REVISTA ELETRÔNICA DA PGE RJ
ISSN 2595-0630
(DOI): 10.46818/pge.v4i1.205

L.E.V.I: INTELIGÊNCIA ARTIFICIAL E INTEGRAÇÃO WEB SERVICE PARA AUTOMATIZAÇÃO DA DISTRIBUIÇÃO DE AÇÕES EM MEIO À CRISE DA COVID-19.

L.E.V.I: ARTIFICIAL INTELLIGENCE AND WEB SERVICE INTEGRATION TO AUTOMATE THE DISTRIBUTION OF LAWSUITS IN THE MIDST OF THE COVID-19 CRISIS.

Celso Araujo Fontes[1]

RESUMO: O presente trabalho tem como objetivo apresentar os esforços aplicados em meio à pandemia da COVID-19 para automatização da distribuição de ações no âmbito da Advocacia Pública Fluminense. O trabalho evidencia a premência da utilização de recursos automatizados que pudessem mitigar a complexidade na categorização das ações frente à grandeza do contencioso da Procuradoria Geral do Estado do Rio de Janeiro. A Inteligência Artificial, associada ao uso do Modelo Nacional de Interoperabilidade (MNI) surgiu como uma ferramenta para agilizar diversas atividades da Advocacia Pública, possibilitando maior agilidade e precisão no trabalho do Procurador de Estado e evitando a exposição dos Servidores da Procuradoria ao vírus Sars-CoV-2.

PALAVRAS-CHAVE: Advocacia Pública. Machine Learning. Inteligência Artificial. Web Service. Interoperabilidade. Tecnologia.

ABSTRACT: This paper aims to present the efforts applied in the midst of the COVID-19 pandemic to automate the distribution of lawsuits within the scope of the Fluminense's Public Advocacy. The article highlights the urgency of using automated resources that could mitigate the complexity in the categorization of lawsuits in face of the greatness of the litigation of the Attorney General of the State of Rio de Janeiro. Artificial Intelligence, linked to the use of the National Interoperability Model, emerged as a tool to streamline various activities of the Public Advocacy, allowing greater agility and precision in the work of the State Attorney and avoiding the exposure of Civil Servants to the Sars-CoV-2 virus.

KEYWORDS: Public Advocacy. Machine Learning. Artificial intelligence. Web Service. Interoperability. Technology.

[1] Mestre em Sistemas da Computação pelo IME-RJ e servidor do quadro da PGE-RJ.

das comunicações. 4.1.1. Afastamento do Procurador 4.2. Resultado dos envios de comunicações por e-mail 5. Conclusão. Referências.

SUMMARY: Introduction. 1. Integration with the Judiciary through the MNI. 2. Monitoring of lawsuits through administrative processes. 3. Prediction of the legal matter through Machine Learning. 3.1. Machine Learning for Document Classification. 3.2. Data collection for Machine Learning. 3.2.1. Recovery of the original lawsuit. 3.2.2. Identification of the complaint in the lawsuit documents tree. 3.2.3. Extraction of the text of the initial petition. 3.2.4. Optical Character Recognition for Scanned Documents. 3.2.5. Pre-processing of the text of the initial petition. 3.3. Classification 3.4. Prediction 3.4.1. Assessment of prediction 4. Sending communications via e-mail. 4.1. Identification of the initial recipient of the communications. 4.1.1. State attorney's leave 4.2. Result of summons by e-mail 5. Conclusion. References.

Introdução

Em 11 de Março de 2020, quando mais de uma centena de países no globo registraram casos de infecção por um então novo tipo de coronavírus (o Sars-CoV-2), a Organização Mundial da Saúde (OMS) declarou a situação como uma pandemia. Neste momento, o mundo, infelizmente, já contabilizava mais de 118 mil (cento e dezoito mil) casos e mais de 4.200 (quatro mil e duzentos) mortos[2]. Como não existiam vacinas ou tratamentos eficazes na época, uma das mais importantes estratégias de controle da contaminação foi a adoção de medidas de distanciamento social, que, apesar dos benefícios a ela associados, afetou (e até o momento em que escrevemos este artigo ainda tem afetado) grandemente o exercício regular de atividades em todos os setores da sociedade, o que inclui a advocacia pública fluminense.

De maneira a zelar pela integridade dos Procuradores e Servidores da Procuradoria Geral do Estado do Rio de Janeiro (PGE-RJ) em meio a essa pandemia, o então Procurador Geral Marcelo Lopes da Silva veiculou no Diário Oficial do Rio de Janeiro a resolução nº 4527 de 2020, estabelecendo parâmetros para a flexibilização de jornada de trabalho, afastamentos e até a possibilidade de execução remota de tarefas (teletrabalho)[3].

Com o propósito de atender os requisitos dessa resolução, o Gabinete da Procuradoria Geral do Estado do Rio de Janeiro (PGE-RJ) estabeleceu um plano de distribuição

[2] GHEBREYESUS, Tedros. *WHO Director-General's opening remarks at the media briefing on COVID-19*. WHO, 11 mar. 2020. Disponível em: https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020. Acesso em: 20 jan. 2021.

[3] RIO DE JANEIRO. Procuradoria Geral do Estado. Resolução PGE nº 4.527 de 16 mar. 2020. [Institui Medidas de Prevenção ao contágio da Covid-19, e dá outras providências]. *Diário Oficial do Estado do Rio de Janeiro*: parte 1: Poder Executivo, Rio de Janeiro, ano 46, n. 049, p. 37, 16 mar. 2020.

automatizada das comunicações judiciais (intimações e citações) por correio eletrônico, de maneira que, não apenas o Procurador do feito pudesse receber o documento em sua caixa de e-mail, mas também as equipes de triagem das Especializadas no caso de novas ações endereçadas ao Estado.

Para que tal fosse possível, foi necessário desenvolver um *software* utilizando um conjunto de tecnologias, serviços e aplicações de forma que a recuperação, identificação e distribuição das comunicações judiciais, na forma de arquivos PDF (*Portable Document Format*), pudessem ser enviadas para os responsáveis sem qualquer tipo de intervenção humana.

O desenvolvimento deste software foi basicamente dividido em três etapas:

A primeira etapa é a recuperação, através da integração MNI (Modelo Nacional de Interoperabilidade)[4], das informações dos processos judiciais, bem como dos documentos digitais que compõem as petições iniciais destes processos.

A segunda etapa é a identificação do estado da ação, isto é, se a ação já possui prévio acompanhamento por parte da PGE-RJ, através de um processo administrativo ou não. Em caso de acompanhamento preexistente, o Procurador do feito (ou seu possível substituto) é identificado. Por outro lado, na hipótese de uma nova ação, utiliza-se Inteligência Artificial e um conjunto de heurísticas para identificar a Especializada de competência frente à matéria da ação. Esta etapa também é dependente de um processo de identificação da petição inicial, pois nem sempre é trivial encontrar a mesma na árvore do processo.

A terceira e última etapa é o disparo em massa de e-mails contendo o documento (comunicação judicial) em anexo, além de um conjunto de informações sobre o processo e o possível estado de acompanhamento que visam facilitar a identificação rápida do teor da ação por parte do destinatário.

Nos tópicos a seguir serão detalhados os conceitos, tecnologias e recursos utilizados em cada um desses passos, bem como uma exposição sobre os esforços para otimizar a precisão da Inteligência Artificial.

1. Integração com o Judiciário através do MNI

[4] CONSELHO NACIONAL DE JUSTIÇA. Termo de Acordo de Cooperação Técnica nº58/2009. Presidente do Supremo Tribunal Federal e do Conselho Nacional de Justiça Gilmar Mendes.

Nos últimos anos, diariamente, todas as comunicações processuais oriundas do Tribunal de Justiça do Rio de Janeiro (TJRJ) são recuperadas de forma automatizada utilizando integração MNI (Modelo Nacional de Interoperabilidade). Este processo permitiu a substituição de um longo processo manual que envolvia o download unitário de cada comunicação no Portal do TJRJ.

A tecnologia envolvida neste processo é conhecida como *Web Service* (WS), que visa proporcionar um serviço para consumo e envio de informações entre sistemas de maneira agnóstica à linguagem de programação[5] utilizada por ambos.

A *Application Program Interface* (API) do MNI possui um conjunto de métodos que visam permitir a recuperação de dados dos processos judiciais, o que inclui seus metadados[6] (exemplos: Partes, Data de Ajuizamento, Órgão Julgador etc.) e também seus documentos (autos) e até suas respectivas movimentações. Além disso, também é possível protocolar manifestações judiciais e consultar os avisos pendentes, isto é, as comunicações judiciais endereçadas ao destinatário atrelado ao usuário MNI. Esse último recurso é exatamente o início da automatização dos processos de uma Procuradoria, pois, a maior parte do trabalho contencioso tem como principal atividade a defesa do Estado.

No total, o MNI[7] possui 6 (seis) métodos, são eles:

I. **ConfirmarRecebimento**: utilizado para recuperar a confirmação da tramitação de um processo entre um tribunal destinatário e seu respectivo remetente;
II. **ConsultarAlteracao**: possibilita a verificação de alterações em um processo, seja nos dados, documentos e/ou movimentações através da utilização de *hashes*;
III. **ConsultarAvisosPendentes**: recupera a lista de comunicações judiciais consideradas pendentes, isto é, sem prévia ciência do destinatário;
IV. **ConsultarProcesso**: recupera os metadados e documentos de um processo judicial através de seu número padrão CNJ[8];

[5] Linguagem de programação é um conjunto de recursos com uma sintaxe pré-definida utilizada para o desenvolvimento de programas através da interpretação e/ou compilação de um código fonte.
[6] Descritores e seus respectivos valores.
[7] MNI versão 2.2, utilizada pelo TJRJ até o momento do fechamento deste trabalho.
[8] CONSELHO NACIONAL DE JUSTIÇA. Resolução nº 65 de 16/12/2008. Ministro Gilmar Mendes.

