contexto internacional, Lu *et al*. (2011) propõem o uso de segmentação de tópicos de textos legais para então usar técnicas de agrupamento e SVM para mesclar os grupos sob a ótica de seus tópicos. Wagh (2013) construiu técnicas de *text mining* para criar tópicos relevantes acerca de processos judiciais. Walter *et al*. (2017) propõem um modelo baseado em regras e conectivos semânticos para a análise de documentos jurídicos.

Em todos esses os trabalhos é possível identificar a preocupação dos autores em propor regras de análise semântica entre os documentos jurídicos ou representações textuais. Neste trabalho propõe-se uma análise empírica das técnicas de classificação de texto como forma de captura de semântica em decisões judiciais para otimizar a tarefa de análise de processos judiciais sob uma ótica não-supervisionada, isto é, quando não se conhece previamente nenhuma característica, rótulo ou classificação dos dados iniciais. Trata-se, portanto, de uma aplicação prática de técnicas à análise de processos judiciais brasileiros no âmbito da Justiça do Trabalho.

3. FUNDAMENTAÇÃO

Nesta seção são descritas as técnicas computacionais de representação de textos, agrupamento e classificação de dados utilizados nos experimentos.

3.1 BERT

O *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin *et al*., 2019) é baseado em um *Transformer* (Vaswani *et al*., 2017), que aprende relações contextuais entre palavras de uma sentença (frase). O modelo foi construído para pré-treinar as representações de texto com uma etapa não supervisionada e uma etapa supervisionada de refinamento da representação a partir de dados rotulados. Como resultado, as representações geradas a partir do BERT tendem a ser mais refinadas do que aquelas geradas apenas por uma camada de saída de uma rede neural.

O BERT consiste, portanto, em duas etapas sequenciais: *pre-training* e *fine-tuning*. Primeiramente, o modelo é treinado com dados não rotulados sob diferentes tarefas de treinamento. Para o *fine-tuning,* o modelo BERT é inicializado com os parâmetros pré-treinados e então todos os parâmetros são ajustados usando dados rotulados previamente (Devlin *et al*., 2019). Ao contrário das Redes Neurais Artificiais, tais como LSTM (Gers *et al*., 1999), Word2Vec (Mikolov *et al*., 2013), Doc2Vec (Le & Mikolov, 2014), o BERT pode gerar representações diferentes para palavras iguais, capturando o contexto em que cada uma de suas instâncias esteja inserida. Desta forma, no uso de modelos como Word2Vec são necessárias apenas as representações vetoriais das palavras, visto que elas são únicas. Por outro lado, no caso do BERT, é necessário o uso do modelo completo de treinamento, visto que a representação de um termo será diretamente ligada ao seu contexto (incluindo, características de posicionamento das palavras).

A arquitetura do BERT segue Vaswani et al (Vaswani *et al*., 2017) e foi implementada utilizando *Tensorflow* (Abadi *et al*., 2016). O trabalho reportou dois modelos chamados de $\text{BERT}_{\text{BASE}}$ (com 12 camadas, 768 unidades e 12 unidades de intra-atenção) e $\text{BERT}_{\text{LARGE}}$ (com 24 camadas, 1024 unidades e 16 unidades de intra-atenção), totalizando 110 mil e 340 mil parâmetros respectivamente.

Para este trabalho foi utilizado a implementação disponível publicamente[5]. Além disso, foi utilizado um modelo pré-treinado estendido chamado *BERT*$_{BASE}$ *Multilingual Cased* (com 104 idiomas, 12 camadas, 768 unidades, 12 unidades de intra-atenção, 110 mil parâmetros) disponível publicamente[6]. Em linhas gerais, o BERT é capaz de receber como entrada um conjunto de sentenças e retornar uma representação em forma de vetor de números, de tal forma que frases semanticamente próximas tenham representações vetoriais próximas (por uma métrica euclidiana, por exemplo). O BERT tem ocupado papel de destaque quando se trata de representação vetorial de texto.

Neste contexto, é importante destacar a ferramenta BERTimbau de Souza, Nogueira e Lotufo (2020) que treinaram o modelo BERT para o português, usando dados do brWaC (Wagner Filho *et al*., 2018), um grande e diversificado corpus de páginas da web. O modelo de representação de textos proposto pelo BERTimbau apresenta resultados expressivos, pois em cenários controlados atinge um valor de F1-Score[7] de 74,8% contra 72,1% de modelos usando o *BERT* multilinguagem[8]. A ferramenta BERTTimbau demonstra a penetração da abordagem BERT em diferentes idiomas, comprovando sua aplicabilidade no problema apresentado neste trabalho.

3.2 *K-Means*

O *K-Means* (MacQueen *et al*., 1967) é um algoritmo de agrupamento consolidado na literatura computacional e tem por objetivo particionar um conjunto de dados em grupos (ou *clusters*). Seu algoritmo segue duas etapas (Wagstaff *et al*., 2001):

1. A cada elemento (ou instância) a ser agrupado (ou *cluster*izada) é atribuído um centroide próximo (um ponto dentro do espaço de representação dos

[5] Disponível em: https://github.com/google-research/bert

[6] Disponível em: https://huggingface.co/bert-base-multilingual-cased

[7] Usando um modelo de classificação baseado em LSTM.

