REVISTA ESMAT 18

RESUMO AUTOMÁTICO DE TEXTOS JURÍDICOS USANDO GRAFOS COM VOCABULÁRIO CONTROLADO E ALGORITMO K-MEANS COM WORDS EMBEDDING

AUTOMATIC SUMMARY OF LEGAL TEXTS USING GRAPHS WITH CONTROLLED VOCABULARY AND K-MEANS ALGORITHM WITH WORDS EMBEDDING

Rogério Nogueira de Sousa

Analista de Tecnologia da Informação. Mestrando do Programa de Mestrado em Modelagem Computacional de Sistemas, pela Universidade Federal do Tocantins. roger@uft.edu.br

David Nadler Prata

Professor Programa de Pós-Graduação em Modelagem Computacional de Sistemas, pela Universidade Federal do Tocantins. ddnp@uft.edu.br.

RESUMO

O processo judicial eletrônico é uma realidade no Brasil, onde 70% dos casos novos em todo Poder Judiciário são virtuais. Fazer uso adequado dessa realidade e aprimorá-la é primordial para dar vazão à demanda de aproximadamente 25 milhões de processos novos por ano. Nesse contexto, buscar facilitar o dia a dia dos operadores da justiça brasileira, responsáveis pelas análises do crescente volume de informação digital presentes nos autos, é primordial para a eficiência na prestação jurisdicional e amplo acesso à justiça. Usar resumos/textos é uma forma ágil de se inteirar sobre o assunto do texto, podendo ser um meio de empregar mais agilidade à tramitação processual. Assim, esta pesquisa visa ao desenvolvimento de uma ferramenta capaz de gerar automaticamente resumos de textos jurídicos, fazendo uso de técnicas de Processamento de Linguagem Natural e teoria de grafos em conjunto com o vocabulário jurídico advindo do Supremo Tribunal Federal.

PALAVRAS-CHAVE: Processamento de Linguagem Natural. Sumarização. Grafos. Prestação Jurisdicional. Acesso à Justiça.

ABSTRACT

Today, the electronic process of the Judiciary Branchhas 70% of all new cases in virtual form. It is paramount to this reality an improvement in celerity to provide output to a growth demand of approximately 25 million of new cases per year. In this context, the seeking to facilitate the day-to-day operations of the Brazilian justice system is critical for efficiency in judicial provision and wide access to justice. The use of text summaries is a promissory way to speed finding out about the subject of documents, contributing to the celerity of law suit procedures. The purpose of this research is to develop a hybrid methodology capable of automatically generating summaries of

legal documents, making use of techniques of Natural Language Processing and graph theory with words embedding, in conjunction with the Legal vocabulary coming from the Federal Supreme Court.

KEYWORDS: Natural Language Processing. Summarization. Graph Theory. Legal Assistance. Access to Justice.

1 INTRODUÇÃO

Em 2016, o Poder Judiciário gastou R$ 2.248.734.431,00 (dois bilhões duzentos e quarenta e oito milhões setecentos e trinta e quatro mil quatrocentos e trinta e um reais) com Tecnologia da Informação (TI) e contava com uma força de trabalho composta por 442.345 colaboradores, divididos entre magistrados, servidores e auxiliares, para operar os 79,7 milhões de processos que estavam pendentes naquele ano na justiça brasileira (CONSELHO NACIONAL DE JUSTIÇA, 2017). Em 2017, o gasto com Tecnologia da Informação reduziu para R$ 2.207.995.675,00 (dois bilhões duzentos e sete milhões novecentos e noventa e cinco mil quatrocentos e trinta e um reais) e o número de processos em tramitação passaram os 80 milhões, com praticamente a mesma quantidade de colaboradores de 2016 (CONSELHO NACIONAL DE JUSTIÇA, 2017).

Diante desse cenário, com números expressivos que nos apresentam uma situação preocupante, de aumento de demanda judicial e escassez de recursos, as buscas por soluções cada vez mais eficientes, que possam maximizar a capacidade de trabalho dos colaboradores, bem como reduzir custos por processos, tornam-se imperiosas para a prestação jurisdicional no Brasil.

A Tecnologia da Informação é então acessada como uma das formas de imprimir mais celeridade às atividades judiciais, com menos dispêndio de tempo dos profissionais envolvidos e, via de consequência, com mais economia de recursos (FELIPE; PERROTA, 2018). O dever de eficiência implica a exigência de que a Administração Pública incorpore os progressos tecnológicos em sua atividade (JUSTEN FILHO, 2016). A justiça brasileira tem total consciência da importância da Tecnologia da Informação para a prestação jurisdicional, tanto que destina em torno de 25% do seu orçamento (retirando gasto com pessoal) à informática (CONSELHO NACIONAL DE JUSTIÇA, 2017). Entre as soluções tecnológicas voltadas ao sistema de justiça, destacamos o uso de processos judiciais eletrônicos, uma vez que 70% dos novos processos judiciais são eletrônicos.

