Instituto Serzedello Corrêa
Escola Superior do Tribunal de Contas da União

MARCOS DAVID DRACH

APLICAÇÃO DE TÉCNICAS DE RECONHECIMENTO E CLASSIFICAÇÃO DE ENTIDADES NOMEADAS PARA A EXTRAÇÃO DE INFORMAÇÕES EM ACÓRDÃOS

Brasília

2020

MARCOS DAVID DRACH

APLICAÇÃO DE TÉCNICAS DE RECONHECIMENTO E CLASSIFICAÇÃO DE ENTIDADES NOMEADAS PARA A EXTRAÇÃO DE INFORMAÇÕES EM ACÓRDÃOS

Trabalho de conclusão do curso de pós-graduação *lato sensu* em Análise de Dados para o Controle realizado pela Escola Superior do Tribunal de Contas da União como requisito para a obtenção do título de especialista.
Orientador: Prof. Dr. Thiago de Paulo Faleiros

Brasília

2020

REFERÊNCIA BIBLIOGRÁFICA

DRACH, Marcos David. **Aplicação de técnicas de Reconhecimento e Classificação de Entidades Nomeadas para a extração de informações em acórdãos**. 2019. Trabalho de Conclusão de Curso (Especialização em Análise de Dados para o Controle) – Escola Superior do Tribunal de Contas da União, Instituto Serzedello Corrêa, Brasília DF.

CESSÃO DE DIREITOS

NOME DO AUTOR: Marcos David Drach
TÍTULO: Aplicação de técnicas de Reconhecimento e Classificação de Entidades Nomeadas para a extração de informações em acórdãos.
GRAU/ANO: Especialista/2019

_____

Marcos David Drach
mddrach@gmail.com

Ficha catalográfica

Drach, Marcos David

Aplicação de técnicas de Reconhecimento e Classificação de Entidades Nomeadas para a extração de informações em acórdãos / Marcos David Drach; orientador, Thiago de Paulo Faleiros, 2019.

82 p.

Monografia (especialização) - Escola Superior do Tribunal de Contas da União, Curso de Especialização em Análise de Dados para o Controle, Brasília, 2019.

Inclui referências.

1.Análise de Dados. 2. Mineração de Texto. 3. Aprendizado de Máquina. 4. Reconhecimento de Entidades Nomeadas. I. Faleiros, Thiago de Paulo. II. Escola Superior do Tribunal de Contas da União. Especialização em Análise de Dados para o Controle. III. Título.

MARCOS DAVID DRACH

APLICAÇÃO DE TÉCNICAS DE RECONHECIMENTO E CLASSIFICAÇÃO DE ENTIDADES NOMEADAS PARA A EXTRAÇÃO DE INFORMAÇÕES EM ACÓRDÃOS

Trabalho de conclusão do curso de pós-graduação *lato sensu* em Análise de Dados para o Controle realizado pela Escola Superior do Tribunal de Contas da União como requisito para a obtenção do título de especialista.

Brasília, 30 de março de 2020.

Banca Examinadora:

_____

Prof. Dr. Thiago de Paulo Faleiros

Orientador

Instituto Serzedello Corrêa - TCU

_____

Prof. Dr. Edans Flavius de Oliveira Sandes

Instituto Serzedello Corrêa - TCU

Dedico aos meus pais, que me deram a base para sonhar,
e a minha esposa, por cujo amor continuo sonhando.

AGRADECIMENTOS

À minha esposa, Penha, pelo incentivo, carinho e paciência, sem os quais este projeto seria inviável.

Aos meus pais, Abel e Irinéa, pelos ensinamentos e inspiração.

Ao TCU e em especial aos dirigentes e organizadores do Curso de Especialização em Análise de Dados para o Controle, pela viabilização de relevante iniciativa para o alargamento da capacitação tecnológica dos auditores.

Aos professores, pelos conhecimentos transmitidos e pela dedicação à honrosa missão de ensinar.

Ao Professor Dr. Thiago de Paulo Faleiros, pela orientação na elaboração deste trabalho.

À Mauro Giacobbo, pelas sugestões e suporte ao longo desta jornada.

"*Quanto mais aumenta nosso conhecimento, mais evidente fica nossa ignorância.*"

(John F. Kennedy, 1962)

RESUMO

O Reconhecimento e Classificação de Entidades Nomeadas (*Named Entity Recognition and Classification* – NERC) é uma área do processamento de linguagem natural que cuida da identificação, extração e categorização de nomes próprios e termos de interesse em textos não estruturados. Surgido na década de 90 como uma tarefa de extração de informação, o NERC tem se beneficiado da constante evolução das técnicas de análise de dados e mineração textual. Dos primeiros sistemas baseados em regras ao estado-da-arte em algoritmos de aprendizado de máquina, o NERC tem sido utilizado em diversas aplicações, tais como pesquisa e classificação de conteúdo, tradução de idiomas, normalização de vocabulários, *chatbots*, sistemas de recomendação e pesquisa semântica, entre outros. No entanto, para que a técnica possa produzir melhores resultados, é necessário o treinamento do modelo de aprendizagem sobre o tipo de corpus desejado. Este trabalho teve por objetivo avaliar a eficácia da técnica aplicada sobre os acórdãos do Tribunal de Contas da União, utilizando-se modelo próprio treinado com uso da biblioteca spaCy. Os resultados demonstraram acurácia acima de 85% em 7 das 9 classes de entidades pesquisadas. Entre os benefícios, espera-se reduzir a carga manual atualmente requerida para o cadastramento das deliberações, bem como possibilitar a extração, a consolidação, o cruzamento e a análise não triviais dos dados de deliberação com vistas à produção de conhecimento institucional.

Palavras-chave: Análise de Dados. Mineração de Texto. Aprendizado de Máquina. Reconhecimento de Entidades Nomeadas.

ABSTRACT

The Recognition and Classification of Named Entities (NERC) is an area of natural language processing that takes care of identifying, extracting and categorizing proper names and terms of interest in unstructured texts. Emerged in the 90s as an information extraction task, NERC has benefited from the constant evolution of data analysis and text mining techniques. From the first rule-based systems to state-of-the-art machine learning algorithms, NERC has been used in several applications, such as content search and classification, language translation, vocabulary standardization, chatbots, recommendation systems and semantic research, among others. However, for the technique to produce better results, it is necessary to train the learning model on the type of corpus desired. This work aims to evaluate the effectiveness of the technique applied on the judgments of the Federal Court of Accounts, using its own model trained using the spaCy library. The results showed accuracy above 85% in 7 of the 9 classes of entities evaluated. Among the benefits, it is expected to reduce the manual load currently required for registering the deliberations, as well as enabling the extraction, consolidation, crossing and non-trivial analysis of the deliberation data with a view to the production of institutional knowledge.

Keywords: Data Analysis. Text Mining. Learning Machine. Named Entity Recognition.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BiLSTM	Bidirectional Long Short-Term Memory
BOW	Bag of Words
BTCU	Boletim do Tribunal de Contas da União
CBOW	Continuous-bag-of-words
CGU	Controladoria-Geral da União
CRF	Conditional Random Field
CRISP-DM	Cross Standard Process for Data Mining
DOU	Diário Oficial da União
EL	Entity Linking
GED	Gerenciamento Eletrônico de Documentos
IBGE	Instituto Brasileiro de Geografia e Estatística
KDD	Knowledge Discovery in Databases
LSTM	Long Short-Term Memory
MLP	Multi Layer Perceptron
MUC	Message Understanding Conferences
NEL	Named-Entity Linking
NER	Named Entity Recognition
NERC	Named Entity Recognition and Classification
NLTK	Natural Language Toolkit
PLN	Processamento de Linguagem Natural
POS	Part-of-Speech
PPA	Plano Plurianual
RNN	Recurrent neural network
SBD	Sentence Boundary Detection
SECINF	Serviço de Cadastramento de Informação
SEPROC	Secretaria de Gestão de Processos
SESES	Secretaria de Sessões
TCU	Tribunal de Contas da União
TF-IDF	Term Frequency - Inverse Document Frequency
UD	Universal Dependencies

SUMÁRIO

1 INTRODUÇÃO

Segundo o Relatório de Atividades do TCU – edição 2018, o TCU apreciou conclusivamente mais de 5.200 processos de controle externo e expediu mais de 24.200 acórdãos por ano, em média, entre os anos de 2014 e 2018 (TCU, 2019, p. 19 e 39). As deliberações do Tribunal produzidas por meio de seus acórdãos possuem uma riqueza de dados não estruturados, tais como funções de governo, políticas públicas, programas temáticos e ações fiscalizadas.

Tal volume de informação é consumida pelo próprio Tribunal nos processos de trabalho subsequentes à fase de julgamento (pós-julgamento), em especial na expedição de comunicações, no monitoramento das deliberações e nas tratativas decorrentes de condenações e sanções, a exemplo do recolhimento de dívidas e multas no âmbito administrativo e da autuação de processos de cobrança executiva.

1.1 PROBLEMA E JUSTIFICATIVA

Embora faça uso intensivo de tecnologia da informação em toda a sua cadeia de produção, o TCU, assim como os órgãos com função jurisdicional, vale-se de procedimento essencialmente manual na redação dos votos consignados pelos ministros relatores de forma a se garantir plena liberdade de expressão.

Com efeito, os julgados do Tribunal contêm informações produzidas em formato textual e não estruturado. Assim, na fase de pós-julgamento, faz-se necessária a identificação e o registro dos elementos constituintes de cada acórdão com o propósito de assegurar a devida persistência daquelas informações em bases de dados e a sua utilização posterior, conforme supracitado.

Ocorre que tal atividade é laboriosa e onerosa. Atualmente, ela é exercida pelo Serviço de Cadastramento de Informações (Secinf), subunidade da Secretaria de Gestão de Processos (Seproc). O Serviço é composto de 10 servidores e 4 estagiários, dedicados integralmente àquela tarefa. O perfil predominantemente manual da atividade, aliado à crescente restrição da força de trabalho e, portanto, de sobrecarga operacional, propicia um cenário de maior vulnerabilidade a falhas, com prejuízo à integridade dos dados.

Ademais, os dados da produção do Tribunal são regularmente divulgados à sociedade por meio de painéis executivos, relatórios e publicações diversas, tais como o Relatório Anual

de Atividades do TCU (TCU, 2019a) e o Relatório de Políticas e Programas de Governo (TCU, 2019b).

Estes dados, além de necessários às atividades inerentes ao pós-julgamento, produzem um conjunto de informações de alta significância estratégica para a Casa. Em especial, quando compiladas e relacionados a outras fontes (benefício do controle, orçamento federal, indicadores sociais, etc.), produz-se informações de valor agregado com potencial de subsidiar a criação de conteúdo abrangente acerca da atuação do Tribunal nas mais diversas dimensões de interesse organizacional, tais como financeira-orçamentária, política-administrativa ou social. A extração, consolidação, cruzamento e análise não triviais dos dados de deliberação têm, portanto, o condão de produzir conhecimentos potencialmente úteis para uma melhor compreensão a respeito da efetividade dos trabalhos realizados pela Casa, com efeitos auspiciosos no aperfeiçoamento do planejamento institucional, bem como na divulgação dos resultados institucionais.

Assim, a presente pesquisa busca prospectar, por meio do Processamento de Linguagem Natural (PLN), em especial da técnica de Reconhecimento e Classificação de Entidades Nomeadas (*Named Entity Recognition and Classification* – NERC[1]), modelo que permita identificar e extrair entidades de interesse institucional a partir das deliberações produzidas pelo Tribunal.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Pretende-se avaliar a aplicabilidade de utilização da técnica de NERC para a extração de entidades nomeadas, presentes no conteúdo das determinações e recomendações expedidas pelo TCU a órgãos e entidades, por meio da biblioteca spaCy com uso de modelo próprio treinado em corpus formado pelo conjunto das deliberações expedidas pelo Tribunal desde 2011.

[1] Também conhecida apenas como *Named Entity Recognition* (NER).