[8] Modelo de treinamento do BERT para diversas línguas que não o inglês.

dados). Este ponto pode ser um dos dados a serem agrupados, ou pode ser criado aleatoriamente;

2. Cada centroide é atualizado para ser a média das instâncias a ele atribuídas. O algoritmo finaliza (ou converge) quando não há mudanças adicionais na atribuição de instâncias aos *clusters* ou até que um número paramétrico de iterações seja alcançado. Para este trabalho foi utilizado a implementação disponível publicamente[9].

Via de regra, o *K-Means* é capaz de agrupar representações vetoriais de tal forma que haja uma relação semântica entre vetores semelhantes (e próximos no espaço vetorial) e dissociar vetores que não possuem essa característica. Em suma, o algoritmo recebe como entrada um conjunto de vetores e retorna um rótulo para cada vetor, de forma que vetores com o mesmo rótulo estão no mesmo grupo semântico.

3.3 MULTIPLE LAYER PERCEPTRON

As Multilayers Perceptron são um subconjunto de redes neurais *feed-forward*. As MLP são compostas de um simples sistema de neurônios artificiais (*perceptrons*) interconectados no qual o modelo final representa um mapeamento não linear entre um vetor de entrada e um vetor de saída. Os perceptrons (ou nós) são ponderados e os sinais de saída são gerados por uma função de soma das entradas que, por sua vez, são modificados por uma simples função de transferência ou ativação não-linear (Gardner & Dorling, 1998). Para este trabalho foi utilizado a implementação disponível publicamente[10].

3.4 RANDOM FOREST

[9] Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[10]Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

O *Random Forest (RF)* é um classificador *ensemble* composto por um conjunto de classificadores baseados em árvores de decisão (Breiman, 2001). Em outras palavras, um RF é um meta classificador que combina um conjunto pré-definido de árvores de decisão que:

1. Realiza uma amostragem aleatória de pontos do conjunto de dados de treinamento ao construir árvores de decisão;
2. Utiliza subconjuntos aleatórios de características quando gera os nós de cada árvore. Com isso, é possível melhorar a capacidade de predição e reduzir o risco de *overfitting* (situação no qual o modelo computacional 'decora' os dados de entrada em vez de 'aprender' e generalizar os dados de entrada) quando utilizado apenas uma árvore de decisão.

3.5 SUPPORT VECTOR MACHINE

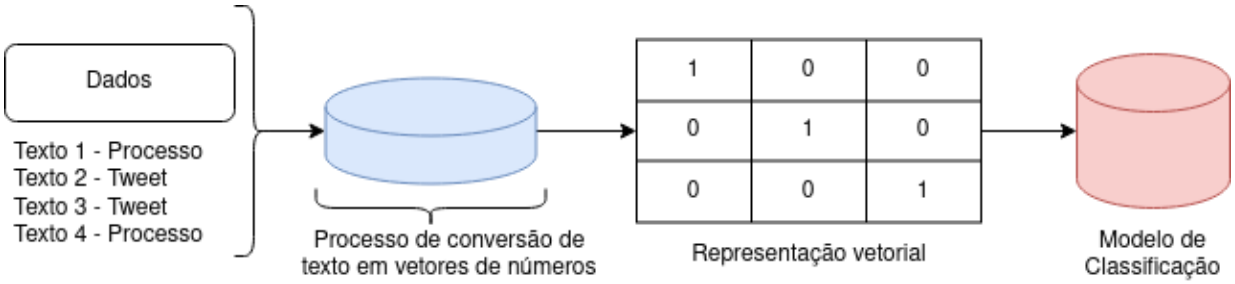
O *Support Vector Machine* (SVM) é um algoritmo de inteligência artificial capaz de encontrar um hiperplano ideal no espaço N-dimensional que melhor separa o conjunto de dados de entrada com base no princípio de minimização de risco estrutural (Vapnik, 2013). O principal objetivo é encontrar tal hiperplano que possua uma margem maximizada, isto é, a distância máxima entre pontos de cada distinta classe do problema. O método possui boa performance em problemas de grandes dimensões, mesmo que a quantidade de dimensões seja maior do que a quantidade de instâncias. Para fazê-lo o SVM realiza transformações nas variáveis de entrada aplicando funções de kernel. Para este trabalho foi utilizado a implementação baseada no SVM disponível publicamente[11].

Em geral, os modelos de classificação (como MLP, RF e SVM) recebem como entrada um conjunto de dados (representados por vetores numéricos) chamados na literatura de instâncias, dados, *input*, *features* e tem como retorno um ou mais rótulos para cada vetor. A Figura 1 demonstra um modelo genérico de classificação.