Alguns tribunais brasileiros se destacam por possuírem 100% de processos eletrônicos nos dois graus de jurisdição, entre eles (CONSELHO NACIONAL DE JUSTIÇA, 2017) o Tribunal de Justiça do Tocantins, que, na vanguarda do processo judicial eletrônico, implantou o e-Proc/TJTO, em 2011. Neste mesmo ano, 100% dos casos novos passaram a ser virtuais. Após 4 anos, todos os processos em tramitação foram digitalizados, tonando-se, em 2015, o primeiro Tribunal a ter todo o acervo de processos judiciais em formato digital (TJTO, 2015).

A digitalização de dados jurídicos constitui uma mega tendência, transformando fluxos de trabalho e modelos de negócios. O volume de dados utilizados no aconselhamento jurídico aumentou exponencialmente (VEITH et al., 2016), gerando maior demanda por seleção, análise e interpretação de uma quantidade de dados sem precedentes. Em contrapartida, tal virtualização facilita o processo de automação, permitindo o crescimento da produtividade e ainda reduzindo custos, ampliando a qualidade e minimizando o tempo de inatividade dos operadores do direito.

Estamos vivendo uma nova era de automação, na qual robôs e computadores podem não apenas executar uma série de atividades de trabalho físico de rotina de forma mais eficiente e barata que os humanos, mas também cada vez mais capazes de realizar atividades que abrangem capacidades cognitivas (MCKINSEY GLOBAL INSTITUTE, 2017). Com os recentes desenvolvimentos em robótica, inteligência artificial e aprendizado de máquina, as tecnologias não apenas realizam atividades que pensávamos que apenas humanos pudessem fazer, como também podem fazê-las cada vez mais em níveis sobre-humanos de desempenho (MCKINSEY GLOBAL INSTITUTE, 2017).

Atualmente, as fases processuais que requerem análises por parte dos operadores do direito são as mais demoradas, pois, com a virtualização dos processos, não há mais gargalos nos atos de autuar e tramitar processos. Para o analista, os textos contidos nos autos são suas principais ferramentas de trabalho. Não raramente o analista se vê obrigado a ler todo o conteúdo de uma peça, apenas para saber do que se trata. Um resumo do conteúdo lhe promoveria mais eficiência, pois facilita e acelera a extração de informações relevantes contidas dentro de documentos dos processos. Um dos desafios em trabalhar com textos no campo legal é a complexidade do domínio, pois a terminologia específica e interpretações legais ocasionam muitas ambiguidades (FARZINDAR; LAPALME, 2004).

Nesse contexto, o desenvolvimento de uma ferramenta capaz de gerar automaticamente resumos extraídos de peças judiciais em formato de texto tem o potencial de impactar diretamente na agilidade dos colaboradores do judiciário responsáveis pela análise dos autos. Tal aumento da produtividade e agilidade nas análises documentais servirá para melhorar a efetividade nas atividades jurídicas diárias, o que impactará significativamente para o alcance das metas jurisdicionais estabelecidas pelo Conselho Nacional de Justiça a todo o Poder Judiciário atualmente.

Em consequência, teremos o aprimoramento do acesso à justiça, pela presteza na concessão da tutela jurisdicional pelo Estado, que tem como objetivo a chamada "ordem jurídica justa", que, além do acesso formal ao órgão jurisdicional, fornece os meios para que o conflito de interesses ocorrido seja solucionado de forma satisfatória. Com isso, pretendemos contribuir para superar entraves econômicos ou jurídicos que possam surgir para a efetiva prestação da justiça (CINTRA; DINAMARCO; GRINOVER, 2010).

O ato de resumir o texto de forma automatizada é também conhecido como Sumarização Automática (SA); no caso, foram aplicadas duas técnicas advindas do

modelo extrativo de sumarização, uma usando grafos com vocabulário controlado, e outra com algoritmo k-means com wordsembedding. Com isso, nas técnicas aplicadas, teremos resumos compostos por sentenças extraídas do corpo do texto, selecionados com base na sua relevância.