V. **ConsultarTeorComunicacao**: recupera o conteúdo da comunicação judicial através do número do aviso pendente;

VI. **EntregarManifestacaoProcessual**: permite protocolar um documento em uma ação pré-existente ou uma petição inicial (ambos, opcionalmente, com seus respectivos anexos);

Para acessar os métodos de um WS, é necessário obter o endereço *endpoint*[9] do mesmo. No caso do padrão de WS utilizado pelo MNI, o padrão SOAP (*Simple Object Access Protocol*)[10], o *endpoint* é um arquivo WSDL (*Web Service Description Language*), responsável por explicitar todas as informações sobre os métodos, entidades, parâmetros de entrada e os possíveis valores de retorno. Comumente, as WSDL são disponibilizadas na internet em forma de endereço URL (*Uniform Resource Locator*) para que o serviço tenha maior abrangência de cobertura na rede.

No contexto do TJRJ, por exemplo, o endereço da WSDL é o https://webserverseguro.tjrj.jus.br/MNI/Servico.svc?wsdl.

De posse do endereço WSDL, o usuário necessitará utilizar algum *software* ou biblioteca *SOAP Client* que seja capaz de "consumir" os recursos do serviço, isto é, invocar os métodos disponíveis no WS e receber os dados de acordo com os parâmetros enviados. Ademais, esse software ou biblioteca precisa ser capaz de manipular a linguagem XML (*Extensible Markup Language*), utilizada para expressar a comunicação entre o cliente e o servidor SOAP. Esta linguagem possui como principal característica a utilização de *tags* (marcadores delimitados pelos caracteres "menor que" e "maior que") para expressar estruturas de dados. O exemplo a seguir utiliza o XML para representar uma entidade "pessoa" e sua entidade filha "endereço":

Figura 1 - XML Representando uma pessoa e seu respectivo endereço

[9] Ponto de acesso de servidor web.