[11]Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

Figura 1

Fluxo de um modelo de classificação genérico



Nota: De forma geral, os dados iniciais (textos, por exemplo) passam por um processo de conversão para vetores de números (processo em azul). O modelo de classificação recebe esses vetores e realiza um processo de treinamento, no qual aprende padrões e relacionamentos entre os vetores, ajustando os chamados parâmetros do modelo (Aggarwal, 2018).

4. METODOLOGIA

Nesta seção é apresentado como cada técnica foi organizada no fluxo de trabalho. O experimento realizado neste trabalho pode ser dividido em três etapas sequenciais: 1) preparação; 2) representação e *cluster*ização; 3) classificação.

4.1 PREPARAÇÃO

A coleta do conjunto de dados, consistente em 430 mil decisões judiciais, se deu pelo acesso disponibilizado pelo Tribunal Regional do Trabalho do Paraná, em atenção ao requerimento para fins de pesquisa, que possibilitou a extração da base de dados da corte de todas as decisões proferidas do período de 2006 a 2018. O arquivo cedido, continha os acórdãos em arquivos em formatos pdf e html. Os dados são abrangentes e representativos, pois contemplam a totalidade dos acórdãos, produto de decisões finais proferidas em processos trabalhistas por essa corte recursal, exceto decisões interlocutórias, que correspondem à segunda instância da Justiça do Trabalho no âmbito do Estado do Paraná. Cada documento,

após a remoção de elementos de marcação, continha tamanho médio de 7.000 caracteres.

A abordagem proposta neste trabalho segue (Kim, 2014), desta forma, as decisões judiciais foram tokenizadas, as stop words (palavras desnecessárias para análise de contexto, com pouca ou nenhuma carga semântica) definidas em NLTK descartadas e por fim realizou-se o processo de *stemming* para remover afixos morfológicos das palavras a fim de obter os lemas de cada palavra, diminuindo o impacto de flexões linguísticas. Características do domínio legal (tais como vara, comarca ou juízo) são adicionadas ao processo de preparação na medida que estes elementos aparecem no acórdão. Em seguida foram criados 30 subconjuntos de 5, 10, 20 e 50 mil decisões judiciais, selecionados aleatoriamente, sem reposição (isto é, ao selecionar uma decisão para um conjunto, ela tem a possibilidade de ser selecionada novamente para outro conjunto) do conjunto de 430 mil decisões, cada subconjunto é chamado aqui de *folder*. No contexto deste trabalho entende-se *modelo* como uma aplicação de uma técnica de agrupamento a um *folder*, por exemplo, o modelo MLP 5.000 diz respeito a aplicação do classificador MLP ao *folder* com 5.000 decisões judiciais.

4.2 REPRESENTAÇÃO E CLUSTERIZAÇÃO

Para cada *folder* foi criada uma representação vetorial utilizando o *framework* BERT. Como parâmetro foram utilizados os modelos pré-treinados e conjunto de dados de múltiplos idiomas. O BERT gera uma representação vetorial de cada documento em cada camada da rede, podendo ser utilizada qualquer uma destas como representação. Para este experimento foram utilizadas as quatro últimas camadas, calculando-se o vetor médio entre os vetores do primeiro *token* de saída ([CLS]) de cada camada, processo semelhante ao utilizado em (Devlin *et al*., 2019). Ainda que o tamanho do documento ou texto de entrada seja um fator importante no desempenho da arquitetura utilizada para a classificação textual (Martins & Silva, 2021), o BERT é capaz de realizar uma representação vetorial com tamanho fixo dos textos consideravelmente acurada devido ao seu mecanismo de *fine-tunning*.

Para a *cluster*ização, a cada modelo de representação foi aplicado o algoritmo de *cluster*ização *K-Means*, a fim de descobrir *clusters* semânticos entre os vetores. O *K-Means* espera como parâmetro a quantidade de *clusters*, isto é, o valor k, portanto o algoritmo foi executado empiricamente com valores de k entre 2 e 30, selecionando-se, por fim, o modelo base para classificação que apresentasse o melhor índice *silhouette* (Rousseeuw, 1987). Como resultado desta subetapa foi possível definir um rótulo artificial para cada decisão judicial, que consistiu no *label* do *cluster* ao qual aquela decisão foi associada.

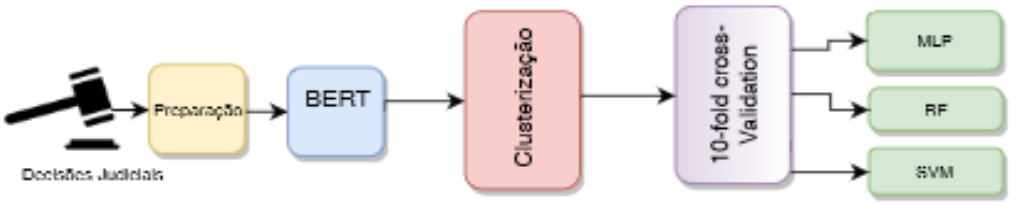
4.3 CLASSIFICAÇÃO

Nesta etapa os algoritmos de classificação foram executados por 30 rodadas independentes sobre cada um dos modelos de representação de texto, tendo como entrada a representação vetorial da decisão judicial gerada pela etapa de representação (utilizando o BERT) e como rótulos os valores gerados pela etapa de *cluster*ização (utilizando o *K-Means*).

