Instituto
Legislativo
Brasileiro

ILB

LEGIMÁTICA: extração automatizada de informação legislativa e jurídica

João Alberto de Oliveira Lima e Lauro César Araujo (editores)

Legimática:
extração automatizada de informação legislativa e jurídica

Brasília
2017

João Alberto de Oliveira Lima e Lauro César Araujo (editores)

Legimática:
extração automatizada de informação legislativa e jurídica

Relatório com resultados apresentados ao Instituto Legislativo Brasileiro (ILB) do Senado Federal como compromisso da conclusão do Edital número 4/2016 do Grupo de Estudos em Tecnologia da Informação Legislativa.

Senado Federal
Instituto Legislativo Brasileiro (ILB)

Brasília
2017

Lima, João Alberto de Oliveira; Araujo, Lauro César (Editores)
Legimática: extração automatizada de informação legislativa e jurídica/ João Alberto de Oliveira Lima e Lauro César Araujo (editores). – Brasília, 2017-
127 p. : il. (algumas color.) ; 30 cm.

Livro – Senado Federal
Instituto Legislativo Brasileiro (ILB), 2017.
Inclui bibliografia.

1. Informática Legislativa. 2. Inteligência Artificial. 3. Semântica textual. 4. Reconhecimento de Imagens. 5. Ontologia. I. Autores. II. Título

CDD 006.7

"Em cada bloco de mármore vejo uma estátua; vejo-a tão claramente como se estivesse na minha frente, moldada e perfeita na pose e no efeito. Tenho apenas de desbastar as paredes brutas que aprisionam a adorável aparição para revelá-la a outros olhos como os meus já a vêem."

(atribuída a Michelangelo Buonarroti)

Agradecimentos

Ao Instituto Legislativo Brasileiro (ILB) e à Diretoria Geral do Senado Federal, pela iniciativa de criar o grupo de estudos especializado em informática legislativa e conceder espaço para investigação e experimentação de oportunidades e inovação no âmbito da Administração Pública.

À Secretaria de Tecnologia da Informação (Prodasen), especialmente à Coordenação de Infraestrutura de Tecnologia da Informação, por permitir o uso de equipamentos para computação de alguns dos experimentos realizados neste trabalho, e à Coordenação de Informática Legislativa e Parlamentar e ao Núcleo de Qualidade e Padronização de Processos e Produtos de Software, pelo apoio às atividades de pesquisa.

Ao Márcio Fonseca, servidor da Câmara dos Deputados, e ao Alexandre Rademaker, professor da Fundação Getúlio Vargas, pelas valiosas palestras e contribuições.

Ao Eneias Cordeiro da Silva, integrante da composição original do grupo, pela contribuição na codificação do processo de *parser* das normas jurídicas. Ao Klause Alvarenga do Nascimento, pela palestra sobre aplicações de técnicas de Inteligência Artificial no reconhecimento de faces de pessoas.

Sumário

Apresentação

O Grupo de Estudo Acadêmico *Legimática*, criado pelo processo seletivo instituído pelo Edital nº 4/2016 do Instituto Legislativo Brasileiro do Senado Federal, tem por objetivo investigar de forma inovadora e interdisciplinar a aplicação de modernas tecnologias da informação no contexto do processo legislativo. Além da Ciência da Computação e Inteligência Articial, utilizou-se conhecimentos provenientes da Linguística Computacional, da Ciência da Informação, da Ontologia e do Direito.

Normalmente, no processo legislativo os computadores são utilizados apenas como editores de textos modernos, assemelhando-se, em muitos aspectos, às antigas máquinas de datilografia. Em todo ciclo do processo legislativo, desde a iniciativa até a promulgação de novas normas, são inúmeras as oportunidades proporcionadas pela aplicação de novas tecnologias da informação. Na atualidade, fazer mais e melhor com menos recursos é o desafio cotidiano de cada gestor. As iniciativas que contribuam com esse fim se justificam pelo princípio da eficiência na administração pública, inscrito no caput do art. 37 da Constituição Federal, considerado por Hely Lopes Meirelles (2007, p. 98) como "o mais moderno princípio da função administrativa, que já não contenta em ser desempenhada apenas com legalidade".

Este documento apresenta a compilação dos textos preliminares que descrevem os resultados do trabalho de estudos dos componentes do Grupo de Estudo Acadêmico *Legimática*. Cada texto está organizado em um capítulo, produzido majoritariamente por um membro do grupo. Porém, todos assinam como autores de todos os textos, pois os trabalhos foram desenvolvidos em conjunto e colaborativamente. O responsável pelo artigo aparece como primeiro autor. Os textos são preliminares porque ou indicam parcialmente os resultados obtidos, ou estão em estágio inicial de redação, eventualmente apenas com o resumo do trabalho realizado.

No primeiro capítulo, apresenta-se a contribuição "Base de Normas Jurídicas Brasileiras: uma iniciativa de *Open Government Data*" liderada por Hudson de Martim. Descreve-se como foram criados sete *datasets* de normas jurídicas. A partir do texto bruto em RTF, realiza-se o *parser* LexML para estruturar o texto em dispositivos identificados individualmente. Os dispositivos foram agrupados por caputs de artigos e parágrafos, para

submissão ao serviços de Processamento de Linguagem Natural do *Google Cloud Services*. Os resultados desse processamento foram disponibilizados no formato *JavaScript Object Notation* (JSON). Cada *dataset* contém as normas jurídicas em formatos pré-processados adequados para diversos tipos de análises e investigações.

O trabalho liderado por Daniel Viero, "Reconhecimento de entidades no auxílio à indexação de textos legislativos brasileiros", utiliza-se dos resultados dos *datasets* de normas jurídicas e identifica oportunidades de indexação de normas jurídicas realizadas pelo Senado Federal bem como indica formas para o aprimoramento de vocabulário controlado de entidades usado para essa indexação.

A contribuição liderada por Wagner Teixeira, "Exemplo de Extração de Definições em Textos Articulados de Normas Jurídicas com o apoio do Processamento de Linguagem Natural" é mais uma aplicação dos *datasets* de normas jurídicas. Combinando expressões regulares e sofisticados filtros baseados na análise morfossintática dos serviços de processamento de linguagem natural do *Google Cloud Services*, o trabalho mostra como extrair definições em textos articulados, que podem ser utilizadas tanto na criação de glossários como na elicitação de conceitos para uma ontologia de domínio.

Além dos avanços na análise morfossintática de textos, a Inteligência Artificial também tem sido aplicada com alta taxa de sucesso na análise de imagens para identificação de faces de pessoas. No capítulo "Aplicação de uma rede neural convolucional para o reconhecimento facial dos senadores brasileiros da 55ª legislatura 2015-2019", Fabrício Santana apresenta um conjunto de imagens classificadas e um modelo treinado para o reconhecimento facial dos senadores. Este modelo viabiliza o uso da biometria, mesmo nos casos de parlamentares que não possuam digitais reconhecíveis por máquina, trazendo mais segurança ao processo de registro de presença e votação.

Com apoio na Teoria Geral do Processo de von Büllow e na Ontologia de Fundamentação de Giancarlo Guizzardi, Flávio Heringer apresenta conceitos de uma ontologia de domínio do Processo Legislativo. Para von Bullow, "processo judicial" é uma relação jurídica. O mapeamento desse conceito para o processo legislativo como uma relação material na ontologia de fundamentação *Unified Foundational Ontology* (UFO) é a principal contribuição do trabalho "Uma proposta de ontologia sobre processo no âmbito do Processo Legislativo".

Por fim, Jideão Vieira Filho, no capítulo "Construção de um Sistema de Gestão de Normas metamodelado baseado em Ontologia de Fundamentação",

apresenta características do domínio da informação jurídica que motivaram a criação de um sistema metamodelado para a descrição de normas e seus relacionamentos. A flexibilidade na descrição da ontologia de domínio permite a alteração do modelo do sistema de normas sem alterações em tabelas ou programas. O capítulo conta com colaboração da professora Dra. Edna Dias Canedo.

Brasília, dezembro de 2017.

João Alberto de Oliveira Lima e Lauro César Araujo

Base de Normas Jurídicas Brasileiras: uma iniciativa de *Open Government Data*

Hudson de Martim* João Alberto de Oliveira Lima
Lauro César Araujo Daniel de Mello Viero
Fabrício Fernandes Santana Flávio Roberto de Almeida Heringer
Jideão José Vieira Filho Wagner Rodrigues Teixeira

Resumo

As normas jurídicas, produzidas por meio do Processo Legislativo, são a base formal de regulação da convivência em sociedade. Por isso, são naturalmente redigidas de forma técnica com objetivo de serem interpretadas juridicamente. Neste trabalho, porém, apresenta-se uma série de transformações automáticas aplicadas ao arcabouço de leis federais de modo a estruturar a informação descrita nesses documentos com intuito de prepará-las para diferentes tipos de interpretações automáticas. Isso visa auxiliar atividades de informação que vão além da própria interpretação jurídica, e vão ao encontro da *Open Government Data*. O artigo descreve uma série de *datasets* contendo os resultados de transformações da base de normas jurídicas brasileira, que contemplam os textos articulados das normas em representação LexML, CoNLL-U, representações sintáticas de sentenças obtidas com a *Google Natural Language Processing API*, entre outras.

Palavras-chave: Norma Jurídica. Processamento de Linguagem Natural. LexML. CoNLL-U. Google Natural Language Processing API.

1 Introdução

No contexto contemporâneo de produção e disponibilidade de grande quantidade de dados processados por máquina, um dos principais desafios enfrentados por cientistas da informação e pesquisadores em geral é o acesso organizado a esses dados. O movimento *Open Data* surgiu nos anos 2000 para, dentre outros objetivos, ajudar a suprir essas necessidades. Sendo um

*hudson.martim@gmail.com

braço do *Open Science*, o *Open Data* prega a ideia dos dados abertos, compartilhados, livres para serem acessados, usados e redistribuídos por qualquer pessoa, para qualquer propósito, sem qualquer restrição (AUER et al., 2007). Uma especialização do *Open Data* é o *Open Goverment Data* (OGD) (GRAY, 2014), que promove a ideia de que, tornando públicos os dados do governo, pode-se alcançar maior transparência das ações governamentais, estimulando um aumento da participação da sociedade na vida pública e contribuindo com o progresso em pesquisas baseadas nessas informações, como em Santos Neto et al. (2013), que propõe uma abordagem para interligar dados abertos de bibliotecas, arquivos e museus baseada em tecnologias e princípios para publicação de dados abertos estruturados na Web.

O Poder Legislativo coleta e produz diariamente uma grande quantidade de dados que são utilizados na execução de suas funções constitucionais de legislar, fiscalizar o governo e representar a sociedade. Dentre esses dados, as normas jurídicas, produzidas através do Processo Legislativo, constituem uma importante fonte de informação para a sociedade, por definir direitos, deveres, competências e imunidades, regulando as relações entre as pessoas e entre as pessoas e o Estado (HOHFELD, 2008). As normas jurídicas são naturalmente utilizadas como textos para interpretação jurídica no caso concreto. Porém, é desejável estruturar a informação descrita nesses documentos de modo que possa ser processada por máquinas para auxílio a diferentes tipos de atividades além da própria interpretação jurídica, a exemplo da obtenção do texto vigente para determinada data, da compreensão das alterações de ordenamento jurídico, da navegabilidade entre remissões expressas, entre outros.

O conjunto de normas jurídicas historicamente produzidas pelo Legislativo Federal representa um volume expressivo de textos que descrevem a evolução do ordenamento jurídico federal do país, sendo uma base de dados valiosa para cientistas da informação e pesquisadores em geral. Diante disso, o objetivo do presente trabalho é a construção de *datasets* abertos contendo os dados de normas jurídicas brasileiras de forma bruta, textual e estruturada.

Como resultado deste trabalho, foram produzidos oito datasets, conforme a seguir:

a) o *dataset* "Textos Articulados das Normas" (subseção 2.1): contempla os textos articulados[1] da publicação original de cada norma jurídica produzida pelo Processo Legislativo Federal a partir de um marco histórico definido;

[1] Textos estruturados com base na técnica legislativa brasileira, advindo da tradição do Império (ordenações portuguesas Filipinas, Manuelinas e Afonsinas) e da Lei Complementar Nº 95, de 26 de Fevereiro de 1998.

b) o *dataset* "Representação LexML dos Textos Articulados das Normas" (subseção 2.2): contém os textos articulados de cada norma do *dataset* anterior estruturados em formato LexML;

c) o *dataset* "Sentenças da Epígrafe, Ementa, Preâmbulo, Dispositivos e Fecho das Normas" (subseção 2.3): separa em arquivos distintos a epígrafe, a ementa, o preâmbulo, os dispositivos[2] e o fecho de cada norma do *dataset* anterior;

d) o *dataset* "Sentenças dos Dispositivos das Normas com Enumerações Agrupadas" (subseção 2.4): agrupa os incisos e alíneas como enumerações da sentença de cada dispositivo do *dataset* anterior;

e) o *dataset* "Representação CoNLL-U das sentenças dos Dispositivos das Normas" (subseção 2.5): contempla a representação CoNLL-U dos dispositivos do terceiro *dataset*;

f) o *dataset* "Representação Sintática das sentenças dos Dispositivos das Normas" (subseção 2.6): contém o resultado da análise sintática de cada dispositivo do quarto *dataset* realizada por processamento de linguagem natural por meio da *Google Natural Language Processing API*;

g) o *dataset* "Textos da Articulação e da Ementa das Normas" (subseção 2.7): agrupa as sentenças dos dispositivos de cada norma criando um arquivo com o texto completo da articulação da norma. É ainda oferecido, para cada norma, um arquivo com o texto da sua ementa;

h) o *dataset* "Representação Sintática dos Textos da Articulação e da Ementa das Normas" (subseção 2.8): contém o resultado da análise sintática do texto da articulação e do texto da ementa de cada norma realizada por processamento de linguagem natural por meio da *Google Natural Language Processing API*.

O seção 2 contempla a descrição de cada um dos *datasets*. A seção 3 apresenta os métodos utilizados para produzi-los e as limitações encontradas. Considerações finais são apresentadas na seção 4.

2 Descrição dos Datasets

As normas jurídicas representam não apenas a saída, mas também a entrada do Processo Legislativo, pois as normas vigentes são constantemente alteradas. Para apoiar o Processo Legislativo Federal, o Senado Federal mantém uma base de dados de normas jurídicas com textos articulados da

[2] Exemplos de dispositivos são Capítulos, Artigos, Incisos, entre outros.

publicação original, republicações e retificações obtidas do Diário Oficial da União (DOU). As normas publicadas no DOU contemplam os textos vigentes na época de sua publicação, ou seja, são os textos com validade jurídica. Os *datasets* produzidos nesta pesquisa foram construídos a partir das bases do Senado Federal. Portanto, referem-se aos textos originais válidos das normas jurídicas. Entretanto, como ressaltado em Lima (2013), "os textos publicados nas bases de dados de legislação na internet no Brasil, mesmo que armazenadas em sítios oficiais das instituições do Estado, não possuem nenhum caráter oficial". Por isso, a construção de defesas jurídicas a partir de documentos derivados da publicação original devem observar a necessidade de consulta ao DOU.

Os *datasets* são compostos de textos de Leis e Leis Complementares federais do período entre 4 de outubro de 1946 e 12 de abril de 2017. Procurou-se um recorte metodológico que selecionasse uma quantidade razoável de normas com maior probabilidade de estarem vigentes. Isso não exclui normas expressamente revogadas, pois o histórico dos textos é importante para a obtenção do texto vigente em uma determinada data[3]. Não estão incluídas as Emendas Constitucionais, Decreto-Lei, Decretos, entre outros atos normativos, tão pouco os textos das proposições legislativas.

As subseções a seguir descrevem cada *dataset* oferecido e apresentam, como exemplo, o resultado do processamento da Lei n. 13.416 de 23 de Fevereiro de 2017.

2.1 Dataset 1: Textos Articulados das Normas

O *dataset* "Textos Articulados das Normas" contém, para cada lei, e lei complementar, do período definido, um arquivo *Rich Text Format* (RTF) com o texto articulado original, extraído do DOU. Há 13.567 arquivos de normas, cujo nome é formatado como `<tipo da norma>-<ano da norma>-<número da norma>.rtf`. Os arquivos RTF são a base das entradas dos demais *datasets*.

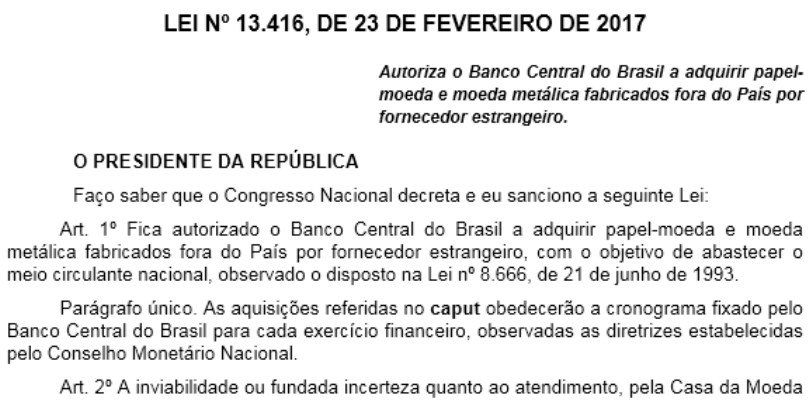
A Figura 1 apresenta o texto da Lei n. 13.416/2017, composto pela epígrafe, ementa, preâmbulo e parte articulada.

2.2 Dataset 2: Representação LexML dos Textos Articulados das Normas

O LexML é um portal administrado pelo Senado Federal especializado em informação jurídica e legislativa que pretende reunir leis, decretos, acór-

[3] O princípio do direito civil "*Tempus regit actum*" define que o ato jurídico se rege pela lei da época, daí a importância de se ter o histórico de textos.

Figura 1 – Exemplo de conteúdo de arquivo do *Dataset 1* (arquivo `LEI-2017-13416.rtf`)



LEI Nº 13.416, DE 23 DE FEVEREIRO DE 2017

Autoriza o Banco Central do Brasil a adquirir papel-moeda e moeda metálica fabricados fora do País por fornecedor estrangeiro.

O PRESIDENTE DA REPÚBLICA

Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:

Art. 1º Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda metálica fabricados fora do País por fornecedor estrangeiro, com o objetivo de abastecer o meio circulante nacional, observado o disposto na Lei nº 8.666, de 21 de junho de 1993.

Parágrafo único. As aquisições referidas no **caput** obedecerão a cronograma fixado pelo Banco Central do Brasil para cada exercício financeiro, observadas as diretrizes estabelecidas pelo Conselho Monetário Nacional.

Art. 2º A inviabilidade ou fundada incerteza quanto ao atendimento, pela Casa da Moeda

Fonte: Os autores

dãos, súmulas, projetos de leis entre outros documentos das esferas federal, estadual e municipal dos Poderes Executivo, Legislativo e Judiciário de todo o Brasil. O LexML oferece um esquema XML[4] para a estruturação dos textos de normas, julgados e projetos de normas através de vocabulário unificado. Outro produto desse projeto é o Parser LexML que, a partir do texto articulado de uma norma em formato RTF, gera um texto em formato `XML` utilizando-se o schema LexML.

O *dataset* "Representação LexML dos Textos Articulados das Normas" é o resultado da execução do Parser LexML sobre o *dataset* 1 e contém um arquivo no formato LexML para cada norma da entrada, em caso de sucesso na conversão. Há 12.569 arquivos, cujo nome é formatado como `<tipo da norma>-<ano da norma>-<número da norma>.xml` – na seção "Limitações Encontradas", há uma descrição sobre as causas de o número de arquivos de saída (*dataset* 2) ser menor que o número de arquivos de entrada (*dataset* 1).

A Figura 2 apresenta o texto da Lei n. 13416/2017 estruturado em XML, esquema LexML. Nota-se que os elementos epígrafe, ementa, preâmbulo e parte articulada estão precisamente delimitados por *tags* do vocabulário LexML. O vocabulário incorpora denota elementos semânticos de organização da informação normativa estruturada. O uso padrão LexML

[4] A documentação do esquema LexML se encontra em <http://projeto.lexml.gov.br/documentacao/Parte-3-XML-Schema.pdf>, acesso de 11.mar.2018.

vai ao encontro do que RIBEIRO e PEREIRA (2015) defendem:

> Para que os princípios de dados abertos sejam completamente atendidos, é preciso que haja uma organização semântica dos dados publicados [...], por exemplo, através do uso de padrões abertos e de vocabulários estruturados, que possibilitem a padronização terminológica, visando à comunicação e compreensão dos dados descritos por máquinas.

2.3 Dataset 3: Sentenças da Epígrafe, Ementa, Preâmbulo, Dispositivos e Fecho das Normas

Em trabalhos de extração semântica de normas, pode ser necessário analisar o texto da norma como um todo, por exemplo, para extrair entidades nomeadas, ou analisar individualmente a sentença de cada elemento (epígrafe, ementa, preâmbulo, dispositivos, fecho) para, por exemplo, tipificar remissões ou extrair definições.

O *dataset* "Sentenças da Epígrafe, Ementa, Preâmbulo, Dispositivos e Fecho das Normas" é o resultado do processamento do *dataset* 2 e contém um diretório para cada norma de entrada, e, dentro de cada diretório de norma, um arquivo separado para cada elemento da norma. Há 362.030 arquivos nesse *dataset*.

Na Figura 3, é apresentada a sentença do *caput* do artigo 1 da Lei n. 13.416/2017.

2.4 Dataset 4: Sentenças dos Dispositivos das Normas com Enumerações Agrupadas

Este *dataset* apresenta os incisos e alíneas nos textos articulados das normas são estruturados como dispositivos distintos, apesar de serem enumerações de um dispositivo agregador. Os incisos 1 e 2 do parágrafo 1 do artigo 2 da Lei n. 13.416/2017, apresentados na Figura 4, são um exemplo dessa situação.

Com essa separação, as sentenças no *dataset* 3 de dispositivos que possuem incisos e alíneas ficam mal-formadas, incompletas, geralmente terminadas com um ":" (dois pontos). Exemplo desse tipo de sentença é apresentado no conteúdo do parágrafo 1 do artigo 2 da Lei n. 13.416/2017 da Figura 4.

Assim, a partir do *dataset* 3, foi produzido o *dataset* "Sentenças dos Dispositivos das Normas com Enumerações Agrupadas", que possui arquivos apenas para os dispositivos agregadores, tendo acrescentadas ao final de

Figura 2 – Exemplo de conteúdo de arquivo do *Dataset 2* (arquivo `LEI-2017-13416.xml`)

```
▼<LexML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.lexml.gov.br/1.0"
 xsi:schemaLocation="http://www.lexml.gov.br/1.0 ../xsd/lexml-br-rigido.xsd">
 ▼<!--
          xsi:schemaLocation="http://www.lexml.gov.br/1.0 http://projeto.lexml.gov.br/esq
  -->
 ▼<Metadado>
    <Identificacao URN="urn:lex:br:senado.federal:lei:2017;13416@data.evento;leitura;2017-
    04-23t10.02"/>
  </Metadado>
 ▼<ProjetoNorma>
  ▼<Norma>
   ▼<ParteInicial>
      <Epigrafe id="epigrafe"/>
     ▼<Ementa id="ementa">
      ▼<b>
         <span xlink:href="urn:lex:br:federal:lei:2017-02-23;13416">LEI Nº 13.416, DE 23
         DE FEVEREIRO DE 2017</span>
        </b>
      ▼<b>
        ▼<i>
           Autoriza o Banco Central do Brasil a adquirir papel-moeda e moeda metálica
           fabricados fora do País por fornecedor estrangeiro.
         </i>
        </b>
      </Ementa>
     ▼<Preambulo id="preambulo">
        <p>O PRESIDENTE DA REPÚBLICA</p>
      ▼<p>
         Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:
        </p>
      </Preambulo>
    </ParteInicial>
   ▼<Articulacao>
    ▼<Artigo id="art1">
       <Rotulo>Art. 1º</Rotulo>
      ▼<Caput id="art1_cpt">
         <!-- Link: urn:lex:br:federal:lei:1993-06-21;8666 -->
        ▼<p>
          ▼<b>
             Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda
             metálica fabricados fora do País por fornecedor estrangeiro, com o objetivo
             de abastecer o meio circulante nacional, observado o disposto na
             <span xlink:href="urn:lex:br:federal:lei:1993-06-21;8666">Lei nº 8.666, de
             21 de junho de 1993</span>
             .
           </b>
         </p>
       </Caput>
      ▼<Paragrafo id="art1_par1u">
         <Rotulo>Parágrafo único.</Rotulo>
```

Fonte: Os autores

Figura 3 – Exemplo de conteúdo de arquivo do *Dataset 3* (arquivo `LEI-2017-13416/0005-art1_cpt.txt`)

```
Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda metÃ¡lica fabricados fora do PaÃs
por fornecedor estrangeiro, com o objetivo de abastecer o meio circulante nacional, observado o disposto na
Lei nÂº 8.666, de 21 de junho de 1993.
```

Fonte: Os autores

suas sentenças, como enumerações, as sentenças de seus incisos e alíneas. Há 128.547 arquivos nesse *dataset*.

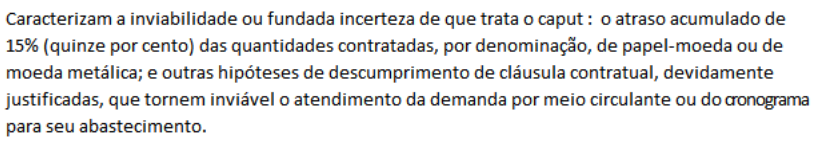
Na Figura 5 é apresentada a sentença do parágrafo 1 do artigo 2 da Lei n. 13.416/2017 com as sentenças dos seus incisos 1 e 2 concatenadas como enumerações, conforme o arquivo gerado no *dataset* 4 para o parágrafo em questão. Não foram gerados arquivos para os incisos 1 e 2 no *dataset* 4.

Figura 4 – Conteúdo LexML do parágrafo 1 do artigo 2 da Lei 13.416/2017, com seus incisos

```
▼<Paragrafo id="art2_par1">
   <Rotulo>§ 1º</Rotulo>
  ▼<p>
   ▼<b>
      Caracterizam a inviabilidade ou fundada incerteza de que trata o caput :
    </b>
  </p>
 ▼<Inciso id="art2_par1_inc1">
    <Rotulo>I -</Rotulo>
   ▼<p>
    ▼<b>
       o atraso acumulado de 15% (quinze por cento) das quantidades contratadas, por denominação,
       de papel-moeda ou de moeda metálica; e
     </b>
    </p>
  </Inciso>
 ▼<Inciso id="art2_par1_inc2">
    <Rotulo>II -</Rotulo>
   ▼<p>
    ▼<b>
       outras hipóteses de descumprimento de cláusula contratual, devidamente justificadas, que
       tornem inviável o atendimento da demanda por meio circulante ou do cronograma para seu
       abastecimento.
     </b>
    </p>
  </Inciso>
 </Paragrafo>
▶<Paragrafo id="art2_par2">...</Paragrafo>
</Artigo>
```

Fonte: Os autores

Figura 5 – Sentença do parágrafo 1 do artigo 2 da Lei 13.416/2017 com seus seus incisos concatenados como enumerações



Fonte: Os autores

2.5 Dataset 5: Representação CoNLL-U das sentenças dos Dispositivos das Normas

O processamento de textos por máquina com objetivo de indexação, de extração de definições, de tipificação de remissões, por exemplo, não traz resultados satisfatórios em geral quando realizado diretamente sobre sentenças em linguagem natural. Esse tipo de processamento, que busca a reconhecimento da semântica das sentenças, exige uma abordagem mais avançada do que a de processamento textual simples. Estudos de Batista et al. (2011), Nakamura, Ogawa e Toyama (2013) demonstram que, em um processamento de extração da semântica, um passo intermediário de análise sintática das sentenças contribui de maneira relevante para a acurácia dos resultados, contribuindo também na simplificação da implementação do processamento.

O formato CoNLL-U (BUCHHOLZ; MARSI, 2006), utilizado no projeto *Universal Dependences* (UD) (NIVRE et al., 2016), estrutura cada palavra/token de uma sentença em uma linha com colunas separadas por *tab*. Cada coluna, em termos gerais, representa uma aspecto da palavra/token, como a forma, o lema, a tag "part-of-speech", características morfológicas etc. Essa estruturação de características e dependências sintáticas/morfológicas tem como objetivo facilitar o processamento de linguagem natural multi-linguagem, além de, por exemplo, suportar aprendizado e avaliações comparativas através de línguas diferentes.

Tendo o *dataset* 3 como entrada, foi produzido o *dataset* "Representação CoNLL-U das sentenças dos Dispositivos das Normas", que contém um diretório para cada norma de entrada e, dentro de cada diretório de norma, um arquivo texto (TXT) para cada dispositivo, cujo conteúdo é a sentença do dispositivo em formato CoNLL-U. Há 337.520 arquivos nesse *dataset*.

Na Figura 6, é apresentada a sentença do *caput* do artigo 1 da Lei n. 13.416/2017 estruturada em CoNLL-U.