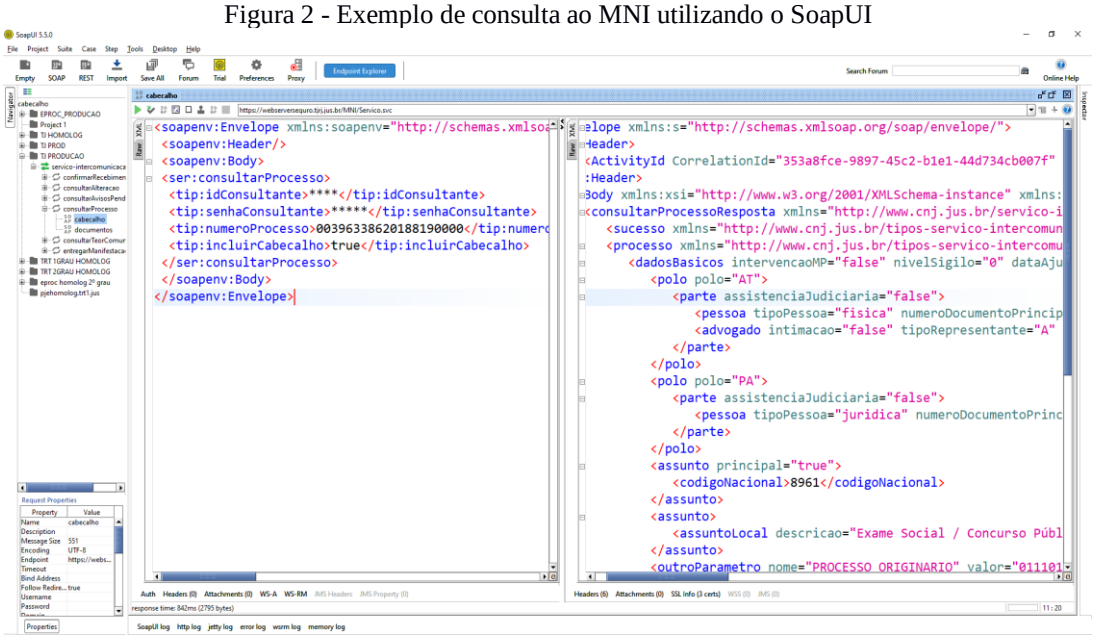
[10] Protocolo para troca de informações entre servidor e cliente que utiliza o XML para a comunicação.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<pessoa cpf="99999999999">
    <nome>Fulano da Silva</nome>
    <sexo>M</sexo>
    <endereco>
        <rua>Rua XPTO</rua>
        <bairro>Feliz</bairro>
        <cidade>Rio de Janeiro</bairro>
        <uf>RJ</uf>
    </endereco>
</pessoa>
```

Fonte: O autor, fevereiro de 2021.

Uma possibilidade de software *Soap Client* bastante popular e capaz de cumprir os requisitos supracitados é o SoapUI[11], cuja interface gráfica é demonstrada na imagem seguinte, onde a consulta de dados de um processo judicial é efetuada através do método *consultarProcesso* cujos parâmetros de entrada são: *idConsultante* e *senhaConsultante*, que representam o login e senha do usuário MNI; *numeroProcesso*, que identifica o número no padrão CNJ no qual se deseja recuperar as informações; *incluirCabecalho*, parâmetro booleano (verdadeiro ou falso) que indica a necessidade (ou não) da recuperação de todos os metadados do processo judicial.

Figura 2 - Exemplo de consulta ao MNI utilizando o SoapUI



Fonte: O autor, fevereiro de 2021.

[11] SMARTBEAR. *SoapUI Open Source*. Versão 5.5.0. [S.I.]: SmartBear Software, 2021. Disponível em https://www.soapui.org. Acesso em: 28 fev. 2021.

Apesar das facilidades da ferramenta, o SoapUI é limitado no que tange a recuperação programática do resultado da execução de múltiplas requisições sequenciais a um WS. Por isso, no contexto da PGE-RJ, optou-se por utilizar a biblioteca *SoapClient*[12], nativa da linguagem de programação PHP[13], que permitiu a implementação de um módulo de integração para o consumo em lote dos dados deste *Web Service*. Tal medida é justificada pelo volume das comunicações endereçadas à PGE-RJ pelo TJRJ via MNI, uma média de 1.382 (mil trezentas e oitenta e duas) diariamente[14]. Cada Comunicação recuperada através do método *consultarTeorComunicacao* tem seus metadados, documentos e informações sobre os processos judiciais persistidos em registros de tabelas em um Sistema Gerenciador Banco de Dados Relacional. No caso da PGE-RJ, o Microsoft SQL Server[15], permitindo que todo este conjunto de dados possa ser processado para uma distribuição interna, de acordo com as matérias jurídicas delegadas para cada Especializada[16].

A partir do momento em que a comunicação de um processo judicial é acautelada no ambiente da PGE-RJ, o próximo passo é o acompanhamento da ação por meio de um processo administrativo.

2. Acompanhamento de ações através de processos administrativos

A PGE-RJ adota o uso de processos administrativos como forma de acompanhamento de processos judiciais. Esta abordagem permite maior celeridade, lisura e organização do caso de maneira que todos os atos internos realizados para o tratamento da ação judicial sejam preservados para que qualquer Procurador possa assumir o feito.

Um processo administrativo pertence a um conjunto de processos denominado acervo, este, por sua vez, é de responsabilidade de um Procurador específico lotado em uma Especializada da PGE-RJ. Os acervos podem representar processos de matérias similares ou apenas a divisão equânime de um total de processos de uma Especializada por Procurador.

---

[12] ZEND. *SOAP CLIENT PHP*. Disponível em: https://www.php.net/manual/en/class.soapclient.php. Acesso em: 28 fev. 2021.
[13] ZEND. *PHP*. Versão 7.0. [S.I.]: Zend. Disponível em: https://www.php.net/. Acesso em: 28 fev. 2021.
[14] Valores recuperados entre maio de 2020 e fevereiro de 2021.
[15] MICROSOFT. *SQL Server*. Disponível em: https://www.microsoft.com/pt-br/sql-server/sql-server-downloads Acesso em: 28 fev. 2021.
[16] PGE-RJ. *Estrutura*. 2021. Disponível em: https://pge.rj.gov.br/institucional/estrutura. Acesso em: 1 mar. 2021.

Até novembro de 2019, o acompanhamento de processos administrativos na PGE-RJ era feito exclusivamente através de uma abordagem híbrida utilizando o sistema Sicaj, para o registro digital dos metadados dos processos, e pastas de papel, para o armazenamento físico dos documentos dos processos.

A partir de dezembro de 2019 foi introduzido na PGE-RJ um novo sistema de acompanhamento processual denominado PGE Digital. Esse sistema foi desenvolvido internamente pela equipe de Tecnologia da Informação do órgão[17], e possui como principal característica a utilização do MNI para possibilitar uma completa informatização das atividades do contencioso, desde o recebimento das comunicações judiciais até a protocolização de petições. Este sistema está sendo gradualmente implantado nas Especializadas da PGE-RJ e substituindo tanto o Sicaj quanto as pastas de papel para acompanhamento de processos.

Apesar dos recursos disponibilizados pelo sistema PGE Digital, fez-se necessário desenvolver uma nova ferramenta que pudesse ser capaz de identificar a Especializada competente para novas ações recebidas via MNI, de maneira que as triagens das Especializadas pudessem receber digitalmente as comunicações processuais sem a intervenção humana, mitigando os impactos causados pela pandemia. A solução encontrada para este problema é o tema do tópico a seguir.

3. Predição da Especializada de competência através de Machine Learning

Atualmente, uma média de 500[18] (quinhentas) novas ações são endereçadas ao Estado do Rio de Janeiro por dia. Cada ação é, normalmente, composta por uma petição inicial. Através da interpretação dos fatos e do pedido em si, descritos na petição inicial, é possível determinar uma ou mais matérias do Direito, as quais o processo judicial pode ser classificado. Por seu turno, cada matéria possui uma Especializada de competência na PGE-RJ. As Especializadas, como o próprio nome já evidencia, são departamentos "especializados" da PGE-RJ onde a atuação dos Procuradores é focada em processos limitados ao conjunto de matérias de competência da Especializada.

[17] Com apoio de consultoria da Fundação COPPETEC – UFRJ.
[18] Valores recuperados entre maio de 2020 e fevereiro de 2021.

Em virtude dessa quantidade de novas ações, é inevitável um esforço significativo para a tarefa de identificar a Especializada de competência. Durante anos, esta tarefa foi responsabilidade única de um departamento da PGE-RJ cognominado Central de Mandados.

Através de uma equipe de aproximadamente 10 (dez) colaboradores, diariamente, os profissionais deste departamento precisavam acessar a árvore de documentos do portal do TJRJ para identificar e ler a petição inicial de cada nova ação, muitas vezes composta por centenas de páginas, para verificar a atribuição correspondente a cada Especializada.

Então, face aos dados apresentados, bem como o agravante causado pela situação do risco eminente de contaminação pelo vírus Sars-CoV-2, aventou-se a possibilidade de encontrar uma nova tecnologia ou ferramenta que pudesse ser capaz de simular a atividade humana desempenhada pela equipe da Central de Mandados de maneira a ser hábil em designar uma Especializada da PGE para uma nova ação.

A primeira possibilidade levantada foi do uso dos assuntos (matérias) dos processos judiciais recuperados através da consulta de processos ao MNI. Entretanto, essa abordagem mostrou-se ineficiente para a maioria dos casos, pois, verificou-se que uma quantidade significativa dos processos judiciais endereçados à PGE-RJ utilizava assuntos abrangentes (ex.: Antecipação de Tutela), o que inviabilizaria uma definição precisa da Especializada de competência.

Outra estratégia aventada foi a possibilidade de uma relação de exclusividade entre determinadas serventias (órgão julgador no MNI) e Especializadas da PGE-RJ. Essa técnica mostrou-se eficiente para algumas Especializadas como, por exemplo, a PG14[19], pois esta possui uma relação de exclusividade com serventias de “órfãos e sucessões" e “família”. Contudo, a maioria das Especializadas não gozam de nenhuma relação de exclusividade similar, tornando essa estratégia improfícua.

Neste contexto, constatou-se que a utilização dos metadados do processo judicial disponibilizados pelo MNI não seriam o suficiente para suceder integralmente a atividade manual da identificação da Especializada de competência através da leitura do texto contido na petição inicial.

Decidiu-se então encontrar uma tecnologia capaz de, através do conjunto de palavras de um texto, determinar uma classe ou categoria a qual o texto pertenceria. Dentre as tecnologias pesquisadas, constatou-se uma abordagem predominantemente utilizada pela

[19] Procuradoria de sucessões (PG-14).

comunidade científica como solução eficiente para esse tipo de problema: o Aprendizado de Máquina (*Machine Learning*)[20].

A próxima seção irá detalhar a utilização de *Machine Learning* para classificar documentos, bem como as tecnologias aplicadas para implementação de um *framework* que utiliza *Machine Learning* para classificar automaticamente as ações judiciais endereçadas à PGE-RJ.

3.1. Machine Learning Para Classificação de Documentos

El Naqa e Murphy definem o *Machine Learning* como "um ramo em evolução de algoritmos computacionais projetados para emular a inteligência humana aprendendo com o ambiente circundante"[21]. Koteluk *et al*[22] especifica o *Machine Learning* com um "subcampo da ciência da inteligência artificial que permite que a máquina se torne mais eficaz com a experiência de treinamento". Por sua vez, Alpaydın destaca a necessidade da utilização de "dados de exemplo" em sua definição de *Machine Learning*:

> Machine Learning é a programação de computadores para otimizar um critério de desempenho usando dados de exemplo ou experiências anteriores. Precisamos de casos de aprendizagem onde não podemos escrever diretamente um programa de computador para resolver um determinado problema, mas precisamos de dados de exemplo ou experiência[23].

E no caso específico de *Machine Learning* (ML) para classificação de documentos, Sebastiani (2002) detalha que "documentos pré-classificados" atuam como dados de exemplo para aprendizagem, onde a abordagem de ML depende da disponibilidade de um *corpus*[24] inicial $\Omega = \{d_1,...,d_{|\Omega|}\} \subset D$ de documentos pré-classificados em $C = \{c_1,...,c_{|C|}\}$, sendo C o conjunto de classes às quais cada documento pertence.

---

[20] SEBASTIANI, Fabrizio. *Machine learning in automated text categorization*. In: ACM computing surveys (CSUR), 2002.

[21] EL NAQA, Issam; MURPHY, Martin J. *What is machine learning?*. In: Machine learning in radiation oncology. Cham: Springer, 2015, p. 3-11.

[22] KOTELUK, Oliwia *et al*. *How Do Machines Learn? Artificial Intelligence as a New Era in Medicine*. In: Journal of Personalized Medicine, v. 11, n. 1, 2021, p. 3.

[23] ALPAYDIN, Ethem. *Introduction to machine learning*. 2º edição. Londres: The MIT Press, 2010, p. 32.

[24] Termo utilizado para se referir a grandes coleções de textos que representam uma amostra de uma variedade particular ou uso de linguagens que são apresentados em forma legível por máquina (ESSEX, 1998).

Por sua parte, cada documento *d* precisa ser subdivido em um conjunto de fragmentos $T = \{t_i,...,t_n\}$, através de um processo chamado de *tokenização*, que basicamente consiste em dividir o texto de um documento em um conjunto de palavras (*tokens*) através de um divisor (exemplo: caractere de espaço).

A *tokenização* é o início de uma subfase importante da atividade de classificação de textos conhecida como pré-processamento, responsável por "converter os dados textuais originais em uma estrutura adequada para a mineração de dados, onde os recursos de texto mais significativos que servem para diferenciar as categorias são identificados"[25].

Após a *tokenização* dos documentos, outras três etapas significativas podem ser aplicadas para cada *token*, de maneira que a quantidade de termos irrelevantes possa ser reduzida significativamente.