Para o MLP, foi construída uma rede neural com 6 camadas densamente conectadas com 768, 384, 192, 96, 48, e 7 neurônios cada (sequencialmente da primeira à última camada). Cada camada foi configurada com a função de ativação ReLu intercaladas por uma camada de *dropout* de 0.25, a última camada fora configurada com a função *softmax*. O modelo foi executado por 200 épocas. O RF foi executado variando-se a quantidade de árvores de decisão internas entre 2 e 50. Para o SVM foram realizados experimentos nos quais o valor de C era modificado entre 2 e 50 tendo a função rbf (amplamente utilizada na literatura) como função de *kernel* e o algoritmo foi executado até se encontrar uma solução factível, ou seja, não foi definido valor máximo de iterações. As configurações dos modelos de classificação são análogas às presentes na literatura, tais como (Simanjuntak *et al*. 2010; Criminisi *et al*., 2012; Wan & Gao, 2015). A Figura 2 mostra o fluxo realizado no experimento.

Figura 2

Fluxo do experimento



Fonte: https://github.com/dimmykarson/analiseprocessos

Foram coletadas as seguintes medidas de avaliação da classificação: acurácia, F1-Score e erro médio quadrático. Foram executadas 30 rodadas independentes em 10 conjuntos de treinamento/teste gerados por processo de *10-fold cross-validation* reportando-se a média das métricas selecionadas.

5. RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os principais resultados e discussões dos experimentos. A Tabela 1 descreve os modelos de agrupamento originados do processo de *cluster*ização e utilizados no processo de classificação. A tabela descreve quantidade de decisões no modelo, o valor de K selecionado (isto é, a quantidade de grupos), o valor de *silhouette* alcançado (que descreve o quanto o agrupamento é coeso) e a quantidade de decisões judiciais em cada rótulo (ou classe).

Tabela 1

Descrição dos modelos de clusterização utilizados

				Classes							
#D	**K**	**S**	**B**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
5000	7	0.689127 (0.0001)	Não	562	957	1498	1297	439	175	72	-

10000	7	0.655430 (0.0012)	Não	1890	2965	1155	888	2655	309	138	-
20000	8	0.664418 (0.0009)	Não	5263	5995	2220	966	3877	690	248	741
50000	7	0.649934 (0.0007)	Não	13136	14879	5695	9467	4348	1637	658	-

Nota: #D: Número de decisões avaliadas; K: Número de agrupamentos; S: Valor *silhouette* encontrado (entre parênteses o desvio padrão de todas as execuções); B: Informa se as classes se encontram balanceadas