O objetivo geral do presente trabalho é apresentar o desenvolvimento de uma solução capaz de automatizar a geração de resumos de textos jurídicos, por meio do levantamento e extração de sentenças que apresentem mais relevância para identificação da ideia central do texto, agilizando, assim, a contextualização dos operadores do direito, de modo a otimizar as tomadas de decisões necessárias no decorrer do processo.

Os objetivos específicos são sintetizar textos jurídicos; usar teoria de grafos e algoritmo k-means para aferir a relevância das sentenças contidas no texto; agilizar a análise de processos judiciais; extrair o vocabulário jurídico do portal do Supremo Tribunal Federal (STF); aferir os ganhos do vocabulário jurídico.

2 DESENVOLVIMENTO

O desenvolvimento da solução foi dividido basicamente em: a) extração do vocabulário jurídico; b) processamento do texto; c) geração do grafo que representa as sentenças do texto e clusterização do texto usando o algoritmo K-means; e d) exposição e aferição das sentenças mais relevantes.

Foi utilizado um vocabulário de termos jurídicos para auxiliar no levantamento das sentenças mais relevantes em textos contidos em processos judiciais. Para sua formação, optamos por utilizar os termos jurídicos presentes no tesauro do Supremo Tribunal Federal, que tem a função de padronizar a informação, sendo um mecanismo de controle terminológico (STF, 2019).

Figura 1. Vocabulário Jurídico Supremo Tribunal Federal

Fonte: ("Vocabulário Jurídico (Tesauro) : Supremo Tribunal Federal (STF)")

Para extração das informações contidas no referido tesauro, foi desenvolvido um programa que navega na *web* de maneira automática, copiando dados das páginas visitadas, conhecido como *web crawler*. Este foi configurado para extração dos termos contidos no portal do Supremo Tribunal Federal, resultando na formação de uma lista com 15.434 termos jurídicos.

O programa fora implementado em Python fazendo uso principalmente dos módulos *Beautiful Soup* na versão 4, para análise e extração do conteúdo das páginas HTML (RICHARDSON, 2015), e o Selenium, na versão 3.14, para navegação entre as páginas do vocabulário.

A linguagem Python, na versão 3.7, foi escolhida por ser de alto nível, orientada a objetos, capaz de ser utilizada em diversas plataformas pelo fato de ser interpretada (PYTHON.ORG, 2019), e vem se mostrando uma boa escolha pela velocidade de desenvolvimento e manutenção; está se estabelecendo como uma das linguagens mais populares da computação científica (PEDREGOSA et al., 2011).

2.1 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural, conhecido também na academia como Linguística Computacional, vem crescendo rapidamente, pois suas teorias e métodos estão sendo aplicados em uma gama de novas tecnologias (BIRD; KLEIN; LOPER, 2009). Essa área de estudo objetiva fornecer ferramentas para que um sistema computacional seja capaz de lidar com linguagens naturais em diversos níveis, como morfológico, sintático e semântico (COPPIN, 2017). Para construção de rotinas que implementam métodos de Processamento de Linguagem Natural, utilizamos a biblioteca Natural Language Toolkit (NLTK), (NLTK PROJECT, 2019), inicialmente projetada para o ensino; na atualidade, é adotado pelo mercado devido à sua usabilidade e abrangência (PERKINS, 2010).

Os textos a serem resumidos passaram por um conjunto de técnicas de Processamento de Linguagem Natural para processá-los previamente com intuito de abstraí-los para facilitar a compreensão computacional.

De início o texto será segmentado em sentenças; estas por sua vez serão tokenizadas, ação onde sentenças passarão por um processo de separação das palavras em termos, também conhecidos como tokens, gerando uma lista de termos por sentença formada a partir de cada sentença. A simples tarefa de criação de lista de palavras advindas de uma sentença é uma parte essencial para todo o processamento de texto (PERKINS, 2010).

O vocabulário representa o conjunto de palavras (tokens) que será usado no processamento do texto (LANE; HOWARD; HAPKE, 2017). Logo, o tamanho do vocabulário implica diretamente a complexidade computacional e a memória requerida para o devido processamento. O uso de técnicas que reduzam o vocabulário é imprescindível para o ganho de performance e pode proferir mais generalidade ao processamento.