1.2.2 Objetivo específico

Com o propósito de se alcançar o objetivo definido para este estudo, estabelece-se como objetivos específicos:

- Conduzir estudos e aprofundar a fundamentação teórica em técnicas de aprendizado de máquina e mineração textual;

- Realizar análise exploratória dos acórdãos do TCU e identificar padrões de formação das entidades de interesse;

- Produzir um corpus anotado tipo *gold standard* de deliberações do Tribunal;

- Avaliar modelo pré-treinado de extração e classificação de entidades nomeadas;

- Gerar modelo customizado para a extração e a classificação de entidades nomeada a partir do treinamento de corpus anotado de deliberações, com uso da biblioteca spaCy;

- Avaliar o desempenho do modelo e identificar a técnica de NERC mais adequada para cada classe de entidade pesquisada.

2 TRABALHOS RELACIONADOS

A pesquisa da literatura buscou identificar estudos relacionados ao uso da técnica de NERC aplicada à língua portuguesa e, preferencialmente, em corpora jurídicos. Embora tratando-se de um universo restrito, foram elencados dois trabalhos na área, destacados a seguir.

Em artigo produzido por servidor da Casa, DUTRA E SILVA (2016) apresenta um breve histórico, discorre sobre as técnicas de aprendizado de máquina à época e, na sequência, cita potenciais aplicações da tecnologia em ações de controle externo e combate à corrupção. Um dos exemplos mencionados refere-se à classificação dos tipos de decisão e à extração de entidades nomeadas na base textual de deliberações, com uso de rede neural simples de Perceptores de Múltiplas Camadas (*Multi Layer Perceptron* – MLP).

Em outro trabalho, ARAUJO et al. (2018) apresentam um conjunto de dados para reconhecimento de entidades nomeadas em documentos legais brasileiros, composto inteiramente de documentos legais. Além de entidades indicativas de pessoa, local, tempo e organização, o conjunto de dados contém rótulos específicos para legislação e jurisprudência. A arquitetura consistiu do modelo bidirecional *Long Short-Term Memory* (LSTM) (HOCHREITER e SCHMIDHUBER, 1997), seguida por uma camada *Conditional Random Field* (CRF) (LAFFERTY et al., 2001), conforme proposto em (LAMPLE et al., 2016). A entrada do modelo utilizou uma sequência de representações vetoriais de palavras individuais construídas a partir de concatenações de palavras e caracteres incorporados (*word and character level embeddings*[2]). A partir de um corpus jurídico próprio, foram anotadas 6 classes de entidades (organização, pessoa, tempo, local, legislação e jurisprudência) em um conjunto de dados composto de 60 documentos, 10 mil sentenças e 270 mil termos (*tokens*). O modelo LSTM-CRF foi treinado, preliminarmente, no corpus em língua portuguesa Paramopama, fornecendo resultados significativamente melhores do que os relatados anteriormente. Na sequência, o modelo foi treinado na base de dados proposta e produziu, como resultado, um F1-score médio de 92,53.

[2] Outros termos encontrados na literatura são modelo semântico de distribuição, representação distribuída, espaço vetorial semântico ou simplesmente espaço de palavras (*word vector*).

3 REFERENCIAL TEÓRICO

A linguagem humana é ambígua e variável. As pessoas são ótimas em produzir e entender a linguagem, e são capazes de expressar, perceber e interpretar significados muito elaborados e sutis. O uso de computação para a compreensão e o tratamento da linguagem é, portanto, altamente desafiador (GOLDBERG, 2017).

Com o avanço dos algoritmos de modelagem estatística, atualmente o aprendizado de máquina supervisionado, no qual um padrão é inferido a partir de pares de entrada e saída pré-anotados, se apresenta como a tecnologia mais promissora para lidar com dados de linguagem.

Enquanto a tarefa de classificação de documentos por meio de regras elaboradas a partir de padrões observados no corpus pode se tornar complexa ou até mesmo inviável, leitores podem facilmente categorizar algumas centenas de exemplos de forma a treinar um algoritmo de aprendizado de máquina. Ainda segundo Goldberg, "os métodos de aprendizado de máquina são excelentes em domínios problemáticos, nos quais é muito difícil definir um bom conjunto de regras, mas a anotação da saída esperada para uma determinada entrada é relativamente simples" (p. 1).

As duas abordagens, juntamente com o método de pesquisa, serão detalhadas na próxima seção.

3.1 RECONHECIMENTO E CLASSIFICAÇÃO DE ENTIDADE NOMEADA

O reconhecimento e a classificação de entidades nomeadas tratam da identificação de diferentes classes de nomes próprios, como nomes de pessoas e empresas, ou tipos especiais, como datas e horas, que podem ser facilmente identificados usando padrões textuais (DALE et al., 2000).

O termo "Entidade Nomeada" foi cunhado para a Sexta Conferência de Compreensão da Mensagem (*Message Understanding Conferences* – MUC-6) (GRISHMAN e SUNDHEIM, 1996) e decorreu da necessidade de se identificar nomes próprios (de pessoas, de organizações, de locais, etc.) e expressões numéricas (hora, data, moeda, percentuais, etc.) de textos não estruturados, como artigos de jornal, para fins comerciais e de defesa. A identificação de referências a essas entidades no texto foi então reconhecida como uma das subtarefas importantes da Extração de Informações e foi denominada "*Named Entity Recognition and Classification*

(NERC)" ou "Reconhecimento e Classificação de Entidades Nomeadas" (NADEAU e SEKINE, 2007).

São três as abordagens para o reconhecimento de entidades nomeadas: pesquisa em fontes externas, baseada em regras e por meio de modelos estatísticos. Esses métodos também podem ser combinados em sistemas híbridos (DOZIER et al., 2010).

O método de pesquisa (*lookup method*) consiste em utilizar uma fonte externa que contenha as entidades de interesse, ou criar uma para este fim, e simplesmente marcar todas as menções no documento a tais entidades com a classe em questão. Por exemplo, se "TCU" estiver em uma lista de nomes de órgãos e aparecer em um documento, ele é marcado como tal. Nomes comuns ou com múltiplos significados, a depender do contexto, podem ser eliminados para garantir a desambiguação. As vantagens da abordagem de pesquisa residem na sua simplicidade de implementação e manutenção, na possibilidade de fazer uso de listas preexistentes e de não requerer nenhum dado de treinamento. As desvantagens são que ele pode gerar muitos falsos positivos, se a lista contiver muitas palavras ambíguas, ou falsos negativos se a lista não é abrangente o suficiente.

A abordagem baseada em regras contextuais codifica regras dedutivas a partir de padrões encontrados nas entidades de interesse. Por exemplo, a palavra "Ministério" seguida de palavras com iniciais em caixa alta pode identificar a ocorrência de um órgão. Ao analisar o corpus, pode-se desenvolver um conjunto dessas regras que reconheça a maioria das instâncias nos dados e não produz muitos falsos positivos. A vantagem do método está no alto nível de precisão que pode ser obtido. No entanto, o desenvolvimento de tais regras pode exigir uma grande quantidade de esforço e se tornar extremamente complexa e até mesmo inviável, em casos de alta variabilidade dos padrões identificados. Ainda, a manutenção desses conjuntos de regras pode ser um desafio, pois muitas vezes as regras possuem interdependências complexas que são fáceis de serem esquecidas, de forma a tornar quaisquer modificações muito arriscadas.

Os modelos estatísticos oferecem uma alternativa às regras contextuais para a codificação de padrões. Características (*features*) de entrada que correspondem aos padrões são desenvolvidas, um modelo estatístico apropriado é escolhido e o modelo treinado usando dados previamente marcados. Assim como as regras contextuais, os modelos estatísticos podem alcançar alta precisão. Por outro lado, o desenvolvimento de tais modelos de aprendizagem supervisionada requer um grande conjunto representativo de dados de treinamento, o que, por sua vez implica considerável esforço de marcação ou anotação.

Em resumo, as vantagens da abordagem estatística sobre as demais são (MISHRA et al., 2016):

- Ponderação de possíveis respostas diferentes em vez de apenas uma, como ocorre na abordagem por regras, produzindo assim resultados mais confiáveis;

- Natural prevalência para os casos mais comuns – os procedimentos de aprendizado usados durante o aprendizado de máquina se concentram automaticamente nos casos mais comuns, enquanto que ao escrever regras manualmente, muitas vezes não é absolutamente óbvio para onde o esforço deve ser direcionado;

- Capacidade de inferência para entradas desconhecidas ou errôneas – os algoritmos de aprendizado de máquina podem fazer de inferência estatística para produzir modelos que lidem com entradas desconhecidas (palavras não presentes na base de treinamento) ou erradas (palavras com erros ortográficos ou omitidas acidentalmente); e

- Precisão do modelo dependente apenas de mais dados de entrada – os algoritmos de aprendizado de máquina podem ser mais precisos simplesmente fornecendo mais dados de entrada. No entanto, sistemas baseados em regras manuscritas só podem ser mais precisos aumentando a complexidade das regras, o que é uma tarefa muito mais difícil. Em particular, há um limite para a complexidade dos sistemas baseados em regras artesanais, além das quais os sistemas se tornam cada vez mais incontroláveis.

Atualmente, a maioria das abordagens é baseada em aprendizado de máquina, onde documentos já classificados são usados para aprender automaticamente uma função de decisão. A maneira como os documentos são representados deriva do modelo de espaço vetorial e os diferentes esquemas de ponderação.

3.2 REDES NEURAIS E APRENDIZAGEM DE MÁQUINA

O aprendizado profundo, um ramo do aprendizado de máquina, baseado no conceito redes neurais, é definido por GOLDBERG (2017) como uma família de técnicas de aprendizado

inspiradas no funcionamento do cérebro e que podem ser caracterizadas como o aprendizado de funções matemáticas diferenciadas, parametrizadas e dispostas em múltiplas camadas encadeadas.

Seu objetivo é aprender a fazer previsões com base em observações passadas. A partir de um grande conjunto de mapeamentos de entrada e saída, os dados de entrada alimentam uma rede que produz transformações sucessivas até que uma transformação final preveja a saída mapeada. Enquanto ao projetista cabe a definição da arquitetura da rede e o provimento de um conjunto adequado de exemplos de entrada e saída, muito do trabalho pesado de aprender a correta representação é realizada automaticamente pela rede.

Ainda segundo o autor, as redes neurais fornecem uma poderosa ferramenta de aprendizado muito atraente para uso em PLN. Existem dois tipos principais de arquiteturas de redes neurais que podem ser combinadas de várias maneiras: redes *feed-forward* e redes recorrentes / recursivas. As redes *feed-forward*, em particular MLPs, permitem trabalhar com entradas de tamanho fixo ou com entradas de comprimento variável nas quais podemos desconsiderar a ordem dos elementos. Ao alimentar a rede com um conjunto de componentes de entrada, ela aprende a combiná-los de maneira significativa. MLPs podem ser usados sempre que um modelo linear foi usado anteriormente. A não linearidade da rede, bem como a facilidade de integrar incorporação de palavras pré-treinada, geralmente produzem uma precisão superior para classificação.

Redes *feed-forward* convolucionais são arquiteturas especializadas que se destacam na extração de padrões locais nos dados: elas são alimentadas com entradas de tamanho arbitrário e são capazes de extrair padrões locais significativos que são sensíveis à ordem das palavras, independentemente de onde elas apareçam na entrada. Eles funcionam muito bem para identificar frases indicativas ou expressões de até um comprimento fixo em frases longas ou documentos.

Redes neurais recorrentes (*Recurrent Neural Network* – RNN) são modelos especializados para dados sequenciais. Estes são componentes de rede que recebem como entrada uma sequência de itens e produzem um vetor de tamanho fixo que resume essa sequência. Como "resumir uma sequência" significa coisas diferentes para tarefas diferentes (ou seja, as informações necessárias para responder a uma pergunta sobre o sentimento de uma frase é diferente da informação necessária para responder a uma pergunta sobre sua gramaticalidade), redes recorrentes raramente são usadas como componente autônomo, e seu poder está em funcionar como

componentes treináveis que podem ser alimentado com outros componentes de rede e treinado para trabalhar em conjunto com eles. Por exemplo, a saída de uma rede recorrente pode ser alimentada em uma rede *feed-forward* que tentará prever algum valor. Redes recorrentes são modelos muito impressionantes para sequências e são, sem dúvida, a opção mais interessante de redes para PLN. Eles permitem abandonar a suposição de Markov (dado o presente, o futuro não depende do passado) prevalecente na PLN por décadas e projetar modelos que podem condicionar frases inteiras, considerando a ordem das palavras quando necessário, e não sofrendo muito com problemas de estimativa estatística decorrentes da escassez de dados. Essa capacidade leva a ganhos impressionantes na modelagem de linguagem – a tarefa de prever a probabilidade da próxima palavra em uma sequência (ou, equivalentemente, a probabilidade de uma sequência), que é a pedra angular de muitos aplicativos de PLN. Redes recursivas estendem redes recorrentes de sequências até árvores.