Fonte: https://github.com/dimmykarson/analiseprocessos

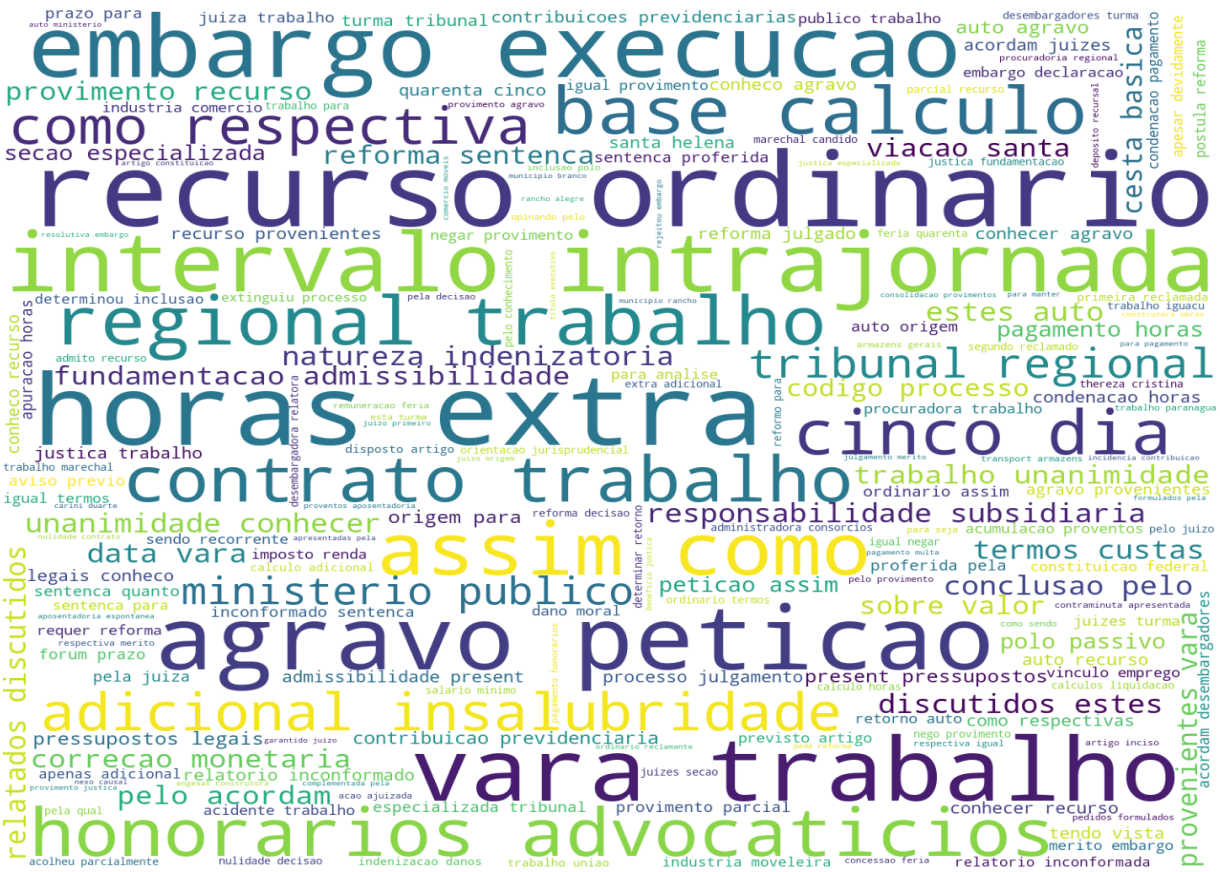
Os resultados obtidos com a *cluster*ização foram acompanhados e apresentados à especialista em Direito Trabalho por amostras dos *clusters*. O especialista, um advogado e professor universitário, com experiência de mais de 25 anos em atuação de casos judiciais e ensino de prática jurídica no ensino superior, conseguiu validar os testes gerados. Para a validação, o especialista com experiência em análise não automatizada de decisões, comparou os dados obtidos com algumas amostras de decisões judiciais para identificar a correção e a coerência das respostas geradas. Neste caso se escolheu verificar se as classes retornadas do processo de agrupamento possuem algum sentido na área judicial. Constatamos que mesmo não se tratando de tema específico da mesma norma trabalhista, as decisões judiciais verificadas têm, entre si, assuntos em interseção, o que justificaria seu agrupamento. Decisões judiciais, por exemplo, que tratam de "hora extra de trabalho" foram associadas em um mesmo *cluster,* mesmo que o termo seja recorrente em vários agrupamentos, sua maior representatividade é no *cluster* da classe 1. A Figura 3 apresenta uma nuvem de palavras do agrupamento

1 para a clusterização com 5.000 acórdãos. As outras nuvens de palavras, para cada um dos conjuntos de acórdãos, estão disponíveis publicamente[12].

Para a tarefa de classificação, foi utilizado o melhor modelo de agrupamento em 7 *clusters* encontrados pela tarefa de agrupamento. A Tabela 2 exibe os resultados médios (e desvio padrão) alcançados para cada algoritmo de classificação. Os resultados mostram que o algoritmo SVM obtém os melhores resultados considerando as métricas utilizadas conforme a representação selecionada em todos os grupos de decisões judiciais.

Figura 3

Nuvem de palavras do *cluster* 1



Fonte:https://github.com/dimmykarson/analiseprocessos

[12]Disponível em: https://github.dimmykarson/nuvensdepalavrasacordaos.

O modelo com RF apresentou relativa sensibilidade ao desbalanceamento do conjunto de dados. Por outro lado, a rede neural apresentou resultados aquém dos outros modelos e, também, observou-se neste experimento um crescente aumento da taxa de perda, possivelmente justificada pela quantidade de camadas de *dropout* que a rede possuía e pela baixa quantidade de camadas.

A Figura 4 mostra a média dos F1-score alcançados por algoritmo em cada classe do modelo. Constata-se que o SVM é pouco sensível ao desbalanceamento do conjunto de dados, apresentando valores de F1-score semelhantes para todas as classes. Foi realizado teste de múltiplas comparações utilizando Kruskal-Wallis (Kruskal & Wallis, 1952) entre todos os algoritmos e todos os modelos, com *p-value* de 0.05. Em todos os testes foi possível observar que existem diferenças estatísticas entre os algoritmos, sendo que os resultados apresentados pelo SVM são estatisticamente melhores que os outros classificadores. Os resultados completos estão disponíveis publicamente[13], bem como todo o código desenvolvido.

Tabela 2

Resultados médios encontrados para a tarefa de classificação

Algoritmo	Modelo	Acurácia	F1-Score	MSE
MLP	5000	0.1914 (0.00011)	0.1914 (0.00011)	3.1582 (0.00597)
	10000	0.2965 (1.63e-10)	0.2964 (0.00012)	3.8885 (2.29e-10)
	20000	0.2997 (1.31e-11)	0.2990 (0.00011)	4.5077 (3.32e-10)
	50000	0.2975 (3.26e-12)	0.2975 (0.00022)	2.7730 (3.38e-11)
RF	5000	0.8799 (0.00011)	0.8796 (0.00011)	0.8603 (0.00275)
	10000	0.8817 (1.26e-04)	0.8816 (0.00012)	0.9640 (1.20e-03)
	20000	0.8789 (2.21e-02)	0.8780 (0.02219)	1.5265 (2.78e-10)
	50000	0.8837 (5.01e-02)	0.8838 (0.05011)	0.8521 (3.27e-01)
SVM	5000	0.9984 (0.00013)	0.9985 (0.00011)	0.0201 (0.00190)
	10000	0.9993 (6.56e-04)	0.9992 (0.00065)	0.00541 (7.53e-03)
	20000	0.9981 (1.38e-04)	0.9982 (0.00013)	0.0207 (1.78e-03)
	50000	0.9997 (2.98e-05)	0.9994 (0.00003)	0.0010 (1.19e-04)