Figura 6 – Estrutura CoNLL-U da sentença do artigo 1 da Lei 13.416/2017

```
1   Fica        _   VERB    VERB    _   2   nsubj       _   _
2   autorizado      _   VERB    VERB    _   0   ROOT    _   _
3   o           _   DET     DET     _   4   det         _   _
4   Banco       _   PROPN   PNOUN   _   2   obj         _   _
5   Central     _   PROPN   PNOUN   _   4   amod        _   _
6   do          _   CCONJ   CONJ    _   7   cc          _   _
7   Brasil      _   PROPN   PNOUN   _   4   conj        _   _
8   a           _   ADP     ADP     _   9   mark        _   _
9   adquirir        _   VERB    VERB    _   2   advcl   _   _
10  papel-moeda     _   ADJ     ADJ     _   9   xcomp:adj   _   _
11  e           _   CCONJ   CONJ    _   12  cc          _   _
12  moeda       _   NOUN    NOUN    _   10  conj        _   _
13  metálica        _   ADJ     ADJ     _   12  amod    _   _
14  fabricados      _   VERB    VERB    _   9   acl:part    _   _
15  fora        _   ADV     ADV     _   14  advmod      _   _
16  do          _   X       ADPPRON _   17  case        _   _
17  País        _   PROPN   PNOUN   _   15  nmod        _   _
18  por         _   ADP     ADP     _   19  case        _   _
19  fornecedor      _   NOUN    NOUN    _   14  nmod    _   _
20  estrangeiro     _   ADJ     ADJ     _   19  amod    _   _
21  ,           _   PUNCT   .       _   26  punct       _   _
22  com         _   ADP     ADP     _   26  mark        _   _
23  o           _   DET     DET     _   22  fixed       _   _
24  objetivo        _   NOUN    NOUN    _   22  fixed   _   _
25  de          _   ADP     ADP     _   22  fixed       _   _
26  abastecer       _   VERB    VERB    _   14  advcl   _   _
27  o           _   DET     DET     _   28  det         _   _
28  meio        _   NOUN    NOUN    _   26  obj         _   _
29  circulante      _   NOUN    NOUN    _   28  appos   _   _
30  nacional        _   ADJ     ADJ     _   29  amod    _   _
31  ,           _   PUNCT   .       _   32  punct       _   _
32  observado       _   VERB    VERB    _   26  acl:part    _   _
33  o           _   DET     DET     _   34  det         _   _
34  disposto        _   NOUN    NOUN    _   32  obj     _   _
35  na          _   ADJ     ADJ     _   34  amod        _   _
36  Lei         _   PROPN   PNOUN   _   34  appos       _   _
37  nº          _   NUM     NUM     _   38  nummod      _   _
```

Fonte: Os autores

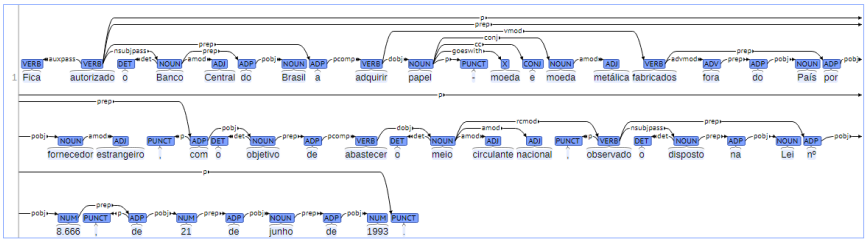
Na Figura 7 é apresentada a sentença do *caput* do artigo 1 da Lei n. 13.416/2017 no grafo da sua árvore sintática, renderizado a partir de sua estrutura CoNLL-U.

2.6 Dataset 6: Representação Sintática das sentenças dos Dispositivos das Normas

A Google oferece uma *API Cloud*[5] para processamento de linguagem natural (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), que é apresentada como um conjunto de serviços web/rest que utilizam modelos

[5] A API de Processamento de Linguagem Natural da Google para português-br encontra-se em <https://cloud.google.com/natural-language/?hl=pt-br>, acesso de 11.mar.2018.

Figura 7 – Renderização da árvore sintática da sentença do artigo 1 da Lei 13.416/2017



Fonte: Os autores

e estratégias de aprendizado de máquina para realizar análise avançada de textos. Dentre os serviços oferecidos por essa *API*, há o serviço de Análise Sintática, que extrai do texto tokens e frases, identifica classes gramaticais (*Parts of Speech* - PoS), e cria uma árvore de análise sintática para cada frase. Além dele, há o serviço de Análise de Entidades que identifica entidades que aparecem no texto e as classifica como pessoa, organização, local, eventos, produtos, mídia.

Com base no *dataset* 4, foi produzido o *dataset* "Representação Sintática das sentenças dos Dispositivos das Normas", que contém um diretório para cada norma de entrada e, dentro de cada diretório de norma, um arquivo *json* para cada dispositivo, cujo conteúdo é o resultado do processamento da sentença do dispositivo pela API da Google, utilizando-se os serviços Análise Sintática e Análise de Entidades. Há 128.547 arquivos nesse *dataset*.

Na Figura 8, é apresentada a sentença do *caput* do artigo 1 da Lei n. 13.416/2017 estruturada em elementos *json* da API de processamento de linguagem natural da Google.

2.7 Dataset 7: Textos da Articulação e da Ementa das Normas

Como consta na descrição do *dataset* 3, "em trabalhos de extração semântica de normas, pode ser necessário analisar o texto da norma como um todo, por exemplo, para extrair entidades nomeadas, ou analisar individualmente a sentença de cada elemento (epígrafe, ementa, preâmbulo, dispositivos, fecho), para, por exemplo, tipificar remissões ou extrair definições".

A construção do *dataset* 3, por exemplo, é focada em trabalhos que precisam analisar individualmente a sentença de cada elemento da norma. Já a construção do *dataset* 7 tem como objetivo contribuir com trabalhos em que

Figura 8 – Exemplo de conteúdo de arquivo do *Dataset 6* (arquivo `LEI-2017-13416/0005-art1_cpt.json`)


```
{
  "sentences": [
    {
      "text": {
        "content": "Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda metÃ¡lica
fabricados fora do PaÃs por fornecedor estrangeiro, com o objetivo de abastecer o meio circulante
nacional, observado o disposto na Lei nÂº 8.666, de 21 de junho de 1993.",
        "beginOffset": 0
      }
    }
  ],
  "tokens": [
    {
      "text": {
        "content": "Fica",
        "beginOffset": 0
      },
      "partOfSpeech": {
        "tag": "VERB",
        "aspect": "IMPERFECTIVE",
        "case": "CASE_UNKNOWN",
        "form": "FORM_UNKNOWN",
        "gender": "GENDER_UNKNOWN",
        "mood": "INDICATIVE",
        "number": "SINGULAR",
        "person": "THIRD",
        "proper": "NOT_PROPER",
        "reciprocity": "RECIPROCITY_UNKNOWN",
        "tense": "PRESENT",
        "voice": "VOICE_UNKNOWN"
      },
      "dependencyEdge": {
        "headTokenIndex": 1,
        "label": "AUXPASS"
      },
      "lemma": "Ficar"
    },
    {
      "text": {
        "content": "autorizado",
        "beginOffset": 5
      },
      "partOfSpeech": {
        "tag": "VERB",
        "aspect": "PERFECTIVE",
        "case": "CASE_UNKNOWN",
```


Fonte: Os autores

é necessário analisar o texto da norma como um todo, mais especificamente, o texto da articulação da norma. Por isso, com base no *dataset* 4, foi produzido então o *dataset* "Textos da Articulação e da Ementa das Normas", que agrupa as sentenças dos dispositivos de cada norma criando, para cada norma, um arquivo com o texto completo da sua articulação e um arquivo com o texto da sua ementa. Há 25.944 arquivos nesse *dataset*, sendo uma metade com arquivos das articulações e a outra com arquivos das ementas das normas.

Na Figura 9, é apresentado o texto da articulação da Lei n. 13416/2017.

Figura 9 – Texto da articulação da Lei $LEI - 2017 - 13416$

```
Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda
metálica fabricados fora do País por fornecedor estrangeiro, com o objetivo
de abastecer o meio circulante nacional, observado o disposto na Lei nº
8.666, de 21 de junho de 1993.
As aquisições referidas no caput obedecerão a cronograma fixado pelo Banco
Central do Brasil para cada exercício financeiro, observadas as diretrizes
estabelecidas pelo Conselho Monetário Nacional.
A inviabilidade ou fundada incerteza quanto ao atendimento, pela Casa da
Moeda do Brasil, da demanda por meio circulante ou do cronograma para seu
abastecimento, em cada exercÃcio financeiro, caracteriza situação de
emergência, para efeito de aquisição de papel-moeda e de moeda metálica de
fabricantes estrangeiros, na forma do inciso IV do caput do art. 24 da Lei
nº 8.666, de 21 de junho de 1993.
Caracterizam a inviabilidade ou fundada incerteza de que trata o caput :
o atraso acumulado de 15% (quinze por cento) das quantidades contratadas,
por denominação, de papel-moeda ou de moeda metálica; e
outras hipóteses de descumprimento de cláusula contratual, devidamente
justificadas, que tornem inviável o atendimento da demanda por meio
circulante ou do cronograma para seu abastecimento.
Para fins da caracterização da situação de emergência de que trata este
artigo, o Banco Central do Brasil fica obrigado a enviar o Programa Anual
de Produção à  Casa da Moeda do Brasil, até 31 de agosto de cada ano, no
qual serão indicadas as projeções de demandas de papel-moeda e de moeda
metálica para o exercício financeiro seguinte.
Esta Lei entra em vigor na data de sua publicação.
```

Fonte: Os autores

2.8 Dataset 8: Representação Sintática dos Textos da Articulação e da Ementa das Normas

Com base no *dataset* 7, foi produzido o *dataset* "Representação Sintática dos Textos da Articulação e da Ementa das Normas", que contém dois arquivos *json* para cada norma, um para sua articulação e outro para sua ementa, cujos conteúdos são o resultado do processamento do texto correspondente pela API de processamento de linguagem natural da Google,

utilizando-se os serviços Análise Sintática e Análise de Entidades – conforme apresentados na descrição do *dataset* 6. Há 25.944 arquivos nesse *dataset*, sendo uma metade com arquivos das articulações processadas e a outra com arquivos das ementas processadas das normas.

Na Figura 10 é apresentado o texto da articulação da Lei n. 13.416/2017 estruturada em elementos *json* da API de processamento de linguagem natural da Google.

Figura 10 – Exemplo de conteúdo de arquivo do *Dataset 8* (arquivo `LEI-2017-13416-dispositivos.json`)


```
{
  "sentences": [
    {
      "text": {
        "content": "Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda metálica
fabricados fora do País por fornecedor estrangeiro, com o objetivo de abastecer o meio circulante nacional,
observado o disposto na Lei nº 8.666, de 21 de junho de 1993.",
        "beginOffset": 1
      }
    },
    {
      "text": {
        "content": "As aquisições referidas no caput obedecerão a cronograma fixado pelo Banco Central do
Brasil para cada exercício financeiro, observadas as diretrizes estabelecidas pelo Conselho Monetário
Nacional.",
        "beginOffset": 257
      }
    },
    {
      "text": {
        "content": "A inviabilidade ou fundada incerteza quanto ao atendimento, pela Casa da Moeda do
Brasil, da demanda por meio circulante ou do cronograma para seu abastecimento, em cada exercício
financeiro, caracteriza situação de emergência, para efeito de aquisição de papel-moeda e de moeda metálica
de fabricantes estrangeiros, na forma do inciso IV do caput do art. 24 da Lei nº 8.666, de 21 de junho de
1993.",
        "beginOffset": 461
      }
    },
    {
```


Fonte: Os autores

3 Métodos Utilizados

Para realizar o processamento da análise sintática das sentenças dos dispositivos das normas, optou-se por utilizar *framework* ou *API Cloud* oferecidos de forma gratuita.

Iniciou-se a pesquisa por *frameworks* e foi identificado que um dos mais lembrados *Parsers* sintáticos para a língua portuguesa é o Palavras (BICK, 2000), um analisador automático (tagger-parser) para português, baseado em regras gramaticais, que foi desenvolvido por Eckhard Bick no contexto de um projeto de doutoramento (1994-2000) na Universidade de Århus (Dinamarca) e utilizado pelo projeto Floresta Sintática (FREITAS;

ROCHA; BICK, 2008). O que impediu a utilização do Palavras neste trabalho foi o seu caráter não-gratuito.

Depois de mais pesquisas, optou-se por utilizar o SyntaxNet, um *framework* aberto e gratuito oferecido pela Google, baseado em redes neurais e implementado sobre o TensorFlow[6]. O SyntaxNet fornece uma fundação para sistemas *Natural Language Understanding* (NLU) e provê um *Parser* sintático que pode ser treinado com corpus em CoNLL-U[7].

A Google oferecia um modelo pré-treinado para português brasileiro baseado no *corpus* da UD. O SyntaxNet, como qualquer framework baseado em *Deep Learning*, precisa de uma base de treinamento expressiva para alcançar níveis de acurácia satisfatórios. O *corpus* português-Br oferecido pela UD tinha em torno de 9.000 sentenças na época dos nossos processamentos. Os resultados da análise sintática sobre as sentenças do *dataset* 3 usando o SyntaxNet com o *corpus* de treinamento UD (*dataset* 5) não se mostraram muito promissores em avaliação por amostragem de sentenças com características utilizadas com frequência em textos legislativos. Isso se deve basicamente à quantidade pequena de sentenças da base da UD utilizada no treinamento.

Durante o período de testes com o SyntaxNet, a Google lançou uma versão beta da sua *Cloud Natural Language API* para o idioma português brasileiro (anteriormente, só estava disponível para inglês, francês, japonês e espanhol). A Google oferecia um crédito por conta de usuário para testar sua API Cloud. Foi calculado que, com o volume de sentenças do *dataset* 4, o limite de crédito de apenas uma conta seria suficiente para o processamento de todas as sentenças. Mesmo tendo saldo suficiente para todas as sentenças, para economizar nas chamadas à API da Google, foi criada uma estrutura de *cache*, aplicando um *Message-Digest algorithm 5* (MD5) em cada sentença com o objetivo de não submeter sentenças repetidas à API. Foram encontradas 26.795 repetições de sentenças, uma economia de 21% no total das potenciais chamadas.

Os resultados da análise sintática sobre as sentenças do *dataset* 4 usando a API Cloud da Google (*dataset* 6) se mostraram promissores, baseando-se na mesma avaliação por amostragem realizada em relação ao processamento com o SyntaxNet usando o *corpus* da UD.

---

[6] O TensorFlow é uma biblioteca de código aberto desenvolvida pela Google para computação numérica usando grafos de dados e que vêm sendo amplamente utilizada na área de aprendizado de máquina.

[7] Uma visão geral do SyntaxNet encontra-se em <https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html>, acesso de 11.mar.2018.

4 Limitações Encontradas

A massa de dados de entrada nunca foi formatada tendo em vista o processamento automático da estrutura via Parser. Por isso, durante os processamentos, foram encontradas situações em normas específicas que impediram a geração do texto estruturado em XML.

Na produção do *dataset* 2 (formato LexML), 998 normas de entrada do *dataset* 1 (em torno de 7%) não tiveram arquivos produzidos no *dataset* 2, pelo fato de o Parser LexML não ter reconhecido de forma correta a articulação do texto da norma de entrada. Um desses casos é a Lei 10.406/2002 ("Novo Código Civil"), que apresenta técnica legislativa incompatível com a Lei Complementar 95/1997, na qual o Parser LexML é baseado.

Considerações finais

Seguindo a ideia promovida pela OGD de tornar públicos os dados do governo para promover transparência e contribuir com o progresso em pesquisas baseadas nessas informações (VICTORINO et al., 2017), este trabalho permitiu oferecer oito *datasets* de forma aberta contendo textos originais e processados com análise sintática de normas federais brasileiras. Os *datasets* estão publicados na plataforma *Figshare* de repositórios digitais especializados em pesquisas acadêmicas[8].

No intuito de contribuir com cientistas da informação e pesquisadores em geral, esses *datasets* podem ser entendidos como uma plataforma de dados para trabalhos futuros, como por exemplo na construção de um extrator de definições ou de tipificação de remissões, que, em vez de utilizar como entrada o texto original de cada norma, poderia utilizar por exemplo o *dataset* 6 como entrada, que contém a análise sintática da sentença de cada dispositivo.

Conforme apresentado no seção 2, adotado-se como recorte metodológico o período entre 4 de outubro de 1946 e 12 de abril de 2017. Uma possibilidade de novas pesquisas seria atualizar os *datasets* com normas de um período de publicação ampliado como, por exemplo, a partir de 1824, ano em que foi outorgada a Primeira Constituição Brasileira.

Outro filtro metodológico utilizado foi o de considerar apenas as normas federais do tipo Lei ou Lei Complementar. Trabalhos futuros poderiam expandir os *datasets* para contemplarem mais tipos de norma.

Gerar um *dataset* das sentenças dos dispositivos em formato CoNLL-U a partir do *dataset* 6, que contém um *json* de análise sintática da Google

[8] Disponível em <https://doi.org/10.6084/m9.figshare.c.4029253.v1>, acesso de 11.mar.2018.

para cada dispositivo, poderia ser uma contribuição para a formação de um *corpus* UD com foco em textos legislativos. Seria necessária uma revisão, por pessoas com experiência em linguística, das estruturas sintáticas geradas.

Como são produzidas continuamente novas normas jurídicas através do Processo legislativo, trabalhos futuros poderiam implementar um mecanismo de atualização periódica ou contínua dos *datasets* produzidos neste trabalho.

O Processamento de linguagem natural tem evoluído rapidamente nos últimos anos, muito pelo aplicação de técnicas de inteligência artificial. A extração da semântica de textos utiliza-se, por padrão, de passos intermediários de análise sintática para alcançar uma acurácia satisfatória. Por exemplo, recentemente a Google publicou o trabalho de criação do SLING (RINGGAARD; GUPTA; PEREIRA, 2017), um *Parser* para anotação semântica em textos, baseado em redes neurais que, partindo apenas dos *tokens* textuais de entrada, produz grafos de *frames* semânticos sem qualquer representação simbólica interveniente. Ainda é cedo para conclusões definitivas, mas a análise sintática de textos e sentenças para fins de extração semântica pode se tornar um passo desnecessário. Assim, trabalhos futuros poderiam acompanhar a evolução dessas técnicas para avaliar se alguns dos *datasets* produzidos neste trabalho ainda serão relevantes e se outros precisarão ser produzidos.

Referências

AUER, S. et al. Dbpedia: A nucleus for a web of open data. In: *In 6th Int'l Semantic Web Conference, Busan, Korea*. [S.l.]: Springer, 2007. p. 11–15.

BATISTA, A. H. et al. Extração automática de definições: um estudo de caso em textos legislativos. Universidade Católica de Brasília, 2011.

BICK, E. The parsing system palavras. *Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*, University of Arhus, 2000.

BUCHHOLZ, S.; MARSI, E. Conll-x shared task on multilingual dependency parsing. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the Tenth Conference on Computational Natural Language Learning*. [S.l.], 2006. p. 149–164.

FREITAS, C.; ROCHA, P.; BICK, E. Um mundo novo na floresta sintá (c) tica–o treebank do português. *Calidoscópio*, v. 6, n. 3, p. 142–148, 2008.

GRAY, J. Towards a genealogy of open data. 2014.

HOHFELD, W. N. *Os conceitos jurídicos fundamentais aplicados na argumentação judicial*. Tradução de Margarida Lima Rego. Avenida de Berna, Lisboa: Fundação Calouste Gulbenkian, 2008. 192 p.

LIMA, J. A. d. O. Apuração do texto original da lei geral de orçamento (lei n. 4.320/64): um estudo de caso sobre a acurácia de bases de dados de legislação federal. *Boletim de Direito Administrativo*, NDJ, 2013.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, BMJ Group BMA House, Tavistock Square, London, WC1H 9JR, v. 18, n. 5, p. 544–551, 2011.

NAKAMURA, M.; OGAWA, Y.; TOYAMA, K. Extraction of legal definitions from a japanese statutory corpus–toward construction of a legal term ontology. In: *Proceedings of 2013 Law via the Internet Conference*. [S.l.: s.n.], 2013. p. 11.

NIVRE, J. et al. Universal dependencies v1: A multilingual treebank collection. In: *LREC*. [S.l.: s.n.], 2016.

RIBEIRO, C. J. S.; PEREIRA, D. V. A publicação de dados governamentais abertos: proposta de revisão da classe sobre previdência social do vocabulário controlado do governo eletrônico. *Transinformação*, Pontifícia Universidade Católica de Campinas, v. 27, n. 1, 2015.

RINGGAARD, M.; GUPTA, R.; PEREIRA, F. C. N. SLING: A framework for frame semantic parsing. *CoRR*, abs/1710.07032, 2017. Disponível em: <http://arxiv.org/abs/1710.07032>.

SANTOS NETO, A. L. dos et al. Tecnologias de dados abertos para interligar bibliotecas, arquivos e museus: um caso machadiano. *Transinformação*, SciELO Brasil, v. 25, n. 1, 2013.

VICTORINO, M. de C. et al. Uma proposta de ecossistema de big data para a análise de dados abertos governamentais concetados. *Informação & Sociedade*, Universidade Federal da Paraíba-Programa de Pós-Graduação em Ciência da Informação, v. 27, n. 1, 2017.

Reconhecimento de entidades no auxílio à indexação de textos legislativos brasileiros

Daniel de Mello Viero* João Alberto de Oliveira Lima
Lauro César Araujo Fabrício Fernandes Santana
Flávio Roberto de Almeida Heringer Hudson de Martim
Jideão José Vieira Filho Wagner Rodrigues Teixeira

Resumo

As leis federais brasileiras, assim como diversos outros tipos de textos legais e jurídicos, são catalogados pelo Senado Federal, onde são indexados de forma manual por especialistas. A questão central deste estudo foi identificar formas de auxiliar esse processo de indexação utilizando técnicas modernas de processamento de linguagem natural. Para tanto, todo o texto da ementa e da parte articulada das leis brasileiras desde 1946 até meados de 2017 foi submetido ao processamento da *Google Cloud Natural Language Processing API* para análise sintática e o reconhecimento de entidades utilizando o idioma português brasileiro. A partir das entidades reconhecidas nos textos, comparou-se a coincidência dessas expressões com os termos de indexação definidos para essas normas e com os termos presentes no tesauro do Senado Federal. Pelos resultados dessa avaliação, chegou-se à conclusão de que o uso dessa tecnologia pode ser útil como mecanismo de sugestão de termos para indexação das normas e para inclusão de novos termos no tesauro da instituição.

Palavras-chave: Texto legislativo. Processamento de linguagem natural. Reconhecimento de entidades. Tesauro.

Introdução

Diariamente, uma equipe especializada do Senado Federal cataloga individualmente todos os textos legais da esfera federal brasileira publicados no Diário Oficial da União e nos Diários do Senado Federal e do Congresso Nacional. Os textos incluem emendas constitucionais, leis complementares, leis

*daniel.viero@senado.leg.br

ordinárias, decretos legislativos, medidas provisórias, além das proposições legislativas em tramitação.

Uma das tarefas mais importantes para a gestão dessas normas e proposições é a indexação, ou seja, a especificação de relacionamentos com termos de um tesauro, visando facilitar classificação, navegação e localização. A existência dessa indexação viabiliza que as pessoas envolvidas no processo legislativo, bem como estudiosos e interessados nos textos normativos e propositivos em geral, encontrem com mais rapidez e precisão os documentos de que necessitam para a adequada realização de suas atividades.

O trabalho de escolha dos termos da indexação de um texto, embora realizado por meio de sistemas informatizados, depende do ser humano. Especialistas do Senado Federal analisam o conteúdo da norma ou proposição e associam a ela as palavras-chave relevantes, conforme sua experiência e práticas adotadas pelo setor. Esse trabalho, realizado de forma manual, é naturalmente propenso a falhas e variações inerentes a escolhas individuais, que porventura podem dificultar a localização de algum texto ou prejudicar o agrupamento de textos similares.

Conforme apresenta Polistchuk e Trinta (2003), "tecnologias permitem ao ser humano ampliar suas potencialidades, estender seus sentidos e controlar o meio natural e social em que vive". Especialmente, a tecnologia da informação possibilita organização e tratamento dos ativos do conhecimento. Dessa forma, é oportuno aproveitar recursos da computação para auxiliar o trabalho de indexação dos textos legislativos e jurídicos realizados no Senado Federal, uma vez que esta é naturalmente uma atividade de e sobre ativos de conhecimento.

Considerando que os textos legislativos são construídos em linguagem não estruturada, ainda que obedeçam às técnicas de redação legislativa e às normas de articulação desse tipo de texto, uma das áreas tecnológicas com maior potencial para contribuir na extração de informação a partir deles é o processamento de linguagem natural. O reconhecimento de entidades é uma das tarefas do processamento de linguagem natural que visa localizar expressões do texto que referem-se a elementos identificáveis, geralmente nomes próprios, e classificá-los em categorias predefinidas como pessoas, lugares ou organizações (NADEAU; SATOSHI, 2007; SANTOS; CARDOSO, 2007). Com base nisso, a hipótese que se levantou foi que o reconhecimento de entidades pode auxiliar, ou complementar, a indexação manual dos textos jurídicos no Senado Federal.

Este trabalho propôs-se, então, a fazer um estudo exploratório, com base no histórico de indexações já existentes, para verificar se a utilização de técnicas de reconhecimento de entidades seria útil no auxílio a futuras indexações de textos legislativos e jurídicos. A expectativa era de que existisse

um número grande de coincidências entre as entidades identificadas no texto e os termos indexados pelo Senado Federal, especialmente considerando as entidades presentes na ementa da norma e aquelas mais frequentes no corpo do texto.

Como método, foram utilizados serviços de processamento de linguagem natural do Google para o idioma português, baseados em inteligência artificial, para reconhecer as entidades no texto articulado e na ementa das leis federais brasileiras. Também foi desenvolvido um algoritmo para identificar as coincidências entre essas entidades e os termos utilizados como indexadores da norma ou presentes no tesauro do Senado Federal.

A seguir, na seção 1, é apresentada a tarefa do Reconhecimento de Entidades como parte do processamento de linguagem natural. A seção 2 detalha todos os passos do estudo exploratório, desde seu planejamento até a totalização dos dados coletados. Na seção 3 esses dados são apresentados, e sua interpretação é discutida na seção 4. As conclusões do estudo são apresentadas na seção 5. Por fim, a seção 6 esclarece algumas limitações deste estudo e propõe trabalhos futuros.

1 Reconhecimento de entidades

Segundo Nadeau e Satoshi (2007), o termo "named entity" foi cunhado para a sexta edição da *Message Understanding Conference* (MUC-6), de 1995. O foco do evento foi a extração de informações (*Information Extraction - IE*) e percebeu-se na ocasião a importância de reconhecer informação como nomes – de pessoas, organizações, lugares – e expressões numéricas – horários, datas, dinheiro, percentuais. Essa tornou-se uma importante subtarefa da extração de informações denominada "*Named Entity Recognition and Classification*" (NERC). A tarefa foi traduzida como "Reconhecimento de Entidades Mencionadas" (REM) pela organização do I HAREM (SANTOS et al., 2006), um evento que propôs desafios relacionados ao reconhecimento de entidades em português e estabeleceu o primeiro marco de estado da arte na área para essa língua. Segundo Santos e Cardoso (2007, p. 3):

> "Entidades mencionadas" (EM) foi a nossa tradução (ou melhor, adaptação) do conceito usado em inglês, *named entities*, e que literalmente poderá ser traduzido para "entidades com nome próprio". A tarefa que nos propusemos avaliar era a de reconhecer essas entidades, atribuindo-lhes uma classificação (dentre um leque de categorias previamente definido e aprovado por todos) que representaria o significado daquela ocorrência específica da entidade no texto em questão.

Eles demonstraram a "convicção de que o REM é parte integrante da maioria dos sistemas inteligentes que processam e interpretam a língua, tais como sistemas de extracção de informação, de resposta automática a perguntas, de tradução automática, ou de sumarização de textos". Segundo eles, "a qualidade do REM nestes sistemas influencia decisivamente o seu resultado final" (ibidem).

São muitas as técnicas utilizadas para reconhecer entidades em textos, partindo desde a utilização de expressões regulares e regras baseadas em análise sintática, até aquelas que utilizam sofisticados mecanismos de inteligência artificial. Nadeau e Satoshi (2007) apresentam um levantamento dessas técnicas até 2006, enquanto Miranda et al. (2011) analisam algumas alternativas baseadas em aprendizagem de máquina. A oferta de opções para reconhecimento de entidades em língua portuguesa é bem mais modesta do que para o inglês, especialmente em termos de eficácia. Amaral et al. (2014) apresentam um comparativo entre algumas dessas opções.

Recentemente a Google se aproveitou de sua imensa base de conteúdo multilíngue e treinou seu mecanismo de inteligência artificial para realizar análise sintática e reconhecimento de entidades em alguns idiomas, entre os quais o português brasileiro. A partir desse trabalho, a Google disponibilizou em sua plataforma em nuvem a *Google Cloud Natural Language API*[1] (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011).

A funcionalidade é apresentada como um conjunto de serviços web/REST que utilizam modelos e estratégias de aprendizado de máquina para realizar análise avançada de textos. Dentre os serviços oferecidos por essa *API*, está a análise sintática e a análise de entidades. A primeira extrai do texto *tokens* e frases, identifica classes gramaticais PoS ("*part of speech*"), e cria uma árvore de análise sintática para cada frase. Já a segunda identifica entidades que aparecem no texto e as classifica como pessoa, organização, local, eventos, produtos, mídia.

