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

Rogério Nogueira de Sousa
Information Technology Analyst. Master's Degree Program in Computational Systems Modeling, Federal University of the State of Tocantins. roger@uft.edu.br

David Nadler Prata
Graduate Program Professor in Computer Modeling Systems, by
Federal University of the State of Tocantins. ddnp@uft.edu.br.

1 INTRODUCTION

In 2016, the Judiciary spent R$ 2,248,734,431.00 (two billion two hundred and forty eight million seven hundred and thirty four thousand four hundred and thirty one reais) on Information Technology (IT) and had a workforce of 442,345 employees, divided among magistrates, servers and auxiliaries, to operate the 79.7 million cases that were pending that year in the Brazilian justice system (NATIONAL COUNCIL OF JUSTICE, 2017). In 2017, spending on Information Technology dropped to R$ 2,207,995,675.00 (two billion two hundred and seven million nine hundred and ninety-five thousand four hundred and thirty-one reals) and the number of lawsuits in progress went up to 80 million, with practically the same number of collaborators as in 2016 (NATIONAL COUNCIL OF JUSTICE, 2017).

Faced with this scenario, with expressive numbers that present us with a worrying situation of increased judicial demand and scarcity of resources, the search for increasingly efficient solutions that can maximize the work capacity of the collaborators, as well as reduce costs for processes, becomes imperative for the judicial provision in Brazil.

Information Technology is then accessed as one of the ways to speed up judicial activities, with less time spent by the professionals involved and, consequently, with more savings of resources (FELIPE; PERROTA, 2018). The duty of efficiency implies the requirement that the Public Administration incorporates the technological progress in its activity (JUSTEN FILHO, 2016). The Brazilian justice system is fully aware of the importance of Information Technology for judicial provision, so much so that it allocates around 25% of its budget (without the personnel expenses) to information technology (National Council of Justice, 2017). Among the technological solutions aimed at the justice system, we highlight the use of electronic lawsuits, since 70% of new lawsuits are electronic.

Some Brazilian courts stand out for having 100% of electronic processes in both degrees of jurisdiction, among them (NATIONAL COUNCIL OF JUSTICE, 2017) the Justice Court of the State of Tocantins, which, at the forefront of the electronic judicial process, implemented the e-Proc/TJTO in 2011. In the same year, 100% of new cases became virtual. After 4 years, all cases in progress were digitized, making it the first court to have the entire collection of court cases in digital format (TJTO, 2015).

The digitalization of legal data is a mega-trend, transforming workflows and business models. The volume of data used in legal counseling has increased exponentially (VEITH et al., 2016), generating greater demand for selection, analysis and interpretation of an unprecedented amount of data. In contrast, such virtualization facilitates the automation process, allowing productivity growth and also reducing costs, increasing quality and minimizing downtime for legal operators.

We are living a new era of automation, in which robots and computers can not only perform a series of routine physical work activities more efficiently and cheaply than humans, but also increasingly capable of performing activities that encompass cognitive skills (MCKINSEY GLOBAL INSTITUTE, 2017). With recent developments in robotics, artificial intelligence and machine learning, technologies not only perform activities that we thought only humans could do, but they can do them increasingly at superhuman levels of performance (MCKINSEY GLOBAL INSTITUTE, 2017).

Currently, the procedural phases that require analysis by the operators of the law are the most time consuming, because with the virtualization of processes, there are no more bottlenecks in the acts of filing and processing processes. For the analyst, the texts contained in the file are his main tools of work. Not rarely the analyst is forced to read the entire content of a piece, just to know what it is about. A summary of the content would promote greater efficiency, since it facilitates and accelerates the extraction of relevant information contained within documents of the processes. One of the challenges in working with texts in the legal field is the complexity of the field, as specific terminology and legal interpretations cause many ambiguities (FARZINDAR; LAPALME, 2004).

In this context, the development of a tool capable of automatically generating summaries extracted from court documents in text format has the potential to have a direct impact on the agility of the judiciary collaborators responsible for the analysis of the files. This increase in productivity and agility in document analysis will serve to improve the effectiveness of daily legal activities, which will

have a significant impact on the achievement of the jurisdictional goals established by the National Council of Justice to the entire Judiciary today.

As a consequence, we will have the improvement of the access to justice, by the promptness in granting the jurisdictional guardianship by the State, which has as objective the so-called "fair legal order", which, besides the formal access to the jurisdictional organ, provides the means for the conflict of interests occurred to be solved in a satisfactory way. With this, we intend to contribute to overcome economic or legal barriers that may arise for the effective provision of justice (CINTRA; DINAMARCO; GRINOVER, 2010).

The act of summarizing the text in an automated way is also known as Automatic Summarization (SA); in this case, two techniques from the extractive model of summarization were applied, one using graphs with controlled vocabulary, and another with k-means algorithm with word-embedding. With this, in the applied techniques, we will have summaries composed of sentences extracted from the body of the text, selected based on their relevance.