Um componente importante nas redes neurais para a linguagem é o uso de uma camada de incorporação, um mapeamento de símbolos discretos para vetores contínuos em um espaço dimensional relativamente baixo. Ao incorporar palavras, elas se transformam de símbolos distintos isolados em objetos matemáticos que podem ser operados. Essa representação de palavras como vetores é aprendida pela rede como parte do processo de treinamento. Maiores detalhes são apresentados na seção seguinte.

3.3 INCORPORAÇÃO DE PALAVRAS

Para que textos possam ser utilizados em algoritmos de aprendizado de máquina, cada palavra no corpus deve ser associada a um vetor de *features*. O vetor representa diferentes aspectos da palavra: cada palavra está associada a um ponto em um espaço de vetor.

As abordagens mais usuais são o *Bag of Words* (BOW), que contabiliza o número de ocorrências de palavras em um documento, e a Frequência do Termo - Inverso da Frequência dos Documentos (*Term Frequency - Inverse Document Frequency* – TF-IDF), que acrescenta uma medida compensatória correspondente à pela frequência da palavra no corpus. Nestes modelos, palavras diferentes têm representações diferentes, independentemente de como são usadas.

Já na incorporação de palavras, palavras com o mesmo significado têm uma representação semelhante (BROWNLEE, 2019).

Incorporações de palavras são de fato uma classe de técnicas em que palavras individuais são representadas como vetores com valor real em um espaço vetorial predefinido. Cada palavra é mapeada para um vetor e os valores do vetor são aprendidos de uma maneira que se assemelha a uma rede neural e, portanto, a técnica é frequentemente agrupada no campo do aprendizado profundo.

A chave para a abordagem é a ideia de usar uma representação distribuída densa para cada palavra. Cada palavra é representada por um vetor com valor real, geralmente dezenas ou centenas de dimensões. Isso contrasta com os milhares ou milhões de dimensões necessárias para representações de palavras esparsas, como numa codificação *one-hot*.

A representação distribuída é aprendida com base no uso de palavras. Isso permite que as palavras usadas de maneira semelhante resultem em representações semelhantes, capturando seu significado. Isso pode ser contrastado com a representações mais simples, como o BOW e o TF-IDF.

Os vetores de palavras podem ser usados como um fim em si mesmo (para calcular semelhanças entre termos) ou como uma base representacional para tarefas posteriores da PLN, como classificação de texto, agrupamento de documentos, parte da marcação de fala, reconhecimento de entidade nomeada, análise de sentimentos e assim por diante. (SAHLGREN, 2015).

3.4 SPACY

SpaCy (SPACY, 2017) é uma biblioteca de código aberto gratuita para processamento avançado e em escala industrial de linguagem natural, desenvolvida em Python pela empresa Explosion[3], especializada em inteligência artificial e PLN. Trata-se de uma biblioteca abrangente, com diversas funcionalidades tais como "tokenização", *Part-of-Speech* (POS) *tagging*, *dependency parsing*, "lematização", segmentação (*Sentence Boundary Detection* –SBD), NERC, *Entity Linking* (EL), similaridade, classificação textual, correspondência baseada em regras, treinamento e serialização. A Figura 1 apresenta uma comparação de seu portfólio de recursos com as bibliotecas *Natural Language Toolkit* (NLTK) e CoreNLP.

[3] Para maiores informações, consulte https://explosion.ai/about.

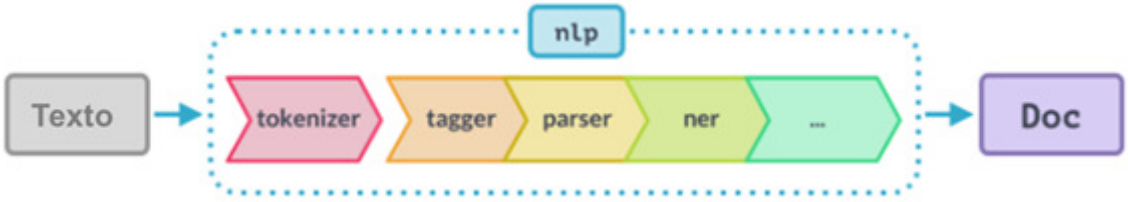
Figura 1– Funcionalidades do spaCy

	SPACY	NLTK	CORENLP
Linguagem de programação	Python	Python	Java / Python
Modelos de rede neural	✓	✗	✓
Vetores de palavra integrado	✓	✗	✗
Suporte a multi-idiomas	✓	✓	✓
Tokenização	✓	✓	✓
Part-of-speech tagging	✓	✓	✓
Segmentação de sentenças	✓	✓	✓
Dependency parsing	✓	✗	✓
Reconhecimento de entidades	✓	✓	✓
Vínculo de entidades	✓	✗	✗
Resolução de coreferências	✗	✗	✓

Fonte: (SPACY, 2017).

Estas funcionalidades, chamadas de componentes, compõem um *pipeline* de processamento, conforme mostrado na Figura 2.

Figura 2 – Pipeline de processamento do spaCy.



Fonte: (SPACY, 2017).

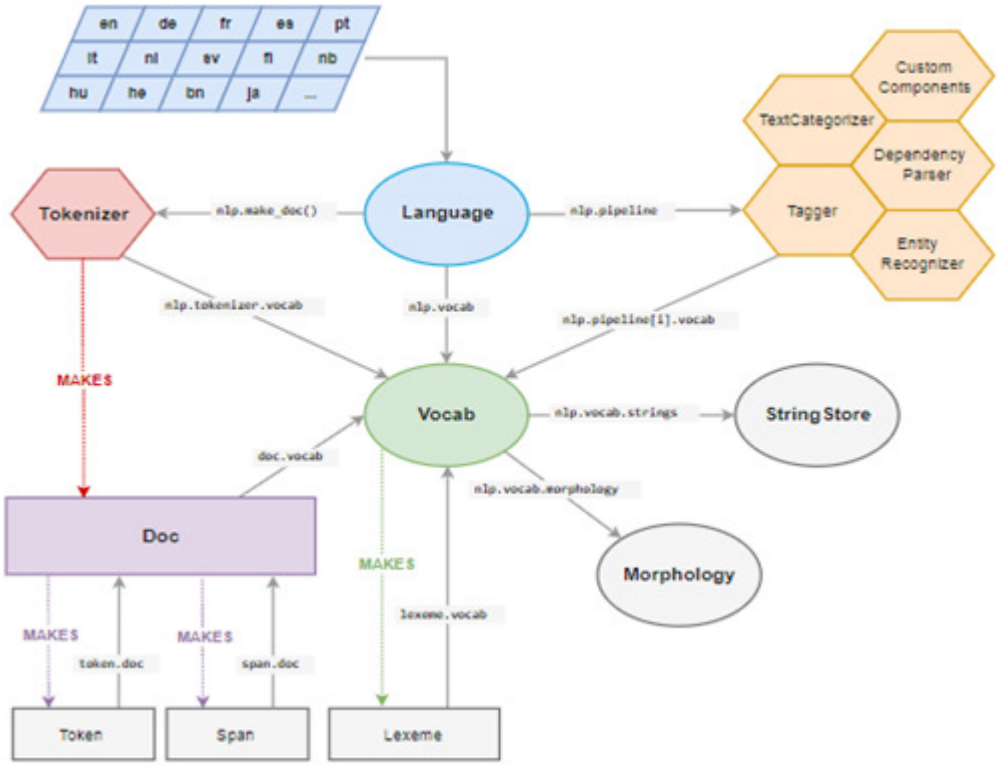
No centro de spaCy está o objeto que contém o *pipeline*, chamada de “nlp", criado pela classe *Language*. Ele coordena os diferentes componentes e organiza o treinamento e a serialização. Também inclui regras específicas do idioma usadas para “tokenizar” o texto em palavras e pontuação. O spaCy suporta uma variedade de idiomas disponíveis.

Ao processar um texto com o objeto nlp, o spaCy cria um objeto “Doc”, abreviação de "documento". O Doc é uma das estruturas centrais de dados da biblioteca e permite acessar informações sobre o texto de maneira estruturada. O objeto se comporta como uma sequência normal de Python e permite iterar sobre seus *tokens* e *spans* (conjunto de *tokens*).

Outra estrutura central é o objeto *Vocab*. Trata-se de um vocabulário compartilhado, no qual o spaCy armazena todos as palavras, suas entidades e características lexicais definidas na classe *Lexeme*. Ao centralizar palavras, vetores e atributos, o spaCy garante economia de memória e uma fonte única de informação.

Na Figura 3 está representada a arquitetura completa da biblioteca.

Figura 3 – Arquitetura da biblioteca spaCy.



Fonte: (SPACY, 2017).

Especificamente quanto ao reconhecimento de entidades, o spaCy fornece um sistema estatístico excepcionalmente eficiente desenvolvido em Python, que pode atribuir rótulos a grupos de *tokens* contíguos. Ele fornece modelos pré-treinados com algumas classes de entidades já disponíveis, bem como permite ainda a adição de novas classes por meio de treinamento.

O sistema foi projetado para oferecer um bom equilíbrio de eficiência, precisão e adaptabilidade. Segundo seus autores, dois artigos revisados por pares em 2015 confirmaram que o spaCy oferece o analisador sintático mais rápido do mundo e que sua precisão está dentro de 1% dos melhores disponíveis. Os poucos sistemas mais precisos são 20x mais lentos ou mais. A Figura 4 mostra um comparativo com outras bibliotecas, relativo ao uso de NERC aplicado ao corpus OntoNotes 5[4] da língua inglesa.

Figura 4 – Quadro comparativo de desempenho entre bibliotecas.

SISTEMA	ANO	TIPO	ACURÁCIA
spaCy `en_core_web_lg` v2.0.0a3	2017	neural	85.85
Strubell et al.	2017	neural	**86.81**
Chiu and Nichols	2016	neural	86.19
Durrett and Klein	2014	neural	84.04
Ratinov and Roth	2009	linear	83.45

Fonte: (SPACY, 2017).

Em novembro de 2017, foi lançada a versão 2.0 (atualmente, 2.2), com 13 novos modelos de redes neurais convolucionais para mais de 7 idiomas. O spaCy v2.0 apresenta novos modelos neurais para marcação, análise e reconhecimento de entidades. Os modelos foram projetados e implementados sob medida, de forma a oferecer um equilíbrio incomparável de velocidade, tamanho e precisão.

[4] O OntoNotes 5.0 é a versão final do projeto OntoNotes, um esforço colaborativo entre a BBN Technologies, a Universidade do Colorado, a Universidade da Pensilvânia e o Instituto de Ciências da Informação da Universidade do Sul da Califórnia, nos EUA, com o objetivo de anotar um grande corpus composto por vários gêneros de texto (notícias, conversas telefônicas, weblogs, etc.).

Embora a arquitetura da rede neural usada não tenha sido publicada, sabe-se que o modelo utiliza uma profunda rede convolucional neutra com conexões residuais, normalização da camada, *maxout* não-linear e uma nova abordagem baseada em transição para análise de entidades nomeadas, proporcionando uma eficiência muito melhor do que a solução de LSTM bidirecional (*Bidirectional Long Short-Term Memory* – BiLSTM) padrão (SPACY, 2017).

O ciclo de treinamento dos modelos no spaCy, descrito a seguir, pode ser construído de forma programática, com uso de suas APIs, ou executado por linha de comando de linha.

Figura 5 – Ciclo de treinamento usado pelo spaCy.