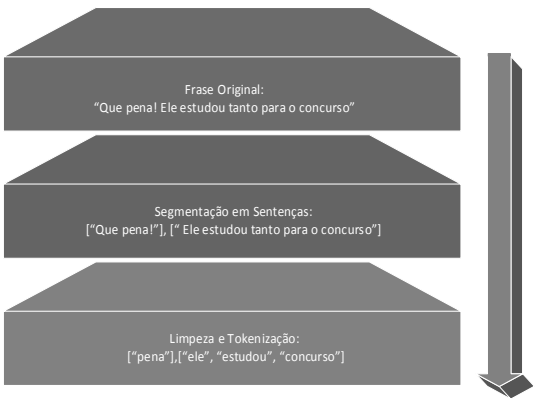
Tais técnicas buscam transformar diversas palavras com significados semelhantes em uma só. Uma dessas técnicas trata-se da conversão de todas as letras do texto para minúsculas. Por ser muito comum, palavras iniciadas com letra maiúscula terão o mesmo significado com a letra inicial minúscula. Mas em alguns casos o significado muda, como, por exemplo, as palavras 'gentil' e 'Gentil', a primeira é usada como adjetivo; a segunda como substantivo, no caso nome próprio. Assim, o uso da técnica de conversão do texto em minúscula deve ser avaliado de acordo com o propósito do processamento, não sendo recomendado quando se almeja detectar no texto entidades nomeadas, como nomes próprios.

No sentido de redução de processamento computacional, são retirados os caracteres especiais das palavras contidas nestes, como arroba, barras e outros símbolos. As acentuações são retiradas para evitar que erros de grafia impactem na interpretação das palavras a serem transformadas em tokens. Também são desconsiderados os números e as pontuações.

Algumas palavras comuns ocorrem com muita frequência em qualquer idioma, mas apresentam baixa relevância para expressar o significado da frase, essas são chamadas de stopwords (LANE; HOWARD; HAPKE, 2017). Geralmente artigos, conjunções, preposições, interjeições, verbos auxiliares e palavras muito repetidas na linguagem natural compõem esse grupo. Tais palavras são retiradas dos textos após a tokenização, com vista a reduzir o esforço computacional, quando se quer extrair informações de um texto.

Cabe salientar que em alguns casos, como processamento de textos curtos, a retirada dos stopwords pode levar à perda de informações relevantes para o significado do texto, situação que não ocorre no estudo em questão, haja vista a natureza dos textos trabalhados (jurídicos) ser normalmente extensa. Logo, a retirada dos stopwords não ocasiona significativos prejuízos ao valor semântico do texto, uma vez que são necessários à linguagem natural pelo seu valor sintático.

Figura 2. Sequência de pré-processamento do texto.



Para dar mais generalidade aos termos, é efetuado um processamento em cada um deles, onde cadeias morfologicamente complexas são identificadas, decompostas em radical e afixos, sendo descartados os afixos, e o termo passa a ser apenas o radical, processo conhecido como *stemming* (LANE; HOWARD; HAPKE, 2017).

Quando se adota a técnica de *stemming* para formação do token, removendo o sufixo e prefixo, temos um termo mais genérico, como, por exemplo, as palavras 'livro', 'livrinho', 'livros' e 'livrecos', todas possuem significados semelhantes ou próximos e em comum a cadeia de caracteres 'livr', sendo esta o elemento base para o significado. Logo, podem-se substituir as quatro palavras pelo radical 'livr' que não há perda considerável de significado. Mesmo que 'livr' não seja uma palavra existente não importa, porque o objetivo é casar as palavras em consultas e em documentos, e não as mostrar ao usuário (COPPIN, 2017).

Os termos são dispostos em vetores contidos em um vetor que representa a sentença. Assim, cada documento passa a ser representado por um vetor de sentenças no qual cada item deste é formado por um de termos; logo, os documentos são representados como uma matriz de termos.

Figura 2. Classe ObjTexto

Uma classe chamada objTexto foi criada com intuito de receber as informações do texto transformado em matriz, bem como as informações do texto original. Acionando o método construtor dessa classe, passando como parâmetro o texto a ser resumido, o atributo "texto" é preenchido com o texto original, posteriormente é acionado o método gerarSentenca, que separa as sentenças e as retorna na forma de um vetor que é adicionado ao atributo "sentencas". Os métodos limpaTokeniza e processaTexto retiram os caracteres especiais e as *stopwords*, realizam o *stemming* dos tokens e atribuem a representação do texto ao atributo "textoProcessado".

2.2 Representação de texto em Grafo

Um grafo é formado por dois conjuntos, sendo um de vértices que representam objetos e outro de arestas que correspondem à relação entre os vértices (COPPIN, 2017). Para produzir um grafo que represente um texto, considerou-se que cada sentença do texto é um vértice, e o valor atribuído à similaridade entre os vértices é a aresta desse grafo. A implementação do grafo utilizou o módulo python, networkx, para criação e manipulação.