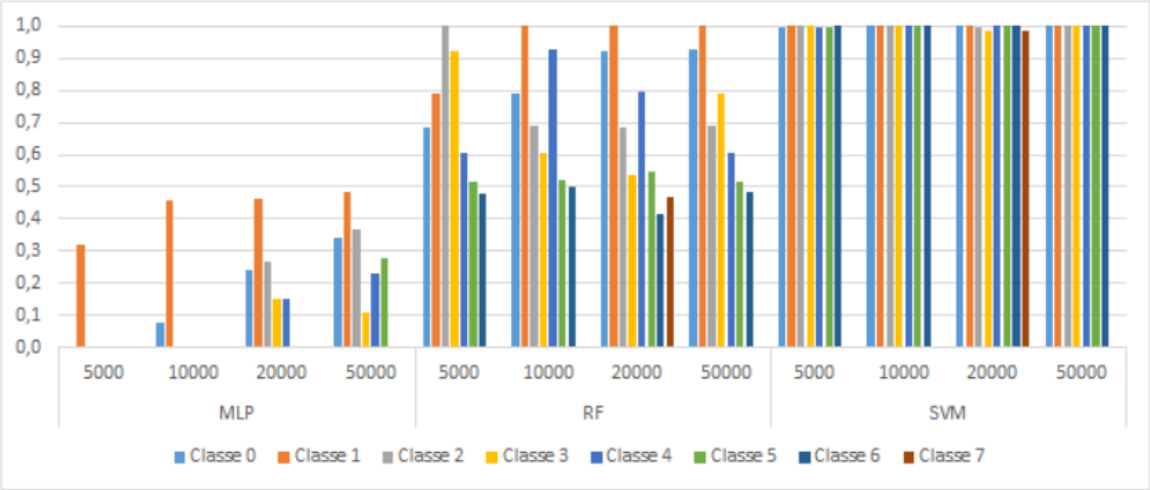
Nota: Experimento realizado com 30 interações independentes para cada modelo de classificação. Entre parênteses o desvio padrão entre as 30 interações.

Fonte: https://github.com/dimmykarson/analiseprocessos

[13] Disponível em: https://github.com/dimmykarson/analiseprocessos.

Figura 4

F1-Score reportado por modelo para cada classe



Nota: F1-score é capaz de medir quanto cada modelo acertou para cada um dos rótulos apresentados.

Fonte: https://github.com/dimmykarson/analiseprocessos

6. CONCLUSÃO

Neste trabalho foram realizados experimentos sobre a representação textual, agrupamento e classificação de decisões judiciais. Foram criados modelos de representação de texto baseado no algoritmo BERT. Em seguida, foi realizada *cluster*ização utilizando o algoritmo *K-Means* a fim de criar um modelo de rotulagem artificial para as decisões. E por fim, realizou-se a classificação destas decisões. O objetivo era encontrar um modelo capaz de capturar automaticamente a semântica nas decisões judiciais. Uma vez que seja capaz de provar que técnicas de aprendizagem de máquina podem ser utilizadas na análise semântica de documentos judiciais com segurança comprovada, seria possível direcionar trabalhos futuros na área de inteligência artificial e Direito brasileiro, tais como sumarização semântica, rotulagem e movimentação de processos judiciais.

A etapa de *cluster*ização, apesar de utilizada como uma fase auxiliar ao experimento, mostrou-se muito eficaz sendo aplicável ao domínio jurídico no fluxo de distribuição de processos entre varas ou secretarias judiciais. Esta etapa é extremamente sensível ao trabalho, pois parte dela os rótulos iniciais de cada uma das decisões judiciais tratadas. A eficácia deste processo é crucial para todo o modelo. Os valores de *silhouette* encontrados indicam que a representação vetorial gerada pelo BERT é perfeitamente agrupável semanticamente no domínio do direito, e que tal modelo pode ser utilizado para trabalhos futuros nesta área.

Com a classificação, foi possível identificar que sob as mesmas representações de texto o modelo baseado em redes neurais (MLP) apresentou valores abaixo do esperado. É possível observar também, utilizando validações e análises estatísticas, que os modelos baseados em RF sofreram *overfitting*. Neste caso o processo de classificação — montagem das árvores de decisão — não conseguiu em todas as suas instâncias generalizar a classificação para instâncias não conhecidas (textos judiciais não apresentados para o treinamento). O modelo, portanto, deve ser aprimorado para ser considerado em experimentos futuros.