Fonte: (SPACY, 2017)

Primeiro, os pesos são inicializados de forma aleatória. O modelo então realiza as previsões em um lote de exemplos, avalia o resultado com base nas respostas corretas e decide como alterar os pesos para obter melhores previsões na próxima iteração. Finalmente, uma pequena correção nos pesos atuais é realizada e o ciclo se repete com o próximo lote de exemplos, até se alcançar a última iteração (época).

Para evitar que o modelo fique preso em uma solução abaixo do ideal, os dados são misturados (*shuffled*) aleatoriamente para cada iteração.

Outra estratégia muito comum ao se fazer a descida do gradiente estocástico e implementada pelo spaCy é a divisão dos dados de treinamento em lotes de vários exemplos, conhecidos como *minibatching*, o que otimiza a estimativa do gradiente. Caso os parâmetros do treinamento sejam ajustados somente após o consumo de todos os dados, levará mais tempo para a atualização do modelo e maior demanda por recursos de processamento. Por outro lado, se os parâmetros forem ajustados a cada instância de descida do gradiente estocástico, as atualizações do modelo serão muito ruidosas e o processo não será computacionalmente eficiente.

Além do modelo estatístico, vale citar que o spaCy também suporta mecanismos sofisticados de correspondência baseados em regras e que permitem encontrar palavras e frases

de forma análoga ao uso de expressões regulares, com uso de listas de terminologia ou ainda por meio de dicionário de padrões. Tais recursos podem ser usados para melhorar a precisão dos modelos estatísticos, predefinindo *tags*, entidades ou limites de sentença para *tokens* específicos. Os modelos estatísticos geralmente respeitam essas anotações predefinidas o que as vezes, segundo seus autores, melhora a precisão de outras decisões. Os componentes baseados em regras também podem ser usados após um modelo estatístico para corrigir erros comuns. Na Figura 6, os autores apresentam algumas diretrizes para a seleção da melhor abordagem.

Figura 6 – Quadro comparativo de predição estatística x baseada em regras.

Predição estatística x regras

	Modelos estatísticos	Sistemas baseados em regras
Aplicação	generalização baseado em exemplos	dicionário com número finito de exemplos
Exemplos do mundo real	nomes de produtos, nomes de pessoas, relacionamentos sujeito/objeto	países do mundo, cidades, fármacos, raça de cães
Funcionalidades do spaCy	reconhecimento de entidades, dependency parser, part-of-specch tagger	Tokenizer, Matcher, PhraseMatcher

Fonte: (SPACY, 2017).

Para o presente trabalho, decidiu-se pela escolha desta biblioteca em razão dos seguintes fatores:

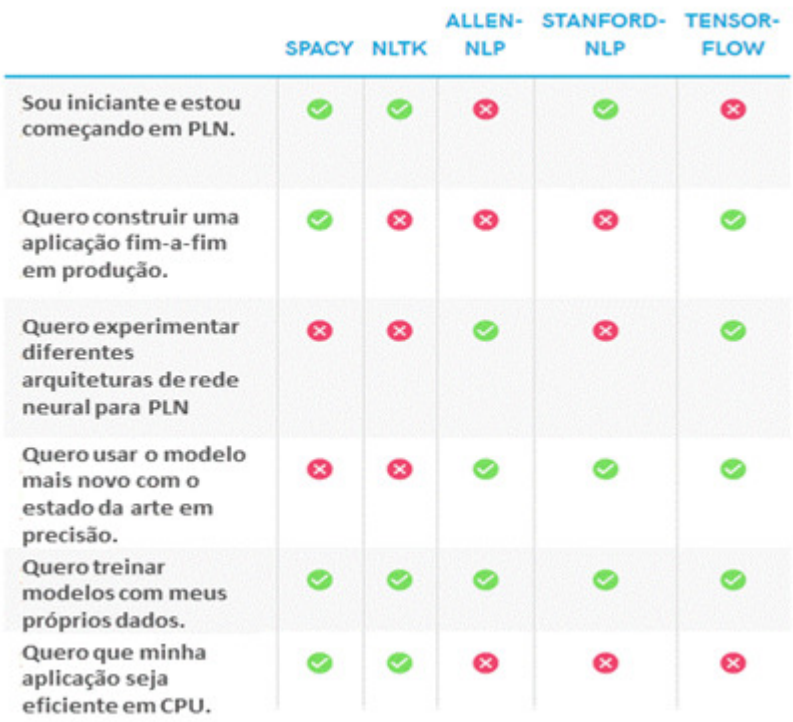
- Amplo conjunto de funcionalidades já disponível, propiciando maior agilidade na construção de produtos escaláveis e robustos para uma variedade de problemas de PLN; além disso, o spaCy possui uma série de rotinas encapsuladas que podem ser executadas diretamente por linha de comando, reduzindo esforço de programação;

- Arquitetura modular, catálogo de API bem documentado e interoperabilidade com TensorFlow, PyTorch, scikit-learn, Gensim, Apache Spark e o resto do ecossistema de IA do Python, possibilitando o desenvolvimento de modelos customizados e sofisticados linguisticamente; e

- Alto desempenho, comparável às melhores bibliotecas gratuitas e disponibilizadas pela comunidade científica.

Além de uma ótima relação custo x benefício, a ferramenta atende aos principais requisitos adotados na pesquisa, ou seja, a possibilidade de treinamento de um corpus próprio, de utilização de algoritmos de alto desempenho e de entrega de um produto final acabado e disponível. Tal fato pode ser corroborado com o sumário apresentado na Figura 7, que apresenta a indicação de uso de algumas bibliotecas frente a diversos cenários de utilização.

Figura 7 – Indicações de bibliotecas x cenários de uso.



Fonte: (SPACY, 2017).

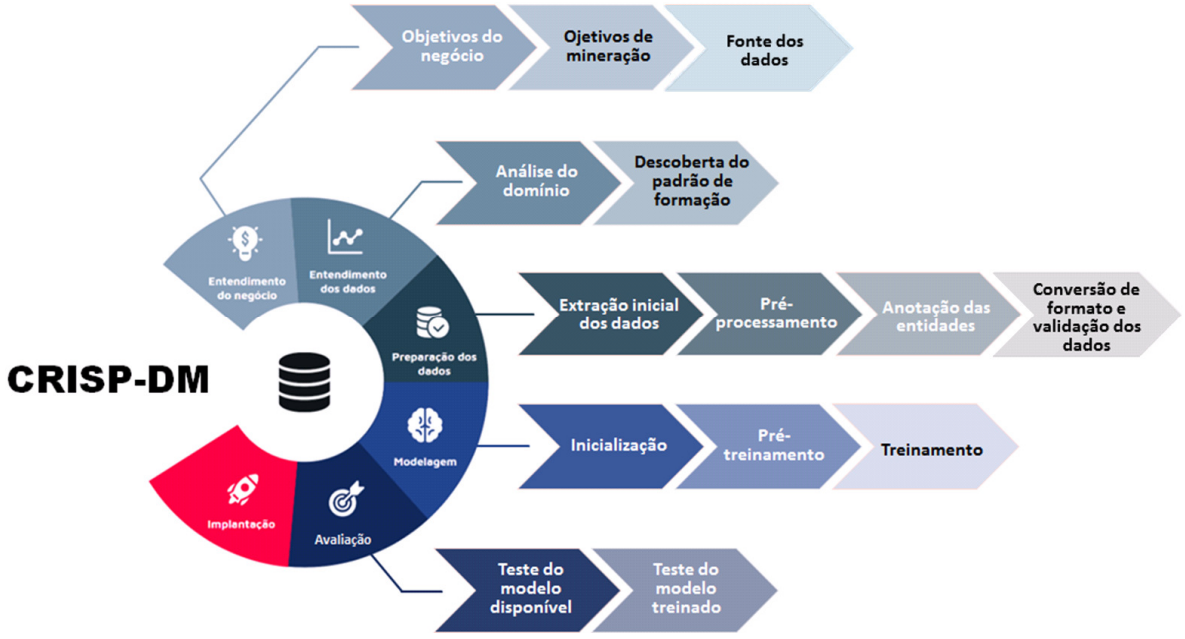
4 DESENVOLVIMENTO

De maneira análoga à mineração de dados, a mineração de texto procura extrair informações úteis de fontes de dados através da identificação e exploração de padrões de interesse. No caso da mineração de texto, no entanto, os dados não se encontram originalmente estruturados. Tarefas relacionadas à preparação dos dados, tais como, limpeza (remoção de dados ruidosos, faltantes ou redundantes), integração (combinação de múltiplas fontes), redução e transformação (redução de dimensionalidade, discretização, agregação, normalização, etc.) somente são possíveis após a etapa de extração.

Assim, as metodologias utilizadas para mineração de dados, tais como o *Knowledge Discovery in Databases* (KDD) (FAYYAD et al., 1996), não se aplicam adequadamente à mineração textual. Neste sentido, para o presente trabalho, preferiu-se adotar, como base, a metodologia *Cross Standard Process for Data Mining* (CRISP-DM) (IBM, 2014), por ser mais abrangente e já consagrada na literatura.

O capítulo está estruturado conforme as fases do modelo e as etapas necessárias à pesquisa, resumidos na Figura 8. Uma breve descrição de cada fase é apresentada no início de cada seção acompanhada do detalhamento das etapas executadas.

Figura 8 – Metodologia de trabalho.



Fonte: (TORRES, 2019), adaptado pelo autor.

4.1 ENTENDIMENTO DO NEGÓCIO

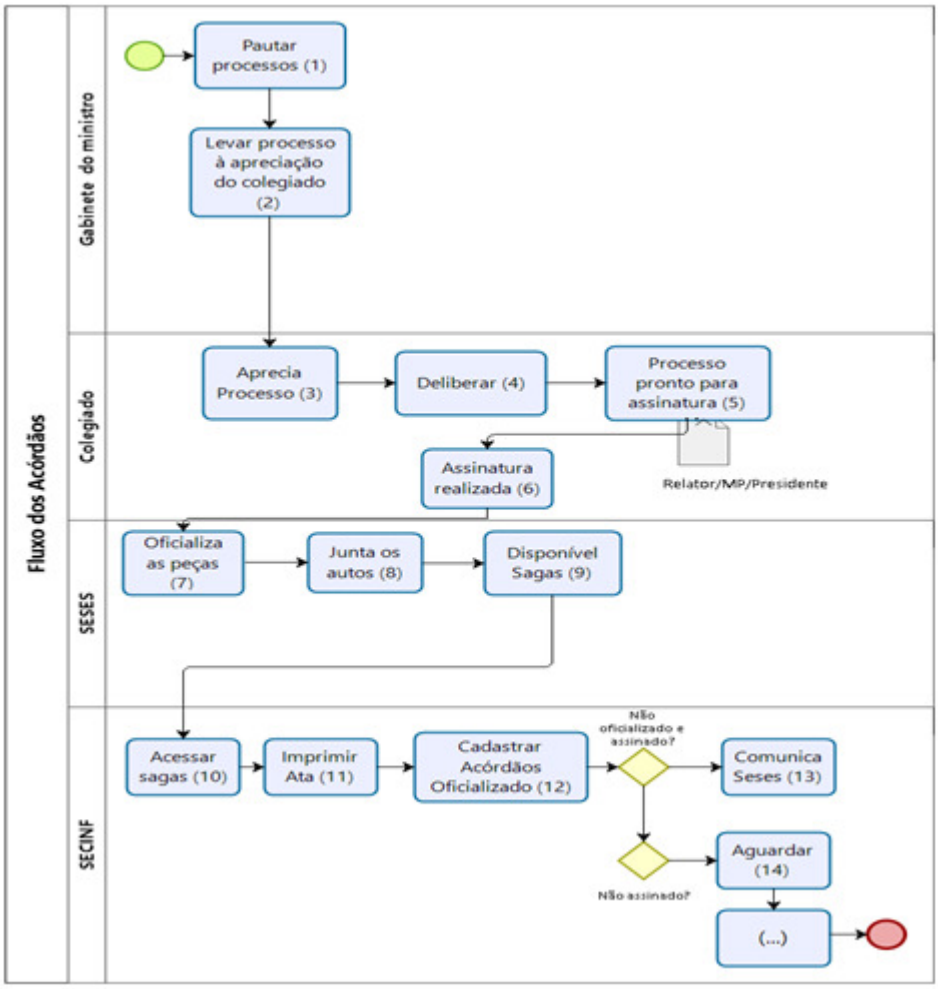
A primeira fase do ciclo contempla a compreensão do cenário, a identificação das necessidades e a definição dos objetivos de negócio a serem alcançados. São mapeadas as condições gerais do projeto, tais como premissas, pré-requisitos, restrições, partes interessadas, normativos e recursos disponíveis. Também são estabelecidos os critérios de mensuração dos resultados e as fontes utilizadas nas tarefas de mineração.