The general objective of this work is to present the development of a solution capable of automating the generation of summaries of legal texts, through the survey and extraction of sentences that are more relevant to the identification of the central idea of the text, thus speeding up the contextualization of the operators of the law, in order to optimize the necessary decision making during the process.

The specific objectives are to synthesize legal texts; to use graph theory and k-means algorithm to gauge the relevance of the sentences contained in the text; to speed up the analysis of judicial proceedings; to extract the legal vocabulary from the portal of the Federal Supreme Court (STF); to gauge the gains of the legal vocabulary.

2 DEVELOPMENT

The development of the solution was basically divided into: a) extraction of the juridical vocabulary; b) text processing; c) generation of the graph representing the text sentences and clusterization of the text using the K-means algorithm; and d) exposure and measurement of the most relevant sentences.

A vocabulary of legal terms was used to assist in the survey of the most relevant sentences in texts contained in court proceedings. For its formation, we chose to use the legal terms present in the thesaurus of the Federal Supreme Court, which has the function of standardizing information, being a mechanism of terminological control (STF, 2019).

Figure 1. Legal Vocabulary Supreme Court

Source: ("Legal Vocabulary (Thesaurus): Supreme Court (STF)")

To extract the information contained in the mentioned thesaurus, a program was developed that navigates the web in an automatic way, copying data from the visited pages, known as web crawler. This was configured for extraction of the terms contained in the Supreme Court portal, resulting in the formation of a list with 15,434 legal terms.

The program had been implemented in Python making use mainly of Beautiful Soup modules in version 4, for analysis and extraction of the content of HTML pages (RICHARD-SON, 2015), and Selenium, in version 3.14, for navigation between vocabulary pages.

The Python language, in version 3.7, was chosen because it is high-level, object-oriented, capable of being used on several platforms because of its interpretation (PYTHON.ORG, 2019), and has proven to be a good choice for its speed of development and maintenance; it is establishing itself as one of the most popular languages in scientific computing (PEDREGOSA et al., 2011).

2.1 Natural Language Processing (PLN)

Natural Language Processing, also known in academy as computational linguistics, is growing rapidly as its theories and methods are being applied to a range of new technologies (BIRD; KLEIN; LOPER, 2009). This area of study aims to provide tools for a computational system to be able to deal with natural languages at various levels, such as morphological, syntactic and semantic (COPPIN, 2017). To build routines that implement Natural Language Processing methods, we use the bi-black Natural Language Toolkit (NLTK), (NLTK PROJECT, 2019), initially designed for teaching; nowadays, it is adopted by the market due to its usability and scope (PERKINS, 2010).

The texts to be summarized have undergone a set of Natural Language Processing techniques to process them previously in order to abstract them to facilitate computational comprehension.

At first the text will be segmented into sentences; these in turn will be tokenized, an action where sentences will go through a process of separating words into terms, also known as tokens, generating a list of terms per sentence formed from each sentence. The simple task of creating a list of words from a sentence is an essential part of all the processing text (PERKINS, 2010).

Vocabulary represents the set of words (tokens) that will be used in text processing (LANE; HOWARD; HAPKE, 2017). Therefore, the size of the vocabulary directly implies computational complexity and the memory required for proper processing. The use of techniques that reduce vocabulary is essential to gain performance and can give more generality to the processing.

Such techniques seek to transform several words with similar meanings into one. One of these techniques is the conversion of all the letters in the text to lower case. Because it is very common, words started with a capital letter will have the same meaning with the initial lowercase letter. But in some cases the meaning changes, such as the words 'kind' and 'gentle', the first is used as an adjective; the second as a noun, in this case proper name. The use of the lowercase text conversion technique should be evaluated according to the purpose of the processing, and is not recommended when you want to detect in the text named entities, as proper names.

In order to reduce computational processing, special characters are removed from the words contained in these, such as arroba, bars and other symbols. Accents are removed to avoid spelling mistakes impacting the interpretation of words to be transformed into tokens. Numbers and punctuations are also disregarded.

Some common words occur very often in any language, but have low relevance to express the meaning of the sentence; these are called stop words (LANE; HOWARD; HAPKE, 2017). Usually articles, conjunctions, prepositions, interjections, auxiliary verbs and much repeated words in natural language make up this group. Such words are taken from the texts after the tokenization, in order to reduce the computational effort, when one wants to extract information from a text.

It should be noted that in some cases, such as short text processing, the removal of stop words may lead to the loss of information relevant to the meaning of the text, a situation that does not occur in the study in question, given the nature of the texts worked (legal) is unusually extensive. Therefore, the removal of the stop words does not cause significant damage to the semantic value of the text, since they are necessary to natural language for their syntactic value.