A similaridade entre as sentenças é calculada pela quantidade de palavras presentes em ambas, as quais também estão contidas no vocabulário de termos jurídicos provido pelo Supremo Tribunal Federal. Quando essa quantidade for zero, considera-se que não há similaridade entre as sentenças; logo, não é criada a aresta entre elas, caso contrário, passa a existir uma ligação entre as sentenças cujo peso é dado pela fórmula:

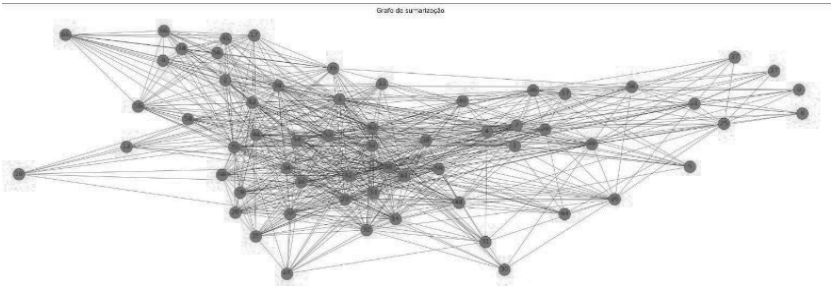
Figura 3. Cálculo de Similaridade

$$S = \frac{Qtd.\ \ palavras\ iguais\ entre\ as\ sentenças}{Log10\ (Tam.\ \ da\ Senteça(1)) + Log10(Tam.\ \ da\ Senteça(2))}$$

Para que sentenças muito grandes não sejam privilegiadas apenas por serem mais extensas que as demais, é feita a divisão da quantidade das palavras iguais nas sentenças pela soma dos logaritmos tamanhos das sentenças.

O método aqui utilizado se baseia no algoritmo TextRank (MIHALCEA; TARAU, 2004), desenvolvido inspirado no PageRank (BRIN; PAGE, 1998), ambos implementados fazendo uso de teoria de grafos e utilizados para ranqueamento com base nas relações entre os vértices (arestas). Enquanto o PageRank é usado pelo buscador da Google para aferir a relevância de páginas *web*, TextRank atribui valor de importância a sentenças dentro de um texto.

Figura 4. Grafo que representa o texto



2.3 Words Embededding

A parte mais importante no processo de clusterização é a métrica aplicada para cálculo de distância entre os elementos (JAIN, 1988); assim, é necessária a transformação do texto numa representação matemática para computar a similaridade entre as sentenças.

Nesse sentido, em 2013, foi desenvolvido o algoritmo word2vec que cria representações vetoriais distribuídas, chamadas *word embedding*, capazes de representar palavras, considerando as relações sintáticas e semânticas (AGUIAR, 2016) de cada palavra em relação ao vocabulário, independentemente do idioma dos textos, dando mais flexibilidade ao processamento dos dados.

Tais vetores são gerados usando redes neurais que fazem uso de uma camada oculta e do algoritmo *backpropagation* para atualizar os pesos dessa camada, quer dizer, gera um vetor por meio de aprendizagem de máquina capaz de capturar propriedades linguísticas indiretamente.

Para esse projeto, utilizou-se o conjunto de representações vetoriais de palavras, pré-treinado, conhecido como Lex2Vec (FONSECA, 2017), gerado a partir de um *corpus* formado por 233.108 normas promulgadas entre os anos 1824 e 2017, oriundas da legislação federal brasileira.

Para gerar os vetores, usou-se a ferramenta word2vec, que implementa dois modelos de representação vetorial. Um gera o vetor com as dimensões pré-definidas, considerando o contexto, buscando informar qual seria a palavra faltante, conhecido como Continuous Bag-of-Word (Cbow), (AGUIAR, 2016), enquanto o Continuous Skip-Gram, por meio de uma palavra, busca informar qual seria o contexto (MIKOLOV et al., 2013).

As representações textuais usando *word embedding* apresentam uma maior gama de informações quando comparadas com representações que fazem uso de contagem de frequência de palavras, sendo verdade até mesmo em comparação com modelos que utilizam parâmetros de compensação para efeitos de frequência (SCHNABEL et al., 2015).