4.1.1 Objetivos de negócio

Registro de deliberações

O processo de trabalho envolvendo o registro de deliberações inicia-se após a oficialização do acórdão, a juntada de peças e a disponibilização dos mesmos em sistema específico (Sagas) pela Secretaria de Sessões (Seses). A partir de então, a responsabilidade passa a ser do Secinf, conforme indica o fluxograma apresentado na Figura 9.

Figura 9 – Fluxo de processamento de acórdãos.



Fonte: (SEPROC, 2019).

O Secinf tem suas competências definidas no artigo 13º da Portaria-Seproc nº 1/2019, cujo extrato segue transcrito adiante (grifo próprio):

> *"Art. 13 Compete ao Serviço de Cadastramento de Informações:*
> *I - promover o registro das informações inerentes às deliberações proferidas pelo Tribunal, de modo a permitir adequada gestão, acompanhamento e produção dos efeitos delas decorrentes;*
> *II - gerenciar e zelar pela atualização de cadastros e bases de dados em função das deliberações do TCU; e*
> *III - desenvolver outras atividades inerentes à sua finalidade."*

Vê-se, claramente, que a principal e praticamente única atividade do Serviço é o registro das deliberações expedidas pelos colegiados do Tribunal. Isto ocorre pelo fato de que as deliberações não são produzidas em meio estruturado e sim lavradas nos acórdãos em formato textual. Assim, faz-se necessária a transcrição dos comandos prolatados para unidades de informação estruturadas, de forma a possibilitar que sejam armazenados em banco de dados e então utilizadas nos processos de trabalho subsequentes ao julgamento (Figura 10).

Figura 10 – Atividades de pós-julgamento.

Fonte: Elaborado pelo autor (2019).

O consumo dessas informações também abrange um vasto público, tanto no âmbito interno quanto externo ao Tribunal, como pode ser visto na Figura 11.

Figura 11 – Deliberações - partes interessadas.

Fonte: Elaborado pelo autor (2019).

O registro das deliberações envolve um processo de trabalho laborioso, composto de diversas atividades agrupadas em três papéis principais: coordenação, cadastramento e conferência.

A Figura 12 apresenta uma síntese do Serviço.

Figura 12 – Síntese do Secinf.

Fonte: Elaborado pelo autor (2019).

O Secinf registra cerca de 35 tipos de deliberação diferentes, em consonância com as possibilidades de decisão do TCU, de acordo com a sua Lei Orgânica e seu Regimento Interno.

A Tabela 1 apresenta o total de deliberações registradas pelo Serviço de 2011 a 2018, com destaque para determinações e recomendações a órgãos e entidades, focos de estudo da presente pesquisa.

Tabela 1 – Tipos de deliberação.

Tipos de deliberação	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**
Abertura de Novo Processo / Apartado	89	87	71	98	79	102	127	104
Acatar/Rejeitar as Alegações de Defesa	983	743	989	1.638	1.572	1.645	1.064	1.483
Acatar/Rejeitar as Razões de Justificativa	1.484	1.216	901	601	380	152	5	11
Apensamento de Outro(s) Processo(s) ao Atual	10	6	5	4	6	10	5	5
Apensamento do Atual Processo a Outro(s)	392	391	628	482	449	378	335	252
Aplicação da Chancela de Sigiloso	4	7	3	5	-	1	-	1
Aplicação de Medida Cautelar a Órgão/Entidade	39	47	37	19	8	-	-	1
Aplicação de Medida Cautelar a Responsável	38	20	28	135	107	94	131	176
Aplicação de Multa a Responsável	2.655	1.734	1.871	2.415	2.541	2.525	2.254	2.332

Aplicação de Outras Sanções (que não multa)	259	120	284	280	283	300	212	323
Arquivamento de Processo	3.865	3.582	3.455	3.143	2.662	2.710	2.604	2.172
Arquivamento por Economia Processual	17	15	12	6	9	5	9	16
Audiência de Responsável	2.803	2.481	379	306	170	247	347	182
Autorização de Recolhimento Parcelado	2.210	1.744	1.751	2.897	2.751	3.286	2.512	2.711
Citação de Responsável	2.323	2.729	3.075	3.431	3.373	3.360	2.942	3.161
Conhecim./Denúncia/Repr./Solic./Consulta	3.486	3.350	3.465	2.880	2.537	2.500	2.749	2.605
Desapensamento de Processo	12	7	16	8	8	3	2	2
Determinação a Órgão/Entidade	18.048	19.623	17.751	17.600	16.291	18.420	16.174	15.993
Determinação de Realização de Fiscalização	1.263	1.229	989	775	806	765	665	605
Diligência a Órgão/Entidade	2.503	2.223	279	176	115	74	60	87
Diligência a Responsável	185	279	32	14	8	-	3	2
Expedição de Quitação a Responsável	10.862	6.828	5.649	6.898	6.993	5.300	5.047	5.324
Expedição de Quitação de Dívida	343	364	361	342	342	285	307	256
Imputação de Débito a Responsável	1.419	923	985	1.987	1.960	2.272	1.754	1.815
Julgamento das contas do Responsável	12.939	8.060	6.999	8.917	9.158	7.804	7.273	7.681
Julgamento de Estágio de Desestatização	38	34	38	21	23	25	3	11
Modificação da Natureza do Processo	106	96	67	91	62	68	62	52
Prorrogação de Prazo de Deliberação	381	275	115	179	197	209	215	217
Prorrogação de Prazo de Deliberação	447	281	18	25	27	15	46	7
Reabertura de Processo	2	-	-	-	-	-	1	1
Recomendação a Órgão/Entidade	1.197	715	688	837	767	773	671	740
Requisição de Serviços Técnicos Especializado	1	-	1	-	-	-	4	3
Retirada da Chancela de Sigiloso	4	5	9	7	8	8	12	10
Sobrestamento do Julgamento	94	84	134	85	38	61	69	91
Tornar Deliberação Sem Efeito	492	514	435	412	448	484	445	400
Trancamento de Contas Iliquidáveis	49	18	32	65	20	16	10	4
Determinação de Providências Internas ao TCU	9.591	8.836	7.831	7.207	6.574	6.764	6.697	6.385
Total	**82.714**	**69.153**	**59.503**	**64.002**	**60.775**	**60.661**	**54.816**	**55.221**

Fonte: Base de dados Radar (TCU).

No ano de 2019, o Tribunal promoveu mudanças significativas de estrutura organizacional e forma de operação. Entre as mudanças, instituiu a Seproc, por meio da Resolução-TCU nº 305/2018, com a finalidade de desenvolver e centralizar os serviços e atividades inerentes à gestão processual e de documentos no âmbito da Secretaria-Geral de Controle Externo. O novo desenho organizacional, paralelamente à absorção das atividades de operação, exigiu remodelagem e reorganização de todos os processos de trabalho, práticas e procedimentos concernentes às funções de apoio à atividade de controle.

No tocante ao Secinf, foram incorporadas ao Serviço duas novas atividades:

- Identificação e instrução de processo relativamente ao saneamento de falha material em acórdãos, antes conduzida pelas equipes de apoio administrativo junto às unidades técnicas;

• Registro de deliberações relativas às fiscalizações de atos de pessoal (admissão, aposentadora, etc.), realizada até então por técnicos da Secretaria de Fiscalização de Pessoal.

Ambas as atribuições implicaram em aumento da carga de operações na equipe, em especial o cadastro de atos de pessoal, que sozinho respondeu por 70% da média de deliberações expedidas pelo Tribunal de 2011 a 2018. Tomando como base a mesma série histórica, o acréscimo de registros equivale, em média, a novos 145.000 registros por ano.

Para fazer frente a tamanho desafio e manter a meta de desempenho estabelecida à área de 3 dias úteis como prazo limite para o registro de um acórdão a partir de sua oficialização, faz-se necessária a adoção de medidas estruturantes, em especial no tocante à automação dos processos de trabalho.

De fato, diversas ações de cunho tecnológico foram iniciadas desde o início das operações da Seproc. Dentre elas, merecem destaque a modernização da plataforma utilizada para o registro das deliberações e a incorporação de sistemas cognitivos, ambos em parceria com a Secretaria de Soluções de Tecnologia da Informação.

Na vertente deste último, já se encontra em produção solução para a identificação, de forma automatizada, das ocorrências de tipificações mais comuns de falha material em acórdãos, tais como CPF e CNPJ inválidos, nomes de pessoas grafados erroneamente e multas aplicadas a pessoas falecidas.

No momento, encontra-se em avaliação iniciativa já relacionada anteriormente que visa à extração automática de deliberações em acórdãos e que igualmente, ao menos em parte, também é o intuito desta pesquisa. Ambos os trabalhos são complementares, vez que não tratam exatamente do mesmo conjunto de entidades e tampouco empregam as mesmas tecnologias.

Fato é que o aporte tecnológico e a automação das atividades de registro se apresentam como imprescindíveis num cenário de restrição crescente de recursos. Cabe frisar que, embora tenha internalizado uma carga adicional de responsabilidades, o Secinf conta, atualmente, com um quantitativo em sua força de trabalho não maior do aquele existente previamente à criação da Seproc.

Por fim, a automação é igualmente essencial para se promover uma desejada transformação no perfil de atuação do Serviço, de menos operacional para mais estratégico, a fim de habilitá-lo a tarefas de maior valor agregado no tocante à efetividade do Tribunal, tais como a

gestão dos monitoramentos quanto ao cumprimento das determinações e o provimento de informações de inteligência, abordado a seguir.

Produção de conhecimento

O Tribunal de Contas da União, seja em atendimento a demandas internas para fins de planejamento ou acompanhamento gerencial, ou externas em cumprimento à exigência legal ou ainda por iniciativa própria, produz e divulga regularmente informações no tocante a sua atuação.

Os conteúdos disponibilizados por meio de relatórios, painéis executivos ou outros canais apresentam, em geral, dados de produção relativos às etapas de instrução e de julgamento. Concernente ao último, há uma profusão de indicadores extraídos do conjunto de decisões emanadas pelo Tribunal e que visam a retratar os diversos aspectos das atividades de controle, tais como apreciação de processos, concessão de medidas cautelares, sustação de atos e contratos, julgamento de contas e condenações e sanções aplicadas.

No entanto, a despeito da amplitude das informações produzidas, há ainda uma riqueza de conhecimento inexplorado nos acórdãos do Tribunal, especialmente quando cruzadas ou complementadas com dados de outras fontes públicas disponíveis.

É o caso, por exemplo, das políticas e programas de governo. Em que pese o levantamento realizado pela Secretaria de Macroavaliação Governamental, por força da elaboração do Relatório de Políticas e Programas de Governo (TCU, 2019b), cuida-se ali de dar transparência à sociedade acerca da atuação do Estado no âmbito das políticas públicas. Trata-se de instrumento pelo qual o TCU apresenta um panorama geral atinente aos riscos, às irregularidades e às deficiências relevantes e recorrentes identificadas por meio de suas fiscalizações sistêmicas nas distintas áreas da atuação governamental.

Embora trate de temas estruturantes, o escopo do levantamento se restringe a um universo restrito de algumas dezenas de fiscalizações. Com foco mais em aspectos qualitativos, o relatório não retrata a real dimensão do esforço fiscalizatório do Tribunal no tocante à execução dos programas de governo, quando considerada a totalidade das ações de controle, e tampouco apresenta alguma correlação deste esforço com a perspectiva financeira-orçamentária de tais programas prevista no Plano Plurianual (PPA). Tal informação, se disponível, poderia ser de

grande valia no auxílio à priorização das ações de controle e no uso mais otimizado de sua força de trabalho.