Figure 2: Pre-processing sequence of the text.

Original Sentence
What a pity!
He has studied so much for the competition

Sentence Segmentation
“What a pity”
He has studied so much for the competition]

Cleaning and Tokenization
“pity]
,he”
has studied”
competition

To give more generality to the terms, a processing is made in each one of them, one of morphologically complex chains are identified, decomposed in radical and affixes, being discarded the affixes, and the term becomes only the radical, process known as stemming (LANE; HOWARD; HAPKE, 2017).

When the stemming technique is adopted to form the token, removing the suffix and prefix, we have a more generic term, such as the words 'book', 'book', 'books' and 'booklets', all have similar or close meanings and in common the string 'boo', this being the basic element for the meaning. Therefore, the four words can be replaced by the radical 'boo', which has no considerable loss of meaning. Even if 'boo' is not an existing word it doesn't matter, because the aim is to match the words in queries and documents, not to show them to the user (COPPIN, 2017).

The terms are arranged in vectors contained in a vector that represents the sentence. Thus, each document is represented by a vector of sentences in which each item of it is matched by one of the terms; thus, the documents are represented as an array of terms.

Figure 2. Class Object Text

A class called objTex was created in order to receive the information of the text transformed into a matrix, as well as the information of the original text. By activating the constructor method of this class, passing the text to be summarized as a parameter, the attribute "text" is filled with the original text, and later the method generate Sentence is activated, which separates the sentences and returns them in the form of a vector that is added to the attribute "sentences". The methods cleanTokeniza and processText remove the special characters and stop words, perform the stemming of the tokens and assign the representation of the text to the "processed text" attribute.

2.2 Text representation in Graph

A graph is formed by two sets, one of vertices representing objects and the other of edges corresponding to the relationship between the vertices (COPPIN, 2017). To produce a graph that represents a text, it was considered that each sentence in the text is a vertex, and the value attributed to the similarity between the vertices is the edge of this graph. The implementation of the graph used

the python module, networkx, for creation and manipulation.

The similarity between the sentences is calculated by the number of words present in both, which are also contained in the vocabulary of legal terms provided by the Federal Supreme Court. When this number is zero, it is considered that there is no similarity between the sentences; therefore, the edge between them is not created, otherwise there is a link between the sentences whose weight is given by the formula:

Figure 3. Similarity Calculation

$$S = \frac{\textit{Quantity of Equal words between the sentences}}{\text{Log10 (Sentence Size (1))} + \text{Log10 (Sentence Size (2))}}$$

So that very large sentences are not privileged just because they are more extensive than the others, the number of equal words in the sentences is divided by the sum of the logarithms sizes of the sentences.

The method used here is based on the TextRank algorithm (MIHALCEA; TARAU, 2004), developed based on PageRank (BRIN; PAGE, 1998), both implemented using graph theory and used for ranking based on the relations between the vertices (edges). While PageRank is used by Google's search engine to gauge the relevance of web pages, TextRank assigns importance to sentences within a text

Figure 4. Graph that represents the text (Page 72)

2.3 Words Embedding

The most important part in the clustering process is the metric applied to calculate the distance between the elements (JAIN, 1988); thus, it is necessary to transform the text into a mathematical representation to compute the similarity between the sentences.

In this sense, in 2013, the word2vec algorithm was developed, which creates distributed vector representations, called word embedding, capable of representing words, considering the syntactic and semantic relations (AGUIAR, 2016) of each word in relation to the vocabulary, regardless of the language of the texts, giving more flexibility to data processing.

Such vectors are generated using neural networks that make use of a hidden layer and the back propagation algorithm to update the weights of this layer, that is, it generates a vector through machine learning capable of capturing linguistic properties indirectly.

For this project, we used the set of vector representations of words, pre-trained, known as Lex2Vec (FONSECA, 2017), generated from a corpus formed by 233,108 norms promulgated between 1824 and 2017, derived from Brazilian federal legislation.

To generate the vectors, the word2vec tool was used, which implements two vector representation models. One generates the vector with the pre-defined dimensions, considering the context, seeking to inform which would be the missing word, known as Continuous Bag-of-Word (Cbow), (AGUIAR, 2016), while the Continuous Skip-Gram, through a word, seeks to inform which would be the context (MIKOLOV et al., 2013).

Textual representations using word embedding present a wider range of information when compared to representations that make use of word frequency counting, being true even when compared to models that use compensation parameters for frequency effects (SCHNABEL et al., 2015).

2.4 Document Clustering with K-means

Another way to represent and organize the text computationally is through clustering, where the sentences are grouped according to their context. For this purpose, we use K-means, which consists of an unsupervised, interactive learning algorithm with low computational complexity (LUO; LI; CHUNG, 2009), in which the number of clusters (groupings) is assigned arbitrarily by means of a constant. Each cluster is formed around a centroid, which is repositioned to each interaction in order to become the most central point of the cluster.