2 Execução do estudo exploratório

A execução do estudo descrito neste artigo foi organizada nas seguintes etapas:

a) delimitação do conjunto de normas a ser estudado;

b) preparação do conjunto de textos;

c) identificação das entidades presentes em cada texto;

[1] A API de Processamento de Linguagem Natural da Google para o português brasileiro encontra-se em <https://cloud.google.com/natural-language/?hl=pt-br>, acesso de 15.mar.2018.

d) obtenção dos termos de indexação de cada norma e do vocabulário controlado completo;

e) normalização dos termos e verificação das coincidências;

f) agregação dos resultados.

Cada uma dessas etapas é detalhada nas seções a seguir.

2.1 Delimitação do conjunto de normas a ser estudado

Como escopo do estudo, foram selecionadas as leis federais publicadas a partir de 4 de outubro de 1946, quando passou a vigorar a padronização no formato dos textos normativos brasileiros, advinda da "Constituição de 1946", instituída pelo "Estado Novo", ou "Terceira República", do então presidente Getúlio Vargas. Assim, a primeira norma processada é justamente a Lei nº 1, de 1946. A norma mais recente considerada foi a Lei nº 13.435, de 12 de abril de 2017.

Abaixo é apresentado o texto completo da Lei nº 13.416, de 23 de fevereiro de 2017, que foi uma das normas analisadas, com estrutura típica, e será utilizado como exemplo nas próximas seções.

Exemplo 1 – Norma completa

LEI Nº 13.416, DE 23 DE FEVEREIRO DE 2017.

Autoriza o Banco Central do Brasil a adquirir papel-moeda e moeda metálica fabricados fora do País por fornecedor estrangeiro.

O PRESIDENTE DA REPÚBLICA Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:

Art. 1º Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda metálica fabricados fora do País por fornecedor estrangeiro, com o objetivo de abastecer o meio circulante nacional, observado o disposto na Lei no 8.666, de 21 de junho de 1993.

Parágrafo único. As aquisições referidas no caput obedecerão a cronograma fixado pelo Banco Central do Brasil para cada exercício financeiro, observadas as diretrizes estabelecidas pelo Conselho Monetário Nacional.

Art. 2º A inviabilidade ou fundada incerteza quanto ao atendimento, pela Casa da Moeda do Brasil, da demanda por meio circulante ou do cronograma para seu abastecimento, em cada exercício financeiro, caracteriza situação de emergência, para efeito de aquisição de papel-moeda e de moeda metálica de fabricantes estrangeiros, na forma do inciso IV do caput do art. 24 da Lei nº 8.666, de 21 de junho de 1993.

§ 1º Caracterizam a inviabilidade ou fundada incerteza de que trata o caput:

I - o atraso acumulado de 15% (quinze por cento) das quantidades contratadas, por denominação, de papel-moeda ou de moeda metálica; e

II - outras hipóteses de descumprimento de cláusula contratual, devidamente justificadas, que tornem inviável o atendimento da demanda por meio circulante ou do cronograma para seu abastecimento.

§ 2º Para fins da caracterização da situação de emergência de que trata este artigo, o Banco Central do Brasil fica obrigado a enviar o Programa Anual de Produção

à Casa da Moeda do Brasil, até 31 de agosto de cada ano, no qual serão indicadas as projeções de demandas de papel-moeda e de moeda metálica para o exercício financeiro seguinte.

Art. 3º Esta Lei entra em vigor na data de sua publicação.

Brasília, 23 de fevereiro de 2017; 196º da Independência e 129º da República.

MICHEL TEMER

Ilan Goldfajn

2.2 Preparação do conjunto de textos

O Senado Federal já havia processado o texto de todas as leis federais com o parser do projeto LexML[2], que identifica as partes do texto legal a partir de padrões dentro dos documentos originais (em geral nos formatos RTF ou DOCX) e os organiza em arquivos estruturados no formato XML.

Nessa estrutura, foi possível obter separadamente a epígrafe, a ementa, o preâmbulo, a parte articulada (rótulos e textos de títulos, capítulos, seções, artigos, parágrafos, alíneas e incisos) e o fecho de cada norma, conforme exemplo a seguir.

Exemplo 2 – Norma estruturada pelo parser do LexML

```
<LexML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.lexml.gov.br/1.0"
  xsi:schemaLocation="http://www.lexml.gov.br/1.0 ../xsd/lexml-br-rigido.xsd">
<Metadado>
  <Identificacao URN=
    "urn:lex:br:senado.federal:lei:2017;13416@data.evento;leitura;2017-04-23t10.02"/>
</Metadado>
<ProjetoNorma>
  <Norma>
    <ParteInicial>
      <Epigrafe id="epigrafe"/>
      <Ementa id="ementa">
        <b><span xlink:href="urn:lex:br:federal:lei:2017-02-23;13416">
          LEI Nº 13.416, DE 23 DE FEVEREIRO DE 2017
        </span></b><b><i>
          Autoriza o Banco Central do Brasil a adquirir papel-moeda e moeda
          metálica fabricados fora do País por fornecedor estrangeiro.
        </i></b>
      </Ementa>
      <Preambulo id="preambulo">
        <p>O PRESIDENTE DA REPÚBLICA</p>
        <p>
          Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:
        </p>
      </Preambulo>
    </ParteInicial>
```

[2] A documentação do esquema LexML se encontra em <http://projeto.lexml.gov.br/documentacao/Parte-3-XML-Schema.pdf>, acesso de 11.mar.2018

```
<Articulacao>
  <Artigo id="art1">
    <Rotulo>Art. 1º</Rotulo>
    <Caput id="art1_cpt"><!-- Link: urn:lex:br:federal:lei:1993-06-21;8666 -->
      <p><b>Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e
      moeda metálica fabricados fora do País por fornecedor estrangeiro, com o
      objetivo de abastecer o meio circulante nacional, observado o disposto na
      <span xlink:href="urn:lex:br:federal:lei:1993-06-21;8666">Lei nº 8.666,
      de 21 de junho de 1993</span>.</b></p>
    </Caput>
    <Paragrafo id="art1_par1u">
      <Rotulo>Parágrafo único.</Rotulo>
      <p><b>As aquisições referidas no caput obedecerão a cronograma fixado
      pelo Banco Central do Brasil para cada exercício financeiro, observadas
      as diretrizes estabelecidas pelo Conselho Monetário Nacional.</b></p>
    </Paragrafo>
  </Artigo>
  <Artigo id="art2">
  ...
```

Algumas das normas definidas no escopo deste estudo estavam ausentes do conjunto de dados gerado pelo parser LexML por problemas técnicos diversos que impediram seu processamento. Estiveram disponíveis para o estudo 12.971 das normas a serem estudadas. Considerou-se que a quantidade faltante, de cerca de 500 leis, não prejudicou a observação.

A partir da versão estruturada de cada norma, foi necessário extrair apenas os elementos textuais da ementa e do corpo do texto, que são os trechos que têm relação semântica com o seu conteúdo, ou seja, cuja análise pode contribuir para a indexação da norma. Descartaram-se assim a epígrafe, o preâmbulo e o fecho da norma, bem como os rótulos e numeração da sua parte articulada, que constituem elementos meramente estruturais ou meta-informação.

Gerou-se então um arquivo em formato texto puro para a ementa e outro para o corpo da norma. Nesse último, foram concatenados os textos de todos os dispositivos da parte articulada da norma, de modo que cada artigo, parágrafo, inciso ou alínea fique em uma linha do arquivo texto, como se fosse um texto não articulado.

Exemplo 3 – Arquivo de ementa em texto puro

> Autoriza o Banco Central do Brasil a adquirir papel-moeda e moeda metálica fabricados fora do País por fornecedor estrangeiro.

Exemplo 4 – Arquivo de corpo da norma em texto puro

> Fica autorizado o Banco Central do Brasil a adquirir papel-moeda e moeda metálica fabricados fora do País por fornecedor estrangeiro, com o objetivo de abastecer o meio circulante nacional, observado o disposto na Lei no 8.666, de 21 de junho de 1993.

As aquisições referidas no caput obedecerão a cronograma fixado pelo Banco Central do Brasil para cada exercício financeiro, observadas as diretrizes estabelecidas pelo Conselho Monetário Nacional.

A inviabilidade ou fundada incerteza quanto ao atendimento, pela Casa da Moeda do Brasil, da demanda por meio circulante ou do cronograma para seu abastecimento, em cada exercício financeiro, caracteriza situação de emergência, para efeito de aquisição de papel-moeda e de moeda metálica de fabricantes estrangeiros, na forma do inciso IV do caput do art. 24 da Lei nº 8.666, de 21 de junho de 1993.

Caracterizam a inviabilidade ou fundada incerteza de que trata o caput:

o atraso acumulado de 15% (quinze por cento) das quantidades contratadas, por denominação, de papel-moeda ou de moeda metálica; e

outras hipóteses de descumprimento de cláusula contratual, devidamente justificadas, que tornem inviável o atendimento da demanda por meio circulante ou do cronograma para seu abastecimento.

Para fins da caracterização da situação de emergência de que trata este artigo, o Banco Central do Brasil fica obrigado a enviar o Programa Anual de Produção à Casa da Moeda do Brasil, até 31 de agosto de cada ano, no qual serão indicadas as projeções de demandas de papel-moeda e de moeda metálica para o exercício financeiro seguinte.

Esta Lei entra em vigor na data de sua publicação.

2.3 Identificação das entidades

Os arquivos de ementa e texto das normas selecionadas foram submetidos, um a um, ao processamento da *Google Cloud Natural Language API*, versão 1-beta2, que era a mais recente no momento do estudo.

O processamento foi feito por meio de chamadas HTTP no estilo REST, por meio de *scripts* de linha de comando, utilizando-se uma conta de usuário previamente cadastrado na Google Cloud.[3]

O resultado das chamadas foi um arquivo no formato JSON para cada texto submetido, contendo a sua segmentação em sentenças e em *tokens*, a sua análise morfo-sintática e o resultado do reconhecimento e classificação de entidades pelo mecanismo de inteligência artificial do serviço. O formato dessas respostas está documentado em Google Cloud Platform (2017).

Exemplo 5 – Trecho do JSON retornado pela Natural Language API

```
{ "sentences": [ ... ],
  "tokens": [ ... ],
  "entities": [
    {
      "name": "Casa da Moeda do Brasil",
      "type": "ORGANIZATION",
      "metadata": {
        "mid": "/m/0764lm",
        "wikipedia_url":
            "https://en.wikipedia.org/wiki/Casa_da_Moeda_do_Brasil"
```

[3] À época desta etapa do trabalho, cada novo usuário do ambiente de nuvem do Google recebeu US$ 300 em créditos de livre utilização nos serviços da plataforma, o que foi suficiente para realizar com sobras todo o processamento deste estudo.

```
      },
      "salience": 0.010533761,
      "mentions": [
        {
          "text": {
            "content": "Casa da Moeda do Brasil",
            "beginOffset": 526
          },
          "type": "PROPER"
        }
      ]
    },
    {...}
  ],
  "language": "pt-BR"
}
```


Uma das informações interessantes retornadas pela API é um índice denominado "saliência", que indica a frequência relativa com que aquela entidade aparece no texto. Além disso, as entidades são classificadas em categorias predefinidas, como pessoa, localidade, organização, evento e obra de arte, entre outras.

O objetivo de processar a ementa e o corpo do texto individualmente foi poder analisá-los em separado, dada a hipótese de que o texto das ementas seria mais relevante para a indexação do que o corpo completo da norma. Além disso foi possível obter o índice de saliência de cada entidade dentro daquele escopo separadamente, o que acreditou-se que poderia gerar melhores possibilidades de análise.

2.4 Obtenção dos termos de indexação e do vocabulário controlado completo

A indexação das normas, no sistema do Senado, não é feita pela associação direta aos termos do vocabulário, mas a objetos de mais alto nível que representam pessoas, organizações, locais, conceitos, entre outros. Esses, por sua vez, são associados aos termos. Os termos, por sua vez, podem ter sigla e sinônimos. Assim, foi necessário montar uma relação de todas as expressões textuais que podem corresponder aos indexadores de cada norma para que se ampliasse a probabilidade de encontrar o indexador com base nas entidades identificadas no texto.

Por exemplo, o tesauro do Senado Federal contém um registro referente à organização "Banco Central do Brasil (BACEN)", uma autarquia federal. Essa entrada tem associada a si um termo (expressão textual) "Banco Central do Brasil", que por sua vez tem a sigla "BACEN". Para uma norma que tem esta organização como um de seus indexadores, as três expressões textuais são associados à norma e considerados no momento de verificar as coincidências com as entidades reconhecidas no texto. Da mesma forma, o

conceito “papel moeda” possui um termo alternativo “cédula monetária”, sendo ambas as expressões consideradas equivalentes.

Além disso, para analisar a possibilidade de sugestão de novos termos para o tesauro do Senado, foi necessário conhecer todos os termos já cadastrados, mesmo que não tenham sido ainda utilizados como indexadores. Para tanto, todo o vocabulário, contendo um total de 38.642 termos, foi considerado na análise.

2.5 Normalização dos termos e verificação das coincidências

O passo seguinte do processamento foi percorrer todas as normas e, para cada uma delas, verificar quais das entidades reconhecidas pela *Google Natural Language API* coincidiam com alguma das expressões textuais relacionadas aos indexadores da norma. Essa etapa foi executada por meio de um programa desenvolvido na linguagem Java.

Nesse momento não foi suficiente realizar uma mera comparação textual. Havia diferenças e limitações em ambos os lados da operação, como a diferença entre maiúsculas e minúsculas, ausência de acentuação ou agrupamento entre sigla e nome de organizações, que faziam com que não se identificassem correspondências claras. Necessitou-se, por isso, normalizar os termos antes da comparação, aplicando-se os seguintes filtros:

a) desconsiderar acentuação e caixa do texto. Por exemplo, “Publicação” equivale a “publicacao”;

b) extrair a sigla e a denominação completa de uma entidade ou termo, quando aparecem separados com travessão ou parênteses. Por exemplo, “Ministério da Saúde - MS” gera mais dois termos: “Ministério da Saúde” e “MS”. A expressão “Imposto de Renda (IR)” expande-se em “Imposto de Renda” e “IR”;

c) desconsiderar hífen separador de substantivos compostos. Por exemplo: “papel-moeda” equivale a “papel moeda” (uma única expressão, mas sem considerar o hífen na comparação);

d) desconsiderar apóstrofos, pontuações e outros caracteres especiais.

Para cada norma, identificaram-se os indexadores em que algum dos termos coincidiram com alguma das entidades reconhecidas na ementa ou no corpo do texto. Identificaram-se também os termos do vocabulário aos quais cada uma das entidades reconhecidas pode ser associada com base na comparação textual.

O resultado do processamento de cada norma foi armazenado de forma estruturada em um arquivo no formato JSON, contendo todos os dados necessários para a análise “offline” posterior.

Exemplo 6 – Trecho do resultado do processamento em formato JSON


```
{
    "id": 17648375,
    "descricao": "Lei nº 13.416 de 23/02/2017",
    "ano": 2017,
    "tamanhoSentencas": 6,
    "tamanhoTokens": 296,
    "indexadores": [
        {
            "nome": "CEDULA MONETARIA, PAPEL MOEDA",
            "termoSigen": {
                "s": "UP",
                "texto": "CEDULA MONETARIA",
                "n": "1",
                "id": 255947,
                "p": "Objeto > Objeto Social > Conceito"
            },
            "encontradoNaEmenta": true,
            "encontradoNoCorpo": true
        },
    ],
        ...
    "entidades": [
        {
            "id": "papel-moeda",
            "textos": "papel-moeda",
            "salienciaEmenta": 0.17949101,
            "salienciaCorpo": 0.0759014166,
            "encontradoNaEmenta": true,
            "encontradoNoCorpo": true,
            "indexador": true,
            "vocabulario": true,
            "termosSigen": [
                {
                    "s": "UP",
                    "texto": "CEDULA MONETARIA",
                    "n": "1",
                    "id": 255947,
                    "p": "Objeto > Objeto Social > Conceito"
                }
            ],
            "tipoEntidade": "OTHER",
            "tipoSubstantivo": "COMMON"
        },
        ...
}
```


2.6 Agregação dos resultados

Para realizar totalizações e agregações, os dados comparativos foram importados para uma base de dados NoSQL MongoDB, por tratar de forma nativa os dados gerados no formato JSON. Os agrupamentos foram executados usando a linguagem de consulta da ferramenta.

Foram realizadas as contagens e percentuais apresentados na seção 3 Resultados.

3 Resultados

Foram processados com sucesso os textos de 12.964 normas, de um total de 13.566 leis federais publicadas no período abordado.

Em média, o corpo de cada norma analisada tem 12 sentenças e 539 *tokens*. As maiores normas são a Lei 5.869 de 11/01/1973 (Código de Processo Civil), que tem 1.986 sentenças, com 66.074 *tokens* no total, e a Lei 11.907 de 02/02/2009 (que trata das carreiras da Administração Pública Federal), com um total de 76.255 *tokens* em 1.299 sentenças.

Cada norma possui em média 9 termos de indexação, sendo que o maior número é de 165 indexadores para a mesma norma (Lei 9.069 de 29/06/1995, conversão da Medida Provisória que estabeleceu o Plano Real).

O processamento da *Google Natural Language API* resultou no reconhecimento de 904.897 entidades no total (69,8 por norma, em média), entre ementas e corpos das normas. A Tabela 1 apresenta a distribuição dessas entidades e os índices de coincidências observadas com relação ao tesauro do Senado e aos indexadores das normas.

Tabela 1 – Quantidade de entidades reconhecidas e índices de coincidência.

	Total	% Tesauro	% Indexadores
Corpo da norma	865.419	51,36%	4,98%
Ementa	93.798	48,87%	19,09%
Total	904.897	51,05%	5,33%

Fonte – os autores.

O índice de saliência de cada entidade, calculado pelo mecanismo do Google, identifica a frequência relativa da entidade no texto em questão. A Tabela 2 ilustra o índice de coincidência com os termos do tesauro e os indexadores considerando-se apenas as entidades mais frequentes do corpo de cada norma, mas desconsiderando as entidades já presentes na ementa.

Além de reconhecer as entidades, a API da Google atribui uma classificação a cada uma delas. A Tabela 3 mostra a frequência de cada tipo de entidade e o seu índice de coincidência com os indexadores da norma e com os termos do tesauro.

Todas as entidades categorizadas como "UNKNOWN" (desconhecido) são também identificadas na análise morfológica como nomes próprios.

Com relação aos indexadores, foram identificados no total 116.524 termos de indexação utilizados. Desses, 15,27% coincidiram com alguma

Tabela 2 – Índices de coincidência considerando somente as entidades mais frequentes do corpo de cada da norma.

Entidades por norma	Total	% Tesauro	% Indexadores
1	12.956	36,87%	11,25%
2	25.888	38,96%	10,70%
3	38.786	41,01%	10,12%
4	51.501	42,47%	9,54%
5	64.084	43,33%	9,08%

Fonte – os autores.

Tabela 3 – Quantidade de entidades reconhecidas e índices de coincidência por tipo de entidade.

Tipo	Quantidade	% Tesauro	% Indexadores
OTHER	496.887	42,73%	3,69%
EVENT	143.978	76,92%	6,74%
ORGANIZATION	87.880	49,33%	10,38%
PERSON	74.823	40,46%	3,35%
LOCATION	58.597	68,07%	6,10%
UNKNOWN	24.465	63,65%	17,72%
CONSUMER_GOOD	10.527	34,24%	4,81%
WORK_OF_ART	7.740	79,10%	0,89%

Fonte – os autores.

entidade reconhecida na ementa da respectiva norma, enquanto 33,99%, com entidade presente no corpo dos seus dispositivos. Em 12,66% dos casos a entidade estava presente em ambos os trechos da norma. Considerando a ementa e o corpo da norma juntos, o índice de indexadores que coincidiram com entidades reconhecidas foi de 36,61%.

4 Discussão

Embora o índice de termos de indexação encontrados entre as entidades reconhecidas seja maior que 33%, a quantidade de entidades não coincidentes com nenhum termo de indexação é bastante alto, em especial no corpo da

norma. Isso significa que são necessárias muitas sugestões para se atingir esse percentual de aproveitamento, o que talvez não traga a facilidade esperada ao especialista que cataloga a norma.

Com base na Tabela 1, é possível considerar que, em média, para cada 5 termos sugeridos com base nas entidades da ementa, um seria selecionado, enquanto 20 sugestões seriam necessárias para que uma fosse aceita com base na totalidade das entidades do corpo do texto. Se as sugestões de indexação forem restritas às entidades mais frequentes no corpo da norma e que não estavam na ementa, o aproveitamento dos termos sugeridos fica próximo aos 10%, como demonstra a Tabela 2.

Percebe-se, com isso, que a eficiência das sugestões de indexação com base nas entidades da ementa seria cerca de quatro vezes maior do que baseada na totalidade das entidades do corpo do texto, ainda que seu alcance seja de apenas 15% dos termos de indexação. Ao se ampliarem as sugestões baseadas na ementa acrescentando os cinco termos mais frequentes no corpo do texto, a eficiência média cai, já que as novas sugestões teriam cerca de 9,08% de aproveitamento, mas amplia-se o potencial de alcance dos termos de indexação.

A classificação feita pela Google Natural Language API é precária, pois a grande maioria dos termos acaba por ser agrupada na categoria "OTHER" (outros). Ainda assim, observa-se que a categoria "ORGANIZATION" (organizações) possui índice de mais de 10% de correspondência com os indexadores das normas, e, por incrível que pareça, na categoria "UNKNOWN" (desconhecido), esse índice chega a mais de 17%. Assim, a associação a uma dessas categorias se torna bom indicativo de que o termo pode ser um indexador.

No caso da categoria "UNKNOWN", uma explicação provável para seu alto grau de coincidência com os indexadores é que todos os termos associados a ela são nomes próprios, ou seja, representam uma pessoa, instituição, evento ou outro tipo de objeto social nomeado que a API de processamento textual não conseguiu encaixar em alguma das outras categorias.

Cabe considerar ainda que existe uma quantidade significativa de entidades reconhecidas presentes no vocabulário mas não utilizadas como indexadores. Talvez esses sejam termos úteis para a indexação, mas que os analistas não costumem selecionar por questões operacionais. A sugestão desses termos poderia facilitar sua adoção e ampliar a quantidade de termos utilizados para cada norma, aumentando ainda mais o índice de eficiência das sugestões baseadas no reconhecimento de entidades do texto.

5 Conclusões

Conforme a discussão dos dados apresentada na seção anterior, pode-se concluir que o reconhecimento de entidades é uma tecnologia útil para criar mecanismo de sugestão de termos de indexação para os especialistas do Senado Federal que trabalham na catalogação de normas.

Em primeiro lugar, confirmou-se a hipótese de que as entidades encontradas na ementa de cada lei têm a capacidade de gerar melhores sugestões de indexação do que as entidades encontradas no corpo da norma, mas percebeu-se que essas sugestões alcançariam no máximo 15% dos indexadores utilizados atualmente.

Além disso, identificou-se que restringir as entidades do corpo da norma às mais frequentes naquele texto aumenta bastante a probabilidade de serem boas sugestões de termo de indexação.

Descobriu-se ainda que as entidades classificados na categoria "UNKNOWN" pelo mecanismo de processamento de linguagem natural do Google correspondem sempre a nomes próprios, e que também podem ser bons candidatos a termo de indexação.

Conclui-se, assim, que uma combinação desses critérios (entidades da ementa, entidades mais frequentes no corpo do texto e nomes próprios) pode ser um bom ponto de partida para a construção de um mecanismo de sugestão aos especialistas do Senado que trabalham na catalogação das textos legais. A incorporação desse recurso ao sistema informatizado já existente provavelmente resultaria em maior padronização e precisão, menores custos e menor esforço dos responsáveis pela tarefa.

6 Limitações e trabalhos futuros

Apesar dos resultados positivos, algumas limitações ou deficiências das ferramentas e dados utilizados precisam ser consideradas ao analisar os resultados obtidos.

O reconhecimento de entidades do Google Cloud NL API, por exemplo nem sempre está correto. Na Lei 13.416 de 2017, utilizada como exemplo este artigo, a entidade "Casa da Moeda do Brasil" foi reconhecida corretamente em uma de suas ocorrências, mas na outra foi agregada a uma entidade precedente, gerando equivocadamente a entidade "Programa Anual de Produção à Casa da Moeda do Brasil".

A classificação das entidades feita pelo serviço do Google também é imprecisa com frequência, como se percebe pela grande quantidade de expressões categorizadas como "UNKNOWN" e "OTHER". Além disso, na mesma lei já citada, a entidade "Banco Central do Brasil" é classificada cor-

retamente como "ORGANIZATION" em um caso, mas como "UNKNOWN" em outra ocorrência.

Em algumas situações, até mesmo a segmentação de sentença não é feita corretamente, como no caso da ementa da Lei 13.285 de 2016, que diz "Acrescenta o art. 394-A ao Decreto-Lei nº 3.689, de 3 de outubro de 1941 - Código de Processo Penal.". O analisador não processou corretamente a abreviação "art." e quebrou a sentença em duas logo após o ponto final dessa expressão. Esse tipo de equívoco provavelmente afetou o reconhecimento de entidades tanto nas ementas quanto no restante dos textos.

Existem também questões relacionadas ao próprio conteúdo do conjunto de textos analisado. Por exemplo, o trecho "Esta Lei entra em vigor na data de sua publicação.", ou outras cláusulas de vigência semelhantes, aparecem em uma quantidade muito grande de normas. Com isso, as entidades "Lei", "vigor" e "publicação" estão quase sempre presentes, sendo que duas delas – "Lei" e "publicação" – estão catalogadas no Sigen e apareceriam como sugestões com muito mais frequência do que o desejado.

Com relação às fontes de dados, existem também algumas imperfeições. Por exemplo, o arquivo XML gerado pelo *parser* do projeto LexML, em alguns casos, agrupou o fecho da norma junto com seu último artigo, especialmente em leis mais antigas. De forma semelhante, há casos em que o preâmbulo foi mantido junto com a ementa. Com isso, trechos que deveriam estar excluídos da análise acabaram entrando, fazendo surgir, por exemplo, nomes próprios dos signatários daquelas leis que são reconhecidos como entidades.

Por fim, identificaram-se ainda questões a serem aprimoradas no tesauro do Senado Federal. Algumas das normalizações de textos usadas na comparação de entidades com termos do vocabulário foram necessárias para contornar a ausência de termos alternativos, como foi o caso das expressões "papel-moeda" e "papel moeda". Outro caso é o nome das unidades da Federação, que estão catalogadas no tesauro como "Estado da Bahia", por exemplo, enquanto a expressão "Bahia" não aparece como termo alternativo e, por isso, não consegue ser associada à respectiva entidade reconhecida no texto quando aparece dessa forma.

Como trabalhos futuros, a implementação do mecanismo de sugestões no sistema informatizado do Senado Federal e a sua utilização pelos especialistas do Senado Federal é fundamental para se confirmar as conclusões deste trabalho.

Outro ponto de continuidade do estudo seria procurar combinações de tipos e classificações de entidades que tenham maiores índices de coincidência com os termos de indexação. Por exemplo, analisar a probabilidade de uma entidade ser indexador caso seja nome próprio e classificada como "ORGANIZATION", ou como "EVENT".

Além disso, vislumbram-se algumas alternativas para aperfeiçoar o algoritmo de associação entre entidades reconhecidas e termos do tesauro, como:

a) utilizar o *lemma* ou outra forma canônica de cada entidade e dos termos do vocabulário na comparação, de modo a se eliminarem as diferenças entre singular/plural e outros aspectos morfológicos;

b) Desambiguar os termos com base na ontologia passível de ser deduzida do vocabulário controlado/tesauro;

c) Considerar comparação fonética entre termos e entidades.

Por fim, uma oportunidade interessante seria repetir este estudo utilizando-se um outro mecanismo de reconhecimento de entidades nomeadas, para que se possa comparar com o Google NL API, utilizado neste estudo.

Referências

AMARAL, D. et al. Comparative Analysis of Portuguese Named Entities Recognition Tools. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Islândia: European Language Resources Association (ELRA), 2014. p. 2554–2558. ISBN 978-2-9517408-8-4.

GOOGLE Cloud Platform. *Princípios básicos da Natural Language API*. 2017. Disponível em: <https://cloud.google.com/natural-language/docs/basics>.

MIRANDA, N. et al. Named entity recognition using machine learning techniques. In: *EPIA-11, 15th Portuguese Conference on Artificial Intelligence*. Lisboa (Portugal): [s.n.], 2011.

NADEAU, D.; SATOSHI, S. A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, v. 30, n. 1, 2007. Disponível em: <http://nlp.cs.nyu.edu/sekine/papers/li07.pdf>.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, BMJ Group BMA House, Tavistock Square, London, WC1H 9JR, v. 18, n. 5, p. 544–551, 2011.

POLISTCHUK, I.; TRINTA, A. R. *Teorias Da Comunicação*: O pensamento e a prática da comunicação social. Rio de Janeiro, Brasil: Elsevier Editora Ltda, 2003. 184 p.

SANTOS, D.; CARDOSO, N. (Ed.). *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área.* 1. ed. Lisboa e Oslo: Linguateca, 2007. ISBN 978-989-20-0731-1.

SANTOS, D. et al. HAREM : an Advanced NER Evaluation Contest for Portuguese. In: *Proceedings of LREC'2006*. Gênova, Itália: [s.n.], 2006. p. 1986–1991.