Igualmente pertinente seria demonstrar o alcance das determinações no espectro político-administrativo do país, em especial no âmbito das municipalidades. Realçar a abrangência geográfica da capacidade fiscalizatória do TCU reforça a mensagem quanto a aplicação, *de facto*, de sua jurisdição em todo o território nacional, acentuando assim o efeito inibidor.

Todo o conhecimento produzido a partir das inferências supracitadas e de, potencialmente, outras mais aqui não elencadas, têm origem no vasto e pouco explorado corpus formado pelo conjunto de deliberações do Tribunal. Essa grande base de informações, se adequadamente perscrutada mediante a utilização de tecnologias de mineração textual, tem o potencial de se tornar uma rica e relevante fonte de conhecimento para a Casa.

4.1.2 Objetivos de mineração

As métricas de avaliação mais comum em modelos NERC são a precisão, a revocação (*recall*) e a medida F (*F-score,* também *F1-score* ou *F-measure*), dadas pelas Fórmulas 1, 2 e 3:

$$Precisão = \frac{VP}{FP + VP} \tag{1}$$

$$Revocação = \frac{VP}{FN + VP} \tag{2}$$

$$Medida\ F = \frac{2 \times Precisão \times Revocação}{(Precisão + Revocação)} \tag{3}$$

Essas métricas levam em consideração algumas medidas básicas relacionadas ao grau de predição e à sensibilidade dos modelos, conforme definição a seguir:

• **VP (Verdadeiro Positivo)**: classificação correta da classe Positivo;

• **VN (Verdadeiro Negativo)**: classificação correta da classe Negativo;

• **FP (Falso Positivo)**: erro em que o modelo previu a classe Positivo quando o valor real era a classe Negativo; e

• **FN (Falso Negativo)**: erro em que o modelo previu a classe Negativo quando o valor real era a classe Positivo.

Relevante mencionar que, em sistemas NERC, as métricas descritas são aplicadas em nível de *token*. Embora sejam úteis, essa abordagem pode levar a resultados imprecisos nos casos de falha de previsão quanto à classe da entidade ou da fronteira de entidades compostas de múltiplos *tokens*. Por exemplo, no caso da entidade "Lei de Acesso à Informação", se apenas os três primeiros *tokens* forem capturados ("Lei de Acesso"), o sistema poderá indicar, erroneamente, uma revocação de 3/5, ao invés de zero. Pode-se obter maior precisão com o uso de métricas que operem no nível completo da entidade nomeada ou outras mais sofisticadas, tais como aquelas indicadas por BATISTA (2018). Tais alternativas, entretanto, estão além do escopo deste trabalho.

4.1.3 Fonte de dados

As deliberações do TCU, tanto do Plenário quanto das Câmaras, assumem a forma de acórdãos, que são publicados, conforme o caso, no Diário Oficial da União (DOU) e/ou no Boletim do Tribunal de Contas da União (BTCU).

Os acórdãos podem conter um ou mais itens de deliberação. Cada item, por sua vez, é composto de um conjunto de metadados, tais como o nome do órgão ou entidade a quem se destina a deliberação e o prazo de vencimento, e de um excerto textual relativo à decisão adotada.

Para efeito de registro, cada item é cadastrado separadamente, em um sistema denominado Radar, a partir do qual os metadados são armazenados em banco de dados Oracle e o texto em solução de Gerenciamento Eletrônico de Documentos (GED). Ambas as informações são disponibilizadas por meio de visões para consumo por outras soluções a fim de atender aos processos de trabalho da fase de pós-julgamento.

Entre elas, está o Sismonitoramento, sistema criado para dar suporte ao acompanhamento sistemático, pelas unidades técnicas, das determinações e recomendações exaradas pelo Tribunal às unidades jurisdicionadas.

A base de dados desse sistema foi, portanto, a fonte escolhida para a extração dos excertos textuais relativos às deliberações e utilizadas no treinamento do modelo.

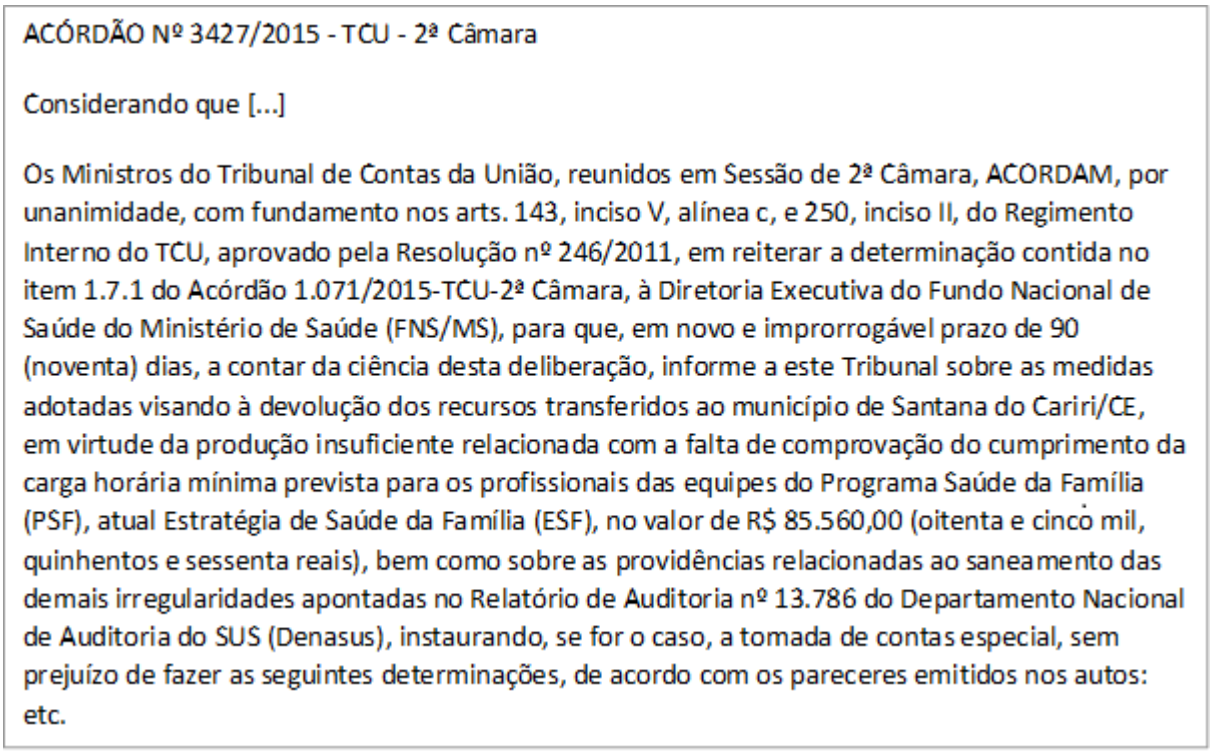
4.2 ENTENDIMENTO DOS DADOS

Nesta fase, os dados brutos são coletados e explorados a fim de permitir maior compreensão de sua natureza, qualidade e propriedades. Na mineração de textos com vistas à extração de entidades, deve-se proceder uma análise de domínio das informações de interesse, buscando-se identificar o padrão de formação (vocabulário, regras de formação, etc.) e o grau de dispersão dos termos, bem como eventuais inter-relacionamentos existentes.

4.2.1 Análise de domínio

Para fins deste estudo, entende-se como item de deliberação, ou simplesmente deliberação, a parte do acórdão que carrega a decisão do colegiado. A Figura 13 ilustra um exemplo típico.

Figura 13 – Exemplo de acórdão.



ACÓRDÃO Nº 3427/2015 - TCU - 2ª Câmara

Considerando que [...]

Os Ministros do Tribunal de Contas da União, reunidos em Sessão de 2ª Câmara, ACORDAM, por unanimidade, com fundamento nos arts. 143, inciso V, alínea c, e 250, inciso II, do Regimento Interno do TCU, aprovado pela Resolução nº 246/2011, em reiterar a determinação contida no item 1.7.1 do Acórdão 1.071/2015-TCU-2ª Câmara, à Diretoria Executiva do Fundo Nacional de Saúde do Ministério de Saúde (FNS/MS), para que, em novo e improrrogável prazo de 90 (noventa) dias, a contar da ciência desta deliberação, informe a este Tribunal sobre as medidas adotadas visando à devolução dos recursos transferidos ao município de Santana do Cariri/CE, em virtude da produção insuficiente relacionada com a falta de comprovação do cumprimento da carga horária mínima prevista para os profissionais das equipes do Programa Saúde da Família (PSF), atual Estratégia de Saúde da Família (ESF), no valor de R$ 85.560,00 (oitenta e cinco mil, quinhentos e sessenta reais), bem como sobre as providências relacionadas ao saneamento das demais irregularidades apontadas no Relatório de Auditoria nº 13.786 do Departamento Nacional de Auditoria do SUS (Denasus), instaurando, se for o caso, a tomada de contas especial, sem prejuízo de fazer as seguintes determinações, de acordo com os pareceres emitidos nos autos: etc.

Fonte: Diário Oficial da União.

Conforme pode ser observado na Tabela 1, o Secinf registra cerca de 35 tipos de deliberação diferentes. Para efeito desta pesquisa, optou-se por utilizar apenas os registros alusivos às determinações e às recomendações a órgãos e entidades. Além de suas óbvias relevâncias no contexto da Administração Pública, os dois tipos oferecem uma diversidade de potenciais entidades, além de representam, juntos, o segundo maior volume de deliberações, atrás apenas dos julgados relativos à legalidade de atos de pessoal.

As classes de entidade selecionados para a pesquisa foram:

- ÓRGÃO OU ENTIDADE
- PRAZO DE VENCIMENTO
- PROGRAMA DE GOVERNO
- LOCALIDADE
- VALOR MONETÁRIO
- NORMATIVO LEGAL
- ACÓRDÃO

- DECISÃO NORMATIVA
- PARECER

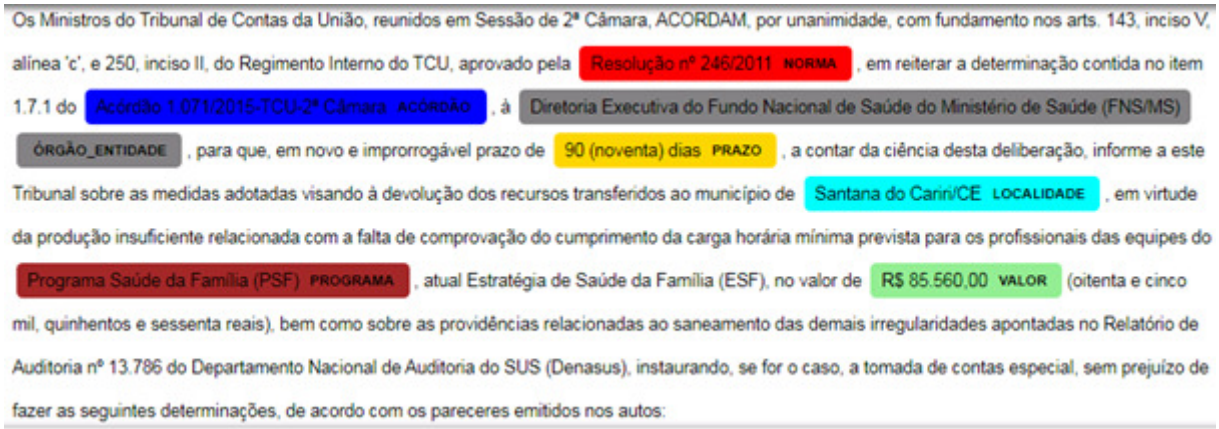
As classes ÓRGÃO OU ENTIDADE e PRAZO DE VENCIMENTO são, atualmente, capturados de forma manual pelo Secinf para fins de registro e, obviamente, de interesse no contexto de uma futura automação da atividade.

As classes PROGRAMA DE GOVERNO, LOCALIDADE e VALOR MONETÁRIO são interessantes para propósitos de produção de informação, conforme exposto na seção anterior.

Finalmente, as classes vinculadas à temática jurídica, NORMATIVO LEGAL, ACÓRDÃO, DECISÃO NORMATIVA e PARECER, pertinentes por natureza em um órgão com atribuição de judicatura administrativa, podem ser úteis para a identificação dos normativos mais referenciados, com vistas a proposição de aperfeiçoamentos regulatórios, por exemplo, ou ainda para aprimorar a pesquisa integrada do TCU relativa à jurisprudência e aos atos normativos.