A primeira é a remoção de *stop-words*, que visa eliminar as palavras sem peso semântico significativo em um determinado idioma, como artigos, conjunções e preposições[26]. Alguns exemplos de stop-words do idioma português seriam palavras como "de", "para", "mas", "a" e "o".

A segunda etapa é conhecida como *stemming*: um algoritmo que se propõe a reduzir todas as palavras com a mesma raiz a uma forma comum. Geralmente, esse procedimento é feito através da remoção dos sufixos derivacionais e flexionais de cada palavra[27]. Um exemplo de *stemming* seria reduzir palavras em português como "pedra", "pedreira" e "pedraria" para "pedr".

A terceira etapa é a utilização de expressões regulares, que permitem a localização de sequências de caracteres em textos que satisfaçam um determinado padrão (expressão). No contexto do pré-processamento, expressões regulares podem ser utilizadas para remover determinados termos considerados irrelevantes para classificação. Alguns exemplos neste contexto seriam: "25/11/1985" (data), gti@pge.rj.gov.br (e-mail), "exmo." e "sra." (pronomes de tratamento).

Contudo, *tokens* relevantes não são o suficiente para que um classificador de ML seja capaz de aprender a identificar uma categoria. Basarkar destaca em seu trabalho que para

[25] SRIVIDHYA, V.; ANITHA, R. *Evaluating preprocessing techniques in text categorization.* In: International journal of computer science and application, v. 47, n. 11, 2010, p. 49-51.
[26] RAULJI, Jaideepsinh K.; SAINI, Jatinderkumar R. *Stop-word removal algorithm and its implementation for Sanskrit language*. In: International Journal of Computer Applications, v. 150, n. 2, 2016, p. 15.
[27] LOVINS, Julie Beth. *Development of a stemming algorithm*. Mech. Transl. Comput. Linguistics, v. 11, n. 1-2, 1968, p. 22-31.

"realizar a classificação de documentos por meio de algoritmos, os documentos precisam ser representados de forma que sejam compreensíveis para o classificador de aprendizado de máquina"[28].

Esta representação é comumente implementada através de uma matriz esparsa M = i x j, onde i é o identificador de um dos documentos em D e j o identificador de um dos *tokens* presentes no conjunto total de palavras extraídas e pré-processadas de todos os documentos (formando um vocabulário). Por seu turno, os valores desta matriz seriam a quantidade de vezes (frequência) em que um termo t_i aparece em um documento d_i[29].

Uma matriz esparsa é computacionalmente ideal pois a maioria dos documentos contém apenas um pequeno subconjunto das palavras no vocabulário, o que significa que a maioria das entradas na matriz serão 0 (zero)[30].

De posse dessa matriz, opcionalmente, é comum também a execução de uma etapa adicional denominada de normalização da frequência. Essa etapa é importante para o uso de ML em textos pois, palavras de baixa frequência costumam ser mais discriminativas do que palavras de alta frequência[31].

A implementação desta etapa consiste em efetuar um novo cálculo sobre a frequência dos termos nos documentos, levando em consideração a frequência de sua ocorrência normalizada pelo comprimento do documento e a contagem normalizada pela frequência inversa do documento da palavra[32].

Com a matriz esparsa normalizada M, o passo seguinte é a etapa de classificação que consiste em utilizar algum "classificador" responsável por aplicar uma função f para cada item de M levando em consideração sua classe c_i pré-classificada em C. E o resultado desta classificação é um modelo de treinamento, capaz de ser utilizado posteriormente para a predição em outros documentos, determinando uma das classes em C para os mesmos.

---

[28] BASARKAR, Ankit. *Document classification using machine learning*. Master's Projects San Jose State University. 2017, p. 4.

[29]PEDREGOSA, Fabian *et al*. *Working With Text Data*. 2021. Disponível em: https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html. Acesso em 28 fev. 2021.

[30] MÜLLER, Andreas C.; GUIDO, Sarah. *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016.

[31] AGGARWAL, Charu C. *Machine learning for text*. Cham: Springer International Publishing, 2018, p. 6.

[32] IKONOMAKIS, M.; KOTSIANTIS, Sotiris; TAMPAKAS, V. *Text classification using machine learning techniques*. In: WSEAS transactions on computers, v. 4, n. 8, 2005, p. 966-974.

Atualmente, existem diversas implementações de ferramentas e bibliotecas de ML, destaca-se entre elas a biblioteca *Sckit-learn*[33], implementada na linguagem de programação *python*, que disponibiliza um conjunto de recursos de ML que abrangem desde o pré-processamento até a predição por diferentes tipos de classificadores.

É possível, outrossim, utilizar bibliotecas de processamento de linguagem natural para auxiliar a etapa de pré-processamento. Dentre as bibliotecas disponíveis para a linguagem *python*, destaca-se a NLTK (*Natural Language Toolkit*)[34], que provê recursos como *stemming*, *tokenização* e remoção de *stopwords* no idioma português.

E através da utilização destas bibliotecas de *Machine Learning* e processamento de linguagem natural, foi desenvolvido um *framework* denominado de L.E.V.I.[35] (*Legal Enviroment Virtual Inteligence*), cuja finalidade é proporcionar a classificação de processos judiciais através do texto de suas respectivas petições iniciais.

Na próxima seção, será descrita a sistemática utilizada para recuperação desses documentos.

3.2. Coleta dos dados para o Machine Learning

Algoritmos de *Machine Learning* supervisionados necessitam de dados pré-classificados para "aprender" melhor sobre cada categoria. Para cumprir esse pré-requisito, foi necessário recuperar um conjunto de processos previamente classificados, isto é, processos judiciais que já tiveram suas petições iniciais analisadas e suas Especializadas de competência definidas.

Para maior equidade dos dados, foram recuperados um total de 6.000 (seis mil) números no padrão CNJ por Especializada (oito Especializadas[36]), totalizando 48.000 (quarenta e oito mil) números únicos de processos judiciais.

---

[33] PEDREGOSA, Fabian *et al*. *Scikit-learn: Machine learning in Python*. In: The Journal of machine Learning research, 2011, p. 2825-2830.

[34] BIRD, Steven; KLEIN, Ewan; LOPER, Edward. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, 2009.

[35] Homenagem ao filho do autor deste artigo.

[36] Especializadas que recebem processos do TJRJ, são elas: Tributária (PG03), Pessoal (PG04), Dívida Ativa (PG05), Patrimônio e do Meio Ambiente (PG06), Previdenciária (PG07), Serviços Públicos (PG08), Sucessões (PG14) e Serviços de Saúde (PG16).

Porém, muitos desses números de processos dizem respeito aos autos de outros incidentes processuais (como agravo de instrumento, cumprimento de sentença etc.) que não àqueles da ação originária onde se encontra a petição inicial, peça fundamental para a classificação do processo.

Faz-se necessário, então, utilizar os dados MNI para recuperar corretamente o número do processo originário. A metodologia utilizada para realizar esta atividade está descrita na próxima seção.

3.2.1. Recuperação do processo originário

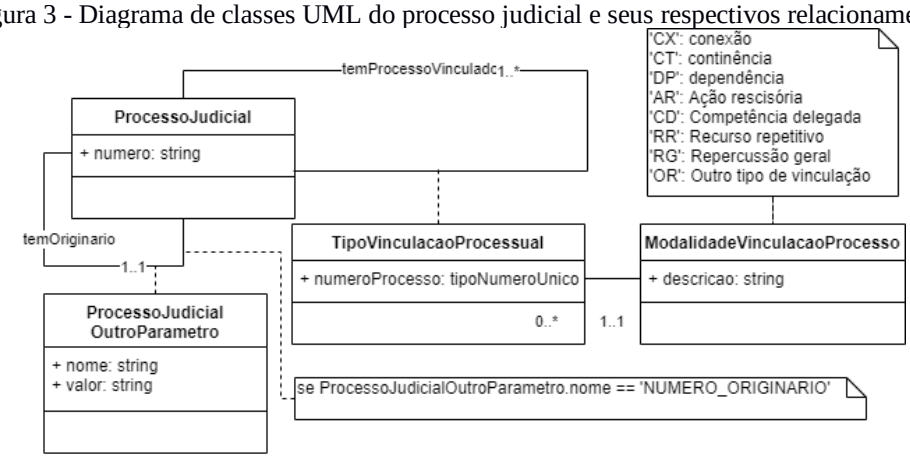
Durante o desenvolvimento deste trabalho, foram encontradas duas possibilidades de recuperação do processo originário por meio de:

I. **Processos vinculados**: o MNI define um tipo de vinculação entre processos chamado de "VinculacaoProcessual". Este recurso é bastante utilizado em situações como, por exemplo, Embargos, de maneira a permitir com que seja explícito a vinculação entre o Embargo e seu processo originário através do tipo de modalidade de vinculação processo denominado de "DP" (dependência);
II. **Processos originários**: como esta versão do MNI (2.2) não possui um metadado explícito para definição de processo originário, nem mesmo uma vinculação deste tipo, o TJRJ optou por utilizar o metadado "outroParametro" do cabeçalho do Processo Judicial para informar, através de um parâmetro chamado NUMERO_ORIGINARIO, o número do CNJ (valor) para indicar a vinculação entre dois processos. Observou-se que esse recurso é bastante utilizado para explicitar a vinculação entre agravos e seus originários.

O diagrama de classes UML (*Unified Modeling Language*)[37] a seguir demonstra as entidades e relacionamentos utilizados para a recuperação do processo do processo originário através do MNI:

[37] Linguagem de Modelagem Unificada.

Figura 3 - Diagrama de classes UML do processo judicial e seus respectivos relacionamentos



Fonte: O autor, fevereiro de 2021.

Então, de posse do número do processo originário, a etapa subsequente foi a identificação da petição inicial na árvore do processo originário.

3.2.2. Identificação da petição inicial na árvore do processo

Conforme citado anteriormente, através do MNI é possível recuperar a lista de documentos de um processo judicial. Por sua vez, cada documento possui um conjunto de metadados associados como descrição, tipo de documento e outros parâmetros. Apesar de a definição local do TJRJ[38] possuir um tipo local chamado "Petição Inicial", observou-se que uma quantidade significativa de processos não faz uso explícito desse tipo em suas petições iniciais, e, em muitos casos, as mesmas são definidas utilizando apenas o tipo "petição" e, no metadado "descrição", texto livre para descrever brevemente o documento, a informação "petição inicial" ou "inicial" é utilizada, além de outras situações.

Tendo em vista as consideráveis possibilidades de definição da petição e uma priorização dos casos "ideais" (petição inicial explícita) para os "não ideais" (sem petição inicial explícita), decidiu-se por implementar um componente baseado em um *Design pattern* denominado de "*Chain-of-responsability*", que permite a fácil manutenção evolutiva de uma cadeia de decisão apenas acrescentando mais uma classe (elo) no grupo (corrente) de classes[39].

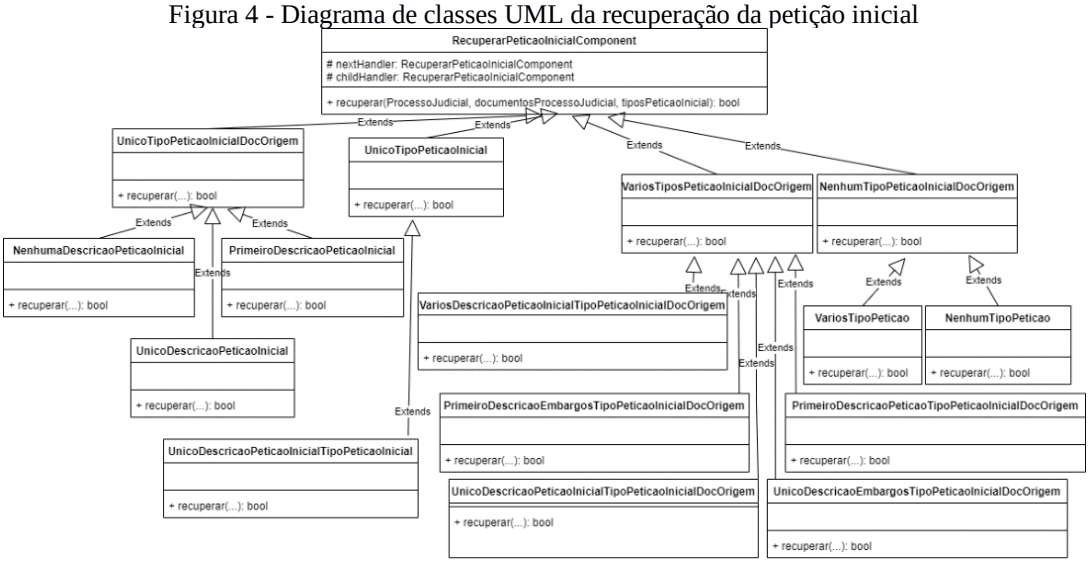
[38] HINKEL, Angélica. Anexo II do Manual do Modelo Nacional de Interoperabilidade. Rio de Janeiro: TJRJ, 2016. Disponível em: http://portaltj.tjrj.jus.br/documents/10136/3067978/anexo-ii-manual-modelo-nacional-interoperabilidade.pdf. Acesso em: 28 fev. 2021.

[39] GAMMA, Erich *et al*. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley

O algoritmo desse componente percorre os elos (executando o método recuperar) até que algum destes retorne verdadeiro de maneira que identifique qual dos documentos da árvore do processo tem maior probabilidade de conter o pedido inicial da ação.

O diagrama UML a seguir descreve as classes e subclasses utilizadas nesta implementação para recuperar a petição inicial na árvore.

Figura 4 - Diagrama de classes UML da recuperação da petição inicial



Fonte: O autor, fevereiro de 2021.

Após o processo de definição de qual, dentro os documentos da árvore do processo judicial, é a petição inicial, a ação subsequente é a recuperação do conteúdo do deste documento. Para a realização deste feito, faz-se necessário utilizar o método *consultarProcesso* do MNI utilizando os parâmetros *idConsultante*, *senhaConsultante*, *numeroProcesso* e *documento*. O parâmetro *documento* é o valor do *idDocumento* presente na lista de documentos do processo.

Os XML a seguir exemplificam a recuperação de um documento específico no MNI do TJRJ:

Figura 5 - XML (simplificado) de consulta de processo do judicial com id de documento

Professional, 1994.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope>
   <soapenv:Header/>
   <soapenv:Body>
      <ser:consultarProcesso>
         <tip:idConsultante>******</tip:idConsultante>
         <tip:senhaConsultante>******</tip:senhaConsultante>
         <tip:numeroProcesso>00667272420108190021</tip:numeroProcesso>
         <tip:documento>58813729</tip:documento>
      </ser:consultarProcesso>
   </soapenv:Body>
</soapenv:Envelope>
```