Exemplo de Extração de Definições em Textos Articulados de Normas Jurídicas com o apoio do Processamento de Linguagem Natural

Wagner Rodrigues Teixeira*
João Alberto de Oliveira Lima
Lauro César Araujo
Daniel de Mello Viero
Fabrício Fernandes Santana
Flávio Roberto de Almeida Heringer
Hudson de Martim
Jideão José Vieira Filho

Resumo

As tecnologias de Processamento de Linguagem Natural baseadas em técnicas de Inteligência Artificial possibilitam a extração de definições de textos articulados de normas jurídicas de forma automática. O presente artigo exemplifica essa estratégia por meio da implementação de um processo de extração de definições que atendam a uma determinada fórmula linguística em um *corpus* com anotações morfossintáticas. A extração de conceitos em *corpus* normativos pode ter múltiplas aplicações, servindo, por exemplo, de subsídio para elaboração de glossários, tesauros ou ontologias de domínio.

Abstract

Technologies of Natural Language Processing based on Artificial Intelligence techniques allow the extraction of definitions of articulated texts of legal norms automatically. This article exemplifies this strategy by means of the implementation of a process of extracting definitions that meet a certain linguistic formula in a corpus with morphosyntactic annotations. The extraction of concepts in normative corpus can have multiple applications, for example, use of subsidies for the development of glossaries, thesauri, or domain ontologies.

Palavras-chave: Extração de definições. Legislação brasileira. Inteligência artificial.

*wagner@senado.leg.br

1 Introdução

A definição é um enunciado (*definiendum*) que explica o significado de um termo (*definiens*). Por exemplo, o art. 3º da Lei 8.112 de 1990 define que "Cargo público é o conjunto de atribuições e responsabilidades previstas na estrutura organizacional que devem ser cometidas a um servidor". Nessa definição, "Cargo Público" é o termo (*definiens*) e "o conjunto de atribuições e responsabilidades previstas na estrutura organizacional que devem ser cometidas a um servidor" é o enunciado (*definiendum*).

Atienza e Manero (1997, p. 68), na obra "*A Theory of Legal Sentences*", consideram a definição como um dos quatro tipos de disposições normativas que incluem ainda as normas mandatórias, as normas de competência e as normas puramente constitutivas. As *normas mandatórias* funcionam, na prática, como imperativos categóricos e, caso sejam seguidas, oferecem razões para agir conforme prescrito. Já as *normas de competência* e as *normas puramente constitutivas* funcionam como imperativos hipotéticos e também oferecem razões para agir caso o agente deseje criar, alterar ou modificar relações jurídicas. As *definições*, por sua vez, não oferecem razões para agir, mas critérios que habilitam o entendimento das normas. Dessa forma, nota-se que as definições desempenham um importante papel no entendimento do ordenamento jurídico de uma determinada jurisdição.

Em estudo específico sobre definições legislativas, Sgarbi (2007, p. 9-10) elenca os seguintes propósitos para a definição: eliminar ambiguidades; explicar algo; reduzir informações; influenciar atitudes e evitar repercussões emocionais. Esse mesmo autor (Ibid., p. 20) atribui à "definição legislativa" o seguinte significado: "todo enunciado que, sendo parte de um texto normativo (uma disposição) indica o significado de alguma expressão da linguagem jurídica, ou seja, o seu *definiendum*".

Registre-se ainda que o uso de definições em textos legislativos, principalmente no Direito Civil, não é pacífico. É conhecida a máxima presente no Digesto (50, 17, 202) e atribuída ao jurisconsulto Caio Prisco que assevera "*Omnis definitio in iure civili periculosa est; parum est enim, ut non subverti posset*" ("toda definição no direito civil é perigosa pois são raras as que não podem ser subvertidas"). Não obstante, as normas, que a cada dia passam a regular cada vez mais os diversos aspectos da vida social, são terrenos férteis para se colher definições.

O ordenamento jurídico federal brasileiro é formado por milhares de normas jurídicas e devido a esse volume a extração manual de definições seria uma tarefa dispendiosa. Este trabalho se insere na tentativa de auxiliar o gestor da informação jurídica a extrair definições de textos de normas jurídicas para apoiar a elaboração de sistemas de organização do conhecimento,

tais como, glossários, tesauros e ontologias.

Após uma visão geral da metodologia proposta, nas seções seguintes, detalha-se cada passo do processo de extração de definições. Na sequência, apresentam-se exemplos dos resultados obtidos com o método proposto, trabalhos correlatos e as considerações finais. Os códigos-fonte utilizados no processo proposto encontram-se como apêndices deste trabalho.

2 Metodologia

A metodologia proposta neste trabalho utiliza como entrada o *corpus* de leis federais estruturado e anotado com etiquetas morfossintáticas utilizando o *Google Natural Language Processing API*, conforme produzido por Martim et al. (2018). O processo de extração de definições proposto é formado por filtros executados de forma sucessiva em cinco passos. A Figura 1 ilustra a relação de cada etapa do processo:

a) Passo 1: Filtrar sentenças candidatas (subseção 2.1);

b) Passo 2: Isolar as Leis cujas sentenças não puderam ser produzidas pelo *parser* LexML (subseção 2.2);

c) Passo 3: Filtrar sentenças com o verbo "ser" na *tag* `ROOT` (subseção 2.3);

d) Passo 4: Retirar as leis autorizativas (subseção 2.4);

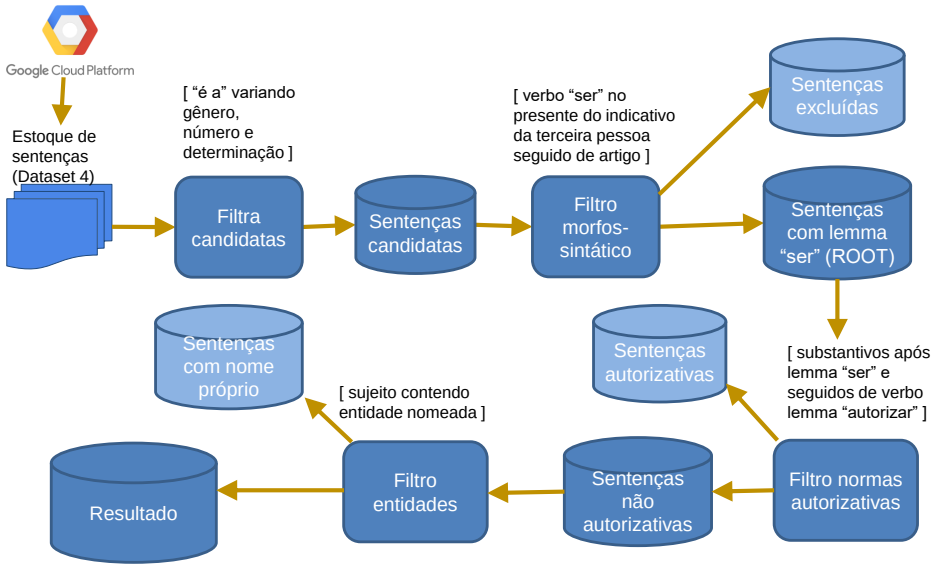
e) Passo 5: Remover definições de entidades nomeadas (subseção 2.5).

2.1 Passo 1: Filtrar sentenças candidatas

As definições legislativas aparecem de diversas formas no texto de normas jurídicas. Uma das mais usuais é a que combina a conjugação do verbo "ser" seguindo de um artigo, ambos no plural ou singular. Para os fins deste trabalho, isto é, exemplificar o processo de extração de definições tomando-se por base o corpus com anotações morfossintáticas, decidiu-se por implementar esse tipo específico de filtro. Para extrair todas as definições, seria necessário considerar também outras fórmulas textuais que ocorrem na legislação.

O *dataset* de sentenças em formato de texto puro, ainda sem marcações morfossintáticas, foi submetido ao filtro que seleciona todas as ocorrências da expressão "é a" com variações de gênero ("é o"), número ("são as", "são os") ou artigo determinante ("é um", "são umas"). Note-se que este passo não se beneficia das anotações morfossintáticas, servindo apenas para extrair sentenças candidatas de acordo com a ocorrência de determinadas expressões literais.

Figura 1 – Visão geral da metodologia Proposta



Fonte: Os autores

Por exemplo, a sentença do art. 1º da Lei 8.112 de 1990, "Esta Lei institui o Regime Jurídico dos Servidores Públicos Civis da União, das autarquias, inclusive as em regime especial, e das fundações públicas federais", não foi selecionada por não atender aos requisitos definidos no filtro de sentenças candidatas.

O código que implementa este filtro e a forma de como utilizá-lo são apresentados no Apêndice A.

2.2 Passo 2: Isolar as Leis cujas sentenças não puderam ser produzidas pelo *parser* LexML

A cadeia de transformação e processamento da informação original pode, eventualmente, não conseguir aplicar determinada norma a todas as etapas do *pipeline*.

As sentenças das normas são agrupadas em diretórios (pastas de arquivos) com arquivos individuais por sentença. As normas que não foram processadas com êxito até o presente estágio produzem diretórios de arquivos vazios, que devem ser retirados do grupo em processamento. A identificação das normas com algum erro é dada pela falha do comando da etapa anterior, conforme ilustrado a seguir:

```
grep: /opt/normas/txt-sentencas/1963/LEI-1963-04295/*:
No such file or directory}
```

O processo de filtragem nesta etapa é executado pelos seguintes comandos:

```
grep "^grep" defs0.v1.txt | sort > leisausentes.txt
grep -v "^grep" defs0.v1.txt | sort > defs0.v2.txt
```

A partir destes comandos são produzidos dois lotes de referências às leis: um contendo as leis que não estão prontas para a próxima etapa (em `leisausentes.txt`) e outro contendo as leis que estão prontas para a próxima etapa (em `defs0.v2.txt`).

2.3 Passo 3: Filtrar sentenças com o verbo "ser" na *tag* `ROOT`

O terceiro passo de filtragem é o primeiro a se beneficiar da etiquetagem morfossintática (*POS tagging*), que pode ser entendida como o processo pelo qual são atribuídas etiquetas a partes de um texto em linguagem natural, contendo indicação de sua função gramatical no contexto da frase (GÜNGOR, 2010, p. 205).

Os serviços do *Google Natural Language Processing API* permitem gerar, para cada sentença, um arquivo em formato JSON com estrutura específica contendo informações sobre *POS tagging*. Para o nosso propósito, analisamos em sequência os objetos no *array* chamado "tokens" procurando o verbo "ser" que seja o elemento raiz da sentença, na flexão de tempo presente do indicativo na terceira pessoa, tanto no plural quanto no singular. Uma vez localizada a palavra com tais características, ela deve ser seguida por um artigo.

O teste completo considera os parâmetros:

```
dependencyEdge.label == 'ROOT' &&
partOfSpeech.tag == 'VERB' &&
lemma == 'ser' &&
partOfSpeech.tense == 'PRESENT' &&
partOfSpeech.person == 'THIRD'
```

O termo subsequente é testado da seguinte forma para ser identificado como artigo:

```
partOfSpeech.tag == 'DET'
```

O programa que realiza o teste de identificação de sentença candidata, e o exemplo de como utilizá-lo, estão no Apêndice B.

2.4 Passo 4: Retirar as leis autorizativas

Algumas leis relativas ao orçamento público foram editadas utilizando-se em sua redação estrutura pouco convencional para o idioma português, impactando o critério adotado pela técnica apresentada neste artigo.

A estrutura oracional mais comumente adotada pelos idiomas, incluindo o Português, é chamada de Sujeito-Verbo-Objeto (SVO). No entanto, algumas leis que autorizam o Poder Executivo a executar determinada ação, particularmente abertura de créditos orçamentários, utilizam a estrutura conhecida por Verbo-Sujeito-Objeto (VSO), produzindo, de acordo com a técnica em exposição, falsos identificadores de definição de termos.

Tal estrutura oracional era mais comumente utilizada até o final da década de 1960, mas eventualmente foi utilizada até o ano 2011. Para exemplificar tal contexto, a Lei nº 4.741, de 15 de Julho de 1965, traz o seguinte texto no caput de seu artigo 7º:

> Para atender às despesas decorrentes desta Lei, no exercício de 1964, **é o** Poder Executivo autorizado a abrir ao Poder Judiciário - Justiça do Trabalho - Tribunal Regional do Trabalho da 1ª Região - o crédito especial de Cr$655.546.871 (seiscentos e cinqüenta e cinco milhões, quinhentos e quarenta e seis mil, oitocentos e setenta e um cruzeiros), que será registrado pelo Tribunal de Contas da União e distribuído ao Tesouro Nacional.

Por ser uma construção incomum no idioma, a rede neural utilizada não obteve sucesso em identificar o termo-raiz da frase no verbo autorizar, tendo identificado o verbo auxiliar como tal.

Identificou-se que todas as formações com a estrutura oracional VSO são autorizações, por esta razão o filtro desta etapa do *pipeline* separa do conjunto de sentenças selecionadas, o que chamamos de *leis autorizativas*.

O componente de software que realiza este filtro é chamado de `isautorizativo.js` e o script de controle de sua chamada com exemplo de utilização é chamado de `filterautorizativos.sh` e estão disponíveis no Apêndice C.

2.5 Passo 5: Remover definições de entidades nomeadas

Por fim, a última etapa de processamento do conjunto de Leis Federais brasileiras trata o caso de quando o texto define não um termo, mas uma entidade, ou que no sujeito da sentença contenha uma entidade nomeada, como por exemplo o caput do Art. 1º da Lei número 7.314, de 23 de maio de 1985 (grifos nossos):

> Os vencimentos e respectiva representação dos cargos do Ministério Público junto ao Tribunal de Contas do Distrito Federal **são os** constantes da Tabela anexa, mantidos os atuais direitos e vantagens.

Nota-se que no exemplo extraído acima, existe a identificação primordial do critério discutido para identificação de uma definição, que é o verbo ser seguido de artigo. No entanto, no sujeito, por conseguinte *definiendum*, ocorre o nome de uma entidade, o que torna a expressão a definição não de um termo, mas de uma entidade, que é objeto de outro estudo.

Os programas que executam esta etapa e exemplo de uso são apresentados no Apêndice D.

O resultado desta etapa final é uma lista das sentenças identificadas como definições de termos, gravada no arquivo `defs0.v5.txt`, resultado da execução do script de controle do programa que implementa este último filtro.

3 Resultados

A Tabela 1 contempla algumas definições obtidas com a aplicação do método descrito na seção 2 sobre o universo das leis federais pré-processadas por Martim et al. (2018).

Tabela 1 – Exemplos de definições extraídas pelo método proposto.

Definiendum	*Definiens*	Referência (Lei)
Os riscos cobertos pelo seguro de crédito à exportação	são os "riscos comerciais" e os "riscos políticos e extraordinários".	4.678/1965, art. 2°
Bravura	é o ato meritório que, ultrapassando o cumprimento do dever militar, é praticado com desprendimento e risco de vida.	5.020/1966, art. 42
Cargo público	é o conjunto de atribuições e responsabilidades previstas na estrutura organizacional que devem ser cometidas a um servidor.	8.112/1990, art. 3°
Para os fins deste artigo, valor originário	é o correspondente ao débito principal, com exclusão de quaisquer parcelas acessórias como juros, multa e correção monetária, bem assim de custas processuais e honorários advocatícios.	8.665/1993, art. 1º
O planejamento governamental	é a atividade que, a partir de diagnósticos e estudos prospectivos , orienta as escolhas de políticas públicas.	12.593/2012, art. 2º
Designer de interiores e ambientes	é o profissional que planeja e projeta espaços internos, visando ao conforto, à estética, à saúde e à segurança dos usuários, respeitadas as atribuições privativas de outras profissões regulamentadas em lei.	13.369/2016, art. 2º

Fonte – Produzido pelos autores.

4 Trabalhos relacionados

A empresa IBM desenvolveu a ferramenta GlossEx (PARK; BYRD; BOGURAEV, 2002) que permite a extração automática de termos candidatos de glossário utilizando um corpus com etiquetas morfossintáticas. Essa ferramenta consegue identificar formas variantes de termos de um mesmo conceito e oferece ainda, por meio de informações estatísticas, um ranking dos conceitos candidatos para o especialista da área realizar a análise pós-processamento.

A tese de doutoramento de Gaudio (2013) investigou a extração automática de definições por meio de um conjunto de heurísticas e métodos que foram testados em corpus da língua portuguesa, inglesa e holandesa. A

autora tratou de três formas de definições: cópula, verbais e definições de pontuação. A estratégia proposta aplica-se a corpus de qualquer área e em qualquer idioma.

No caso específico de extração de definições em textos articulados de normas jurídicas na língua portuguesa, Batista (2011) utilizou-se do processamento de linguagem natural com etiquetagem morfossintática para a definição de um conjunto de funções extratoras de características relevantes à tarefa de extração de definições. O pesquisador utilizou um corpus da área de Direito das Telecomunicações e conseguiu avaliar os seus resultados comparando-os com um glossário da área temática que havia sido previamente elaborado de forma manual.

5 Conclusão

As definições veiculadas em normas jurídicas desempenham um importante papel no entendimento do Direito. Normalmente posicionadas na parte inicial dos textos articulados, as definições jurídicas determinam como um conceito deve ser entendido no contexto da norma.

Este trabalho teve por objetivo exemplificar a extração de definições em textos articulados de normas jurídicas com apoio de técnicas de processamento de linguagem natural. A sistemática aqui proposta utilizou-se de filtros e identificou sentenças que seguiam uma determinada fórmula linguística.

O resultado da extração automática de definições em textos articulados de normas jurídicas deve ser revisto por um especialista e auxilia primordialmente na elaboração de sistemas de organização do conhecimento, tais como glossários, tesauros, taxonomias e ontologias.

Trabalhos futuros podem aperfeiçoar a estratégia adotada neste trabalho, aplicando mais elementos heurísticos de obtenção de definições, bem como treinar redes neurais a partir dos resultados obtidos.

Referências

ATIENZA, M.; MANERO, J. R. *A theory of legal sentences*. [S.l.]: Springer Science and Business Media, 1997.

BATISTA, H. *Extração Automática de Definições*: um estudo de caso em textos legislativos. Dissertação (Mestrado) — Universidade Católica de Brasília, 2011.

GAUDIO, R. del. *Automatic Extraction of Definition*. Tese (Doutorado) — Universidade de Lisboa, 2013.

GÜNGOR, T. Part-of-speech tagging. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.). *Handbook of Natural Language Processing*. [S.l.: s.n.], 2010. p. 205–235.

MARTIM, H. de et al. *Datasets da Base de Normas Jurídicas Brasileira*. 2018. Publicadas como Open Data no Figshare. Disponível em: <https://doi.org/10.6084/m9.figshare.c.4029253.v1>. Acesso em: 10 abr 2018.

PARK, Y.; BYRD, R.; BOGURAEV, B. K. Automatic glossary extraction: Beyond terminology identification. In: *Proceedings of the 19th International Conference on Computational Linguistics*. USA: [s.n.], 2002. v. 1.

SGARBI, A. Definições legislativas. *Direito, Estado e Sociedade*, n. 31, p. 6–32, jul/dez 2007.

APÊNDICE A – Filtro 0

O Código A.1 pode ser utilizado para filtrar o repositório de sentenças no formato texto utilizando expressões regulares para selecionar todas as sentenças que possuam "é o" e suas variações de gênero e número na terceira pessoa do indicativo. O resultado da chamada ao filtro deve produzir a primeira lista de referências às sentenças, no arquivo `defs0.v1.txt`.

```
./finddefs0.sh | sort > defs0.v1.txt
```

Código A.1 – Código do arquivo `finddefs0.sh`

```
#!/usr/bin/env bash

# comando para localizar os padrões de sequencia de string
# basicos para o criterio 0 - caracteristicas morfossintaticas.
#
# O termo central e' o verbo SER seguido de artigo.
#

# como o repositorio de leis contem tambem projetos, e' necessario
# que se faca um filtro que execute a pesquisa apenas nos textos das leis
```

```
BASE=/opt/normas/txt-sentencas
verbose=0
ano=""

help( )
{
    echo "$(basename ${0}):"
    echo "--ano {ano} - ano com 4 digitos. Default todos"
    echo "--verbose - mostra o arquivo entre colchetes sozinho na linha e
     ↪  nas linhas abaixo o conteudo"
    echo "--noverbose - mostra apenas o nome do arquivo - default"
}

while [ "${#}" -ge 1 ]
do
    case "${1}" in
    --ano)
        shift
        ano="${1}"
        ;;
    --verbose)
        verbose=1
        ;;
    --noverbose)
        verbose=0
        ;;
    --help|-h|-?|*)
        help
        exit 0
        ;;
    esac
    shift
done

find "${BASE}/${ano}" -type d -name "LEI*" -print | while read lei
do
    for f in ${lei}/*
    do
        if egrep "(( é (o|a|um|uma) )|( são (os|as|uns|umas) ))" ${f} 2>&1
         ↪  > /dev/null
        then
            if [ "${verbose}" -ne "0" ]
            then
                echo "[${f}]"
                cat ${f}
            else
                echo "${f}"
            fi
        fi
    done
done
```

B Testar a sentença processada pelo Google Compute Engine

Código B.1 – Código de `hasdefs.js`

```
#!/usr/bin/env nodejs

var fs = require( 'fs' );
var path = require( 'path' );
var conll = require( './conll' );
var gce = require( './gce' );

var printJSON = false;

function makeDef( o, tokenN, rule, fName )
{
    // refazer a getText para atuar apenas na sentenca referenciada (ver
    ↪  o.sentences[{text.content, text.beginOffset}]
    return( {
        "definiendum": conll.getText( o, 0, tokenN ),
        "definiens": {
            "text": conll.getText( o, tokenN, -1 ),
            "rule": rule,
            "src": fName
        }
    } );
}

function testJSONFile( fName )
{
    var o, tokenN;

    o = gce.load( fName );
    if( ( tokenN = gce.testForRule0( o ) ) >= 0 ) {
        if( printJSON )
            console.log( JSON.stringify( makeDef( o, tokenN, "linguistica",
            ↪  fName ) ) );
        return( 0 );
    }
    return( 1 );
}

var nArg;

for( nArg = 2; nArg < process.argv.length; ++nArg ) {
    switch( process.argv[ nArg ] ) {
    case "--json":
        printJSON = 1;
        break;
```

```
    case "--help":
    case "-h":
    case "-?":
        console.log( "%s [--json] gce.json", path.basename( process.argv[ 1
        ↪  ] ) );
        console.log( "--json - imprime o registro da definicao na saida
        ↪  padrao (default nao)" );
        console.log( "gce.json = arquivo JSON de resultado do POS tagging
        ↪  de sentencas da Google Compute Engine." );
        process.exit( 3 );

    default:
        process.exit( testJSONFile( process.argv[ nArg ] ) );
        break;
    }
}

/*
    Codigos de retorno:
        0: arquivo possui uma definicao
        1: arquivo nao possui uma definicao
        2: arquivo nao informado
        3: apenas imprimir ajuda e encerrar
*/
process.exit( 2 );
```

Arquivo `filterdefs0.sh` (Código B.2) deve ser chamado da forma:

```
./filterdefs0.sh > defs0.v3.txt
```

Código B.2 – Código de `filterdefs0.sh`

```
#!/usr/bin/env bash

# comando para localizar os padrões de sequencia de string
# basicos para o criterio 0 - caracteristicas morfossintaticas.
#
# O termo central e' o verbo SER seguido de artigo.
#

# como o repositorio de leis contem tambem projetos, e' necessario
# que se faca um filtro que execute a pesquisa apenas nos textos das leis

verbose=0
database="defs0.v2.txt"
firstline=1
linecount=0

help( )
```

```
{
    echo "$(basename ${0}):"
    echo "--db {arquivo com lista das sentencas} - os arquivos apontados
     ↪  devem"
    echo "            conter as sentencas em txt originadoras das base
     ↪  JSON."
    echo "            Default: \"${database}\""
    echo "--start {n} - linha inicial de processamento [1..n]"
    echo "            Default: \"${firstline}\""
    echo "--count {n} - quantidade de linhas a serem processadas (0 =
     ↪  todas)"
    echo "            Default: \"${linecount}\""
    echo "--verbose - habilita apresentacao de mensagens, que serao
     ↪  precedidas"
    echo "            por \"* \""
    echo "--noverbose - desliga opcao \"--verbose\""
    echo "--help ou -h ou -? - mostra mensagem de ajuda."
}

while [ "${#}" -ge 1 ]
do
    case "${1}" in
    "--db")
        shift
        database="${1}"
        ;;
    "--start")
        shift
        firstline="${1}"
        ;;
    "--count")
        shift
        linecount="${1}"
        ;;
    "--verbose")
        verbose=1
        ;;
    "--noverbose")
        verbose=0
        ;;
    "--help"|"-h"|"-?"|*)
        help
        exit 0
        ;;
    esac
    shift
done

totallines=$(wc -l < "${database}")
lastline=${totallines}
if [ "${linecount}" -gt "0" ]
then
    lastline=$((firstline + linecount - 1))
```

```
fi

currentline="${firstline}"

if [ "${verbose}" -a "${verbose}" -gt "0" ]
then
    echo "* Avaliando listagem \"${database}\""
    echo "* Quantidade de arquivos: ${totallines}"
    echo "* Arquivo inicial: ${currentline}"
    echo "* Quantidade de arquivos: $(( lastline - currentline + 1 ))"
    echo "* lastline=${lastline}, currentline=${currentline}"
fi

while [ "${currentline}" -le "${lastline}" ]
do
    sentencefile=$(sed -n "${currentline}p" "${database}")
    jsonfile=$(./pathtxt2json.sh <<< ${sentencefile} )
    if [ "${verbose}" -a "${verbose}" -gt "0" ]
    then
        echo "* arquivo corrente: ${currentline}"
        echo "* sentencefile=${sentencefile}"
        echo "* jsonfile=${jsonfile}"
    fi
    if ./hasdef.js ${jsonfile}
    then
        echo "${sentencefile}"
    fi
    let ++currentline
done
```

C Filtro para separação das sentenças autorizativas

Código C.1 – Código de `isautorizativo.js`

```
#!/usr/bin/env nodejs

var fs = require( 'fs' );
var path = require( 'path' );
var conll = require( './conll' );
var gce = require( './gce' );

function testJSONFile( fName )
{
    var o, tokenN;

    o = gce.load( fName );
    if( ( tokenN = gce.testForRule0( o ) ) >= 0 ) {
        if( o.tokens[ tokenN + 2 ].partOfSpeech.tag == "NOUN" ) {
```

```
            var nextVerb;

            nextVerb = gce.findNextVerb( o, tokenN + 2 );
            if( nextVerb >= 0 ) {
                if( o.tokens[ nextVerb ].dependencyEdge.headTokenIndex ==
                ↪  tokenN + 2 &&
                    o.tokens[ nextVerb ].lemma == "autorizar" ) {
                    return( 0 );
                }
            }
        }
    }
    return( 1 );
}

var nArg;

for( nArg = 2; nArg < process.argv.length; ++nArg ) {
    switch( process.argv[ nArg ] ) {
    case "--help":
    case "-h":
    case "-?":
        console.log( "%s gce.json", path.basename( process.argv[ 1 ] ) );
        console.log( "gce.json = arquivo JSON de resultado do POS tagging
        ↪  de sentencas da Google Compute Engine." );
        process.exit( 3 );

    default:
        process.exit( testJSONFile( process.argv[ nArg ] ) );
        break;
    }
}

/*
    Codigos de retorno:
        0: arquivo possui uma definicao
        1: arquivo nao possui uma definicao
        2: arquivo nao informado
        3: apenas imprimir ajuda e encerrar
*/
process.exit( 2 );
```

Exemplos de utilização:

Para excluir autorizativos do grupo principal, cuja versão no *pipeline* passa se chamar `defs0.v4.txt`:

```
./filterautorizativos.sh --nomatch > defs0.v4.txt
```

Para produzir o grupo das leis autorizativas e guardá-lo no arquivo `defs0.v3-autorizativas.txt`:

```
./filterautorizativos.sh --match > defs0.v3-autorizativas.txt
```