Na Figura 14, segue um exemplo de resultado esperado, extraído por meio do visualizador integrado da biblioteca "displaCy", com as marcações correspondentes às classes de entidade sobre os termos correspondentes.

Figura 14 – Exemplo de acórdão com as entidades esperadas em destaque.



Fonte: Elaborado pelo autor (2019).

Após a extração das deliberações a partir da base de dados do sistema Sismonitoramento, foram examinadas 30.433 deliberações em 13.065 acórdãos, exarados de 01/01/2011 à 31/07/2019.

O total de entidades apresentado representa a quantidade de termos distintos encontrados no corpus analisado. No entanto, tais quantitativos **não** refletem, de fato, o total de valores distintos, dada a ocorrência de:

- Erros de ortografia, p. ex., "1 ano" e "1 anos";

- Grafias diversas, p. ex., "Acórdão 1.793/2011, do Plenário" e "Acórdão 1.793/2011-Plenário; e

- Abreviações e siglas, p. ex., "Petróleo Brasileiro S.A." e "Petrobras".

A normatização e a resolução de entidades são tarefas subsequentes ao NERC e estão além deste estudo.

4.2.2 Descoberta do padrão de formação

Após exaustiva análise de todo o conjunto de deliberações, chegou-se a padrões heurísticos de formação das entidades composto de três partes: QUALIFICADOR, CHAVE e COMPLEMENTO. Os possíveis valores para cada uma das classes de entidades estudadas estão listados na Tabela 2[5].

[5] Notação utilizada: "< >" - campo obrigatório; "[ ]" - campo opcional; "|" – valores alternativos.

Tabela 2 – Padrão de formação das entidades.

Padrão de Formação das Entidades

ÓRGÃO OU ENTIDADE

Qualificador:	[Núm. ordinal] [Academia | Administração | Advocacia | Advocacia-Geral | Advogado | Agência | Análise | Arquivo | Arsenal | Assessoria | Associação | Auditoria | Autoridade | Banco | Base | Batalhão | Brigada | Cadastro | Câmara | Capitania | Casa | Central | Centro | Cerimonial | Circunscrição | Colégio | Comando | Comando-Geral | Comissão | Comitê | Companhia | Complexo | Confederação | Congresso | Conselho | Consórcio | Construtora | Consulado | Consulado-Geral | Consultoria | Controladora | Controladoria | Controladoria-Geral | Controladoria-Regional | Controle | Coordenação | Coordenação-Geral | Coordenador | Coordenadoria | Coordenadoria-Geral | Corpo | Corregedoria | Corregedoria-Geral | Defensoria | Delegacia | Departamento | Departamento-Geral | Depósito | Desenvolvimento | Diocese | Direção | Direção-Geral | Diretor | Diretor-Geral | Diretoria | Diretoria-Executiva | Diretoria-Geral | Diretório | Distrito | Divisão | Docas | Embaixada | Empresa | Energética | Entidade | Escola | Escritório | Esquadrão | Estabelecimento | Estação | Estado | Estado-Maior | Estaleiro | Estratégia | Faculdade | Federação | Financiadora | Força-Tarefa | Fundação | Fundo | Gabinete | Gerência | Gerência-Executiva | Gestão | Gestor | Governo | Grupamento | Grupo | Hospital | Imprensa | Indústria | Inspetoria | Instituto | Inventariança | Justiça | Laboratório | Marinha | Maternidade | Ministério | Ministro | Ministro-Chefe | Município | Munícipio | Museu | Núcleo | Operador | Ordem | Órgão | Ouvidoria | Parque | Patrimônio | Penitenciária | Planejamento | Plano | Poder | Polícia | Política | Porto | Prefeito | Prefeitura | Presidência | Procurador-Chefe | Procuradoria | Procuradoria-Geral | Procuradoria-Regional | Pró-Reitoria | Receita | Rede | Refinaria | Região | Regimento | Reitor | Representação | Representante | Saneamento | Seção | Seccional | Secretaria | Secretaria-Executiva | Secretaria-Geral | Secretário | Sede | Serviço | Setor | Sindicato | Sistema | Sociedade | Subdiretoria | Subsecretaria | Subsecretaria-Geral | Superintendência | Superintendente | Supremo | Tecnologia | Tesouro | Transmissora | Transportadora | Tribunal | União | Unidade | Universidade | Vice-Presidência]
Chave:	<nome | sigla do órgão ou entidade>
Complemento:	[órgão ou entidade superior] [localidade]

NORMATIVO LEGAL

Qualificador:	[ABNT | Ato | CF | Código | Consolidação | Constituição | Contrato | Decisão | Decreto | Decreto-Lei | Despacho | DN | Emenda | Enunciado | Estatuto | IN | Instrução | Jurisprudência | LC | Lei | Medida | Memo | Memorando-Circular | MP | NBR | Norma | Nota | ON | Ordem | Orientação | Portaria | Projeto | RDC | Recomendação | Regimento | Regulamento | Resolução | RI | RLC | SFC | Súmula | Termo]
Chave:	<identificador | sigla>
Complemento:	[data de publicação] [órgão expedidor]

ACÓRDÃO

Qualificador:	Acórdão
Chave:	<número> <ano>
Complemento:	[órgão julgador] [colegiado] [relator]

DECISÃO NORMATIVA	
Qualificador:	Decisão
Chave:	<identificador>
Complemento:	[órgão decisor] [colegiado] [data]

PARECER	
Qualificador:	Parecer
Chave:	<identificador> <data>
Complemento:	[órgão emissor]

PROGRAMA DE GOVERNO	
Qualificador:	[Programa]
Chave:	[código] <nome | sigla do programa>
Complemento:	-

LOCALIDADE	
Qualificador:	-
Chave:	<nome do município> <nome | sigla do estado>
Complemento:	-

VALOR MONETÁRIO	
Qualificador:	-
Chave:	<moeda> <quantidade> | <quantidade> <unidade>
Complemento:	-

PRAZO DE VENCIMENTO	
Qualificador:	-
Chave:	<quantidade> <unidade>
Complemento:	-

Fonte: Elaborado pelo autor (2019).

As variações observadas na formação das entidades, em especial em ÓRGÃOS OU ENTIDADES, têm implicações na etapa de anotação da base de treinamento, detalhadas na seção seguinte.

4.3 PREPARAÇÃO DOS DADOS

Nesta fase, os dados são tratados para que possam ser utilizados pelos algoritmos de mineração. Na mineração de dados, são realizadas tarefas como seleção de registros, escolha e criação/derivação de atributos, limpeza e saneamento, integração e formatação de dados. Em mineração de texto, são empregadas técnicas próprias de pré-processamento textual, tais como normalização, "tokenização", correção ortográfica, remoção de *stop words*, "stemização" e "lematização"; na sequência, converte-se dos dados em representações numéricas, define-se a *feature* de interesse (BOW, TF-IDF ou incorporação de palavras) e, por fim, aplica-se uma redução de dimensionalidade, se necessário.

No tocante à tarefa de NERC, uma das principais atividades preparatória para a utilização do modelo ocorre nesta etapa. Trata-se da anotação (também conhecida como marcação, "tagueamento" ou *tagging*) das entidades para formação da base de treinamento. Embora onerosa, tal atividade é essencial para a acurácia dos resultados em modelos de aprendizagem supervisionada.

4.3.1 Extração inicial dos dados

Conforme citado, foram extraídos mais de 30 mil itens de deliberação de aproximadamente 13 mil acórdãos. Cada item é composto de um parágrafo que representa uma determinação ou recomendação expressa no acórdão.

Os dados foram submetidos à limpeza prévia para eliminação de redundâncias (eventualmente, dois acórdãos distintos contêm a mesma redação no tocante à deliberação) e então exportados para o formato JSON para as rotinas de pré-processamento.

4.3.2 Pré-processamento

A etapa de pré-processamento teve como objetivo a geração dos vetores de palavras utilizados no treinamento. Os vetores de palavras permitem importar o conhecimento do texto bruto para o modelo. O conhecimento é representado como uma tabela de números, com uma linha por termo em seu vocabulário. Se dois termos forem usados em contextos semelhantes, suas linhas são bastante semelhantes, enquanto as palavras usadas em contextos diferentes terão

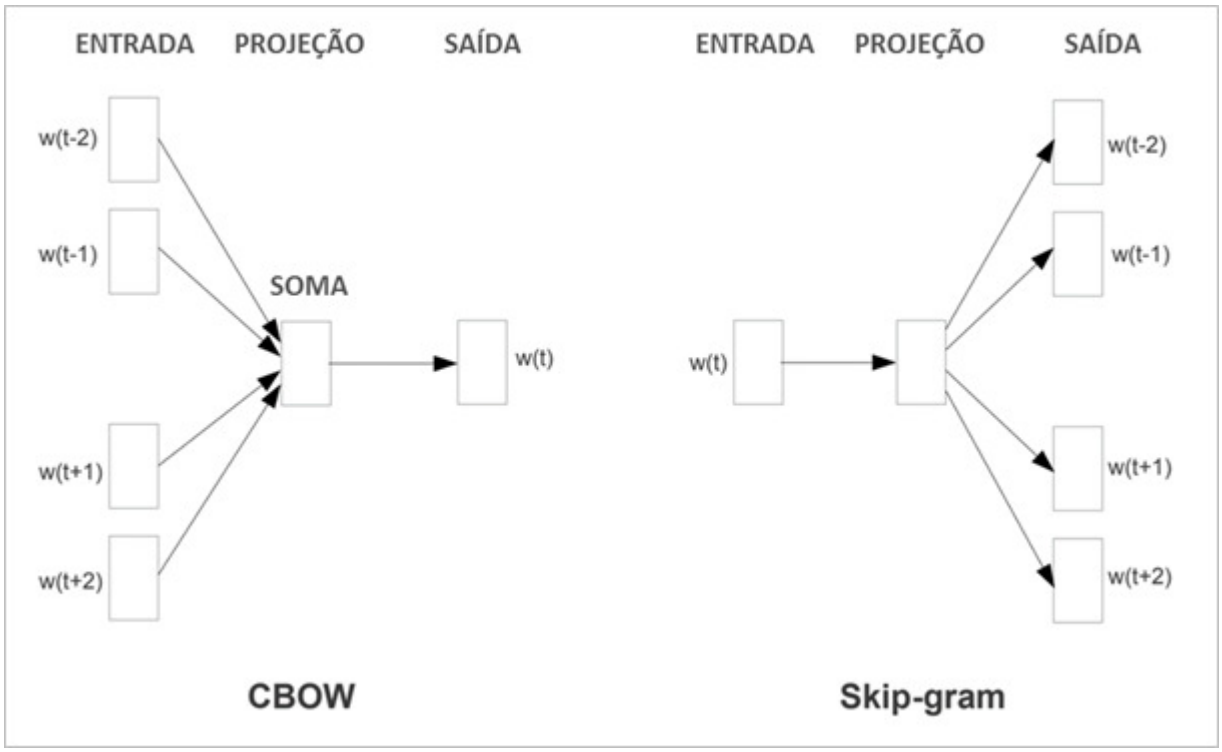
valores bastante diferentes. Isso permite que o algoritmo de NERC utilize os valores de linha atribuídos às palavras como uma espécie de dicionário a fim de aprender algo sobre o significado das palavras no texto e, assim, inferir a classe de palavras que não estão bem representados nos dados de treinamento rotulados (SPACY, 2017).

Preliminarmente, foram executadas rotinas para remoção de pontuação e de *stop words*, com uso de classes do NLTK (BIRD et al., 2009).

Na sequência, foi aplicado o algoritmo Word2Vec da biblioteca Gensim (REHUREK e SOJKA, 2010). O Word2Vec, proposto por MIKLOV et al. (2013), é um modelo que incorpora palavras em um espaço vetorial de menor dimensão usando uma rede neural superficial. O resultado é um conjunto de vetores de palavras em que vetores próximos no espaço vetorial têm significados semelhantes com base no contexto e vetores de palavras distantes entre si têm significados diferentes. Em outras palavras, o Word2Vec aprende os relacionamentos entre as palavras automaticamente, produzindo vetores com notáveis relações lineares.