Fonte: O autor, fevereiro de 2021.

Onde o retorno esperado pelo servidor é apresentado na imagem a seguir:

Figura 6 - XML (simplificado) de resposta do WS MNI para consulta de processo com documento

```xml
<consultarProcessoResposta>
 <sucesso>true</sucesso>
 <processo>
  <documento nivelSigilo="0"
    tipoDocumentoLocal="14"
    descricao="Outros"
    idDocumento="58813729"
    tipoDocumento="58"
    dataHora="20130814145138">
    <conteudo>JVBERi0xLjYNJeLjz9MNCjQgMCBvYm[...]</conteudo>
        <outroParametro nome="IDENTIFICADOR_DOCUMENTO" valor="58813729"/>
        <outroParametro nome="VOLUME" valor="1"/>
        <outroParametro nome="NUMERO_FOLHA_VIRTUAL" valor="1"/>
        </documento>
 </processo>
</consultarProcessoResposta>
```

Fonte: O autor, fevereiro de 2021.

Dentre os atributos e valores retornados na consulta anterior, o valor da *tag* "<*conteúdo*>" é o responsável por armazenar o PDF no formato *base64*. Esse formato possibilita a inclusão de arquivos binários em XML.

Com o arquivo PDF da petição inicial, a etapa seguinte é recuperar o texto livre deste documento.

3.2.3. Extração do texto da petição inicial

Documentos PDF são arquivos binários compostos de diferentes informações que possibilitam desde a inclusão de texto e imagens formatadas até formulários e gráficos em três dimensões. Como não é possível extrair apenas o texto de um PDF utilizando uma ferramenta

ou biblioteca de edição de texto, fez-se necessário utilizar uma ferramenta ou biblioteca capaz de tal feito.

Para este trabalho, utilizou-se inicialmente a ferramenta *pdftotext*, integrante do utilitário *Xpdf*[40], capaz de gerar um arquivo **.txt* (texto livre sem formatação) a partir de um PDF.

Apesar do êxito na maioria das petições iniciais recuperadas utilizando esta ferramenta, diversos documentos **.txt* retornavam completamente vazios. Analisando esta situação, descobriu-se que uma quantidade significativa de petições iniciais eram PDF formados apenas por imagens, mais precisamente, textos digitalizados.

De posse desse fato, fez-se necessário encontrar algum recurso que possibilitasse a extração de texto de documentos digitalizados. A solução encontrada foi a utilização de OCR (*Optical Character Recognition*) e os detalhes da utilização desta tecnologia serão descritos na próxima seção.

3.2.4. Reconhecimento óptico de caracteres para documentos digitalizados

Apesar da iniciativa notória do Judiciário Fluminense em tornar seus processos eletrônicos desde a origem, uma quantidade significativa de processos físicos digitalizados continuam em tramitação. Devido ao formato fotográfico destes documentos, faz-se necessária a utilização de algum recurso que seja capaz de reconhecer os caracteres contidos nestes documentos.

Para tal feito, optou-se por utilizar a *engine* de reconhecimento ótico de caracteres *Tesseract*[41], que possibilita a recuperação parcial ou total do texto contido em uma imagem. Esta imagem pode estar representada, por exemplo, no formato PNG (*Portable Network Graphics*).

Todavia, como o formato dos documentos disponibilizados pelo MNI do TJRJ é o PDF, lançou-se mão de uma etapa adicional para a conversão de cada página de um documento PDF para uma imagem PNG, a fim de possibilitar que o *Tesseract* pudesse ser hábil a extrair o texto dos documentos digitalizados.

---

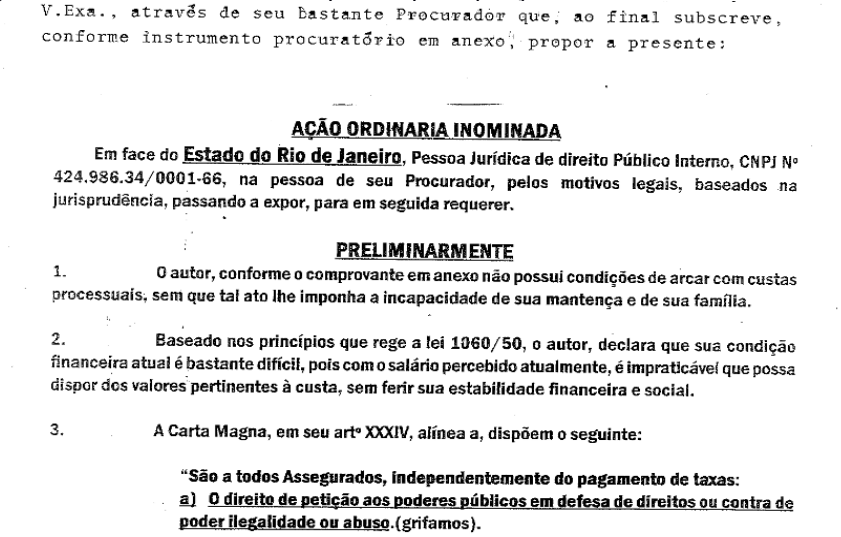
[40] GLYPH & COG, *XpdfReader*. 2021. Disponível em: https://www.xpdfreader.com/. Acesso em: 07 mar. 2021.

[41] SMITH, Ray. *An overview of the Tesseract OCR engine*. In: Ninth international conference on document analysis and recognition (ICDAR 2007). IEEE, 2007, p. 629-633.

Tendo isto em vista, foi utilizada inicialmente uma ferramenta intitulada de *pdftopng*, integrante do utilitário *Xpdf*. Porém, após algumas semanas de uso, foram observadas algumas limitações no mesmo, principalmente no que tange a PDF no formato PDF/A. Então, para sobrepujar tal limitação, decidiu-se por migrar para outra ferramenta chamada *Imagemagick*[42] com a qual os documentos puderam ser convertidos para PNG possibilitando o passo seguinte: o reconhecimento ótico pelo *Tesseract*.

Na imagem a seguir, é demonstrado um exemplo de um processo judicial ajuizado em novembro de 2014, cuja inicial é um documento contendo apenas as imagens digitalizadas de um ou mais documentos. Além disso, ainda é possível notar que os caracteres do documento possuem diferentes fontes, tamanhos e estilos em uma mesma página. Também é possível notar linhas e pequenas rasuras, provavelmente oriundas do processo de impressão aumentando significativamente a dificuldade para o reconhecimento ótico.

Figura 7 - Trecho do documento que compõe a petição inicial de um Processo Judicial



V.Exa., através de seu bastante Procurador que, ao final subscreve, conforme instrumento procuratório em anexo, propor a presente:

AÇÃO ORDINARIA INOMINADA

Em face do **Estado do Rio de Janeiro**, Pessoa Jurídica de direito Público Interno, CNPJ Nº 424.986.34/0001-66, na pessoa de seu Procurador, pelos motivos legais, baseados na jurisprudência, passando a expor, para em seguida requerer.

PRELIMINARMENTE

1. O autor, conforme o comprovante em anexo não possui condições de arcar com custas processuais, sem que tal ato lhe imponha a incapacidade de sua mantença e de sua família.

2. Baseado nos princípios que rege a lei 1060/50, o autor, declara que sua condição financeira atual é bastante difícil, pois com o salário percebido atualmente, é impraticável que possa dispor dos valores pertinentes à custa, sem ferir sua estabilidade financeira e social.