Código C.2 – Código de `filterautorizativos.sh`

```
#!/usr/bin/env bash

#
# comando para localizar sentencas de leis autorizativas
#
# como o repositorio de leis contem tambem projetos, e' necessario
# que se faca um filtro que execute a pesquisa apenas nos textos das leis

verbose=0
database="defs0.v3.txt"
firstline=1
linecount=0
match=1

help( )
{
    echo "$(basename ${0}):"
    echo "--match - gera saida com as sentencas autorizativas"
    echo "--nomatch - gera saida com as sentencas que nao sao autorizativas
    ↪  (default)"
    echo "--db {arquivo com lista das sentencas} - os arquivos apontados
    ↪  devem"
    echo "            conter as sentencas em txt originadoras das base
    ↪  JSON."
    echo "            Default: \"${database}\""
    echo "--start {n} - linha inicial de processamento [1..n]"
    echo "            Default: \"${firstline}\""
    echo "--count {n} - quantidade de linhas a serem processadas (0 =
    ↪  todas)"
    echo "            Default: \"${linecount}\""
    echo "--verbose - habilita apresentacao de mensagens, que serao
    ↪  precedidas"
    echo "            por \"* \""
    echo "--noverbose - desliga opcao \"--verbose\""
    echo "--help ou -h ou -? - mostra mensagem de ajuda."
}

while [ "${#}" -ge 1 ]
do
    case "${1}" in
    "--match")
        match=1
        ;;
    "--nomatch")
        match=0
        ;;
```

```
    "--db")
        shift
        database="${1}"
        ;;
    "--start")
        shift
        firstline="${1}"
        ;;
    "--count")
        shift
        linecount="${1}"
        ;;
    "--verbose")
        verbose=1
        ;;
    "--noverbose")
        verbose=0
        ;;
    "--help"|"-h"|"-?"|*)
        help
        exit 0
        ;;
    esac
    shift
done

totallines=$(wc -l < "${database}")
lastline=${totallines}
if [ "${linecount}" -gt "0" ]
then
    lastline=$((firstline + linecount - 1))
fi

currentline="${firstline}"

if [ "${verbose}" -a "${verbose}" -gt "0" ]
then
    echo "* Avaliando listagem \"${database}\""
    echo "* Quantidade de arquivos: ${totallines}"
    echo "* Arquivo inicial: ${currentline}"
    echo "* Quantidade de arquivos: $(( lastline - currentline + 1 ))"
    echo "* lastline=${lastline}, currentline=${currentline}"
fi

while [ "${currentline}" -le "${lastline}" ]
do
    sentencefile=$(sed -n "${currentline}p" "${database}")
    jsonfile=$(./pathtxt2json.sh <<< ${sentencefile} )
    if [ "${verbose}" -a "${verbose}" -gt "0" ]
    then
        echo "* arquivo corrente: ${currentline}"
        echo "* sentencefile=${sentencefile}"
        echo "* jsonfile=${jsonfile}"
```

```
    fi
    if ./isautorizativo.js ${jsonfile}
    then
        isaut=1
    else
        isaut=0
    fi
    let doecho=isaut^\!match
    if [ "${doecho}" -ne "0" ]
    then
        echo "${sentencefile}"
    fi
    let ++currentline
done
```

D Programa identificador de definições de entidades

Código D.1 – Código de `isentidade.js`

```
#!/usr/bin/env nodejs

var fs = require( 'fs' );
var path = require( 'path' );
var conll = require( './conll' );
var gce = require( './gce' );

function testJSONFile( fName )
{
    var o, tokenN;
    var ent;

    o = gce.load( fName );

    if( ( tokenN = gce.testForRule0( o ) ) >= 0 ) {
//        if( o.entities[ ent ].type == "ORGANIZATION" ) {
            for( ent = 0; ent < o.entities.length; ++ent ) {
                var mention;

                for( mention = 0; mention < o.entities[ ent
                ↪  ].mentions.length; ++mention ) {
                    if( o.entities[ ent ].mentions[ mention
                    ↪  ].text.beginOffset < o.tokens[ tokenN
                    ↪  ].text.beginOffset &&
                        o.entities[ ent ].mentions[ mention ].type ==
                        ↪  "PROPER" )
                        return( 0 );
                }
            }
```

```
//        }
    }
    return( 1 );
}

var nArg;

for( nArg = 2; nArg < process.argv.length; ++nArg ) {
    switch( process.argv[ nArg ] ) {
    case "--help":
    case "-h":
    case "-?":
        console.log( "%s gce.json", path.basename( process.argv[ 1 ] ) );
        console.log( "gce.json = arquivo JSON de resultado do POS tagging
        ↪  de sentencas da Google Compute Engine." );
        process.exit( 3 );

    default:
        process.exit( testJSONFile( process.argv[ nArg ] ) );
        break;
    }
}

/*
    Codigos de retorno:
        0: arquivo possui uma definicao
        1: arquivo nao possui uma definicao
        2: arquivo nao informado
        3: apenas imprimir ajuda e encerrar
*/
process.exit( 2 );
```

Exemplos de utilização:

Para excluir do conjunto resultante as sentenças que definem entidades e guardar o novo conjunto no arquivo `defs0.v5.txt`:

```
./filterentities.sh --nomatch > defs0.v5.txt
```

Para criar um grupo das sentenças que definem entidades a partir do conjunto atualmente selecionado:

```
./filterentities.sh --match > defs0.v4-entidades.txt
```

Código D.2 – Código de `filterentities.sh`

```
#!/usr/bin/env bash
```

```
#
# comando para localizar as sentencas com definicoes de entidades nominadas
#

verbose=0
database="defs0.v4.txt"
firstline=1
linecount=0
match=1

help( )
{
    echo "$(basename ${0}):"
    echo "--match - gera saida com as sentencas autorizativas"
    echo "--nomatch - gera saida com as sentencas que nao sao autorizativas
    ↪  (default)"
    echo "--db {arquivo com lista das sentencas} - os arquivos apontados
    ↪  devem"
    echo "            conter as sentencas em txt originadoras das base
    ↪  JSON."
    echo "            Default: \"${database}\""
    echo "--start {n} - linha inicial de processamento [1..n]"
    echo "            Default: \"${firstline}\""
    echo "--count {n} - quantidade de linhas a serem processadas (0 =
    ↪  todas)"
    echo "            Default: \"${linecount}\""
    echo "--verbose - habilita apresentacao de mensagens, que serao
    ↪  precedidas"
    echo "            por \"* \""
    echo "--noverbose - desliga opcao \"--verbose\""
    echo "--help ou -h ou -? - mostra mensagem de ajuda."
}

while [ "${#}" -ge 1 ]
do
    case "${1}" in
    "--match")
        match=1
        ;;
    "--nomatch")
        match=0
        ;;
    "--db")
        shift
        database="${1}"
        ;;
    "--start")
        shift
        firstline="${1}"
        ;;
    "--count")
        shift
        linecount="${1}"
```

```
        ;;
    "--verbose")
        verbose=1
        ;;
    "--noverbose")
        verbose=0
        ;;
    "--help"|"-h"|"-?"|*)
        help
        exit 0
        ;;
    esac
    shift
done

totallines=$(wc -l < "${database}")
lastline=${totallines}
if [ "${linecount}" -gt "0" ]
then
    lastline=$((firstline + linecount - 1))
fi

currentline="${firstline}"

if [ "${verbose}" -a "${verbose}" -gt "0" ]
then
    echo "* Avaliando listagem \"${database}\""
    echo "* Quantidade de arquivos: ${totallines}"
    echo "* Arquivo inicial: ${currentline}"
    echo "* Quantidade de arquivos: $(( lastline - currentline + 1 ))"
    echo "* lastline=${lastline}, currentline=${currentline}"
fi

while [ "${currentline}" -le "${lastline}" ]
do
    sentencefile=$(sed -n "${currentline}p" "${database}")
    jsonfile=$(./pathtxt2json.sh <<< ${sentencefile} )
    if [ "${verbose}" -a "${verbose}" -gt "0" ]
    then
        echo "* arquivo corrente: ${currentline}"
        echo "* sentencefile=${sentencefile}"
        echo "* jsonfile=${jsonfile}"
    fi
    if ./isentidade.js ${jsonfile}
    then
        isaut=1
    else
        isaut=0
    fi
    let doecho=isaut^\!match
    if [ "${doecho}" -ne "0" ]
    then
        echo "${sentencefile}"
```

```
    fi
    let ++currentline
done
```

E Módulo CONLL

Código E.1 – Código de `conll.js`

```
#!/usr/bin/env nodejs

var util = require( 'util' );
var sprintf = util.format;

var _self_module = {
    getFORM: function( token )
    {
        return( token.text.content );
    },

    getLEMMA: function( token )
    {
        if( 'lemma' in token )
            return( token.lemma );
        return( '_' );
    },

    getUPOSTAG: function( token )
    {
        return( token.partOfSpeech.tag );
    },

    getXPOSTAG: function( token )
    {
        if( _self_module.getUPOSTAG( token ) == 'PUNCT' )
            return( '.' );
        return( '_' );
    },

    /*
    mapFEAT_animacy: function( pos )
    {
        return( undefined );
    },
    */

    mapFEAT_aspect: function( pos )
    {
        var aspectMap = {
```

```
            'IMPERFECTIVE': 'Imp',
            'PERFECTIVE': 'Perf'
        };

        if( ! ( 'aspect' in pos ) || ! ( pos.aspect in aspectMap ) )
            return( undefined );
        return( 'Aspect=' + aspectMap[ pos.aspect ] );
    },

    mapFEAT_case: function( pos )
    {
        var caseMap = {
            'ACCUSATIVE': 'Acc',
            'DATIVE': 'Dat',
            'NOMINATIVE': 'Nom',
    //        'PREPOSITIONAL': ''
        };

        if( ! ( 'case' in pos ) || ! ( pos.case in caseMap ) )
            return( undefined );
        return( 'Case=' + caseMap[ pos.case ] );
    },
    /*
    mapFEAT_definite: function( pos )
    {
        return( undefined );
    },

    mapFEAT_degree: function( pos )
    {
        return( undefined );
    },
    */
    mapFEAT_gender: function( pos )
    {
        var genderMap = {
            'FEMININE': 'Fem',
            'MASCULINE': 'Masc',
            'COMMON': 'Common',
            'NEUTER': 'Neut'
        };

        if( ! ( 'gender' in pos ) || ! ( pos.gender in genderMap ) )
            return( undefined );
        return( 'Gender=' + genderMap[ pos.gender ] );
    },

    mapFEAT_mood: function( pos )
    {
        var moodMap = {
            'IMPERATIVE': 'Imp',
            'INDICATIVE': 'Ind',
            'SUBJUNCTIVE': 'Sub',
```

```
        };

        if( ! ( 'mood' in pos ) || ! ( pos.mood in moodMap ) )
            return( undefined );
        return( 'Mood=' + moodMap[ pos.mood ] );
    },
    /*
    mapFEAT_negative: function( pos )
    {
        return( undefined );
    },

    mapFEAT_numtype: function( pos )
    {
        return( undefined );
    },
    */
    mapFEAT_number: function( pos )
    {
        var numberMap = {
            'PLURAL': 'Plur',
            'SINGULAR': 'Sing'
        };

        if( ! ( 'number' in pos ) || ! ( pos.number in numberMap ) )
            return( undefined );
        return( 'Number=' + numberMap[ pos.number ] );
    },

    mapFEAT_person: function( pos )
    {
        var personMap = {
            'FIRST': '1',
            'SECOND': '2',
            'THIRD': '3'
        };

        if( ! ( 'person' in pos ) || ! ( pos.person in personMap ) )
            return( undefined );
        return( 'Person=' + personMap[ pos.person ] );
    },

    /*
    mapFEAT_poss: function( pos )
    {
        return( undefined );
    },

    mapFEAT_prontype: function( pos )
    {
        return( undefined );
    },
```

```
    mapFEAT_reflex: function( pos )
    {
        return( undefined );
    },
    */

    mapFEAT_tense: function( pos )
    {
        var tenseMap = {
    //        'CONDITIONAL_TENSE',
            'FUTURE': 'Fut',
            'PAST': 'Past',
            'PLUPERFECT': 'Pqp',
            'PRESENT': 'Pres',
        };

        if( ! ( 'tense' in pos ) || ! ( pos.tense in tenseMap ) )
            return( undefined );
        return( 'Tense=' + tenseMap[ pos.tense ] );
    },

    /*
    mapFEAT_verbform: function( pos )
    {
        return( undefined );
    },

    mapFEAT_voice: function( pos )
    {
        return( undefined );
    },
    */

    getFEATS: function( token )
    {
        var feats, pos;
        var featArray;

    // Animacy: animacy
    // Aspect: aspect
    // Case: case
    // Definite: definiteness or state
    // Degree: degree of comparison
    // Gender: gender
    // Mood: mood
    // Negative: whether the word can be or is negated
    // NumType: numeral type
    // Number: number
    // Person: person
    // Poss: possessive
    // PronType: pronominal type
    // Reflex: reflexive
    // Tense: tense
```

```
    // VerbForm: form of verb or deverbative
    // Voice: voice

        pos = token.partOfSpeech;
        featArray = new Array( );
    //    featArray.push( _self_module.mapFEAT_animacy( pos ) );
        featArray.push( _self_module.mapFEAT_aspect( pos ) );
        featArray.push( _self_module.mapFEAT_case( pos ) );
    //    featArray.push( _self_module.mapFEAT_definite( pos ) );
    //    featArray.push( _self_module.mapFEAT_degree( pos ) );
        featArray.push( _self_module.mapFEAT_gender( pos ) );
        featArray.push( _self_module.mapFEAT_mood( pos ) );
    //    featArray.push( _self_module.mapFEAT_negative( pos ) );
    //    featArray.push( _self_module.mapFEAT_numtype( pos ) );
        featArray.push( _self_module.mapFEAT_number( pos ) );
        featArray.push( _self_module.mapFEAT_person( pos ) );
    //    featArray.push( _self_module.mapFEAT_poss( pos ) );
    //    featArray.push( _self_module.mapFEAT_prontype( pos ) );
    //    featArray.push( _self_module.mapFEAT_reflex( pos ) );
        featArray.push( _self_module.mapFEAT_tense( pos ) );
    //    featArray.push( _self_module.mapFEAT_verbform( pos ) );
    //    featArray.push( _self_module.mapFEAT_voice( pos ) );
        feats = "";
        featArray.forEach( function( o, i ) {
            if( o && o != undefined && o.length > 0 ) {
                if( feats.length > 0 )
                    feats += '|';
                feats += o;
            }
        } );
        return( feats.length > 0 ? feats : '_' );
    },

    getHEAD: function( token )
    {
        if( token.dependencyEdge.label == 'ROOT' )
            return( '0' );
        return( 1 + parseInt( token.dependencyEdge.headTokenIndex ) );
    },

    getDEPREL: function( token )
    {
        return( token.dependencyEdge.label );
    },

    getDEPS: function( token )
    {
        return( '_' );
    },

    getMISC: function( token )
    {
        return( '_' );
```

```
    },

    token2Conllu: function( tokenSeq, token )
    {
        // ID, FORM, LEMMA, UPOSTAG, XPOSTAG, FEATS, HEAD, DEPREL, DEPS,
        ↪  MISC
        return( sprintf( '%d\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s',
            tokenSeq + 1,
            _self_module.getFORM( token ),
            _self_module.getLEMMA( token ),
            _self_module.getUPOSTAG( token ),
            _self_module.getXPOSTAG( token ),
            _self_module.getFEATS( token ),
            _self_module.getHEAD( token ),
            _self_module.getDEPREL( token ),
            _self_module.getDEPS( token ),
            _self_module.getMISC( token ) ) );
    },

    getText: function( o, first, last )
    {
        var t;
        var txt;

        if( first === undefined )
            first = 0;
        if( last === undefined || last < 1 )
            last = o.tokens.length;

        txt = "";
        for( t = first; t < last; ++t ) {
            if( txt.length > 0 )
                txt += " ";
            txt += o.tokens[ t ].text.content;
        }
        return( txt );
    },

    dumpText: function( o, first, last )
    {
        console.log( _self_module.getText( o, first, last ) );
    },

    getCONLL: function( fName, o )
    {
        var text, tokenSeq;

        text = sprintf( "# %s\n", fName );
        o.tokens.forEach( function( o, i ) {
            text += _self_module.token2Conllu( i, o ) + "\n";
        } );
        return( text );
    },
```

```
    dumpCONLL: function( fName, o )
    {
        console.log( _self_module.getCONLL( fName, o ) );
    },

    dumpToken: function( token, tokenSeq )
    {
        if( token.partOfSpeech.tag == 'VERB' )
            console.log( '%d: %s: (->%d) %s %s %s %s %s',
                tokenSeq + 1,
                token.text.content,
                token.dependencyEdge.headTokenIndex,
                token.dependencyEdge.label,
                token.partOfSpeech.tag,
                token.lemma,
                token.partOfSpeech.tense,
                token.partOfSpeech.person );
        else
            console.log( '%d: %s: (->%d) %s %s',
                tokenSeq + 1,
                token.text.content,
                token.dependencyEdge.headTokenIndex,
                token.dependencyEdge.label,
                token.partOfSpeech.tag );
    }
}

module.exports = _self_module;
```

F Módulo GCE

Código F.1 – Código de `gce.js`

```
#!/usr/bin/env nodejs

var fs = require( 'fs' );
var util = require( 'util' );
var sprintf = util.format;

var _self_module = {
    load: function( fName ) {
        var o;

        o = JSON.parse( fs.readFileSync( fName, { "encoding": "UTF-8",
        ↪  "flag": "r" } ) );
        return( o );
    },
```

```
    testForRule0 : function ( o ) {
        var n;

        for( n = 0; n < o.tokens.length; ++n ) {
            if( o.tokens[ n ].dependencyEdge.label == 'ROOT' &&
                o.tokens[ n ].partOfSpeech.tag == 'VERB' &&
                o.tokens[ n ].lemma == 'ser' &&
                o.tokens[ n ].partOfSpeech.tense == 'PRESENT' &&
                o.tokens[ n ].partOfSpeech.person == 'THIRD' ) {
                // encontrado verbo SER como ROOT
                // deve ser seguido de artigo e pelo menos mais um token.
                if( n < o.tokens.length - 2 ) {
                    var DETtoken;

                    DETtoken = o.tokens[ n + 1 ];
                    if( DETtoken.partOfSpeech.tag == 'DET' )
                        return( n );
                }
            }
        };
        return( -1 );
    },

    findNextVerb : function ( o, start ) {
        var n;

        for( n = start + 1; n < o.tokens.length; ++n )
            if( o.tokens[ n ].partOfSpeech.tag == "VERB" )
                return( n );
        return( -1 );
    }
}

module.exports = _self_module;
```

G Função de mapeamento de caminhos

Código G.1 – Código de `pathtxt2json.sh`

```
#!/usr/bin/env bash

# Exemplo:
# de:   /opt/normas/txt-sentencas/1947/LEI-1947-00127/00009-art3_cpt.txt
# para:
↪   /opt/normas/txt-sentencas-json/1947/LEI-1947-00127/00009-art3_cpt.json
```

```
sed "s/txt-sentencas/txt-sentencas-json/" | sed "s/\.txt$/\.json/"
```

H Programa de conversão dos arquivos JSON obtidos da GCE para o formato CONLL-U ou texto

Código H.1 – Código de `2text.js`

```
#!/usr/bin/env nodejs

var conll = require( "./conll" );
var gce = require( "./gce" );

var asText = 1;

function printHelp( )
{
    console.log( "Produz forma legivel alternativa para arquivo JSON de
    ↪  saida do processamento CGE (Google Cloud Engine).\n" +
    "2text {params} arquivo\n" +
    "em {params}:\n" +
    "Informe \"--text\" ou \"--type text\" para saida em texto;
    ↪  (DEFAULT)\n" +
    "  ou\n" +
    "Informe \"--conll\" ou \"--type conll\" para saida em CONLL-U." );
}

var nArg;

for( nArg = 2; nArg < process.argv.length; ++nArg ) {
    switch( process.argv[ nArg ] ) {
    case '--text':
        asText= 1;
        break;

    case '--conll':
        asText = 0;
        break;

    case '--type':
        if( ++nArg >= process.argv.length ) {
            console.log( "O parametro '--type' necessita argumento 'text'
            ↪  ou 'conll'." );
            process.exit( 1 );
        }
        switch( process.argv[ nArg ] ) {
        case 'text':
            asText = 1;
```

```
            break;

        case 'conll':
            asText = 0;
            break;

        default:
            console.log( "Tipo '%s' invalido.", process.argv[ nArg ] );
            process.exit( 2 );
        }
        break;

    case '--help':
    case '-h':
    case '-?':
        printHelp( );
        process.exit( 0 );
        break;

    default:
        fName = process.argv[ nArg ];
        conteudo = gce.load( fName );
        if( asText )
            conll.dumpText( conteudo );
        else
            conll.dumpCONLL( fName, conteudo );
        break;
    }
}
```

I Programa para visualizar os arquivo JSON do GCE em um formato derivado do CONLL-U

Código I.1 – Código de `browseconll.sh`

```
#!/usr/bin/env bash

# uso: $(basename ${0}) < database

database="defs0.v4.txt"
first=1

help( )
{
    echo "$(basename ${0}) -- navega uma base de JSON mostrados como CONLL
    ↪  resumido (colunas 1, 2, 4, 7 e 8)"
    echo "Parametros:"
    echo "--db {arquivo} - arquivo contendo caminhos para sentencas-txt"
```

```
    echo "        default: \"${database}\""
    echo "--first {N} - linha a partir da qual iniciar a navegacao"
    echo "        default: ${first}"
    echo "--help ou -h ou -? - mostra esta mensagem de ajuda"
}

prompt( )
{
    _prompted="x"
    if [ "${#}" -ge "1" ]
    then
        while :
        do
            echo -n "${1}"
            read r
            if [ -z "${r}" ]
            then
                _prompted="${3}"
                return
            fi
            if grep ":${r}:" <<< "${2}" 2>&1 > /dev/null
            then
                _prompted="${r}"
                return
            fi
        done
    fi
}

while [ "${#}" -gt "0" ]
do
    case "${1}" in
    "--db")
        shift
        database="${1}"
        ;;
    "--first")
        shift
        first="${1}"
        ;;
    "--help"|"-h"|"-?")
        help
        exit 1
        ;;
    *)
        echo "Parametro  \"${1}\" incorreto."
        help
        exit 2
        ;;
    esac
    shift
done
```

```
for (( line = first, lastline = $(wc -l < ${database}); line <= lastline;
 ↪  ++line ))
do
    sentencefile=$(sed -n "${line}p" "${database}")
    jsonfile=$(./pathtxt2json.sh <<< ${sentencefile} )
    _prompted="R"
    while :
    do
        case "${_prompted}" in
        [Pp])
            break
            ;;
        [Aa])
            if [ "${line}" -gt "1" ]
            then
                let line-=2
            else
                let --line
            fi
            break
            ;;
        [Rr])
            ./2text.js --conll "${jsonfile}" | cut -d $'\t' -f1,2,4,7,8 |
             ↪  less
            ;;
        [Jj])
            less "${jsonfile}"
            ;;
        [Cc])
            break 2
            ;;
        esac
        prompt "${line}/${lastline} - [P]roximo, [A]nterior, [R]ever, ver
         ↪  [J]SON, [C]ancelar: (P)" ":P:p:A:a:R:r:J:j:C:c:" "P"
    done
done
```

Aplicação de rede neural convolucional para reconhecimento facial dos senadores brasileiros da 55ª legislatura 2015-2019

Fabrício Fernandes Santana* João Alberto de Oliveira Lima
Lauro César Araujo Daniel de Mello Viero
Flávio Roberto de Almeida Heringer Hudson de Martim
Jideão José Vieira Filho Wagner Rodrigues Teixeira

Resumo

Este artigo apresenta um modelo de classificação treinado para reconhecimento facial dos senadores brasileiros da 55ª legislatura (2015-2019), o conjunto de imagens previamente classificadas utilizado para treinamento do modelo de classificação, e a acurácia deste modelo.

Palavras-chave: inteligência artificial. aprendizado de máquina. rede neural convolucional. reconhecimento facial. Senado.

1 Introdução

Nos últimos anos, técnicas de inteligência artificial têm sido aplicadas em diversas tarefas de nosso cotidiano, tais como: solução de trabalhos repetitivos, reconhecimento de voz, imagem, processamento de linguagem natural, diagnósticos médicos, carros autônomos, dentre outras (GOODFELLOW; BENGIO; COURVILLE, 2016). A introdução das redes neurais convolucionais (*convolutional neural network*) – tipo de algoritmo de aprendizado de máquina supervisionado (*supervised machine learning*) – melhorou de forma significativa o desempenho de atividades relacionadas à visão computacional, entre elas o reconhecimento facial (ZHONG; CHEN; HUANG, 2017).

O estado da arte na tarefa de reconhecimento facial foi alcançado por Schroff, Kalenichenko e Philbin (2015) ao atingir a acurácia de 99,63%

*fabricio.santana@senado.leg.br

no reconhecimento facial quando aplicada ao *Labeled Faces in the Wild (LFW)*, que é um banco público de imagens utilizado pela academia para *benchmark* de algoritmos de reconhecimento facial (HUANG et al., 2007). Devido a esse significativo índice de sucesso, este trabalho tem por objetivo aplicar a mesma abordagem sugerida pelos pesquisadores para treinar um modelo capaz de reconhecer a face dos senadores brasileiros da 55ª legislatura (2015-2019).

A necessidade de utilização do reconhecimento facial no Senado Federal foi identificada no âmbito do sistema de votação eletrônica das comissões (SVE) que é utilizado pelos parlamentares para o registro de presença e votação nas comissões permanentes e mistas. Estas operações (presença e voto) são registradas no sistema em duas etapas: primeiro o parlamentar informa seu código público e então apresenta sua digital para leitura e identificação. Entretanto, o leitor biométrico não é capaz de capturar a digital de alguns parlamentares, e nesse caso, é necessário que o parlamentar utilize sua senha pessoal e secreta em substituição à impressão digital. Entretanto, para aumentar a segurança e melhorar a usabilidade do sistema, este trabalho propõe a introdução do reconhecimento facial como alternativa ao processo de autenticação dos parlamentares. Com a introdução do reconhecimento facial, o parlamentar não precisaria digitar seu código público, mas sim ter sua face reconhecida pelo sistema, e finalizar o processo de autenticação com a leitura da impressão digital. Isso reduziria a necessidade de informar a senha apenas no caso de não ser possível ler a digital. Com a introdução do reconhecimento facial, o parlamentar sempre irá operar o sistema após ter pelo menos uma identificação biométrica bem sucedida, sendo possível garantir que o usuário do sistema é o próprio parlamentar.

Para apresentar os resultados objetivos, o presente artigo está dividido da seguinte forma: a seção 2 apresenta uma revisão bibliográfica com um breve apanhado dos últimos resultados obtidos na área de reconhecimento facial; a seção 3 apresenta o percurso metodológico adotado que abrange: obtenção e preparação de um conjunto de imagens de Senadores para treinamento de um modelo de reconhecimento facial, utilização do modelo treinado, aplicação prática em ambiente simulado; a seção 4 consolida os principais resultados alcançados; e, as considerações finais e sugestões de aplicações futuras de reconhecimento facial dentro do Senado Federal são apresentadas na seção 5.

2 Revisão Bibliográfica

Esta seção 2 apresenta a síntese de conceitos abordados neste trabalhado a partir da revisão das obras de Russell e Norvig (2010), Goodfellow,

Bengio e Courville (2016), Schroff, Kalenichenko e Philbin (2015), Zhong, Chen e Huang (2017), Abadi et al. (2016), Guo et al. (2016) e Sandberg (2017).

A inteligência artificial é uma disciplina que tem ocupado cientistas que almejam entender como os seres humanos pensam e desejam criar agentes que agem racionalmente. Segundo Russell e Norvig (2010), um agente racional é aquele que age para atingir o melhor resultado, ou, quando há incerteza, o melhor resultado possível, e, este agente alcança tal resultado operando de forma autônoma, percebendo o ambiente, persistindo por um longo período de tempo, adaptando a mudanças, e, criando e perseguindo objetivos.

Goodfellow, Bengio e Courville (2016) afirmam que, na disciplina de inteligência artificial, técnicas de aprendizado de máquina (*machine learning*) estão evoluindo por dar ao computador habilidade de solucionar problemas sem serem explicitamente programados, ou seja, sem a programação de regras pré-definidas, pois os algoritmos aprendem com dados. O autor apresenta em sua obra uma clássica definição de aprendizado de máquina: "*A computer program is said to learn from experience E with respect to someclass of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*". Assevera ainda que apresentam-se como problemas de aprendizado de máquina: classificação, regressão, tradução, entre outros, e que o reconhecimento facial é um tarefa de classificação, e, neste tipo de tarefa, o algoritmo deve inferir em qual categoria/identidade uma determina entrada pertence.

Aprendizado profundo (*deep learning*) é um tipo específico de algoritmo de aprendizado de máquina que teve sua criação motivada em virtude da dificuldade dos algoritmos tradicionais de aprendizado de máquina em generalizar bem para alguns tipos de tarefas, como reconhecimento de fala ou reconhecimento de objetos. O desafio de generalizar para inferir respostas para novos exemplos se torna exponencialmente mais difícil quando se trabalha com dados de muitas dimensões (GOODFELLOW; BENGIO; COURVILLE, 2016).

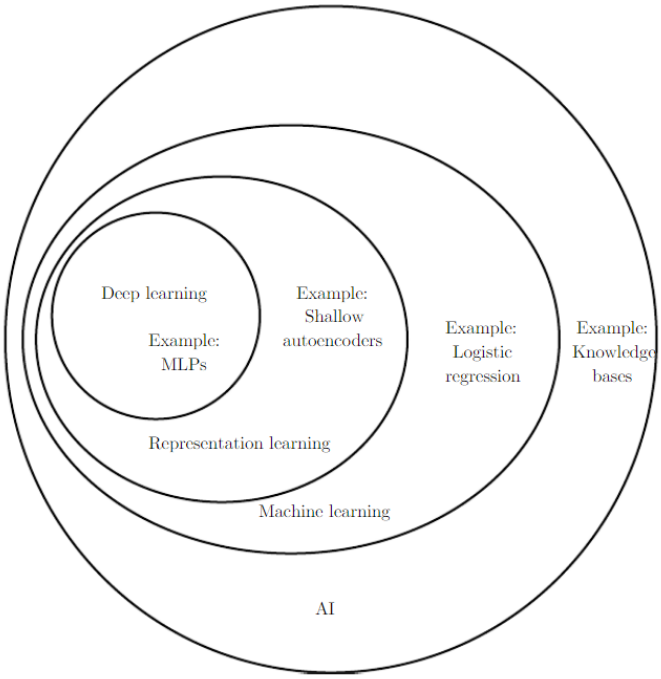
As redes neurais artificiais são um tipo de algoritmo de aprendizado de máquina inspirado na neurociência e nas redes neurais biológicas que compõem o cérebro dos animais. Esta rede é representada por meio da composição de várias funções que são organizadas em camada. O algoritmo de aprendizado deve decidir como usar estas camadas para produzir a saída desejada (GOODFELLOW; BENGIO; COURVILLE, 2016).