2.4 Clusterização do documento com K-means

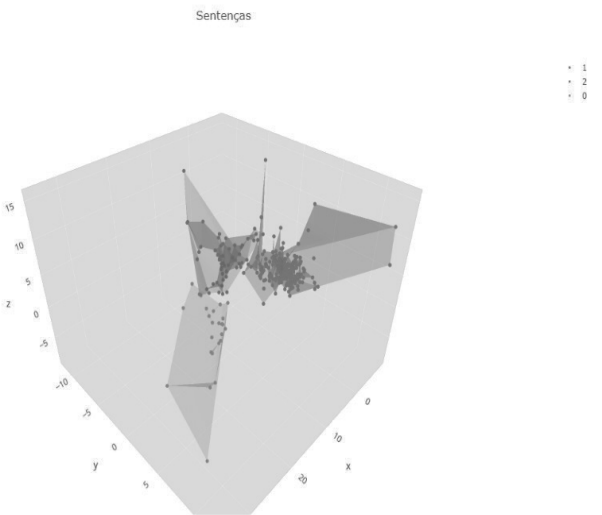
Outra forma de representar e organizar o texto computacionalmente se dá por meio de clusterização, em que as sentenças são agrupadas conforme seu contexto. Para tal, utilizamos neste trabalho o K-means, que consiste em um algoritmo de aprendizagem não supervisionada, interativo, com baixa complexidade computacional (LUO; LI; CHUNG, 2009), em que o número de clusters (agrupamentos) é atribuído de forma arbitrária por meio de um constante. Cada cluster se forma em torno de um centroide, que é reposicionado a cada interação, a fim de se tornar o ponto mais central do cluster.

As sentenças são rotuladas com base na sua relação com os centroides, ou seja, se a sentença é mais próxima de um centroide em relação aos demais, ela passa a

pertencer ao cluster do centroide mais próximo e recebe o rótulo deste cluster, pois os elementos de um cluster tendem a ser similares e diferentes dos não pertencentes ao grupo. O K-means itera até que não haja mais movimentação de elementos entre os clusters ou até que o número máximo de iterações tenha sido atingido (JAIN, 1988).

Para implementar o algoritmo K-means neste trabalho, utilizou-se o módulo python K-Means da biblioteca Scikit-Learn. Ao término das interações é gerada uma matriz com o índice da sentença e o cluster à qual ela foi designada. Como os vetores de representação (words embeddding) dos termos possuem mais de 3 dimensões, é usado o método matemático conhecido como análise de componentes principais para redução a 3 dimensões, possibilitando a visualização gráfica do texto clusterizado.

Figura 5. Visualização 3D das sentenças de um texto em 3 clusters



2.5 Extração do resumo

O conceito de centralidade na teoria dos grafos está associado ao grau de importância de um vértice dentro de um grafo, como, por exemplo, pessoas mais influentes em seu círculo social apresentam maior índice de centralidade (BORBA, 2013). Nesse caso, a centralidade utilizada é a de grau, quer dizer que as sentenças que tiverem mais sentenças relacionadas, serão mais relevantes para o texto.

Nesse contexto, aplicou-se a função "degree_centrality" do módulo *networkx*, que retorna um objeto do tipo dicionário com o rótulo do vértice e o grau de centralização normalizado pela divisão do máximo de possibilidade de um grafo simples de n-1, em que n é o número de vértices desse grafo, de cada vértice (NETWORKX DEVELOPERS, 2014).

Para coletar os vértices com maior grau de centralidade, o objeto é ordenado do maior para o menor, usando como índice o grau de centralidade. A quantidade de sentenças que compõe o resumo pode ser calculada com base em um percentual do texto completo, quer dizer, pode ser usada uma taxa de compressão ou por meio de um valor fixo de número de sentenças.

Como geralmente o resumo equivale de 10 a 20% do texto original (RINO; PARDO, 2003), na solução foi optada pela taxa de compressão de 90%, quer dizer, o resumo possuirá uma quantidade de sentenças equivalente a 10% da quantidade de sentenças do texto original.

Como rótulos dos vértices representam as posições das sentenças no texto original, os vértices de maior centralidade apontam para as sentenças mais relevantes no texto, possibilitando, assim, a extração das sentenças que resumem a informação contida no texto.

No texto devidamente clusterizado com o algoritmo k-means, assim como no grafo, a eleição das sentenças mais relevantes considera a centralidade destas, só que considerando o seu cluster. Logo, as sentenças mais próximas do centroide são as mais relevantes naquele agrupamento.

Para calcular as distâncias entre os vetores centroides e os vetores que representam as sentenças, utilizou-se a distância cosseno (MIKOLOV et al., 2013). Logo, quanto menor o ângulo formado entre os dois vetores, menor é a distância entre ambos e mais similares são.

Figura 6. Distância cosseno

$$Similaridade = cos(A,B) = \frac{A.B}{||A||.||B||} = \frac{\sum_{i=n}^{n} A_i.B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Considerando que a quantidade de sentenças que farão parte do resumo é de 10% da quantidade de sentenças do texto original, é coletada uma sentença de cada cluster, priorizando sua proximidade com o centroide do cluster a que ela está vinculada.