3. A Carta Magna, em seu artº XXXIV, alínea a, dispõem o seguinte:

"São a todos Assegurados, independentemente do pagamento de taxas:
a) O direito de petição aos poderes públicos em defesa de direitos ou contra de poder ilegalidade ou abuso.(grifamos).

Fonte: Documento recuperado via MNI TJRJ do processo nº 0411983-35.2014.8.19.0001 p.3[43]

Na figura a seguir, é descrito *ipsis litteris* o mesmo trecho do texto da imagem anterior, porém, desta vez, em sua versão textual recuperada através de conversão para PNG e leitura ótica:

[42] IMAGEMAGICK DEVELOPMENT TEAM. 2021. ImageMagick. Disponível em: https://imagemagick.org. Acesso em: 07 mar. 2021.

[43] RIO DE JANEIRO. Tribunal de Justiça do Rio de Janeiro. Recurso Extraordinário 0411983-35.2014.8.19.0001. Des. Peterson Barroso Simão. 15 fev. 2021.

Figura 8 - Resultado do OCR utilizando o *Tesseract* com os erros gramaticais destacados

"V.Exa., através de seu bastante Procurador que; ao final subscreve, conforme instrumento ***procuratôrio*** *em anexo) propor a presente: AÇÃO ORDINARIA INOMINADA Em face do Estado do Rio de Janeiro, Pessoa Jurídica de direito Público Interno, CNP Nº 424.986.34/0001-66, na pessoa de seu Procurador, pelos motivos legais, baseados na jurisprudência, passando a expor, para em seguida requerer. PRELIMINARMENTE 1, O autor, conforme o comprovante em anexo não possui condições de arcar com custas processuais, sem que tal ato lhe imponha a Incapacidade de sua mantença e de sua família. 2. Baseado nos princípios que rege a lei 1060/50, o autor, declara que sua* ***condiçãe*** *financeira atual é bastante difícil, pois como salário percebido atualmente, é impraticável que possa dispor dos valores pertinentes à custa, sem ferir sua estabilidade financeira e social. 8. A Carta Magna, em seu artº XXXIV, alínea a, dispõem o seguinte: "são a todos Assegurados, independentemente do pagamento de taxas: a) O direito de petição aos poderes públicos em defesa de direitos ou contra de* ***peder*** *ilegalidade ou abuso.(grifamos)."*

Fonte: O autor, fevereiro de 2021.

Analisando o resultado do OCR é possível verificar que, apesar das condições intrínsecas do documento original, apenas 3 (três) palavras, de um total de 171 (cento e setenta e uma), foram geradas com erros gramaticais, demonstrando a boa eficiência da ferramenta.

Sendo assim, após a recuperação do texto da petição inicial, seja por texto livre ou por reconhecimento ótico, o próximo passo é o pré-processamento do texto.

3.2.5. Pré-processamento do texto da petição inicial

Conforme descrito anteriormente, textos precisam ser convertidos para uma estrutura adequada para o *Machine Learning*. Este processo de conversão é conhecido como "pré-processamento" do texto e o mesmo é normalmente composto pelas etapas de *tokenização*, remoção de termos de irrelevantes, *stemming*, vetorização e normalização.

Para implementação de cada uma destas etapas na implementação deste trabalho, utilizou-se os seguintes recursos:

I. ***NLTK Tokenizer***: classe integrante da biblioteca NLTK, que possibilita efetuar a *tokenização* dos textos através do método *word_tokenize*();

II. ***Regex***[44]: módulo alternativo ao *re* (nativo do *python*), que possibilita a substituição de termos através do método *sub()*. Por meio de um conjunto de 36 (trinta e seis) expressões regulares definidas para este trabalho, o método *sub()* do regex foi utilizado para substituir termos considerados irrelevantes pelo caractere "espaço";

III. ***RSLP Stemmer***: implementação *python*[45] do RSLP *Stemmer* (Removedor de Sufixos da Língua Portuguesa)[46] disponível no pacote de módulos do NLTK, permite reduzir os termos para suas respectivas raízes;

IV. ***TfIdfTransformer***[47]: classe integrante do pacote *Sckit-Learn*, que permite a normalização e a vetorização dos conjuntos de termos através do método *fit_transform*().

Através dos recursos supracitados, é possível obter uma matriz esparsa no formato necessário para o próximo passo: a classificação.

3.3. Classificação

Para a classificação dos documentos, optou-se, inicialmente, por utilizar o classificador *Multinomial Naive Bayes*, cuja implementação é nativa do *Sckit-learn* através da classe *sklearn.naive_bayes.MultinomialNB*.

Contudo, após uma pesquisa mais aprofundada, decidiu-se utilizar o classificador *Xgboost*[48], pois, nos últimos anos, este classificador tem se destacado como o "vencedor" em diversas competições de *Machine Learning*[49].

---

[44] BARNETT, Matthew. *Regex*. Disponível em: https://pypi.org/project/regex/. Acesso em: 28 fev. 2021.

[45] TRESOLDI, Tiago. *Natural Language Toolkit: RSLP Stemmer*. Disponível em: https://ntlk.com. Acesso em: 28 fev. 2021.

[46] ORENGO, Viviane Moreira; HUYCK, Christian R. *A Stemming Algorithm for the Portuguese Language*. In: Eighth International Symposium on String Processing and Information Retrieval. 2001. p. 186-193

[47] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html.

[48] CHEN, Tianqi; GUESTRIN, Carlos. *Xgboost: A scalable tree boosting system*. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016, p. 785-794.

[49] NIELSEN, Didrik. *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?*. Dissertação de Mestrado. NTNU, 2016.

Ao utilizar a classe *XGBClassifier* do *Xgboost*, é possível efetuar a classificação através do método *fit(X,y)* onde *X* é a matriz esparsa, oriunda do pré-processamento, e *y* é a lista contendo as respectivas classes C de cada documento (no contexto deste trabalho, as classes são os nomes das Especializadas de contencioso).

O resultado desta classificação é um objeto dessa mesma classe (*XGBClassifier*), porém, neste momento, hábil para executar o passo seguinte: a predição.

3.4. Predição

A predição é o processo de determinar uma das classes do modelo treinado para uma nova entrada. No contexto deste trabalho, as entradas são números de processos judiciais pois, a partir deles, o L.E.V.I. se encarrega de recuperar o eventual número do originário da ação, o texto da petição inicial (texto livre ou por reconhecimento ótico) e, por fim, transforma o texto em uma matriz esparsa X através das etapas do pré-processamento.

Mediante a recuperação e transformação do texto da petição inicial, é utilizado o método *predict(X)* do classificador, onde o resultado será uma das 8 (oito) Especializadas do contencioso.

Para recuperação dos números dos processos judiciais, alvitrou-se por uma estratégia de utilizar o método "*consultarAvisosPendentes*" do MNI, onde para cada ação, é verificado se a mesma possui acompanhamento prévio ou não. Em caso de ausência de acompanhamento prévio, o L.E.V.I. executa a predição para tentar identificar a Especializada competente.

E o resultado de cada predição é armazenado em uma tabela do banco de dados intitulada de *predicao_especializada_machine_learning*. Essa tabela é composta de campos que identificam o processo judicial, a Especializada predita, a data e hora da predição e a serialização de informações semiestruturadas denominada "*resultado_json*" que permitem o detalhamento do contexto da predição como, por exemplo, a indicação da necessidade da utilização de OCR na petição inicial ou não.

Através da informação persistida nesta tabela, foi possível também criar um componente no PGE Digital, permitindo que processos administrativos pudessem ser formados automaticamente para Especializadas migradas através da resposta da predição do L.E.V.I.

3.4.1. Avaliação da predição

Para avaliar a precisão do L.E.V.I., analisou-se a quantidade de processos administrativos formados automaticamente pelo PGE Digital no período de 01 de janeiro de 2020 a 26 de fevereiro de 2020, cujas predições se encontravam disponíveis na tabela "*predicao_especializada_machine_learning"* no momento anterior à criação dos processos administrativos. O resultado encontrado foi um total de 5.130 (cinco mil cento e trinta) processos administrativos, onde 4.954 (quatro mil novecentos e cinquenta e quatro) mantiveram-se na Especializada predita pelo L.E.V.I. e apenas 176 (cento e setenta e seis) foram redistribuídos, resultando em um total de 96,56% de precisão.

4. Envio das comunicações através de e-mail

Conforme descrito anteriormente, a estratégia para o envio das comunicações judiciais adotada pelo gabinete da PGE-RJ foi a utilização de e-mail. Essa estratégia serve como alternativa para os usuários das Especializadas que ainda não foram migradas para o sistema PGE Digital, pois permite que eles sejam capazes de receber as comunicações judiciais através de suas respectivas contas de e-mail. Tal recurso não é necessário no PGE Digital, pois o mesmo possui uma funcionalidade denominada de "caixa de entrada" que permite ao usuário acesso instantâneo a todas as comunicações endereçadas a ele.

Para implementar o envio de e-mail, foi utilizado a classe *E-mail* do *framework CakePHP*[50] que, através do método *Email::send(),* permite enviar e-mails através de um servidor SMTP[51]. No contexto da PGE-RJ, o servidor de e-mail utilizado é o *Exchange*[52], da Microsoft.