As redes neurais convolucionais são um tipo especial de rede neural para processar dados que possam ser representados em uma topologia conhecida, por exemplo, uma matriz. Este é o caso, por exemplo, de ima-

gens que podem ser representadas por uma matriz de duas dimensões de *pixels*. O nome convolucional significa que a rede emprega uma operação matemática chamada convolucional que é um tipo especial de operação linear. "*Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers*" (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Figura 1 ilustra um diagrama de Venn que apresenta a hierarquia dos conceitos e técnicas de inteligência de IA.

Figura 1 – Diagrama com hierarquia de conceitos e técnicas de IA



Fonte: Goodfellow, Bengio e Courville (2016, p. 9)

Na abordagem convencional, ou seja, sem a utilização de técnicas de inteligência artificial, programar um computador para reconhecer o rosto de uma pessoa nas mais diversas situações de iluminação, posição, qualidade da imagem etc., exigiria uma grande quantidade de regras devido ao alto número de combinações que o computador precisaria conhecer *a priori*, e,

além disso, a cada nova identidade, a programação do computador precisaria ser alterada (ZHONG; CHEN; HUANG, 2017).

Com a inteligência artificial, mais especificamente aprendizado de máquina, o computador aprende a partir das características de imagem cuja identidade seja conhecida e cria um modelo que seja capaz de inferir a identidade de uma nova imagem. No caso de novas identidades, é necessário re-treinar o modelo para torná-lo capaz de reconhecer imagens que não foram apresentadas durante o treinamento. Portanto, o desafio deixa de ser programar um conjunto de regras, conforme a abordagem convencional, e passa a ser criar um modelo matemático a partir da observação de exemplos exibidos para o algoritmo que seja capaz de inferir uma saída a partir de uma entrada (SCHROFF; KALENICHENKO; PHILBIN, 2015).

Algoritmos de reconhecimento facial avançaram muito nos últimos anos. Tradicionalmente, publicações de novos algoritmos de reconhecimento facial avaliam sua performance contra *datasets* públicos de imagens, tais como: *Labeled Faces on the wild (LFW)* (HUANG et al., 2007), *Youtube Faces DB (YTB)* (WOLF; HASSNER; MAOZ, 2011), *CASIA-WebFace* (YI et al., 2014), *MS-Celeb-1M* (GUO et al., 2016) e outros. A Tabela 1 apresenta uma síntese da performance de alguns algoritmos de reconhecimento quando aplicados sobre o *dataset LFW* (ZHONG; CHEN; HUANG, 2017).

Tabela 1 – Performance de diferentes métodos de reconhecimento facial sobre o LFW

Método	DataSet	LFW	Ano publicação
DeepFace	4M	97,35%	2014
Facenet	200M	99,63%	2015
DeepID	0,2M	97,45%	2014
DeepID2+	0.2M	99,47%	2015
VGG Face	2,6M	99,13%	2015
Center face	0,7M	99,28%	2016

Fonte: Zhong, Chen e Huang (2017)

A introdução das redes neurais convolucionais foi fundamental para o estabelecimento de um novo patamar de acurácia na tarefa de reconhecimento facial, pois como esta técnica de aprendizado é baseada em dados, o algoritmo aprende as características da face e lida com variações de posição, iluminação, obstrução etc. (ZHONG; CHEN; HUANG, 2017).

Este trabalho aplica a abordagem implementada pelo método do FaceNet (SCHROFF; KALENICHENKO; PHILBIN, 2015), que atualmente é considerado o estado da arte na tarefa de reconhecimento facial (ZHONG;

CHEN; HUANG, 2017). Esse método foi utilizado neste trabalho para criação de um modelo capaz de identificar a identidade dos senadores brasileiros da 55ª legislatura (2015-2019) a partir de uma imagem, ou seja, apresentada uma imagem espera-se que o modelo treinado infira a identidade do Senador.

3 Percurso metodológico

A implementação de softwares que utilizam inteligência artificial, mais especificamente aprendizado de máquina, passam pelas seguintes etapas: compreensão do problema a ser resolvido, escolha do algoritmo que melhor atende as necessidades do problema, criação/separação de um grande volume de dados para treinamento, treinamento de um modelo de inteligência artificial e verificação da acurácia do modelo. Para essa última etapa, é necessário utilizar dados diferentes dos utilizados para treinamento, ou seja, dados não utilizados na fase de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016). Já um processo típico de reconhecimento facial usualmente consiste em quatro etapas: detectar faces de imagens capturadas, localizar pontos de referências das faces e realizar alinhamento utilizando transformações geométricas, extrair características faciais para reconhecimento, decidir por uma identidade baseada em uma probabilidade (ZHONG; CHEN; HUANG, 2017).

As subseções seguintes apresentam como cada uma dessas etapas foram executadas no contexto específico de reconhecimento de faces de senadores brasileiros da 55ª legislatura.

3.1 Etapa 1: Compreensão do problema a ser resolvido – Reconhecimento facial

A tarefa de reconhecer a face a partir de uma imagem em virtude da combinação de diferentes fatores tais como expressões, posições, iluminação etc. gera um grande número de combinações, e, portanto, utilizar a estratégia de aprendizado de máquina se aplica a essa classe de problema (ZHONG; CHEN; HUANG, 2017).

Com essa estratégia de inteligência artificial, apresenta-se a um algoritmo um conjunto de imagens com suas respectivas identidades para treinar um modelo que seja capaz de reconhecer identidades de novas imagens. Por isso, a tarefa de reconhecimento facial é considerado um problema de classificação (GOODFELLOW; BENGIO; COURVILLE, 2016).

Conforme observado na revisão bibliográfica, atualmente, a melhor estratégia para a solução deste tipo de problema é por meio da implementação

de algoritmos de redes neurais convolucionais (ZHONG; CHEN; HUANG, 2017).

3.2 Etapa 2: Escolha do algoritmo que melhor atende as necessidades do problema – Implementação

Abordagens que utilizam estratégias de aprendizagem de máquina, de forma mais específica as redes neurais convolucionais, atingiram taxa de acerto comparáveis aos seres humanos. Atualmente, o estado da arte no reconhecimento facial com 99.63% de acurácia pertence a Schroff, Kalenichenko e Philbin (2015), por isso, essa abordagem foi a adotada para o treinamento de um modelo capaz de reconhecer a face dos senadores brasileiros da 55ª legislatura.

O FaceNet é a implementação baseada em TensorFlow de uma rede neural convolucional para o reconhecimento facial (SCHROFF; KALENICHENKO; PHILBIN, 2015). TensorFlow é uma biblioteca de código-aberto disponibilizada pelo Google e utilizada para aprendizagem de máquina por oferecer grande suporte a computação matricial (ABADI et al., 2016).

Para o atingir o objetivo deste trabalho foi utilizado o Facenet para treinar um modelo capaz de reconhecer a imagem do Senadores da 55ª Legislatura.

3.3 Etapa 3: Criação/separação de um grande volume de dados para treinamento – Preparação do *dataset*

Desenvolver um algoritmo capaz de criar um modelo de reconhecimento facial demanda uma grande quantidade de imagens. Encontrar conjunto de dados não é uma tarefa simples (GUO et al., 2016).

Para o presente estudo utilizou-se imagens públicas pertencentes a 81 senadores disponíveis na página oficial do Senado no flickr[1]. Após o download, as imagens passaram por uma revisão manual para verificação da sua qualidade e classificação correta, em alguns casos, as tags do flickr estava incorreta. Para garantir a qualidade do *dataset*, imagem com um único rosto com face voltada para frente sem rotação e com boa iluminação

Após esse tratamento, o *dataset* de imagens de senadores para treinamento de um modelo de classificação ficou com cerca de 10.000 imagens.

Antes do treinamento propriamente dito, as imagens passaram por um processo de alinhamento para melhorar a acurácia do reconhecimento automático (ZHONG; CHEN; HUANG, 2017). Portanto, as imagens foram alinhadas antes de serem utilizadas para criar um classificador a partir de

[1] <https://www.flickr.com/photos/49143546@N06/>, acesso de 15.mar.2018

um modelo de reconhecimento facial previamente treinado sobre o *dataset LFW* (SANDBERG, 2017).

3.4 Etapa 4: Treinamento de um modelo de inteligência artificial

Com o modelo treinado, duas abordagens foram realizadas. Primeiro, foi disponibilizado um serviço web capaz de receber uma imagem como entrada e aplicá-la sobre o modelo de reconhecimento facial previamente treinado. Na segunda abordagem, foi utilizado um *crawler* para realizar download de imagens retornadas na busca do Google Images e aplicá-las de forma automática ao modelo de reconhecimento facial treinado. Com o uso do *crawler* foi possível criar uma *dataset* de imagens públicas de forma automatizada, a partir de uma busca no *Google Images* utilizando como termo de busca o nome parlamentar de cada um dos senadores.

4 Etapa 5: Verificação da acurácia do modelo

Este trabalho apresenta como resultado um conjunto de imagens classificadas dos senadores da 55ª legislatura, um modelo treinado utilizando o facenet/tensorflow capaz de reconhecer estes parlamentares, e a confiança alcançada no reconhecimento das imagens utilizando tal modelo.

Como primeiro resultado, foi construído um *dataset* com mais de 10 mil imagens de senadores classificadas que podem ser utilizadas por pesquisadores para que criar novos modelos de classificação. Este *dataset* está disponível para download e utilização[2].

A partir do conjunto de imagens treinou-se um modelo de reconhecimento utilizando o facenet/tensorflow. Este modelo pode ser reutilizado em implementações que necessitem reconhecer o rosto dos parlamentares de imagens da face.

Com o modelo treinado, foi apurado a acurácia do mesmo a partir da inferência da identidade de aproximadamente 2.000 imagens. Nesta etapa as imagens utilizadas para apuração da acurácia são imagens diferentes das que foram utilizadas durante o treinamento. Além da acurácia, foi avaliado também confiança, ou seja, o grau de certeza na inferência realizada em cada imagem, e a confiança média da inferência do modelo considerando todas as imagens analisadas, neste caso é importante notar não apenas a taxa de acerto do modelo, mas o quão confiante ele é sobre estar certo ou não na inferência realizada.

[2] <https://doi.org/10.6084/m9.figshare.5977648.v1>, acesso de 10 abr 2018.

Tabela 2 – Resultado do modelo de reconhecimento aplicado sobre o dataset de teste com imagem dos Senadores

Senador	% Confiança média	% confiança acima 75%	% confiança entre 50%-75%
Acir Gurgacz	89,26%	93%	6%
Aecio Neves	82,26%	74%	25%
Alvaro Dias	75,33%	63%	36%
Ana Amelia	79,55%	64%	35%
Angela Portela	69,62%	25%	75%
Antonio Anastasia	73%	42%	57%
Antonio Carlos Valadares	80,32%	75%	25%
Armando Monteiro	88,71%	81%	18%
Ataides Oliveira	76,8%	66%	33%
Benedito de Lira	79,66%	75%	25%
Cassio Cunha Lima	73,98%	45%	54%
Cidinho Santos	74,72%	57%	42%
Cristovam Buarque	76,13%	60%	40%
Dalirio Beber	65,59%	15%	84%
Dario Berger	78,54%	57%	42%
Davi Alcolumbre	76,88%	40%	60%
Edison Lobao	74,74%	50%	50%
Eduardo Amorim	56,6%	0%	100%
Eduardo Braga	83,52%	73%	26%
Elmano Ferrer	69,82%	40%	60%
Eunicio Oliveira	73,28%	46%	53%
Fatima Bezerra	71,98%	40%	60%
Fernando Bezerra Coelho	71,52%	50%	50%
Fernando Collor	79,78%	50%	50%
Flexa Ribeiro	72,73%	50%	50%
Garibaldi Alves Filho	73,9%	43%	56%
Gladson Cameli	72,31%	50%	50%
Gleisi Hoffmann	80,59%	69%	30%
Helio Jose	71,84%	42%	57%
Humberto Costa	87,29%	78%	21%
Ivo Cassol	76,65%	66%	33%
Jader Barbalho	60,25%	25%	75%
Joao Alberto Souza	78,11%	52%	47%
Joao Capiberibe	70,78%	33%	66%
Jorge Viana	80,98%	63%	36%
Jose Agripino	76,03%	54%	45%
Jose Maranhao	68,75%	40%	59%
Jose Medeiros	73,5%	42%	57%
Jose Pimentel	76,37%	52%	47%
Katia Abreu	73,3%	50%	50%
Lasier Martins	61,85%	12%	87%
Lidice da Mata	68,99%	18%	81%
Lindbergh Farias	78,38%	84%	15%
Lucia Vania	81,11%	75%	25%
Magno Malta	79,87%	77%	22%

Continua...

Tabela 2 – Resultado do modelo de reconhecimento aplicado sobre o dataset de teste com imagem dos Senadores

Senador	% Confiança média	% confiança acima 75%	% confiança entre 50%-75%
Maria do Carmo Alves	58,8%	0%	100%
Marta Suplicy	81,19%	72%	27%
Omar Aziz	72,8%	66%	33%
Otto Alencar	60,06%	0%	100%
Paulo Bauer	79,07%	59%	40%
Paulo Paim	82,95%	81%	18%
Paulo Rocha	83,14%	78%	21%
Pedro Chaves	63,73%	33%	66%
Raimundo Lira	65,4%	30%	70%
Randolfe Rodrigues	84,4%	73%	26%
Regina Sousa	83,66%	83%	16%
Reguffe	66,84%	14%	85%
Renan Calheiros	72,33%	52%	48%
Roberto Muniz	65,82%	18%	81%
Roberto Requiao	75,97%	66%	33%
Romario	82,29%	81%	18%
Romero Juca	81,5%	70%	29%
Ronaldo Caiado	73,56%	50%	50%
Rose de Freitas	64,19%	9%	90%
Sergio Petecao	80,71%	72%	27%
Simone Tebet	71,12%	47%	52%
Tasso Jereissati	59,42%	0%	100%
Telmario Mota	74,17%	61%	38%
Valdir Raupp	86,91%	83%	16%
Vanessa Grazziotin	85,55%	78%	21%
Vicentinho Alves	53,4%	0%	100%
Waldemir Moka	79,95%	66%	33%
Wellington Fagundes	68,53%	50%	50%
Wilder Morais	78,31%	75%	25%
Zeze Perrella	56,78%	0%	100%

Fonte: Produzido pelos autores.

5 Considerações finais

A disciplina de inteligência artificial tem sido amplamente difundida nos últimos anos por meio de empresas e pesquisadores que tem desenvolvido novos produtos e técnicas para a solução de problemas reais de forma muito eficiente. Muitas são os algoritmos de inteligência artificial. Dentro do escopo deste trabalho identificou-se que o aprendizado de máquina é uma das técnica promissora para o reconhecimento facial pois permite a criação

de modelos capazes de identificar a identidade de uma face após treinamento com imagens de identidades conhecidas, ou seja, a partir de combinações característica-resposta é possível criar um modelo que possa inferir identidade de uma pessoa em uma imagem que não foi apresentada ao algoritmo na fase de treinamento.

O artigo apresenta um *benchmark* para a tarefa de reconhecer 81 senadores do Brasil a partir da imagem de suas faces. O método de reconhecimento facial aplicado atinge um resultado muito satisfatório, próximo ao desempenho humano. Além disso, é disponibilizado um modelo de reconhecimento facial treinado e um dataset com mais de 10.000 imagens classificadas dos senadores[3].

Tanto o modelo quanto o *dataset* podem ser utilizados para trabalhos futuros, como por exemplo, classificação (semi)automática de fotos de senadores para efeitos arquivísticos, realidade aumentada reconhecimento em vídeos.

Agradecimentos

A Klause Alvarenga, pela ideação e incentivo.

Referências

ABADI, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2016. Disponível em: <https://arxiv.org/abs/1603.04467>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. Disponível em: <http://www.deeplearningbook.org>.

GUO, Y. et al. MS-celeb-1M: A dataset and benchmark for large-scale face recognition. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [s.n.], 2016. v. 9907 LNCS. ISBN 9783319464862. Disponível em: <https://arxiv.org/abs/1607.08221>.

HUANG, G. B. et al. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. 2007.

[3] <https://doi.org/10.6084/m9.figshare.5977648.v1>, acesso de 10 abr 2018.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A modern approach*. 3rd. ed. [s.n.], 2010. Disponível em: <http://web.cecs.pdx.edu/~mperkows/CLASS_479/2017_ZZ_00/02___GOOD_Russel=Norvig=Artificial%20Intelligence%20A%20Modern%20Approach%20(3rd%20Edition).pdf>.

SANDBERG, D. *Classifier training of inception resnet v1*. 2017. Disponível em: <https://github.com/davidsandberg/facenet/wiki/Classifier-training-of-inception-resnet-v1>.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 07-12-June, p. 815–823, 2015. Disponível em: <https://arxiv.org/abs/1503.03832>.

WOLF, L.; HASSNER, T.; MAOZ, I. Face recognition in unconstrained videos with matched background similarity. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2011. ISBN 9781457703942. ISSN 10636919.

YI, D. et al. Learning Face Representation from Scratch. 2014. Disponível em: <https://arxiv.org/abs/1411.7923>.

ZHONG, Y.; CHEN, J.; HUANG, B. Toward End-to-End Face Recognition Through Alignment Learning. *IEEE Signal Processing Letters*, v. 24, n. 8, 2017. Disponível em: <https://arxiv.org/abs/1701.07174>.

Uma proposta de ontologia sobre processo no âmbito do Processo Legislativo

Flávio Roberto de Almeida Heringer* Lauro César Araujo
João Alberto de Oliveira Lima Daniel de Mello Viero
Fabrício Fernandes Santana Hudson de Martim
Jideão José Vieira Filho Wagner Rodrigues Teixeira

Resumo

Há décadas o processo legislativo brasileiro é alvo de estudos e de avanços na organização e gestão da informação. Desde a década de 1970, com a criação do Prodasen, o Senado Federal investe em tecnologia para gerenciar o processo legislativo. As soluções empregadas na gestão da informação legislativa têm sido apresentadas como soluções a problemas pragmáticos, essencialmente teleológicas, mas não são baseadas em teoria geral que possa explicar o fundamento conceitual dos fenômenos que tratados pelos sistemas. Com base nessa motivação, este artigo reúne fundamentos conceituais com intuito de propor uma ontologia do processo legislativo baseada na ideia de "relação material". A proposta é construída como uma ontologia de domínio fundamentada na *Unified Foundational Ontology* (UFO).

Palavras-chave: Ontologia. Processo Legislativo. Processualística. Unified Fundamental Ontology (UFO). Relação Material.

1 Introdução

Os bancos de dados e sistemas de suporte ao processo legislativo no Brasil remontam à década de 1970 com a criação do Prodasen e o desenvolvimento dos primeiros sistemas de registro de acompanhamento para a tramitação de matérias legislativas na Câmara federal e no Senado Federal. Na época, a solução baseada em banco de dados textual (IBM-Stairs)

*heringer@senado.leg.br

limitava-se a registrar o equivalente a uma ficha ementária com os dados essenciais do projeto e os seus andamentos dentro das Casas Legislativas.

Essa solução acabou por forjar o termo "matéria legislativa" para designar um amplo conjunto de "coisas" que tramitam no Poder Legislativo, desde proposições em sentido estrito (projetos de lei, por exemplo), requerimentos e comunicações de outros Poderes até documentos como um telex ou mesmo o registro de uma solenidade. Em suma, havia uma lacuna conceitual que era utilizada para livremente registrar tudo aquilo sobre o qual se desejava informar ou que se julgava importante para futuras pesquisas, afinal, o sistema é o principal meio de recuperação de informações disponível.

Outra questão importante para análise é a percepção das dificuldades em se manter um padrão de registros ao longo do tempo, cuja causa certamente está vinculada à ausência de uma firme base conceitual. Além disso, o presente diagnóstico não pode deixar de considerar que alguns problemas históricos aqui apontados foram fruto de limitações concretas, como o tempo para tomada de decisão, o custo de novas implementações nos sistemas (ocasionalmente levando à utilização indevida dos sistemas de informação), bem como às limitações dos recursos computacionais e do contingente de recursos humanos para o aperfeiçoamento dos sistemas. Nosso propósito não é abordar todas as causas da situação vigente, mas perceber como uma ontologia de domínio do processo legislativo é necessária e útil para a construção de estruturas de informação que descrevam adequadamente o processo legislativo, sendo, assim, a base para a construção das diversas soluções de Tecnologia da Informação (TI) para suporte a esse processo.

Para tanto é necessário indagar primeiramente a respeito da essência do processo legislativo: o que é o Processo Legislativo? Quantos "Processos Legislativos" existem? Como são instanciados materialmente e contados no âmbito do Poder Legislativo?

Por fim, é importante destacar que o escopo do presente estudo está limitado ao Poder Legislativo Federal Brasileiro, especialmente ao Senado Federal.

2 Elementos e fundamentos conceituais

2.1 O que é o Processo?

Antes de averiguar o conceito de processo legislativo, cabe adentrar numa definição preliminar, qual seja, o que é um processo? Esta indagação pode ser respondida de diversas maneiras conforme o ramo da ciência com o qual se está lidando. Pode-se falar em processos naturais, sejam eles biológicos ou físico-químicos, processos econômicos, sociais ou administrativos,

e inúmeras outras possibilidades. Tudo isso porque a palavra "processo" é oriunda de um conceito latino fundamental: "ação de adiantar-se, movimento para diante, o andar, andamento, marcha; acontecimento, êxito" (Houaiss).

Em seu sentido jurisdicional, atrelado ao ramo do Direito Processual, o processo é o modo pelo qual o poder Estatal de jurisdição (*jurisdictio,onis* isto é, o ato de dizer o Direito, de administrar a justiça) é exercido e, como tal, é aplicado há séculos, desde os tempos da Antiguidade. Se é certo que há normas processuais jurisdicionais tão antigas, é igualmente importante estabelecer que a ciência do Direito Processual ganhou corpo a partir do surgimento dos Estados Modernos com sua concepção de repartição de Poderes e, especialmente, ao final do séc XIX com a obra de Oskar von Bülow.

Em sua obra, von Bülow defendia a ideia de que o processo é uma relação jurídica da qual participam autor, réu e juiz, estabelecendo um conjunto de direitos e obrigações entre essas partes que se desdobram conforme o procedimento (ou rito) adotado. Esta teoria é conhecida como teoria triangular da relação processual, conforme nos ensina o professor José Eduardo Carreira Alvim (ALVIM, 2016).

A teoria do processo como relação jurídica ilumina aspectos essenciais relacionados ao nascimento, ou à constituição, do processo, em especial os pressupostos necessários para sua formação. Alvim nos informa que essas condições envolvem os requisitos necessários para que as pessoas participem do processo, o objeto em discussão, quais são os atos ou fatos que determinam seu surgimento, e quais os agentes aptos a realizar tais atos.

O próprio termo "pressupostos processuais" utilizado por von Bülow, destaca os justamente o conjunto dos "elementos constitutivos da relação jurídica processual". O conceito de relação jurídica processual também serviu para auxiliar na distinção entre os aspectos intrínsecos do processo (qual seja a própria relação que se estabelece entre as partes e o juiz) e os aspectos extrínsecos, quais sejam, os procedimentos pelos quais o processo se manifesta em cada situação concreta. Assim, podemos distinguir as características intrínsecas ao processo como, por exemplo, sua autonomia, complexidade e unicidade, em contraposição aos ritos e procedimentos que são adotados conforme a situação concreta que se pretende apreciar, como, por exemplo, os ritos ordinário, sumário e sumaríssimo.

Nesse ponto, é digno de nota a exposição do professor Carreira Alvim quando afirma que:

> visto *por fora*, o processo se apresenta aos nossos olhos como um conjunto ou complexo de atos que se desenvolvem *preordenadamente*, mas considerado *por dentro* ele constitui uma relação jurídica que interliga os sujeitos processuais,

> impondo a todos uma atuação que, por fim, resultará na resolução do conflito [...] (ALVIM, 2016, p. 226)

É extremamente comum a percepção de que o processo legislativo nada mais é do que um conjunto de eventos que se concatenam para a tentativa de formulação de uma norma. Essa ideia tem influenciado decisivamente muitos agentes que atuam com vistas à aplicação de técnicas de modelagem de processos (BPM, por exemplo), ao desenvolvimento de soluções tecnológicas ou ao tratamento informacional do processo legislativo, em detrimento da perspectiva da relação jurídica estabelecida. Parece-nos que o aspecto exterior do processo, seu fluxo ou tramitação, parece mais facilmente percebido pelos agentes que se debruçam sobre o tema, ao mesmo tempo em que grande parte das perguntas sobre o processo legislativo recai sobre a necessidade de acompanhamento processual, tal como ocorre na esfera judicial.

Outro aspecto a ser notado é que a relação processual se estabelece diante de um objeto (bem jurídico tutelado) tendo em vista um *objetivo* de solução de uma lide. Adiante, trataremos da aplicação desses conceitos ao processo legislativo conforme a ontologia proposta.

Apesar de predominante, a teoria da tríade processual de von Büllow sofreu e ainda sofre contestações por parte de alguns estudiosos, tendo em vista que os doutrinadores ora pretendem resolver omissões ou apontar ênfases distintas presentes na relação processual. Carreira Alvim apresenta aa existência de algumas teorias alternativas, mencionando alguns, como Hellwig, que defendem que não há uma relação entre autor e réu pois tudo precisa da mediação do juiz, o que caracterizaria uma relação meramente angular, e outros, como Chiovenda, que entendem que o processo *contém* uma relação jurídica.

Entretanto, a imensa maioria dos doutrinadores adere à ideia do processo como uma relação jurídica que se inicia quando o autor da ação provoca o juiz natural da causa e se completa quando o réu é devidamente citado e comparece ao processo, dando início ao seu andamento regular e o estabelecimento do contraditório. Conforme o ensino de Cintra, Grinover e Dinamarco (2005) "cada processo, em concreto, tem início quando o primeiro ato processual é praticado..." e "o fim do processo ocorre ordinariamente quando é emitido o provimento jurisdicional invocado..."

A presente descrição, ainda que muito simplificada, busca apenas estabelecer uma base conceitual mínima para os propósitos deste estudo. Portanto, haja vista a presente definição de processo, como o processo legislativo se relaciona com esses conceitos?

2.2 O que é o Processo Legislativo?

O Processo Legislativo comporta diversas definições, conforme a perspectiva do agente definidor. Para os Constituintes originários de 1988, o Processo Legislativo consiste na elaboração de um conjunto de diplomas normativos elencados no artigo 59 da Constituição Federal.

Sob a ótica da ciência Política, o Processo Legislativo é visto como uma dinâmica entre os diversos atores públicos como um "fenômeno dinâmico da realidade social, que se caracteriza por uma concatenação de atos e fatos não necessariamente disciplinada pelo direito", conforme apresenta Nino Olivetti em Bobbio Norberto (1998). A abordagem política, embora pertinente para seus fins, não é o objeto de discussão neste estudo.

Talvez em razão de seu cunho fortemente político e sua complexidade poucos se debruçam sobre os aspectos normativos que sustentam o processo legislativo. O interesse em tela nesse estudo está alinhado com uma análise do Processo Legislativo sob a perspectiva jurídico-normativa, com o objetivo de uma conceituação aplicável a modelos computacionais e sistemas de informação.

Entendemos que essa complexidade não pode ser ignorada, mas enfrentada. Como afirmou o professor e ex-Presidente dos Estados Unidos da América, Wodrow Wilson, o funcionamento do parlamento não pode ser entendido sem esforço e sem um processo cuidadoso e sistemático de análise:

> *Like a vast picture thronged with figures of equal prominence and crowded with elaborate and obtrusive details, Congress is hard to see satisfactorily and appreciatively at a single stand-point. Its complicated forms and diversified structure confuse the vision, and conceal the system which underlies its composition. It is too complex to be understood without na effort, without a careful and systematic process of analysis.* (WILSON, 1981 apud MIKVA; LANE, 2002).

O Professor Manuel Gonçalves Ferreira Filho (FERREIRA FILHO, 1995) acrescenta ainda uma questão relevante para a tarefa de conceituação. Seria o termo legislativo na expressão "processo legislativo" uma referência ao agente do processo — isto é, o Poder Legislativo — ou ao objeto do processo – isto é, a norma legal a ser produzida? Tal questão é interessante pois, apesar da definição constitucional se referir à produção de normas gerais, os regimentos internos das Casas Legislativas tratam de diversas situações que são de competência legislativa cujo resultado final não desemboca na produção legislativa em sentido estrito. Os processos de fiscalização e de investigação são, também, regulados nos regimentos internos e seus procedimentos nele disciplinados sem que resultem em normas jurídicas ao seu término.

2.3 Pode-se transpor os conceitos do processo jurisdicional para o processo legislativo?

Para Floriano de Azevedo Marques Neto (MARQUES NETO, 2008), "existe um campo comum a toda a processualização da atividade estatal" que ele denomina de "teoria geral do processo de exercício do poder estatal", o qual abrange todas as esferas de processo relacionadas às funções do Estado, sejam as funções de legislar, julgar, o exercício do poder de polícia, a execução de políticas públicas etc.