3 CONCLUSÕES

É comum, na seara do direito, haver peças jurídicas extensas, podendo, em alguns momentos, dificultar o trabalho dos analistas jurídicos que atuam na análise de informações para efetivar a prestação da justiça. A linguagem jurídica, por si só, pode

se constituir em barreira para interpretação adequada do intuito perseguido pelo postulante. Portanto, a sumarização automática de textos jurídicos tem como objetivo a transmissão da mensagem central do texto, otimizando, dessa forma, os principais processos de tomadas de decisões que podem surgir no decorrer do processo.

Criar mecanismos que possam agilizar a tramitação processual é imperioso para uma prestação jurisdicional satisfatória, pois automatizar tarefas é o caminho mais rápido e menos custoso para o Poder Judiciário. Ações simples como a geração de um resumo automático podem impactar sensivelmente no dia a dia dos operadores do direito, devido ao imenso volume de informação textual contido nos processos judiciais.

Um aspecto importante foi a utilização do vocabulário de termos jurídicos extraído do *site* do Supremo Tribunal Federal, que promoveu mais assertividade à técnica que faz uso de grafos, ao garantir que se utilizem apenas termos jurídicos durante a análise de similaridade das sentenças, embora reduzindo a capacidade de gerar resumos mais precisos de textos não jurídicos.

O uso de *words embedding* deu mais generalidade ao sistema de sumarização; embora as representações vetoriais dos vocabulários tenham sido geradas a partir de textos oriundos do mundo jurídico, ele é capaz se resumir texto fora desse contexto.

O uso de clusters apresentou diversas vantagens, como, por exemplo, a segmentação do texto com base no contexto, que facilita as análises focadas dentro do documento.

Não raramente temos de analisar textos advindos da extração do conteúdo de documentos formais, com múltiplas páginas, e nos deparamos co m cabeçalhos, com informações não relevantes, que se repetem, sendo necessário desconsiderar tais cabeçalhos para geração do resumo. A clusterização tende a agrupar os cabeçalhos em um único cluster, por terem informação repetida, facilitando, dessa forma, o descarte destes durante a sumarização.

Este trabalho apresentou duas formas tecnicamente viáveis, com um custo de processamento e implementação relativamente baixo. Pois, além de fazer uso de um arcabouço tecnológico aberto, as técnicas de sumarização, implementadas utilizando a teoria de grafos u agrupamento com algoritmo k-means são simples e amplamente disseminadas no mercado, o que facilita consideravelmente a manutenção e o aprimoramento da solução, garantindo mais possibilidade de continuidade e retorno sobre o investimento.

REFERÊNCIAS

AGUIAR, E. M. de. **Aplicação do Word2vec e do Gradiente descendente dstocástico em tradução automática**. 30 maio 2016. Disponível em: <http://bibliotecadigital.fgv.br/dspace/handle/10438/16798>. Acesso em: 9 abr. 2019.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing with Python**. 1. ed. Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472: Julie Steele, 2009.

BORBA, E. M. **Medidas de centralidade em grafos e aplicações em redes de dados**. 2013. Disponível em: <https://lume.ufrgs.br/handle/10183/86094>. Acesso em: 12 fev. 2019.

BRIN, S.; PAGE, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. **Computer Networks and ISDN Systems**, v. 30, n. 1–7, p. 107–117, abr. 1998.

CINTRA, A. C. de A.; DINAMARCO, C. R.; GRINOVER, A. P. **Teoria Geral do Processo**. 26a ed. São Paulo: Malheiros, 2010.

CONSELHO NACIONAL DE JUSTIÇA. Relatório Justiça em Números 2017, ano-base 2016. Disponível em: <http://www.cnj.jus.br/files/conteudo/arquivo/2017/12/b60a659e5d5cb79337945c1dd137496c.pdf>.

CONSELHO NACIONAL DE JUSTIÇA. **Painel.cnj.br**. Disponível em: <https://paineis.cnj.jus.br/QvAJAXZfc/opendoc.htm?document=qvw_l%5Cpainelcnj.qvw&host=QVS%40neodimio03&anonymous=true>. Acesso em: 29 out. 2018.

COPPIN, B. **Inteligência Artificial**. Rio de Janeiro: LTC, 2017.