Ao realizar envio de e-mails, é mister informar um destinatário da mensagem, por isso, no contexto de trabalho, utilizou-se três abordagens distintas de definição de destinatário,

[50]CAKEPHP. *Email*. Versão 3.6. [S.I.]: Cake Software Foundation, 2021. Disponível em: https://book.cakephp.org/3/en/core-libraries/email.html. Acesso em: 28 fev. 2021.
[51] Sigla para *Simple Mail Transfer Protocol*.
[52]MICROSOFT. Exchange. Disponível em: https://www.microsoft.com/pt-br/microsoft-365/exchange/email. Acesso em: 28 fev. 2021.

de acordo com situações também diferentes. Destarte, seguem adiante as situações identificadas juntamente com as opções de destinatário que para estas foram escolhidas:

I. **Novas ações oriundas da capital:** processos cuja a localidade informada no MNI é o município do Rio de Janeiro. Utiliza-se a identificação da Especializada de competência através do L.E.V.I, para que a comunicação seja enviada para o e-mail identificado como o da triagem da Especializada. Quando não houver predição disponível, o destinatário será a Central de Mandados;
II. **Novas ações judiciais oriundas de outros municípios:** identifica uma das unidades regionais da PGE-RJ através de um mapeamento persistido em uma tabela chamada *mapeamento_localidade_regional*, que correlaciona os municípios do Estado do Rio de Janeiro a estas unidades, para que o e-mail de uma dessas unidades seja definido como destinatário;
III. **Processos com prévio acompanhamento:** especificamente para processos físicos, cujos registros de movimentação, distribuição e vinculação ainda se encontram no sistema legado Sicaj.

Quando um processo possui prévio acompanhamento pelo Sicaj, faz-se necessário identificar o "destinatário inicial das comunicações" do acervo do processo, cujo endereço de e-mail será utilizado para comunicação recebida via MNI. A metodologia utilizada para essa identificação é o tema da próxima seção.

4.1. Identificação do destinatário inicial das comunicações

Conforme descrito anteriormente, a PGE-RJ utiliza o conceito de acervos para representar um conjunto de processos administrativos. Esse acervo, por seu turno, precisa de um procurador titular, onde este será o responsável pelos processos do acervo.

No sistema Sicaj, não existe explicitamente o conceito de acervo, apenas uma associação individual de processo e usuário denominada de "vinculação", porém, infelizmente, muitos destes usuários (tabela "*servidor"*) não correspondem de fato ao nome de um Procurador da PGE-RJ, o que inviabilizaria a identificação do destinatário das comunicações.

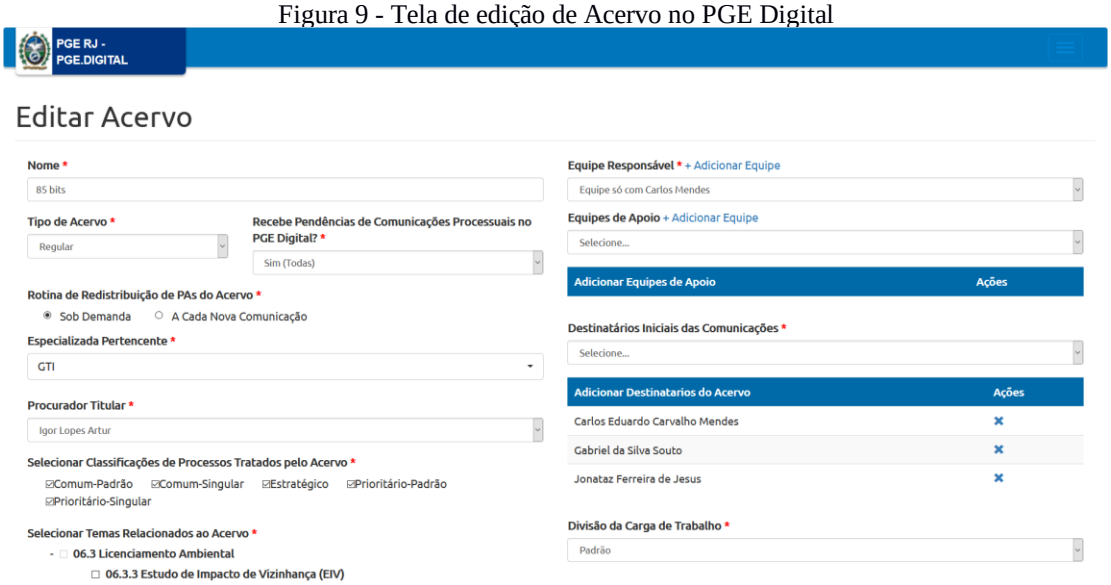
Mediante esses fatos, optou-se, então, por implementar um mapeamento entre os usuários do Sicaj e os usuários da rede "AD" (*Active Directory)*. Esse mapeamento foi implementado através da criação de uma tabela denominada de "*mapeamento_acervo_sicaj*" que contém o nome na tabela *servidor* do Sicaj e o *login* do usuário Procurador correspondente na rede AD.

Entretanto, o mapeamento não foi o suficiente para emular a distribuição dos processos físicos em acervos de massa que possuem a presença dos Técnicos de Suporte Processual, profissionais da área do Direito que auxiliam os Procuradores em acervos com um tamanho considerável de processos.

Para sanar essa situação, disponibilizou-se aos usuários das Especializadas ainda não migradas a permissão para utilizar o cadastro de acervos do PGE Digital. Esse cadastro permite definir um ou mais "destinatários iniciais das comunicações" para que, através de uma estrutura de dados chamada "fila circular", as comunicações possam ser enviadas para os e-mails dos destinatários em uma ordem circular (exemplo: d1 -> d2 -> d3 -> d1 -> d2 etc.).

A imagem a seguir ilustra a edição de um acervo fictício chamado "85 bits" com três destinatários iniciais das comunicações:

Figura 9 - Tela de edição de Acervo no PGE Digital



Fonte: O autor, fevereiro de 2021.

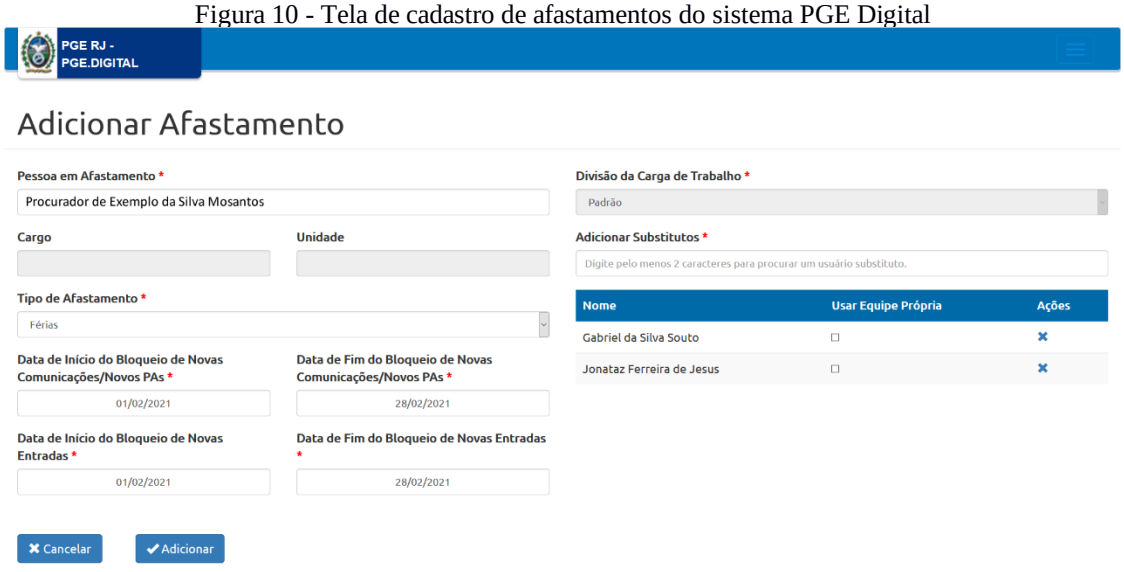
Outrossim, os Procuradores responsáveis pelos acervos podem estar em período de afastamento, o que poderá acarretar o envio das comunicações para os seus substitutos.

4.1.1. Afastamento do Procurador

Os Procuradores do Estado do Rio de Janeiro podem gozar de afastamentos por diversos motivos, desde férias até problemas de saúde. Quando um Procurador se encontra em afastamento, dois procuradores podem, durante este período, dividir o recebimento de novas comunicações nas quais seus processos judiciais são acompanhados por processos administrativos do acervo do Procurador afastado.

Para agilizar o cadastramento dos afastamentos e evitar uma centralização desta atividade no setor de Tecnologia da Informação da PGE-RJ, optou-se por, novamente, disponibilizar aos usuários das Especializadas ainda não migradas a permissão para utilizar outro recurso do PGE Digital, o cadastro de afastamentos. Através desse recurso, o usuário é capaz de informar o período de bloqueio de novas comunicações e os substitutos durante este período. A imagem a seguir demonstra um exemplo de utilização desta tela do sistema, onde os dados preenchidos informam um afastamento de um Procurador no período compreendido entre 01 de fevereiro de 2021 a 28 de fevereiro de 2021 por motivo de férias, bem como a definição de dois substitutos:

Figura 10 - Tela de cadastro de afastamentos do sistema PGE Digital



Fonte: O autor, fevereiro de 2021.

Com o destinatário definido, a comunicação é enviada por e-mail em forma de anexo, além de um conjunto de informações extraídas do próprio Sicaj para facilitar a

contextualização do usuário sobre o estado do processo administrativo e também do processo judicial.

O disparo destes e-mails é efetuado de segunda-feira a sexta-feira, normalmente no período da manhã pela equipe de Operações da PGE-RJ.

Na próxima seção, serão apresentados alguns resultados sobre a utilização do envio de comunicações por e-mail até o fechamento deste trabalho.