Entretanto, o modelo baseado no SVM alcança índices satisfatórios, sobre os quais conclui-se que técnicas de representação de texto através do BERT são viáveis como ferramenta de apoio à classificação de decisões judiciais. Vale ressaltar que os modelos de agrupamento baseados no *K-Means* são especializados em encontrar agrupamentos esféricos e isso em parte justifica a alta performance do SVM no trabalho, visto que este algoritmo também é especializado nesta distribuição de dados.

Em linhas gerais, os experimentos aqui propostos conseguem nortear trabalhos que alinham a classificação automática de texto às técnicas consolidadas da inteligência artificial. Pode-se inferir ainda que tais técnicas podem ser diretamente aplicáveis ao processo judicial completo (não só à decisão), auxiliando em atividades como distribuição, correção de classificação e movimentação de processos judiciais.

A tarefa de agrupamento proposta neste trabalho pode ser diretamente substituída por dados rotulados pelos atuais sistemas eletrônicos do Poder Judiciário. Os processos judiciais podem ser facilmente agrupados por juiz, comarca, assunto, argumentação, tema, tese, data, entre outros aspectos. Esse

agrupamento pode servir como processo de rotulagem especializada, e desta forma o modelo proposto contribuiria para, em última instância, otimizar o trato jurídico.

Alguns pontos podem ameaçar a validade do trabalho, dentre os quais destacam-se: o uso de apenas decisões da classe trabalhista com um vocabulário reduzido frente ao conjunto total de possibilidades jurídicas. Como discutido, a rotulagem das decisões é dita artificial, isto é, um modelo não-supervisionado foi utilizado para agrupar as decisões judiciais de acordo com propriedades computacionais inferidas, o que não garante que os grupos sejam estritamente coesos semanticamente em todas as instâncias.

Reforça-se então que modelos genéricos de representação de textos são capazes de capturar a carga semântica complexa e particular do domínio jurídico e que técnicas novas e consolidadas de inteligência artificial podem trazer um dinamismo ao processo de análise, agrupamento e classificação automatizada de documentos jurídicos. Alguns direcionamentos futuros são listados a seguir:

- Modificação do modelo de representação, aplicando as técnicas BERTTimbau e LegalNLP;
- Uso de modelos consolidados na literatura para classificação de textos, tais como LSTM, TextGCN ou CNN;
- Implementação da abordagem em cenários não controlados, tais como ambientes com processos judiciais de assuntos semelhantes;
- Aplicação da abordagem através de ferramenta auxiliar aos sistemas de processo eletrônico.

AGRADECIMENTOS

Este trabalho foi financiado pela CAPES, Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brasil. Agradecimentos ao Tribunal de Justiça do Estado do Piauí (TJPI) e o Tribunal Regional do Trabalho do Paraná (TRT-PR) pela colaboração na pesquisa.

REFERÊNCIAS

Abadi, M. *et al*. (2001). Random forests. *Machine Learning*, *45*(1), 5-32.

Abadi, M. *et al*. (2016). Tensorflow: A system for large-scale machine learning. In *12th fUSENIXg Symposium on Operating Systems Design and Implementation (fOSDIg 16)* (pp. 265-283).

Charu C. Aggarwal (2018). Machine Learning for Text. *Springer Publishing Company, Incorporated, 1st edition*, 541-543.

Conselho Nacional de Justiça - CNJ (2018). Justiça em números 2018: ano-base 2017. Brasília: CNJ, 2018b. Disponível em: https://www.cnj.jus.br/wp-content/uploads/conteudo/arquivo/2018/08/44b7368ec6f888b383f6c3de40c32167.pdf. Acesso em: 26 dez. 2022.

Conselho Nacional de Justiça - CNJ (2019). Justiça em números 2019: ano-base 2018. Brasília: CNJ, 2019b. Disponível em: https://www.cnj.jus.br/wp-content/uploads/conteudo/arquivo/2019/08/justica_em_numeros20190919.pdf . Acesso em: 26 dez. 2022.

Conselho Nacional de Justiça - CNJ (2020). Justiça em números 2020: ano-base 2019. Brasília: CNJ, 2020b. Disponível em: https://www.cnj.jus.br/wp-content/uploads/2020/08/WEB-V3-Justi%C3%A7a-em-N%C3%BAmeros-2020-atualizado-em-25-08-2020.pdf . Acesso em: 26 dez. 2022.

Criminisi, A. *et al*. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends R in Computer Graphics and Vision*, *7*(2-3), 81-227.

Devlin, J. *et al*. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*, 1, 4171-4186.

Fersini, E. *et al*. (2010). Semantics and machine learning: A new generation of court management systems. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*. Springer, pp. 382–398.

Gardner, M. W., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron) – A review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14-15), 2627-2636.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. *1999 Ninth International Conference on Artificial Neural Networks ICANN 99*, *470*, 850-855.

Hartmann, N. *et al*. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology* (pp. 122-131). Uberlândia: Sociedade Brasileira de Computação.