FARZINDAR, A.; LAPALME, G. LetSum, an Automatic Legal Text Summarizing System. **ResearchGate**, 2004. Disponível em: <https://www.researchgate.net/publication/228980166_Letsum_an_automatic_legal_text_summarizing_system>. Acesso em: 24 fev. 2019.

FELIPE, B. F. da C.; PERROTA, R. P. C. Inteligência Artificial no Direito – uma realidade a ser desbravada. **Revista de Direito, Governança e Novas Tecnologias**, v. 4, n. 1, p. 1–16, 21 ago. 2018.

FONSECA, M. **A word2vec model trained on Brazilian legislation**.: thefonseca/lex2vec. Disponível em: <https://github.com/thefonseca/lex2vec>. Acesso em: 20 maio 2019.

JAIN, A. K. **Algorithms for clustering data**. [s.l.] Englewood Cliffs, N.J.: Prentice Hall, 1988.

JUSTEN FILHO, M. **Curso de Direito Administrativo**. n. 12a, p. 1861, 2016.

LANE, H.; HOWARD, C.; HAPKE, H. M. **Natural Language Processing in Action**. 3. ed. [s.l.] Manning Publications Co., 2017.

LUO, C.; LI, Y.; CHUNG, S. M. Text document clustering based on neighbors. **Data & Knowledge Engineering, Including Special Section**: Conference on Privacy in Statistical Databases (PSD 2008) – Six selected and extended papers on Database Privacy. v. 68, n. 11, p. 1271–1288, 1 nov. 2009.

MCKINSEY GLOBAL INSTITUTE. **A Future That Works**: Automation, Employment and Productivity. Disponível em: <https://www.mckinsey.com/~/media/mckinsey/featured%20insights/Digital%20Disruption/Harnessing%20automation%20for%20a%20future%20that%20works/MGI-A-future-that-works-Executive-summary.ashx>. Acesso em: 30 out. 2018.

MIHALCEA, R.; TARAU, P. TextRank: Bringing Order into Texts. **Proceedings of EMNLP 2004**, p. 8, 2004.

MIKOLOV, T. et al. Distributed Representations of Words and Phrases and their Compositionality. In: BURGES, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems** 26. [s.l.] Curran Associates, Inc., 2013. p. 3111–3119.

NETWORKX DEVELOPERS. **Degree_centrality - NetworkX 1.9 documentation.** Disponível em: <https://networkx.github.io/documentation/networkx-1.9/reference/generated/networkx.algorithms.centrality.degree_centrality.html>. Acesso em: 12 fev. 2019.

NLTK PROJECT. **Natural Language Toolkit - NLTK 3.4 documentation.** Disponível em: <https://www.nltk.org/>. Acesso em: 20 nov. 2018.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python**. Journal of Machine Learning Research**, v. 12, n. Oct, p. 2825–2830, 2011.

PERKINS, J. **Python Text Processing with NLTK 2.0 Cookbook.** 32 Lincoln Road Olton Birmingham, B27 6PA, UK.: Packt Publishing Ltd., 2010.

PYTHON.ORG**. The Python Tutorial - Documentação do Python 3.7.1**. Disponível em: <https://docs.python.org/3/tutorial/index.html>. Acesso em: 18 nov. 2018.

RICHARDSON, Leonard. **Beautiful Soup Documentation - Beautiful Soup 4.4.0 documentation.** Disponível em: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. Acesso em: 24 jan. 2019.

RINO, L. H. M.; PARDO, T. A. S. **A Sumarização Automática de Textos**: Principais Características e Metodologias. p. 43, 2003.

SCHNABEL, T. et al. Evaluation methods for unsupervised word embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal. **Anais**... In: PROCEEDINGS OF THE 2015 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING. Lisbon, Portugal: Association for Computational Linguistics, 2015. Disponível em: <http://aclweb.org/anthology/D15-1036>. Acesso em: 20 maio 2019.

STF. **Vocabulário Jurídico (Tesauro**). Disponível em: <http://www.stf.jus.br/portal/jurisprudencia/listarTesauro.asp>. Acesso em: 24 jan. 2019.

TJTO. **7 anos**: Sistema de Processo Judicial Eletrônico e-Proc/TJTO é referência para Judiciário brasileiro. Disponível em: <http://www.tjto.jus.br/index.php/magistrado/plantao-forense/8-noticias/5553-7-anos-sistema-de-processo-judicial-eletronico-e-proc-tjto-e-referencia-para-judiciario-brasileiro-2>. Acesso em: 29 out. 2018.

VEITH, C. et al. **How Legal Technology Will Change the Business of Law**. n. Report, 2016.

Recebido em: 04/06/2019
Aprovado em:19/06/2019