Afirma, ainda, que a existência de um tronco comum a todos os processos de agentes estatais, os quais devem ser observados para o legítimo exercício de suas atribuições e competências legais, não exclui a diversidade de facetas e especificidades que distinguem, por exemplo, o processo civel do processo penal, bem como os diversos tipos de procedimentos para aprovação de matérias legislativas.

Para fundamentar sua proposição o autor elenca os seguintes aspectos:

a) A função comum do processo: que consiste em estabelecer regras básicas de exercício do poder evitando que o agente público o exerça conforme seu próprio tirocínio e arbítrio, bem como estabelecendo as formas de participação do cidadão no exercício dessa atividade;

b) Os princípios centrais que compõem qualquer tipo de processo estatal, a saber: contraditório, publicidade, equidistância entre interesses, lealdade e boa-fé, instrumentalidade e pluralidade de competências decisórias (instâncias decisórias).

Podemos, de maneira bem sucinta, correlacionar esses princípios ao processo legislativo. Vejamos:

O *contraditório* está presente no cotidiano do processo legislativo, seja nos debates situação/oposição, seja no enfrentamento de questões específicas como as questões pró ou contra porte de armas, ruralistas e ambientalistas e tantos outros possíveis exemplos.

A *publicidade* é expressamente determinada nos Regimentos Internos das Casas Legislativas, razão da existência dos Diários do Senado, do Congresso e da Câmara dos Deputados.

A *equidistância entre interesses* é manifesto pelo papel exercido pela Presidência da Mesa ou das Comissões, evidente na prerrogativa regimental que estabelece que o Presidente não deve votar nas votações ostensivas, exceto em caso de empate (RISF art. 48, XXIII, por exemplo).

A *lealdade e boa-fé* pode ser observada no Regimento Interno do Senado Federal em várias ocasiões, mas citamos os princípios regimentais da impossibilidade de adoção de procedimentos diversos do estabelecido

no regimento sem anuência unânime dos senadores (art. 412, III) ou a divulgação antecipada da pauta de votações (art. 412, XI).

A *instrumentalidade* presente na adoção de procedimentos urgentes que suprimam interstícios e prazos, respeitada publicidade mínima e o conhecimento antecipado das matérias que serão votadas (art. 337).

A *pluralidade de instâncias*, evidente na possibilidade de deliberação terminativa nas comissões cabendo a possibilidade de recurso ao Plenário.

Por fim é mister destacar que o conceito de devido processo legislativo, utilizado em ampla jurisprudência do STF e na doutrina, já é um forte indicador da correlação (ou mesmo do diálogo) entre o estudo da processualística na esfera jurisdicional e sua aplicação no campo do Processo Legislativo. Como afirma o professor Marques Neto, "o processo é condição *sine qua non* para a legitimidade do exercício do poder num Estado Democrático de Direito".

Portanto, entendemos que há um sólido fundamento a amparar a proposta de modelagem do Processo Legislativo como uma relação jurídica análoga ao processo jurisdicional, compartilhando diversos princípios e pressupostos que buscaremos explicitar nas seções seguintes.

3 A Ontologia de Fundamentação Unificada - UFO

Para a construção de uma ontologia de domínio do Processo Legislativo e, em especial, para a conceituação de seus aspectos fundamentais ora propostos é indispensável uma base teórica de sustentação.

O trabalho de modelagem ontológica de um domínio não se trata de uma mera catalogação com base em padrões, uma vez que o modo como usar as classes tem mudado. Cada vez mais elas são mais gerais e construídas com abordagem teórica mais consistente (RUPP; BURKE, 2004). Por isso, a compreensão de questões metafísicas é necessária quanto para entender e descrever o mundo, como para construí-lo. Questões sobre sistematização e classificação de conceitos já são levantadas há séculos no âmbito jurídico. No âmbito brasileiro, por exemplo, o eminente jurista doutrinador Augusto Teixeira de Freitas (1816–1883), no século XVIII, discutia essas questões no âmbito de seus projetos que englobavam o Código Civil brasileiro (FREITAS, 2003):

> Classificar não é simplesmente dividir, não é sómente designar por uma denominação commum os individuos que se assemelhão á certos respeitos. A divisão é instrumento da analyse; mas, terminada esta, e conhecidas as differenças e sememelhanças dos entes ou factos observados; a classificação,instrumento da synthese, os distribue, não em series

> isoladas, mas em classes superiores e inferiores, subordinadas umas ás outras, e fumando um verdadeiro systema, que não é um simples *arranjamento* e *superposição*, mas um tecido, um aggregado de partes reciprocamente unidas
>
> Para haver essa união, bem se vê, que a classificação só pode ser o producto de uma idéa geral, de um principio dominante. Se a classificação não é fundada sobre um principio, não existe systema; porque as classes ja não dependem umas das outras.. A escolha desse principio é a grande difficuldade, e determina as classificações *naturaes* e as *artificiaes* (FREITAS, 1859, p. 52-53).

Posto dessa forma, para explicar e classificar é necessário haver fundamentos que norteiam os sistemas de classificações, além de um sistema geral de classificações que permita que outros conceitos sejam especializados, compostos, relacionados e derivados de conceitos mais abstratos, elementares ou gerais. Conforme argumenta-se em Araujo (2017, p. 68), esse é um papel que, contemporaneamente no âmbito de modelagens conceituais, atribui-se às ontologias de fundamentação: o de servir de fundamento para o desenvolvimento de ontologias mais especializadas. Dessa forma, conforme Oberle (2006, p. 43-49), ontologias de fundamentação tratam-se de ontologias genéricas, usadas como referência para desenvolvimento de ontologias de domínio mais específicos. Elas fornecem classes conceituais que servem para permitir que situações e entidades específicas possam ser explicadas a partir de algum tipo de relação com aqueles conceitos mais gerais, como a partir de conceitos como rigidez, princípio de identidade e relações de inerência, exemplificação, instância, subsunção etc.

Dentre as ontologias de fundamentação utilizadas contemporaneamente para descrever ontologias de domínio, destacam-se a BWW (acrônimo de *Bunge, Wand e Weber*) (WAND; WEBER, 1990; WAND; WEBER, 1995; WEBER, 1997) e a *Unified Foundational Ontology* (UFO) (GUIZZARDI, 2005). De acordo com Verdonck e Gailly (2016), embora a primeira seja a mais utilizada como fundamentos de ontologias de domínio, a UFO tem ganhado destaque nos últimos anos, com uma taxa de crescimento mais acentuada que a primeira.

A UFO é derivada de uma síntese e do aperfeiçoamento de duas outras ontologias de fundamentação: GFO/GOL (DEGEN et al., 2001; DEGEN et al., 2011) e OntoClean/DOLCE (WELTY; GUARINO, 2001; GUARINO; WELTY, 2002; GUARINO; WELTY, 2008), mas é atualmente desenvolvida de forma independente daquelas. A UFO consiste em um sistema de categorias que reflete características da realidade reconhecidas pelo senso comum, como relações parte-todo, princípio de identidade, dependência e relações entre objetos, entre outras. Ela é desenvolvida com uma abordagem

interdisciplinar inspirada na Ontologia Formal, Lógica Filosófica, Linguística e Psicologia Cognitiva. A UFO é organizada em três camadas:

a) **UFO-A** (*Ontology of Endurants*): uma ontologia de endurantes, ou de objetos sem partes temporais, que persistem no tempo e mantêm sua identidade. A principal referência sobre UFO-A é a referida tese de Guizzardi, mas também alguns trabalhos posteriores como Guizzardi (2006), Guizzardi (2009), entre outros. Em Zamborlini (2011) encontra-se uma introdução mais atualizada à UFO-A, além da obra original do autor;

b) **UFO-B** (*Ontology of Perdurants*): uma ontologia que incrementa a UFO-A com a introdução da noção de perdurantes, caracterizados como eventos com duração delimitada, compostos de partes temporais (GUIZZARDI; FALBO; GUIZZARDI, 2008; GUIZZARDI et al., 2013);

c) **UFO-C** (*Ontology of Social and Intentional Entities*): com base na UFO-A e na UFO-B, a UFO-C é uma ontologia de *entidades sociais* e intencionais, incluindo aspectos linguísticos. A UFO-C é desenvolvida essencialmente por Guizzardi (2006), mas também está presente em Guizzardi, Falbo e Guizzardi (2008), Guizzardi e Guizzardi (2010), Bringuente, Falbo e Guizzardi (2011) e em outras obras.

A UFO é também usada como fundamentação ontológica na aplicação de um método de avaliação à Linguagem de Modelagem Conceitual UML 2.0 (*Unified Modeling Language*). Tal processo deu origem à linguagem chamada de OntoUML, que é então uma versão estendida da UML 2.0 ontologicamente bem-fundamentada. Os diagramas presentes neste texto são construídos na linguagem OntoUML conforme apresentada em (GUIZZARDI, 2005) e publicações posteriores.

Nossa escolha de usar UFO justifica-se, portanto, pela aplicação bem-sucedida desta ontologia fundamental em trabalhos anteriores para avaliar, redesenhar e fundamentar ontologias, modelos e estruturas de várias áreas de pesquisa (NARDI et al., 2013), bem como pelo fato de a ontologia se mostrar adequada para descrever conceitos do Processo Legislativo abordados neste artigo.

Dentre os conceitos introduzidos pela UFO-A, a ideia de *Relator* é extensamente explorada em modelos conceituais porque denotam a soma mereológica dos papeis exercidos por ao menos dois indivíduos que são mediados pela relação (GUIZZARDI, 2005, p. 240). Agregam características, como deveres, direitos, privilégios e prerrogativas, inerentes à relação constituída entre as entidades relacionadas. Dessa forma, *Relators* são *moments*, e por isso são indivíduos existencialmente dependentes dos entidades que mediam.

Este artigo desenvolve conceitos endurantes, típicos da UFO-A, ao mesmo tempo que abre a possibilidade de extensão e de desenvolvimento de futuros trabalhos englobando aspectos sociais (UFO-C) e perdurantes (UFO-B) da ontologia de domínio analisada.

4 Uma ontologia de domínio para o processo legislativo brasileiro

À luz da UFO podemos considerar ao menos três alternativas de modelagem do conceito de "processo" como instância de qualquer materialização das espécies definidas nas normas que regulam o Processo Legislativo na esfera federal.

A primeira alternativa considera o "processo" como um *kind*, ou seja como um endurante rígido conforme já descrito na seção anterior . Nesse caso há como que uma confusão entre o processo e sua materialização na forma de um conjunto de pastas e documentos destinados ao registro dos eventos ocorridos e das manifestações textuais. Ao caracterizar o "processo" como um *kind* é inescapável essa associação, ainda que acrescida de alguns metadados informacionais relevantes. Eis a razão pela qual surge um conceito de matéria legislativa extremamente flexível, como mencionamos anteriormente, capaz de abarcar qualquer coisa que se queira fazer tramitar na forma de um "processo físico".

A segunda alternativa é descrever o "processo" como uma sequência de ações, eventos e movimentações ocorridos, adotando, portanto um enfoque perdurante na explicação conceitual do "processo". Ainda que possível, essa modelagem carece de limitações pois somente a soma das descrições de cada evento ocorrido não contém em si mesma a necessária liga que individualize e narre cada processo. O processo legislativo é um conjunto de eventos que podem estar relacionados a um, nenhum ou vários processos. Por exemplo, em uma sessão plenária muitos eventos se sucedam afetando vários processo. Porém, a sucessão de eventos deve estar vinculada àquilo que se quer narrar individualmente sobre cada processo, dando orgnização e perspectiva capaz de descrever cada instância de "processo".

Por fim, a terceira alternativa, que desenvolveremos a seguir, representa o processo como uma relação materializada. Tal alternativa, como visto anteriormente, encontra-se em sintonia com um ramo majoritário da doutrina relacionada à processualística jurídica.

Adotando a perspectiva de que o Processo Legislativo pode ser entendido como todos os processos relacionados às atribuições do Poder Legislativo, conforme preconizado por FERREIRA FILHO (1995), identificamos vários tipos de processo legislativo descritos em dispositivos da Constituição Federal de 1988 (CF88) e do Regimento Interno do senado Federal (RISF) vigentes:

a) Processos legiferantes, incluindo o processo legislativo orçamentário e as normas internas das Casas Legislativas (CF88 arts. 61 a 69);

b) Processos relacionados a atribuições não-legiferantes do Congresso, como as indicações de autoridades, apreciação de concessões de radiodifusão, adoção de tratados internacionais, autorizações de operações financeiras, suspensão de normas inconstitucionais etc (CF88 arts. 48 a 52);

c) Processos fiscalizatórios, incluindo a avaliação de políticas públicas, tomadas de contas, audiências públicas, análise de relatórios de agências e outros órgãos públicos, e o recebimento de comunicações diversas, como as comunicações de decisões do TCU (CF88, arts. 70 a 74);

d) Processos judiciários, como nos casos de *impeachment* e cassação de mandato (CF88, art. 80);

e) Processos investigatórios, como ocorre nas Comissões Parlamentares de Inquérito (CF88, art. 58, § 3º);

f) Processos pré-legiferantes, como a apreciação de sugestões enviadas pela sociedade ou indicações nos termos regimentais (RISF art. 102-E, I)

g) Processos regimentais, cujo objeto é o tratamento de incidentes processuais internos, como a apreciação de requerimentos de natureza processual (RISF, art.258, p. ex.);

h) Processos de caráter hermenêutico, aqueles derivados de questões de ordem que estabelecem novas práticas processuais não positivadas (RISF, arts. 403 a 408);

i) Processos de acompanhamento de mandatos, que autorizam licenças, afastamentos, missões e representações (RISF, arts. 40 a 44-A);

j) Processos de gestão do funcionamento dos colegiados legislativos, como realização de sessões e reuniões, designação de integrantes de colegiados (RISF, arts. 106 e seguintes, p. ex.).

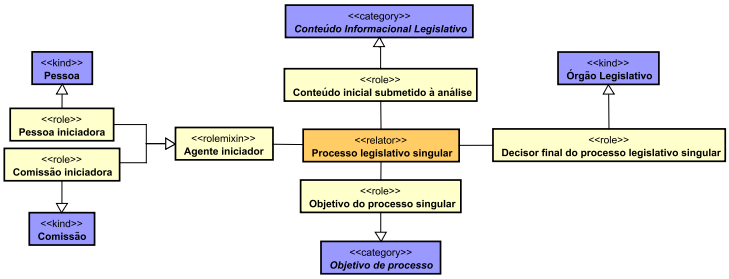
Dada essa grande variedade de processos, é possível identificar um conjunto de elementos definidores capaz de exprimir de maneira unificada o conceito de processo?

A partir da análise das características de cada tipo de processo acima mencionado, à luz da teoria processual anteriormente exposta, é possível descrever o processo legislativo como uma relação material em que participam, de um lado, um **agente iniciador**, seja individual ou colegiado, conforme as regras constitucionais ou regimentais, de outro lado um **órgão**

legislativo apreciador, seja o Congresso Nacional, uma de suas Casas ou quaisquer comissões ou órgãos legislativos regimentalmente estabelecidos, tendo um **objetivo** definido em norma jurídica a ser atingido a partir de um **conteúdo informacional** submetido ao seu conhecimento. A presença desses 4 elementos definidores é capaz de descrever as diversas relações processuais que se materializam como instâncias do processo legislativo.

A Figura 1 consiste no diagrama com a visão geral do processo legislativo fundamental.

Figura 1 – Diagrama com visão geral do processo legislativo fundamental



Fonte: Os autores

Por exemplo, o Processo Legislativo em sentido estrito, ie, a formação das leis, pode ser visto a partir de diferentes granularidades. Em seu aspecto macro é uma relação que consiste de uma **agente iniciador** constitucionalmente definido (um deputado, p.ex.), apresentando à deliberação do Congresso Nacional (**órgão legislativo apreciador**)por meio de uma de suas Casas (a Câmara, p. ex.) um texto (**conteúdo informacional**) preparado conforme as normas vigentes (Lei Complementar nº 95/1998, Regimento Interno, entre outras) com vistas a tornar esse texto uma norma cogente geral (**objetivo**, qual seja, de transformar o texto em uma lei).

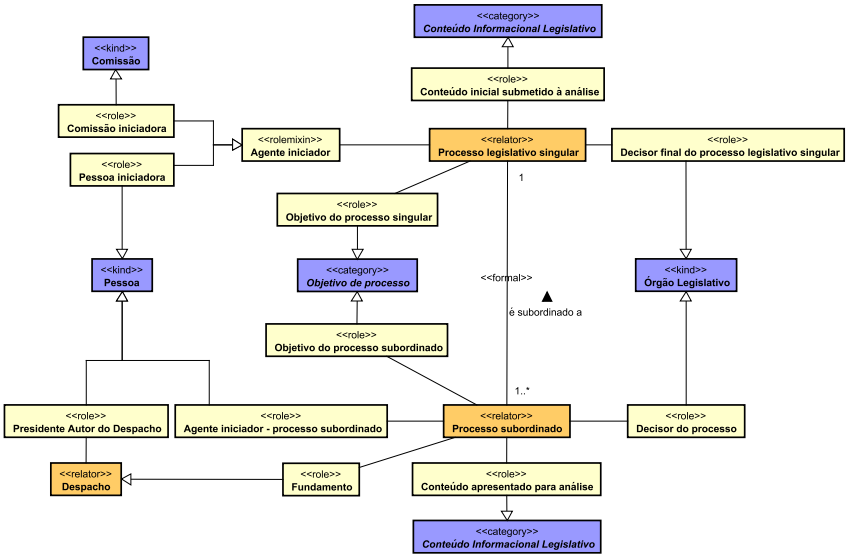
Tal descrição ontológica pode ser aplicada a qualquer caso de Processo Legislativo na esfera mais geral estabelecida na Constituição, em especial aos tipos definidos no artigo 59 da Carta Magna. Mas e as demais espécies de "processo legislativo" como as decorrentes de norma regimental (indicações, sugestões, requerimentos processuais) ou as comunicações ou solicitações diversas que são recebidas pelo Legislativo com vistas ao desempenho de suas atribuições? Também nesses casos os mesmos elementos constituintes da relação material estão presentes, visto que sempre há um agente iniciador, uma órgão legislativo apreciador, um objetivo a ser atingido e um conteúdo informacional submetido pelo agente iniciador.

4.1 A extensão do conceito contempla subprocessos

Do ponto de vista didático, diversos autores buscam sistematizar o Processo Legiferante conforme etapas ou fases, de modo a permitir melhor compreensão do fluxo de tramitação de cada tipo de proposição legislativa. Tais etapas ou fases, normalmente compreendem uma etapa de instrução pelas comissões, emendamento, deliberação em Plenário, publicação, revisão por outra Casa Legislativa e a etapa de promulgação/sanção ou veto.

Entretanto, todas essas tentativas carecem de uma base teórica que claramente as qualifique e distingua uma em relação às outras. A concepção de subprocesso, conforme definido na Figura 2, permite que, de modo flexível, se defina a evolução do processo legislativo a partir da constituição de relações processuais específicas ao longo da apreciação pelos diversos colegiados legislativos.

Figura 2 – Diagrama com visão do processo legislativo fundamental e seus subprocessos, ou processos subordinados



Fonte: Os autores

Assim, a presente solução comporta a possibilidade de se construir os fluxos ou etapas do processo legislativo a partir da construção de relações processuais internas, existencialmente dependentes da relação processual principal, com as mesmas características ontológicas constitutivas da relação

material já definida como "processo". Portanto, os processos legislativos são *relators* que podem ser simples em sua constituição ou compostos por diversos "processos" dentro de "processos".

Cabe ressalvar que não se trata de uma relação mereológica, mas da possibilidade de, conforme o grau de observação desejado, expressar a existência de "subprocessos" dentro de um processo, de maneira coerente (e não-arbitrária) com a definição conceitual ora adotada.

Podemos tomar como exemplo o processo de análise de um projeto de lei pelas comissões permanentes do Senado Federal. Ao despachar a matéria para as comissões o Presidente do Senado atua como **agente iniciador**, apresentando a cada comissão, enquanto **órgão legislativo apreciador**, o **conteúdo informacional** da proposição e das eventuais emendas que venham a ser submetidas ao colegiado, com vistas ao atingimento de um **objetivo** específico em cada comissão, seja a oferta de um parecer para instrução do Plenário, seja um deliberação conclusiva em substituição ao Colegiado Pleno. Observa-se, portanto, que as mesmas características ontológicas da relação material (*relator* definido para a instância de um processo legislativo principal pode ser reproduzida para os processos legislativos subordinados a ele, compondo, assim, uma sucessão dinâmica coerente e didática relativa ao Processo Legsialtivo *in casu*. O *Despacho* é um *relator* que fundamenta os processos subordinados porque atribui ao Órgão Legislativo a obrigação de decidir sobre o conteúdo apresentado para a análise à luz do objetivo do processo subordinado. O *Despacho* possui outros conceitos relacionados que não são apresentados no modelo.

5 Conclusão

O presente estudo procurou estabelecer uma ontologia sobre "processo" no âmbito do Processo Legislativo Federal a partir de fundamentos da doutrina jurídica processual transposta para o Processo Legislativo. Ao estabelecer o processo legislativo *in concretu* como uma espécie de relação material, é possível construir os conceitos mais basilares para o desenvolvimento de soluções computacionais que atinjam ao menos dois grandes objetivos simultâneos:

a) Informar com precisão e clareza os atores interessados no Processo Legislativo, sejam parlamentares, grupos de interesse, agentes públicos e, principalmente a própria sociedade, sobre os seus elementos constitutivos; e

b) Fornecer o suporte ao trabalho para as diversas equipes internas à Casa Legislativa que atuam no apoio ao Processo Legislativo e no registro histórico dos processos, individualmente considerados.

Neste primeiro trabalho, foi dado o primeiro passo para a construção de uma ontologia completa para o Processo Legislativo, considerando ainda, a necessidade, em trabalhos futuros, de expandir o presente estudo para definir conceitualmente a participação dos diversos atores no Processo, a natureza das relações materiais constituídas ao longo do ciclo de vida dos processos, o papel dos eventos legislativos na constituição, modificação ou extinção da relação material processual. Além disso, futuros estudos pode ser desenvolvidos para compreender os resultados do processo legislativo, em especial as políticas públicas derivadas de leis como relações materiais, ou seja, uma futura construção de uma ontologia de domínio para políticas públicas definidas em lei.

Referências

ALVIM, J. E. *Teoria geral do processo*. [S.l.: s.n.], 2016.

ARAUJO, L. C. *Uma Linguagem para Formalização de Discursos com base em Ontologias*. Brasília: Senado Federal, 2017. (Coleção de Teses, Dissertações e Monografias de Servidores do Senado Federal). ISBN 978-85-7018-902-8.

BOBBIO NORBERTO, M. N. P. G. Dicionário de política. *Dicionário de política*, v. 1, p. 382, 1998.

BRINGUENTE, A. C. de O.; FALBO, R. de A.; GUIZZARDI, G. Using a foundational ontology for reengineering a software process ontology. *Journal of Information and Data Management*, v. 2, n. 3, p. 511–526, 2011. Disponível em: <http://seer.lcc.ufmg.br/index.php/jidm/article/view/164>. Acesso em: 13 set 2013.

CARVALHO, K. G. *Técnica Legislativa: Revista, Atualizada e Ampliada*. [S.l.]: Editora del Rey, 2007.

CINTRA, A. d. A.; GRINOVER, A. P.; DINAMARCO, C. R. *Teoria geral do processo*. [S.l.]: Malheiros editores, 2005.

DEGEN, W. et al. GOL: Towards an Axiomatized Upper-Level Ontology. In: SMITH, B.; GUARINO, N. (Ed.). *Proceedings of FOIS'01*. Ogunquit, Maine, USA: ACM Press, 2001.

DEGEN, W. et al. GOL: A general ontological language. In: . [s.n.], 2011. Disponível em: <http://ontology.buffalo.edu/smith/articles/gol_fois2001.pdf>. Acesso em: 8 jun 2014.

FERREIRA FILHO, M. G. Do Processo Legislativo. *São Paulo: Saraiva –3ª ed., atual*, 1995.

FREITAS, A. T. de. *Nova apostila à censura do Senhor Alberto de Moraes Carvalho sobre o Projecto do Código Civil Portuguez*. Rua dos Inválidos, 61B, Rio de Janeiro: Typographia Universal de Laemmert, 1859.

FREITAS, A. T. de. *Consolidação das leis civis*. Ed. fac-sim. Brasília: Senado Federal, Conselho Editorial, 2003. (Coleção história do direito brasileiro. Direito civil). Prefácio de Ruy Rosado de Aguiar.

GUARINO, N.; WELTY, C. Evaluating ontological decisions with OntoClean. *Commun. ACM*, ACM, New York, NY, USA, v. 45, n. 2, p. 61–65, fev. 2002. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/503124.503150>. Acesso em: 21 set 2013.

GUARINO, N.; WELTY, C. A. An overview of OntoClean. In: STAAB, S.; STUDER, R. (Ed.). *Handbook on Ontologies*. Second edition. EUA: Springer, 2008. p. 201–220.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*. Tese (Doutorado) — Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 2005. Disponível em: <http://www.loa.istc.cnr.it/Guizzardi/SELMAS-CR.pdf>.

GUIZZARDI, G. Agent roles, qua individuals and the counting problem. In: GARCIA, A. et al. (Ed.). *Software Engineering for Multi-Agent Systems IV*. EUA: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 3914). p. 143–160.

GUIZZARDI, G. The problem of transitivity of part-whole relations in conceptual modeling revisited. In: *Proceedings of the 21st International Conference on Advanced Information Systems Engineering*. Berlin, Heidelberg: Springer-Verlag, 2009. (CAiSE '09), p. 94–109. ISBN 978-3-642-02143-5. Disponível em: <http://dx.doi.org/10.1007/978-3-642-02144-2_12>. Acesso em: 20 set 2013.

GUIZZARDI, G.; FALBO, R. de A.; GUIZZARDI, R. S. S. Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE software process ontology. In: *Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS)*. Recife: [s.n.], 2008. p. 244–251.

GUIZZARDI, G. et al. Towards ontological foundations for the conceptual modeling of events. In: NG, W.; STOREY, V.; TRUJILLO, J. (Ed.). *Conceptual Modeling*. [S.l.]: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 8217). p. 327–341. ISBN 978-3-642-41923-2.

GUIZZARDI, R. S. S. *Agent-Oriented Constructivist Knowledge Mangament*. Tese (Doutorado) — Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 2006. Disponível em: <http://doc.utwente.nl/56967/1/r_guizzardi.pdf>. Acesso em: 11 set 2013.

GUIZZARDI, R. S. S.; GUIZZARDI, G. Applying the UFO ontology to design an agent-oriented engineering language. In: *Proceedings of the 14th East European Conference on Advances in Databases and Information Systems*. Berlin, Heidelberg: Springer-Verlag, 2010. (ADBIS'10), p. 190–203. ISBN 3-642-15575-8, 978-3-642-15575-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1885872.1885890>. Acesso em: 29 jul 2014.

MARQUES NETO, F. de A. Ensaio sobre o processo como disciplina do exercício da atividade estatal. In: DIDIER JR., F.; JORDãO, E. F. (Coord.). *Teoria do Processo*: Panorama doutrinário mundial. Salvador, BA: Editora Podium, 2008. p. 261–285.

MIKVA, A. J.; LANE, E. *Legislative Process*. [S.l.]: Aspen Publishers, 2002.

NARDI, J. C. et al. Towards a commitment-based reference ontology for services. In: *IEEE EDOC (Enterprise Computing Conference)*. Vancouver: [s.n.], 2013.

OBERLE, D. *Semantic Management of Middleware (Semantic Web and Beyond: Computing for Human Experience)*. Secaucus, NJ, USA: Springer-Verlag, 2006. ISBN 0387276300.

RUPP, N.; BURKE, D. From catalogers to ontologists. *The Serials Librarian*, v. 46, n. 3-4, p. 221–226, 2004. Disponível em: <http://dx.doi.org/10.1300/J123v46n03_04>. Acesso em: 22 out 2014.

VERDONCK, M.; GAILLY, F. Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling. In: COMYN-WATTIAU, I. et al. (Ed.). *Conceptual Modeling*. Cham: Springer International Publishing, 2016. p. 83–97. ISBN 978-3-319-46397-1. Disponível em: <https://doi.org/10.1007/978-3-319-46397-1_7>. Acesso em: 16 mar 2018.

WAND, Y.; WEBER, R. Mario bunge's ontology as a formal foundation for information systems concepts. In: WEINGARTNER, P.; DORN, G. J. W. (Ed.). *Studies on Mario Bunge's Treatise*. Rodopi, 1990, (Poznań studies in the philosophy of the sciences and the humanities). ISBN 9789051831870. Disponível em: <https://books.google.com.br/books?id=w-soo6UdwhAC>.

WAND, Y.; WEBER, R. On the deep structure of information systems. *Information Systems Journal*, John Wiley and Sons, v. 5, 1995.

WEBER, R. *Ontological Foundations of Information Systems*. Melbourne: Coopers & Lybrand Research Methodology Monograph No. 4, Coopers & Lybrand, 1997.

WELTY, C.; GUARINO, N. Supporting Ontological Analysis of Taxonomic Relationships. *Data and Knowledge Engineering*, v. 39, p. 51–74, 2001.

WILSON, W. Congretional government 57. John Hopkins Univ. Press, 1981.