4.2. Resultado dos envios de comunicações por e-mail

No período entre 20 de abril de 2020 a 26 de fevereiro de 2021, 385.140 (trezentos e oitenta e cinco mil cento e quarenta) e-mails, contendo as comunicações judiciais recebidas via MNI, foram enviados para mais de 300 destinatários distintos.

Considerando o cálculo de aproximadamente de uma árvore de eucalipto para cada 20 mil folhas de papel A4 (75 g/m2), e uma média de uma página por comunicação, é possível estimar um total de 19.257 árvores poupadas[53].

5. Conclusão

Diante do exposto, pode-se tanto constatar os benefícios da utilização da integração MNI para recuperação dos dados e documentos de processos judiciais, quantos os do uso de *Machine Learning* para classificação de novos processos.

Ademais, o envio das comunicações por e-mail permitiu amenizar os impactos causados pela pandemia da COVID-19, possibilitando que as triagens das Especializadas e os responsáveis administrativos dos processos pudessem receber as comunicações de forma digital e sem riscos de exposição ao vírus Sars-CoV-2.

Em um mundo de incertezas e temores, a tecnologia deve ser compreendida como uma poderosa ferramenta de consolidação das atividades da advocacia pública, essenciais para o pleno exercício da sociedade. O Exmo. Ministro do STF Dias Toffoli, no prefácio do livro

[53] REVISTA GALILEU. *Quantas árvores de papel dá pra fazer com uma árvore?* 2009. Disponível em: http://revistagalileu.globo.com/Revista/Common/0,,EMI110264-17775,00.html. Acesso em: 28 fev. 2021.

"Tecnologia e Justiça Multiportas: Teoria e prática" disserta sobre a importância da tecnologia nas atividades jurídicas onde afirma que:

> O cenário futuro mostra-se cada vez mais desafiador, tendo em vista a existência de diversas tecnologias consideradas disruptivas: Inteligência Artificial, Computação em Nuvem, Big Data, Internet das Coisas (IoT), Blockchain, Internet 5G, Smart Contracts, dentre várias outras. Embora desafiadoras, essas ferramentas têm se mostrado importantes instrumentos de inovação e aprimoramento das atividades jurídicas, cujas tecnologias se tornam, sucessivamente, mais sofisticadas e consolidadas[54].

Este trabalho é resultado da atuação vanguardista e inovadora da Procuradoria Geral do Estado do Rio de Janeiro que, mais uma vez, assume um papel de pioneirismo no âmbito da advocacia pública brasileira.

Referências

AGGARWAL, Charu. *Machine learning for text*. Cham: Springer International Publishing, 2018.

ALPAYDIN, Ethem. *Introduction to machine learning*. 2 ed. Londres: The MIT Press, 2010, p. 32.

BARNETT, Matthew. *Regex*. Disponível em: https://pypi.org/project/regex/. Acesso em: 28 fev. 2021.

BASARKAR, Ankit. *Document classification using machine learning*. Master's Projects San Jose State University. 2017, p. 4.

BIRD, Steven; KLEIN, Ewan; LOPER, Edward. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, 2009.

CAKEPHP. *Email*. Versão 3.6. [S.I.]: Cake Software Foundation, 2021. Disponível em: https://book.cakephp.org/3/en/core-libraries/email.html. Acesso em: 28 fev. 2021.

[54] FUX, Luiz; ÁVILA, Henrique; CABRAL, Trícia. *Tecnologia e Justiça Multiportas: Teoria e prática*. Editora Foco, 2021.

CHEN, Tianqi; GUESTRIN, Carlos. *Xgboost: A scalable tree boosting system*. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, p. 785-794.

CONSELHO NACIONAL DE JUSTIÇA. *Termo de Acordo de Cooperação Técnica nº58/2009*. Presidente do Supremo Tribunal Federal e do Conselho Nacional de Justiça Gilmar Mendes.

_____. *Resolução nº 65 de 16/12/2008*. Ministro Gilmar Mendes.

EL NAQA, Issam; MURPHY, Martin J. *What is machine learning?*. In: Machine learning in radiation oncology. Springer, Cham, 2015. p. 3-11.

ESSEX, University of. Corpus Linguistics. *W3C Corpus Linguistics Pages*. 1998. Disponível em:https://www1.essex.ac.uk/linguistics/external/clmt/w3c/corpus_ling/content/introduction2.html. Acesso em: 1 fev. 2020.

FUX, Luiz; ÁVILA, Henrique; CABRAL, Xavier (Ed.). *Tecnologia e Justiça Multiportas: Teoria e prática*. Editora Foco, 2021.

GAMMA, Erich *et al*. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.

GHEBREYESUS, Tedros. *WHO Director-General's opening remarks at the media briefing on COVID-19*. WHO, 11 de março de 2020. Disponível em: https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020. Acesso em: 20 jan. 2021.

GLYPH & COG, *XpdfReader*. 2021. Disponível em: https://www.xpdfreader.com/. Acesso em: 07 mar. 2021.

HINKEL, Angélica. *Anexo II do Manual do Modelo Nacional de Interoperabilidade*. Rio de Janeiro: TJRJ, 2016. Disponível em:http://portaltj.tjrj.jus.br/documents/10136/3067978/anexo-ii-manual-modelo-nacional-interoperabilidade.pdf. Acesso em: 28 fev. 2021.

IKONOMAKIS, M.; KOTSIANTIS, Sotiris; TAMPAKAS, V. *Text classification using machine learning techniques*. In: WSEAS transactions on computers, v. 4, n. 8, p. 966-974, 2005.

IMAGEMAGICK DEVELOPMENT TEAM. 2021. ImageMagick. Disponível em: https://imagemagick.org. Acesso em: 07 mar. 2021.

KOTELUK, Oliwia *et al*. *How Do Machines Learn? Artificial Intelligence as a New Era in Medicine*. Journal of Personalized Medicine, v. 11, n. 1, 2021, p. 3.

LOVINS, Julie Beth. *Development of a stemming algorithm*. In: Mechanical Translation and Computational Linguistics, v. 11, n. 1-2, 1968 p. 22-31.

MICROSOFT. *Exchange*. Disponível em: https://www.microsoft.com/pt-br/microsoft-365/exchange/email. Acesso em: 28 fev. 2021.

_____. *SQL Server*. Disponível em: https://www.microsoft.com/pt-br/sql-server/sql-server-downloads. Acesso em: 28 fev. 2021.

MÜLLER, Andreas C.; GUIDO, Sarah. *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016.

NIELSEN, Didrik. *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?*. Dissertação de Mestrado. NTNU. 2016.

ORENGO, Viviane Moreira; HUYCK, Christian R. *A Stemming Algorithm for the Portuguese Language*. In: Eighth International Symposium on String Processing and Information Retrieval. 2001. p. 186-193.

PEDREGOSA, Fabian *et al*. *Scikit-learn: Machine learning in Python*. In: The Journal of machine Learning research, v. 12, 2001, p. 2825-2830.

_____, Fabian *et al*. *Working With Text Data*. 2021. Disponível em: https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html. Acesso em: 28 fev. 2021.

PGE-RJ. *Estrutura*. 2021. Disponível em: https://pge.rj.gov.br/institucional/estrutura. Acesso em: 1 mar. 2021.

RAULJI, Jaideepsinh; SAINI, Jatinderkumar. *Stop-word removal algorithm and its implementation for Sanskrit language*. In: International Journal of Computer Applications, v. 150, n. 2, 2016, p. 15.

REVISTA GALILEU. *Quantas árvores de papel dá pra fazer com uma árvore?* 2009. Disponível em: http://revistagalileu.globo.com/Revista/Common/0,,EMI110264-17775,00.html. Acesso em: 28 fev. 2021.

RIO DE JANEIRO. Procuradoria Geral do Estado. Resolução PGE nº 4.527 de 16 mar. 2020. [Institui Medidas de Prevenção ao contágio da Covid-19, e dá outras providências]. *Diário Oficial do Estado do Rio de Janeiro*: parte 1: Poder Executivo, Rio de Janeiro, ano 46, n. 049, p. 37, 16 mar. 2020.

_____. Tribunal de Justiça do Rio de Janeiro. Recurso Extraordinário 0411983-35.2014.8.19.0001. Comarca da Capital. 2014, p. 3.

SEBASTIANI, Fabrizio. *Machine learning in automated text categorization*. In: ACM computing surveys (CSUR), v. 34, n. 1, p. 1-47, 2002.

SMARTBEAR. *SoapUI Open Source*. Versão 5.5.0. [S.I.]: SmartBear Software, 2021. Disponível em https://www.soapui.org. Acessado 28 fev. 2021.

SMITH, Ray. *An overview of the Tesseract OCR engine*. In: Ninth international conference on document analysis and recognition (ICDAR 2007). IEEE, 2007. p. 629-633.

SRIVIDHYA, V.; ANITHA, R. *Evaluating preprocessing techniques in text categorization*. In: International journal of computer science and application, v. 47, n. 11, p. 49-51, 2010.

TRESOLDI, Tiago. *Natural Language Toolkit: RSLP Stemmer*. Disponível em: https://ntlk.com. Acesso em: 28 fev. 2021.

ZEND. *PHP*. Versão 7.0. [S.I.]: Zend. Disponível em: https://www.php.net/. Acesso em: 28 fev. de 2021.

_____. *SOAP CLIENT PHP*. Disponível em: https://www.php.net/manual/en/class.soapclient.php. Acesso em: 28 fev. 2021.

Recebido em: 13/03/2021.
Aprovado pela coordenadoria editorial.