Junior, E. S. *et al*. (2012). Modelagem de sistema baseado em conhecimento em um tribunal de justiça utilizando commonkads. *Revista Democracia Digital e Governo Eletrônico*, *2*(7), 160-189.

Júnior, M. D. S. (20o1). Proposta de modelo RBC para a recuperação inteligente de jurisprudência na Justiça Federal [Tese de Doutorado, Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal De Santa Catarina].

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746-1751). Doha: Association for Computational Linguistics.

Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, *47*(260), 583-621.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning*, *32*(2), 1188-1196.

Lu, Q. *et al*. (2011). Legal document *cluster*ing with built-in topic segmentation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (pp. 383-392).

MacQueen, J. *et al*. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1 (pp. 281-297).

Maia Filho, M. S., & Junquilho, T. A. (2018). Projeto Victor: Perspectivas de aplicação da inteligência artificial ao direito. *Revista de Direitos e Garantias Fundamentais*, *19*(3), 218-237.

Martins, V. S.; Silva, C. D. (2021). Text Classification in Law Area: a Systematic Review. In *Anais do IX Symposium on Knowledge Discovery, Mining and Learning* (pp. 33-40). Porto Alegre: SBC.

McCarty, L. T. (2017). Deep semantic interpretations of legal texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law* (pp. 217-224).

Mikolov, T. *et al*. (2013). Efficient estimation of word representations in vector space. In *ICLR - Workshop Poster* (eprint 1301.3461).

Polo, F. M. *et al*. (2021). Natural Language Processing methods for the Brazilian Legal Language. In *Computing Research Repository - CoRR* (eprint 2110.15709).

Rajpurkar, P., *et al*. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Computing Research Repository - CoRR* (eprint 1606.05250).

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65.

Simanjuntak, D. A. *et al*. (2010). Text classification techniques used to facilitate cyber terrorism investigation. In *2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies* (pp. 198-200). Jakarta: Bina Nusantara University.

Souza, F., Nogueira, R., & Lotufo, R. (2020). BERTimbau: Pretrained BERT models for Brazilian Portuguese. In R. Cerri, & R. C. Prati (Eds.), *Intelligent Systems. BRACIS 2020. Lecture Notes in Computer Science, Vol. 12319* (pp. 403-417). Springer, Cham.

Vapnik, V. (2013). *The nature of statistical learning theory*. Nova York: Springer Science & Business Media.

Vaswani, A. *et al*. (2017). Attention is all you need. In I. Guyon *et al*. (Eds.), *Advances in neural information processing systems* (pp. 5998-6008).

Wagh, R. S. (2013). Knowledge discovery from legal documents dataset using text mining techniques. *International Journal of Computer Applications*, *66*(23), 32-34.

Wagner Filho, J. A. *et al*. (2018). The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki: European Language Resources Association.

Wagstaff, K. *et al*. (2001). Constrained *K-Means clustering* with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 577-584).

Walker, V. R. *et al*. (2017). Semantic types for computational legal reasoning: Propositional connectives and sentence roles in the veterans' claims dataset. In *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law* (pp. 217-226).

Wan, Y. & Gao, Q. (2015). An ensemble sentiment classification system of Twitter data for airline services analysis. In *2015 IEEE International Conference on Data Mining Workshop* (pp. 1318-1325).

Wang, A. *et al*. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations* (eprint 1804.07461).

Zellers, R. *et al*. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural*

Language Processing (pp. 93-104).

Dimmy Magalhães: Analista de sistema do Tribunal de Justiça do Estado do Piauí desde 2013 e como Professor na Faculdade de Engenharia de Software iCEV desde 2022. Durante este período tem se dedicado ao desenvolvimento de sistemas inteligentes para o tribunal e na formação de graduandos na área de engenharia de software. Ele busca o aprimoramento de tecnologias com objeto de cooperar

com a produção de soluções computacionais área jurídica em prol da melhor prestação jurisdicional.

Aurora Pozo: Professora e pesquisadora no departamento de informática da Universidade Federal do Paraná, sendo atualmente professora titular. Durante este período tem se dedicado à formação de mestres e doutores de forma contínua. As suas contribuições científicas são na área de inteligência computacional. Ela mantém forte ênfase em cooperação nacional e internacional.

Sidnei Machado: Professor Associado de Direito do Trabalho da Faculdade de Direito da Universidade Federal do Paraná (UFPR). É professor permanente do Programa de Pós-Graduação em Direito da Universidade Federal do Paraná. É pesquisador e líder do Grupo de Pesquisa Clínica de Direito do Trabalho (PPGD/UFPR) e colaborador do Programa de Pós-Graduação em Sociologia da UFPR. Tem experiência nas áreas de Direito do Trabalho, Direito Constitucional, Direito da Seguridade Social e Sociologia Jurídica, com ênfase em atuação em pesquisas em temáticas sobre: direitos humanos, democracia, constitucionalismo social, regulação jurídica do trabalho, clínica Jurídica e pesquisa empírica no direito.

Data de submissão: 30/01/2022.
Data de aprovação: 03/09/2022.