The sentences are labeled based on their relationship with the centroids, that is, if the sentence is closer to a centroid than the others, it belongs to the nearest centroid cluster and receives the label of this cluster, because the elements of a cluster tend to be similar and different from those not belonging to the group. The K-means itera until there is no more movement of elements between the clusters or until the maximum number of iterations has been reached (JAIN, 1988).

To implement the K-means algorithm in this work, the python K-Means module of the Scikit-Learn library was used. At the end of the interactions a matrix is generated with the index of the sentence and the cluster to which it was assigned. As the representation vectors (words embedding) of the terms have more than 3 dimensions, the mathematical method known as principal component analysis is used for reduction to 3 dimensions, allowing the graphical visualization of the clustered text.

Figure 5. 3D visualization of a text sentences in 3 clusters (figure Page 74)

2.5 Summary Extraction

The concept of centrality in graph theory is associated with the degree of importance of a vertex within a graph, as, for example, more influential people in their social circle perceive a higher index of centrality (BORBA, 2013). In this case, the centrality used is that of degree, meaning that the sentences that have more related sentences will be more relevant to the text.

In this context, the "degree_centrality" function of the Networkx module was applied, which returns a dictionary-type object with the label of the vertex and the degree of centralization normalized by dividing the maximum possibility of a simple n-1 graph, where n is the number of vertices of this graph, of each vertex (NETWORKX DEVELOPERS, 2014).

To collect the vertices with the highest degree of centrality, the object is ordered from the highest to the lowest, using as index the degree of centrality. The number of sentences that make up the object can be calculated based on a percentage of the whole text, that is, a compression rate can be used or by means of a fixed number of sentences.

As the abstract is usually equivalent to 10 to 20% of the original text (RINO; PARDO, 2003), in the solution the compression rate of 90% was chosen, that is, the abstract will have an amount of sentences equivalent to 10% of the amount of sentences of the original text.

As vertex labels represent the positions of the sentences in the original text, the vertices of greater centrality point to the most relevant sentences in the text, thus making it possible to extract the sentences that summarize the information contained in the text.

In the text properly clustered with the k-means algorithm, as well as in the graph, the election of the most relevant sentences considers the centrality of these, only considering your cluster. Therefore, the sentences closest to the centroid are the most relevant in that cluster.

To calculate the distances between the centroid vectors and the vectors representing the sentences, the cosine distance was used (MIKOLOV et al., 2013). Soon, The smaller the angle formed between the two vectors, the smaller the distance between them and the more simi-ar they are.

Figure 6. Cossene distance

Similarity = Page 75

Considering that the amount of sentences that will be part of the summary is 10% of the amount of sentences in the original text, a sentence is collected from each cluster, prioritizing its proximity to the centroid of the cluster to which it is linked.

3 CONCLUSIONS

It is common, in the field of law, to have extensive legal pieces, which may, at times, hinder the work of legal analysts who act in the analysis of information to effect the provision of justice. The legal language, on its own, can constitute a barrier to the proper interpretation of the intent pursued by the applicant. Therefore, the automatic summarization of legal texts aims at transmitting the central message of the text, thus optimizing the main decision-making processes that may arise during the process.

Creating mechanisms that can speed up the procedural process is imperative for a satisfactory jurisdictional provision, because automating tasks is the fastest and least costly way for the Judiciary. Simple actions such as the generation of an automatic summary can have a significant impact on the day to day life of the law operators, due to the immense volume of textual information contained in judicial processes.

An important aspect was the use of the vocabulary of legal terms extracted from the site of the Federal Supreme Court, which promoted more assertiveness to the technique that makes use of graphs, by ensuring that only legal terms are used during the analysis of similarity of sentences, while reducing the ability to generate more accurate summaries of non-legal texts.

The use of words embedding has given more generality to the summarization system; although the vector representations of vocabularies have been generated from texts originating from the legal world, it is able to summarize text outside this context.

The use of clusters has presented several advantages, such as the segmentation of text based on context, which facilitates focused analysis within the document.

It is not rare to have to analyze texts that come from the extraction of the content of formal documents, with multiple pages, and we come across headlines, with non-relevant information, which are repeated, being necessary to disregard such headlines to generate the summary. Clustering tends to group headers into a single cluster, as they have repeated information, thus facilitating the disposal of these during summarization.

This work presented two technically feasible ways, with a relatively low processing and implementation cost. Because, besides making use of an open technological framework, the techniques of summarization, implemented using the graph theory or grouping with k-means algorithm are simple and widely disseminated in the market, which facilitates considerably the maintenance and improvement of the solution, ensuring more possibility of continuity and return on investment.