ZAMBORLINI, V. C. *Estudo de Alternativas de Mapeamento de Ontologias da Linguagem OntoUML para OWL*: Abordagens para representação da informação temporal. Dissertação (Mestrado) — Universidade do Espírito Santo. Departamento de Informática, Vitória, 2011.

Construção de um Sistema de Gestão de Normas metamodelado baseado em Ontologia de Fundamentação

Jideão José Vieira Filho[*,†,‡] Edna Dias Canedo[†]
João Alberto de Oliveira Lima[*] Lauro César Araujo[*]
Daniel de Mello Viero[*] Fabrício Fernandes Santana[*]
Flávio Roberto de Almeida Heringer[*] Hudson de Martim[*]
Wagner Rodrigues Teixeira[*]

Resumo

Este trabalho trata da utilização de metamodelagem para a construção de um sistema de gestão de normas flexível de modo que uma ontologia de domínio embasada numa ontologia de fundamentação seja suficiente para definir o comportamento do sistema especialista.

Palavras-chave: Gerenciamento de atos normativos. Ontologia de fundamentação. Metamodelagem.

1 Introdução

A descrição de atos normativos e suas relações é uma atividade essencialmente descritiva, em que detalhes de propriedades e relacionamentos podem ser retratados e modelados de formas não previstas. Da mesma forma, instituições jurídicas criadas por atos normativos – tais como leis, decretos, portarias, resoluções etc. – são inumeráveis, uma vez que se consistem na própria materialidade da norma. Igualmente, pessoas físicas e jurídicas, municípios, entidades geográficas, como rios e outros acidentes, são constantemente citados em normas. Diante disso, abstrações são uma forma

[*]Senado Federal
[†]Universidade de Brasília
[‡]jideao@senado.gov.br

de abordar problemas referentes a questões inumeráveis, pois permitem a concentração em aspectos essenciais de um contexto qualquer sem a necessidade de conhecer características marginais ou menos importantes.

Um modelo é uma abstração de um sistema. Metamodelos são modelos com grau de abstração ainda maior e que permitem a representação de outros modelos, ou seja, metamodelos são modelos de modelos. Embora em casos concretos de desenvolvimento de sistemas possa não ser uma regra, idealmente um metamodelo bem fundamentado permite que alterações no modelo representado não exijam modificações no metamodelo. Nesse sentido, o modelo construído sobre o metamodelo deve ser o mais estável possível, mas deve permitir evoluções. Para isso, é importante ser alicerçado em conceitos sólidos. Conceitos sólidos dependem de uma ontologia estável.

Este texto tem como objetivo descrever a construção de um sistema metamodelado como solução para a complexidade do domínio da gestão de normas jurídicas. O modelo flexível, construído para o metamodelo, é embasado na *Unified Foundational Ontology*, uma ontologia bem fundamentada filosoficamente e que vem sendo cada vez mais adotada.

2 Unified Foundational Ontology (UFO)

A UFO é uma ontologia de fundamentação construída e fundamentada em teorias de áreas como ontologia formal na Filosofia, ciência cognitiva, lógica filosófica e linguística, e desenvolvida para fornecer fundamentação ontológica para linguagens gerais de modelagem conceitual. Ela começou como a unificação de duas ontologias: a GFO (*Generalized Formalized Ontology*) e a Ontoclean/DOLCE[1]. Apesar de importantes, essas ontologias possuem problemas relacionados ao desenvolvimento de fundamentos ontológicos para linguagens de modelagem conceitual de uso geral, como EER, UML, ORM. A UFO, por sua vez, unifica essas ontologias de modo a superar as limitações detectadas e utilizando-se de seus pontos positivos (GUIZZARDI; WAGNER, 2010) (GUIZZARDI et al., 2015) (GUIZZARDI, 2005).

A ontologia é dividida em três subontologias que abrangem diferentes aspectos da realidade. São elas:

a) UFO-A: ontologia de endurantes que trata de aspectos de modelagem conceitual de estruturas. É organizada como uma ontologia de quatro categorias, fundamentada no trabalho de Lowe (2007), que engloba teorias de estruturas de tipos e taxonômicas (GUIZZARDI et al., 2004; GUIZZARDI, 2012) conectada a uma teoria de identificadores de objetos (incluindo uma semântica formal

[1] <http://www.ontoclean.org>

em lógica formal quantificada de sortais (GUIZZARDI, 2015)), relações todo-parte (GUIZZARDI, 2007; GUIZZARDI, 2009), propriedades intrinsicamente particularizadas, atributos e espaços de valores de atributos (GUIZZARDI; WAGNER; HERRE, 2004; GUIZZARDI; ZAMBORLINI, 2014), relações e propriedades relacionais particulares (GUIZZARDI; WAGNER, 2008; COSTAL; GÓMEZ; GUIZZARDI, 2011) e papéis (GUIZZARDI; ROLES, 2006);

b) UFO-B: ontologia de perdurantes (eventos e processos) que trata de aspectos tais como mereologia de perdurantes, ordem temporal de perdurantes, participação de objetos em perdurantes, causalidade, mudança e a conexão entre perdurantes e endurantes por meio de disposições (GUIZZARDI et al., 2013);

c) UFO-C: ontologia de entidades sociais, fundamentada nas UFO-A e UFO-B. A UFO-C trata de noções como crenças, desejos, intenções, objetivos, ações, compromissos e reivindicações, papeis sociais e *relators* sociais, dentre outros (GUIZZARDI; GUIZZARDI, 2010).

2.1 Por que escolheu-se a UFO?

Verdonck e Gailly (2016) realizaram um estudo da frequência de referências em artigos científicos às principais ontologias de fundamentação conhecidas pela comunidade acadêmica. Os autores analisaram uma amostra de duzentos artigos e os resultados estão descritos na Tabela 1. Observa-se que dentre as ontologias citadas, a UFO é a segunda mais referenciada, atrás apenas da BWW (acrônimo de Bunge, Wand e Weber) (WAND; WEBER, 1990; WAND; WEBER, 1995; WEBER, 1997), criada por Wand e Weber e fundamentada no trabalho do filósofo argentino Mário Bunge (BUNGE, 1977).

Entretanto, ao se analisar a distribuição temporal das referências, percebe-se, conforme ilustrado na Figura 1, que a partir de 2010, cinco anos após o surgimento da UFO, iniciou-se um processo de progressiva preferência à UFO em detrimento quando comparada à BWW.

Verdonck e Gailly definiram três perspectivas que agrupam entidades modeladas por ontologias. O objetivo desta definição era compreender como as duas ontologias (BWW e UFO) são utilizadas e em que situação. As três perspectivas são:

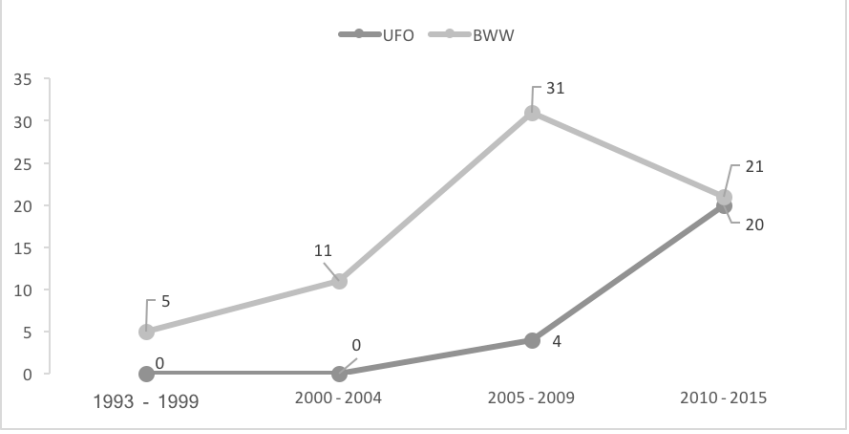
a) *perspectiva estática*: as entidades dessa perspectiva tendem a descrever a estrutura de um sistema. São frequentemente representados como entidade, coisa ou objeto. Possuem identificador único e, frequentemente, possuem atributos, os quais representam

Tabela 1 – Frequência de referências a ontologias de fundamentação na literatura.

Ontologia de fundamentação	**Referências**
BWW	68
UFO	24
General Formal Ontology (GFO)	4
DESO	3
DOLCE	3
Chisholm Ontology	2
SUMO	2
BORO	1
Basic Formal Ontology (BFO)	1
Searle's Ontology	1

Fonte: Adaptada de Verdonck e Gailly (2016)

Figura 1 – Utilização da BWW e da UFO entre 1993 e 2015.



Fonte: Adaptado de Verdonck e Gailly (2016)

valores específicos da entidade. De modo geral, essas entidades estão conectadas a outras por vários relacionamentos;

b) *perspectiva dinâmica*: agrupa fenômenos que representam transições e tempo. Esses fenômenos são geralmente traduzidos em construções que descrevem eventos e processos;

c) *perspectiva comportamental e funcional*: os principais fenômenos

> que pertencem a esta perspectiva são estados sociais e suas relações ou transformações. Fenômenos sociais são representados por entidades como atores ou papeis que eles assumem e ações que eles executam. Regras e objetivos também podem ser categorizados como fenômenos sociais, desde que influenciem o comportamento de um ator. A transformação de um estado pode ser definido como uma atividade, baseada num conjuntos de fenômenos que os transformam em outro conjunto de fenômenos. Outros termos utilizados são função e tarefas.

Tabela 2 – Utilização das ontologias UFO e BWW por tipo de perspectiva.

Perspectiva	BWW	%	UFO	%
Estática	40	52,0%	8	27,6%
Dinâmica	19	24,7%	5	17,2%
Comportamental/funcional	18	23,3%	16	55,2%

Fonte: Adaptada de Verdonck e Gailly (2016)

Conforme sumarizado na Tabela 2, mais da metade dos fenômenos tratados pela BWW são classificados na perspectiva estática, enquanto fenômenos de perspectiva dinâmica e de perspectiva comportamental e funcional representam cerca de 25% dos fenômenos tratados, cada categoria. Na UFO, mais da metade dos fenômenos são classificados como comportamental e funcional.

Provavelmente, a BWW foi a primeira tentativa de desenvolver uma fundamentação ontológica para modelagem conceitual. Como ela foi alicerçada na ontologia de Bunge (BUNGE, 1977), o qual, por sua vez, tinha como principal objetivo construir uma ontologia de ciências, a sua ontologia possui algumas limitações para modelagem conceitual. A ontologia de Bunge foi construída no domínio do mundo material. Ela consiste de objetos materiais que possuem propriedades físicas independentes da percepção humana. Ela não abrange objetos conceituais, os quais são necessários para representar conceitos fundamentais para a comunicação humana (ALLEN; MARCH, 2007). A modelagem conceitual, por sua vez, trata da representação de aspectos do mundo físico e social de modo a melhorar a comunicação e o entendimento de determinado tema (MYLOPOULOS, 1992). Deste modo, para desenvolver fundamentações ontológicas para modelagem conceitual, deve-se observar aspectos de cognição e linguística humana profundamente.

3 Descrição do problema

Os objetos gerenciados pelo sistema de gestão de normas são criados por meio de relacionamentos. Por exemplo, a Advocacia Geral da União (AGU) foi instituída pela Lei Complementar nº 73 de 1993 (LCP 73/1993). Deste modo, existe um relacionamento – aqui chamado de `Declaração de Instituição` – que materializa a conexão entre AGU e LCP 73/1993. AGU participa do relacionamento no papel de `criado por` e LCP 73/1993 no papel de `criador de`.

Na abordagem tradicional de desenvolvimento de sistemas – com bases de dados relacionais –, um sistema gerenciador de atos normativos geraria um modelo extenso, complexo, com muitas tabelas e muitos detalhes transversais. Existem muitas categorias de objetos para se gerenciar e muitas ligações entre eles. É inviável conhecer todos os detalhes desses objetos e das suas relações no momento da modelagem do sistema. A qualquer momento, uma nova propriedade ou um novo objeto não previsto pode ser identificado. Para gerenciar estas novas entidades identificadas, necessitam-se alterações no modelo de dados (criação ou alteração de tabelas), na lógica do *software*, nas telas do sistema etc.

Nesse sentido, caso existisse um sistema capaz de se adaptar às modificações do modelo criado para representar o domínio de descrição e estruturação de atos jurídicos, os problemas citados seriam mitigados.

4 Proposta

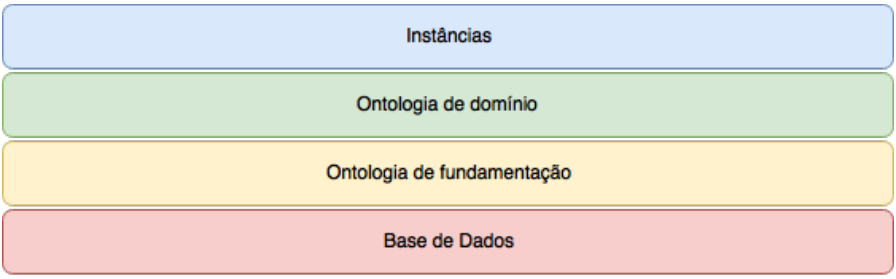
A proposta para solução do problema descrito na seção 3 apresenta um sistema metamodelado apoiado em uma ontologia de fundamentação. Este sistema é capaz de se adequar a evoluções que ocorrem na ontologia que reflete o domínio do sistema.

Toda modelagem de sistema faz algum compromisso ontológico (GUIZZARDI; HALPIN, 2008). Não importa como é feita a modelagem, ela sempre é retratada por uma ontologia. De acordo com Collier (1994), o contrário de uma ontologia não é a ausência de uma ontologia e, sim, uma ontologia ruim. Portanto, para construir uma ontologia de domínio compreensível, adaptável, mas que seja estável e de fácil manutenção, é importante que esta ontologia seja uma descrição coerente do domínio modelado.

A presença de uma ontologia de fundamentação como referência para a construção de uma ontologia de domínio possibilita uma modelagem com conceitos mais consistentes e rígidos. Isso leva a crer que a evolução do sistema aos problemas específicos do domínio será mais coerente com o mundo físico e social representado. Deste modo, as mudanças no modelo do

domínio poderão ser incorporadas ao sistema de modo mais harmônico com os conceitos já presentes no sistema.

Figura 2 – Camadas do sistema proposto



Fonte: Os autores

Desta forma, é proposta uma solução com quadro camadas (Figura 2). Cada camada é brevemente descrita abaixo:

a) *Camada do modelo de banco de dados:* metamodelada de modo a permitir o mapeamento da ontologia de fundamentação, da ontologia de domínio e das instâncias para o banco de dados relacional;

b) *Camada da ontologia de fundamentação:* utilizada como referência para construir uma ontologia de domínio consistente e coerente;

c) *Camada da ontologia de domínio:* contém as classes de indivíduos que representa o domínio modelado. Construída sobre os alicerces da *Unified Foundational Ontology* (UFO);

d) *Camada de instâncias:* representam os objetos que instanciam as classes da ontologia de domínio.

O foco deste trabalho está na descrição da camada da Base de Dados, uma vez que a Ontologia de Fundamentação já é objeto de pesquisa do professor Dr. Giancarlo Guizzardi e as camadas superiores representam os próprios dados cadastrados no sistema.

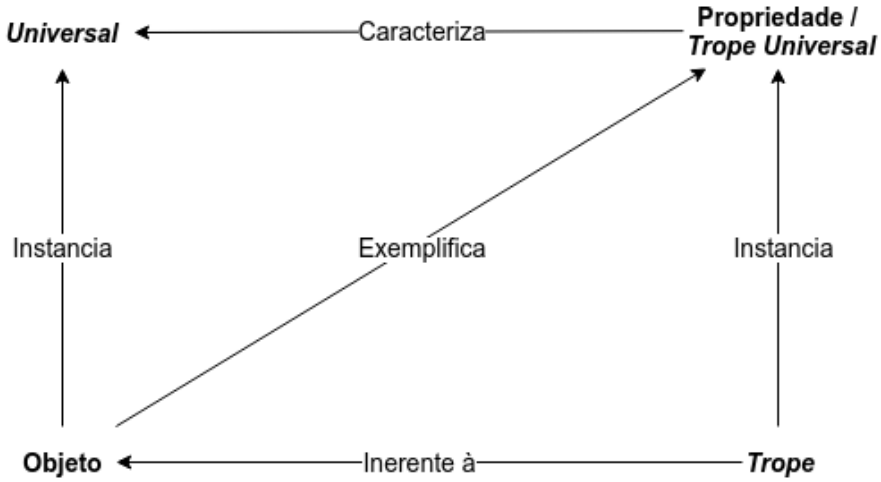
4.1 Metamodelo de banco de dados

O objetivo da abordagem adotada é flexibilizar o sistema construído de tal modo a permitir que novas propriedades e tipos pudessem, a qualquer momento, serem inseridos ou modificados na ontologia de domínio e refletidos no banco de dados. Para isso, precisava-se de um modelo que gerenciasse objetos transversalmente ao gerenciamento de suas propriedades. Assim,

ao se descobrir uma propriedade que não foi modelada em uma classe de objeto, esta propriedade poderia ser adicionada na ontologia de domínio, sendo consequentemente inserida na base de dados (sem a necessidade de criação de novas colunas ou tabelas) e o sistema se adaptaria aos novos dados.

Constrói-se o modelo de dados do sistema proposto a partir do quadrado ontológico aristotélico, o qual é exemplificado pela UFO conforme demonstrado na Figura 3.

Figura 3 – Quadrado aristotélico



Fonte: Adaptado de Guizzardi e Zamborlini (2014)

A UFO distingue as categorias dos Particulares (*Particular*) – ou Individuais – e dos Universais (*Universal*) – ou simplesmente Classes, Tipos. Os particulares existem na realidade, possuem identidade única e exemplificam os Universais.

Os Universais são entidades utilizadas na explicação metafísica do que é, para os particulares, compartilhar uma característica, atributo ou qualidade ou ser categorizado como um tipo ou como uma classe natural (ZIMMERMAN, 2017).

5 Exemplo de aplicação da proposta adotada

Uma lei ordinária é um ato normativo primário que contém muitas propriedades, dentre elas:

a) evento de criação: assinatura que fundamenta a existência da norma, no qual registra-se a propriedade data de assinatura;

b) número: identificação da norma (exemplo: Lei nº 8.112);

c) ementa: descrição que resume o conteúdo da norma;

d) publicação original: a publicação em que a norma foi veiculada para conhecimento público;

e) relacionamentos (não obrigatórios) com outras entidades, inclusive outros atos normativos.

Para se construir um sistema de gestão de normas com uma base de dados tradicional, pode-se criar uma tabela de banco de dados para cada tipo de entidade modelada do sistema. Cada propriedade conhecida, seria então mapeada como uma coluna da tabela do banco de dados.

Assim, para este exemplo, uma abordagem possível seria ter, pelo menos, uma tabela para norma, outra para publicação e outra para cada tipo de entidade que se relaciona com esta norma (exemplo: processo de origem, acórdãos que consideram a norma constitucional/inconstitucional, etc.). A tabela norma precisaria pelo menos das seguintes colunas: data de assinatura, número, ementa, código da publicação original (que faria referência à tabela de publicações), código das instâncias das demais entidades que participam de relacionamentos (outras leis, processos, dentre outras, em que cada entidade seria representada por uma nova tabela). O exemplo dado é uma simplificação considerável de apenas uma entidade. Estima-se que para modelar um sistema de gestão de normas com centenas entidades, seriam necessárias centenas de tabelas, cada uma delas com várias colunas.

Além da quantidade de tabelas e da complexidade do modelo relacional citado, a modelagem tradicional desse sistema exigiria o conhecimento prévio de muitos detalhes das entidades que seriam gerenciadas pelo mesmo. Caso uma nova propriedade fosse identificada após a utilização do sistema em produção, no mínimo uma nova coluna precisaria ser criada. Em casos mais complexos, se esta nova propriedade envolvesse uma entidade não prevista, haveria a necessidade de se criar novas tabelas, reavaliar os relacionamentos existentes, criar novos relacionamentos.

Com a abordagem proposta, não existe a necessidade do conhecimento prévio de todos os detalhes das entidades do domínio do sistema. Cada entidade poderia ser mapeada em tabelas baseadas na figura Figura 3. Os Universais da figura são as entidades do domínio do sistema. Lei numerada, Lei complementar, Conceito, Pessoa, Organização, Partido Político são exemplos de Universais. Os Objetos são as instâncias dos universais. O objeto Lei nº 8112/1990 é instância do universal Lei Numerada; o conceito

Língua Brasileira de Sinais (LIBRA) é instância do universal Conceito; Rui Barbosa é instância do universal Pessoa. E assim por diante.

As propriedades de lei ordinária seriam cadastradas na tabela Propriedade/ *Trope* Universal. São Propriedades de uma lei: número, ementa, publicação original, evento de assinatura. Outro tipo de entidade (por exemplo: Pessoa) teria suas propriedades (nome, data de nascimento, etc) também cadastradas na tabela Propriedade/ *Trope* Universal. Esta tabela teria necessidade de uma coluna com a identificação do Universal ao qual a propriedade caracteriza.

Ademais, os valores atribuídos às propriedades de um indivíduo seriam cadastradas na tabela *Trope*. Por exemplo, a Lei nº 8.112 de 1990 precisa ter valores atribuídos para cada propriedade obrigatória da tabela Propriedade/ *Trope* Universal que caracteriza o Universal Lei Numerada. Desta forma, na tabela Trope os seguintes valores são alimentados: 8.112 para número da norma; 11/12/1990 para data do evento de criação (data de assinatura); o texto: "Dispõe sobre o regime juridico dos servidores publicos civis da união, das autarquias e das fundações publicas federais" para ementa e assim por diante. A tabela *Trope* teria pelo menos duas referências: uma para o Objeto ao qual é inerente e outra para a Propriedade/ *Trope* Universal à qual instancia.

Por fim, caso uma nova Entidade (Universal) ou Propriedade seja identificada, não há necessidade de se criar uma nova tabela, nem promover alteração no software. Basta que os dados da entidade e/ou das propriedades sejam inseridas nas respectivas tabelas. Nenhuma outra modificação será necessária para se adaptar à utilização das entidades/propriedades identificadas.

6 Conclusão

A abordagem adotada mostrou-se benéfica para o domínio da gestão de normas. O metamodelo de banco de dados construído permitiu a evolução progressiva do sistema sem necessidade de alteração de código e sem necessidade de alteração da estrutura do banco de dados. Toda evolução é feita por meio de alterações na ontologia de domínio.

Alterar apenas a ontologia de domínio, neste caso, traz benefícios importantes para o sistema: não existe necessidade de retrabalho para construção de interfaces gráficas, modificação de estruturas de dados; pode-se construir o sistema sem conhecer complemente o seu domínio; permite a criação de uma interface gráfica para possibilitar ao usuário alterar o comportamento do sistema; dentre outros.

Referências

ALLEN, G.; MARCH, S. A critical assessment of the bunge-wand-weber ontology for conceptual modeling. 06 2007.

BUNGE, M. *Ontology I: The Furniture of the World*. Boston, MA: D. Reidel Publishing Company, 1977. v. 3. (Treatise on Basic Philosophy, v. 3).

COLLIER, A. Critical realism: an introduction to roy bhaskar's philosophy. 1994.

COSTAL, D.; GÓMEZ, C.; GUIZZARDI, G. Formal semantics and ontological analysis for understanding subsetting, specialization and redefinition of associations in uml. In: SPRINGER. *International Conference on Conceptual Modeling*. [S.l.], 2011. p. 189–203.

GUIZZARDI, G. *Ontological foundations for structural conceptual models*. [S.l.]: CTIT, Centre for Telematics and Information Technology, 2005.

GUIZZARDI, G. Modal aspects of object types and part-whole relations and the de re/de dicto distinction. In: SPRINGER. *Advanced Information Systems Engineering*. [S.l.], 2007. p. 5–20.

GUIZZARDI, G. The problem of transitivity of part-whole relations in conceptual modeling revisited. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2009. p. 94–109.

GUIZZARDI, G. Ontological meta-properties of derived object types. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2012. p. 318–333.

GUIZZARDI, G. Logical, ontological and cognitive aspects of object types and cross-world identity with applications to the theory of conceptual spaces. In: *Applications of Conceptual Spaces*. [S.l.]: Springer, 2015. p. 165–186.

GUIZZARDI, G.; HALPIN, T. Ontological foundations for conceptual modelling. *Appl. Ontol.*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 3, n. 1-2, p. 1–12, jan. 2008. ISSN 1570-5838. Disponível em: <http://dl.acm.org/citation.cfm?id=1412417.1412423>.

GUIZZARDI, G.; ROLES, A. *Qua Individuals and The Counting Problem, Invited Chapter in Software Engineering of Multi-Agent Systems, vol. IV, P. Giorgini, A. Garcia, C. Lucena, R. Choren*. [S.l.]: Springer-Verlag, 2006.

GUIZZARDI, G.; WAGNER, G. What's in a relationship: an ontological analysis. *Conceptual Modeling-ER 2008*, Springer, p. 83–97, 2008.

GUIZZARDI, G.; WAGNER, G. Using the unified foundational ontology (ufo) as a foundation for general conceptual modeling languages. In: *Theory and Applications of Ontology: Computer Applications*. [S.l.]: Springer, 2010. p. 175–196.

GUIZZARDI, G. et al. Towards ontological foundations for conceptual modeling: The unified foundational ontology (ufo) story. v. 10, 12 2015.

GUIZZARDI, G. et al. Towards ontological foundations for the conceptual modeling of events. In: SPRINGER. *International Conference on Conceptual Modeling*. [S.l.], 2013. p. 327–341.

GUIZZARDI, G. et al. An ontologically well-founded profile for uml conceptual models. In: SPRINGER. *Advanced Information Systems Engineering*. [S.l.], 2004. p. 1–122.

GUIZZARDI, G.; WAGNER, G.; HERRE, H. On the foundations of uml as an ontology representation language. In: SPRINGER. *EKAW*. [S.l.], 2004. v. 3257, p. 47–62.

GUIZZARDI, G.; ZAMBORLINI, V. Using a trope-based foundational ontology for bridging different areas of concern in ontology-driven conceptual modeling. *Science of Computer Programming*, Elsevier, v. 96, p. 417–443, 2014.

GUIZZARDI, R.; GUIZZARDI, G. *Ontology-Based Transformation Framework from Tropos to AORML In in Social Modeling for Requirements Engineering, P. Giorgini, N. Maiden, J. Mylopoulos, E. Yu (eds.) Cooperative Information Systems Series*. [S.l.]: MIT Press, Boston, 2010.

LOWE, E. J. *The Four-Category Ontology: A Metaphysical Foundation for Natural Science*. [S.l.]: Clarendon Press, 2007.

MYLOPOULOS, J. Conceptual modelling and telos. In: LOUCOPOULOS, P.; ZICARI, R. (Ed.). *Conceptual Modeling, Databases, and Case: An Integrated View of Information Systems Development*. New York, NY, USA: John Wiley & Sons, Inc., 1992. cap. 2, p. 49–68.

VERDONCK, M.; GAILLY, F. Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling. In: _____. *Conceptual Modeling: 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings*. Cham:

Springer International Publishing, 2016. p. 83–97. ISBN 978-3-319-46397-1. Disponível em: <https://doi.org/10.1007/978-3-319-46397-1_7>.

WAND, Y.; WEBER, R. Mario bunge's ontology as a formal foundation for information systems concepts. In: WEINGARTNER, P.; DORN, G. J. W. (Ed.). *Studies on Mario Bunge's Treatise*. Rodopi, 1990, (Poznań studies in the philosophy of the sciences and the humanities). ISBN 9789051831870. Disponível em: <https://books.google.com.br/books?id=w-soo6UdwhAC>.

WAND, Y.; WEBER, R. On the deep structure of information systems. *Information Systems Journal*, John Wiley and Sons, v. 5, 1995.

WEBER, R. *Ontological Foundations of Information Systems*. Melbourne: Coopers & Lybrand Research Methodology Monograph No. 4, Coopers & Lybrand, 1997.

ZIMMERMAN, D. W. Universal. In: *Encyclopædia Britannica*. Encyclopædia Britannica, inc., 2017. Disponível em: <https://www.britannica.com/topic/universal>.

Encerramento

Os trabalhos desenvolvidos no grupo de estudos são valiosos para o Senado Federal, porque demonstram a viabilidade de ideias inovadoras que provavelmente não seriam experimentadas em projetos de desenvolvimento de software tradicionais, que naturalmente possuem significativas restrições de escopo, recursos e prazos. As atividades de estudo e pesquisa possibilitam investigar soluções de ponta para problemas conhecidos. Com os resultados obtidos nesta pesquisa, é possível propor projetos que aprimoram significativamente a forma de trabalho com informação legislativa e jurídica. Da mesma forma, os estudos possibilitaram validar tecnologias, como a de reconhecimento de faces, que, devido aos ótimos resultados, são candidatas a serem adotadas no curto prazo em soluções de reuniões de comissões parlamentares.

Do ponto de vista acadêmico, o material produzido pelo grupo de pesquisa vai além deste documento, pois todos os softwares produzidos e os *datasets* com resultados das transformações nas normas jurídicas e nas imagens das faces dos parlamentares, por serem informações públicas, estão disponíveis a toda comunidade acadêmica e servem de base para pesquisas futuras que podem trazer resultados ainda mais significativos.

Após revisões e adequações pertinentes, os capítulos com contribuições acadêmicas mais acentuadas poderão ser formatados pelos autores para submissão a revistas ou eventos especializados nos temas abordados.