Existem duas versões do modelo, ambas implementadas pelo Word2Vec: *Continuous-Bag-of-Words* (CBOW) e Skip-grams. Ambos os modelos aprendem sobre palavras com base no seu contexto local, onde o contexto é definido por uma janela de palavras vizinhas que "desliza" ao longo do texto (*context window*). A diferença entre eles é que o modelo CBOW aprende a incorporação prevendo a palavra atual com base em seu contexto. Já o modelo de Skip-grams faz o oposto, ou seja, aprende prevendo as palavras ao redor, dada a palavra atual (Figura 15).

Figura 15 – Modelos de implementação do algoritmo Word2Vec.



Fonte: (BROWNLEE, 2019)

O tamanho da janela é um parâmetro configurável do modelo e tem um forte efeito nas semelhanças vetoriais resultantes. Janelas extensas tendem a produzir semelhanças de cunho mais temático, enquanto janelas menores tendem a produzir semelhanças mais funcionais e sintáticas (BROWNLEE, 2019).

Os principais parâmetros utilizados no presente estudo estão apresentados na tabela seguinte.

Tabela 3 – Principais parâmetros utilizados no algoritmo Word2Vec.

Parâmetro	**Definição**	**Valor**
min_count	Mínimo de ocorrências abaixo do qual a palavra será ignorada	5
size	Dimensão do vetor de palavras	200
workers	Número de processadores (paralelização)	2
window	Tamanho da janela deslizante de contexto	10
iter	Número de épocas de treinamento	30

Fonte: Elaborado pelo autor (2019).

O código completo, escrito em Python, está disponível no APÊNDICE B – Códigos-fonte.

4.3.3 Anotação das entidades

Para a anotação do corpus, foram realizadas, inicialmente, experimentações em algumas ferramentas *open source* disponíveis, tais como "spaCy NER Annotator", "Brad Rapid Annotation Tool", "WebAnno", "Doccano". Embora todas elas ofereçam recursos interativos que facilitam a tarefa de "tagueamento" do texto, preferiu-se utilizar o Microsoft Excel, dada a sua capacidade de automatização de rotinas por meio de funções nativas, macros e expressões regulares.

Foram adotadas as seguintes premissas durante as anotações:

- ÓRGÃOS OU ENTIDADES foram anotados somente no contexto de destinatários das determinações ou recomendações; além disso, suas marcações se deram na forma mais extensível possível, ou seja, o fragmento "Administração Regional do Serviço Nacional de Aprendizagem Comercial no Estado de Rondônia (Senac/RO)" foi anotado como uma única entidade;

- PRAZO foi anotado somente no contexto de vencimento, ou seja, período limite para o cumprimento da deliberação;

- LOCALIDADE foi anotado somente quando se referia a munícipio; nos casos em que o município foi indicado como destinatário da deliberação, optou-se por classificá-lo como ÓRGÃO OU ENTIDADE; e

- Entidades compostas foram anotadas isoladamente nos casos em que foi possível identificá-las separadamente; assim, "Acórdão n. 3.472/2012-Plenário e Acórdão n. 295/2008-Plenário" receberam anotações próprias, enquanto "Acórdãos ns. 3.472/2012 e 295/2008, ambos do Plenário" foi anotado como uma única entidade.

Com base no entendimento dos dados e, especialmente, na identificação dos padrões de formação das entidades de interesse, desenvolveu-se *scripts* para a identificação e marcação

automática dos *tokens*. Além de facilitar o trabalho de marcação, tal abordagem proporcionou o benefício colateral de fornecer uma percepção sobre quais entidades responderiam melhor à aplicação do método baseado em regras. Como resultado, chegou-se ao quadro apresentado na Tabela 4.

Tabela 4 – Complexidade por classe de entidade.

Classe de entidade	Complexidade
ÓRGÃO OU ENTIDADE	**ALTÍSSIMA**
NORMATIVO LEGAL	**ALTA**
ACÓRDÃO	**MÉDIA**
PROGRAMA DE GOVERNO	**MÉDIA**
LOCALIDADE	**MÉDIA**
PRAZO DE VENCIMENTO	**BAIXA**
VALOR MONETÁRIO	**BAIXA**
DECISÃO NORMATIVA	**BAIXA**
PARECER	**BAIXA**

Fonte: Elaborado pelo autor (2019).

As informações da classe PRAZO DE VENCIMENTO, VALOR MONETÁRIO, DECISÃO NORMATIVA e PARECER têm um padrão de formação razoavelmente simples e, portanto, puderam ser anotadas em boa medida com o uso de scripts. Pela mesma razão, são classes candidatas à aplicação de regras como alternativa ao modelo estatístico para a extração e classificação de suas entidades.

A classe ÓRGÃO OU ENTIDADE, por outro lado, possui comportamento diametralmente oposto. Ele possui tamanha variação de possibilidades que se torna inviável quaisquer automatizações para sua extração. Nas análises realizadas, ficou evidente que o domínio da informação é demais amplo, em função da complexidade organizacional-administrativa pátria. Seja pela posição estatal que ocupam (superiores ou subalternos), pela estrutura em que são dispostos (simples ou compostos), pela atuação funcional que exercem (funções de governo) ou ainda pela vinculação política-administrativa a que estão submetidos (união, estado, munícios ou distrito federal), fato é que os órgãos e as entidades indicados nos acórdãos são tão específicos quanto possíveis, o que gera uma extraordinária gama de possíveis valores para a entidade, constatado pela grande variedade de qualificadores listados na Tabela 2. Como efeito, não foi possível o uso de regras automatizadas para as anotações de órgãos e entidades, restando

um oneroso esforço manual para este fim, procedimento conhecido pela literatura como ***golden standard annotation***.

Em uma escala muito menor, mas ainda complexa o suficiente, encontram-se os NORMATIVOS LEGAIS. Novamente, consultando-se a Tabela 2, percebe-se a diversidade de instrumentos legais citados nas deliberações do Tribunal. Os ACÓRDÃOS, por sua vez, estão situados na mesma faixa em razão das frequentes citações a julgados anteriores e da flexibilidade como isto ocorre nos textos, tanto no grau (referências a um ou mais acórdãos) quanto na forma (referências com ou sem a indicação do relator, do colegiado, etc.). Assim como ocorreu no caso anterior, neste caso as anotações também foram realizadas de forma manual.

Por fim, PROGRAMAS DE GOVERNO e LOCALIDADES possuem outra particularidade. Embora sejam relativamente fáceis de serem capturados por *scripts* quando bem formados ("Programa tal", "município de tal" ou "Município/Estado"), o mesmo não ocorre quando grafados isoladamente, ou seja, sem os qualificadores "Programa" e "Município" ou menção ao Estado. Nestas situações, termos como "PBF" (Programa Bolsa Família) e "Descoberto" (município de Goiás) não puderem ser identificados. Assim, além de uma rotina automatizada, fez-se necessário o uso, em caráter subsidiário, de fontes externas, tal qual a lista de programas de governo disponível no Portal da Transparência da Controladoria-Geral da União (CGU) e a Malha Municipal disponibilizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Para as duas classes, portanto, as anotações foram levadas à cabo de forma semi-supervisionada.

Na Tabela 5 apresenta-se os totais observados.

Tabela 5 – Total de anotações e termos distintos por classe de entidade.

Classe de entidade	**Total de anotações**	**Total de termos distintos**
ÓRGÃO OU ENTIDADE	**22.540**	**7.157**
NORMATIVO LEGAL	**19.645**	**4.816**
PRAZO DE VENCIMENTO	**9.202**	**91**
ACÓRDÃO	**3.204**	**2.454**
LOCALIDADE	**3.539**	**1.647**
VALOR MONETÁRIO	**1.722**	**1.465**
PROGRAMA DE GOVERNO	**1.266**	**592**
PARECER	**91**	**81**
DECISÃO NORMATIVA	**70**	**53**
Total	**61.279**	**18.355**

Fonte: Elaborado pelo autor (2019).

Realizadas as anotações, o resultado gerado foi um *array* de elementos, onde cada elemento, correspondente a um documento, era composto de um dicionário contendo, por sua vez, um texto (deliberação) e um *array* de entidades. A Figura 16 reapresenta o exemplo anterior, agora com as anotações destacadas.

Figura 16 – Exemplo de acórdão com anotações.

{"text": "Os Ministros do Tribunal de Contas da União, reunidos em Sessão de 2ª Câmara, ACORDAM, por unanimidade, com fundamento nos arts. 143, inciso V, alínea 'c', e 250, inciso II, do Regimento Interno do TCU, aprovado pela Resolução nº 246/2011, em reiterar a determinação contida no item 1.7.1 do Acórdão 1.071/2015-TCU-2ª Câmara, à Diretoria Executiva do Fundo Nacional de Saúde do Ministério de Saúde (FNS/MS), para que, em novo e improrrogável prazo de 90 (noventa) dias, a contar da ciência desta deliberação, informe a este Tribunal sobre as medidas adotadas visando à devolução dos recursos transferidos ao município de Santana do Cariri/CE, em virtude da produção insuficiente relacionada com a falta de comprovação do cumprimento da carga horária mínima prevista para os profissionais das equipes do Programa Saúde da Família (PSF), atual Estratégia de Saúde da Família (ESF), no valor de R$ 85.560,00 (oitenta e cinco mil, quinhentos e sessenta reais), bem como sobre as providências relacionadas ao saneamento das demais irregularidades apontadas no Relatório de Auditoria nº 13.786 do Departamento Nacional de Auditoria do SUS (Denasus), instaurando, se for o caso, a tomada de contas especial, sem prejuízo de fazer as seguintes determinações, de acordo com os pareceres emitidos nos autos:", "patterns": [["Acórdão 1.071/2015-TCU-2ª Câmara","ACÓRDÃO"], ["Santana do Cariri/CE","LOCALIDADE"], ["Programa Saúde da Família (PSF)","PROGRAMA"], ["Resolução nº 246/2011","NORMA"], ["90 (noventa) dias","PRAZO"], ["R$ 85.560,00","VALOR"], ["Diretoria Executiva do Fundo Nacional de Saúde do Ministério de Saúde (FNS/MS)","ÓRGÃO_ENTIDADE"]]}

Fonte: Elaborado pelo autor (2019).

4.3.4 Conversão de formato e validação dos dados

A biblioteca spaCy requer que os dados de treinamento estejam no formato JSON serializado e as entidades anotadas no padrão BILUO. Esta notação tem a seguinte configuração:

Tabela 6 – Notação BILOU.

Tag	Descrição
B EGIN	Primeiro *token* de uma entidade *multi-token*
I N	*Token* interno de uma entidade *multi-token*
L AST	Último *token* de uma entidade *multi-token*

U NIT	Único *token* de entidade
O UT	*Token* que não faz parte da entidade

Fonte: Elaborado pelo autor (2019).

À título de ilustração, o exemplo citado anteriormente anotado no formato exigido pelo spaCy é apresentado na Figura 17, com destaque para a entidade "Resolução nº 246/2011" (classe NORMA).

Figura 17 – Exemplo de acórdão com anotações no formato BILOU.

[{"id": 0, "paragraphs": [{"raw": "Os Ministros do Tribunal de Contas da União, reunidos em Sessão de 2ª Câmara, ACORDAM, por unanimidade, com fundamento nos arts. 143, inciso V, alínea 'c', e 250, inciso II, do Regimento Interno do TCU, aprovado pela Resolução nº 246/2011, em reiterar etc.:", "sentences": [{"tokens": [{"id": 0, "orth": "Os", "head": 0, "dep": "", "ner": "O"}, {"id": 1, "orth": "Ministros", "head": 0, "dep": "", "ner": "O"}, {"id": 2, "orth": "do", "head": 0, "dep": "", "ner": "O"}, {"id": 3, "orth": "Tribunal", "head": 0, "dep": "", "ner": "O"}, {"id": 4, "orth": "de", "head": 0, "dep": "", "ner": "O"}, {"id": 5, "orth": "Contas", "head": 0, "dep": "", "ner": "O"}, {"id": 6, "orth": "da", "head": 0, "dep": "", "ner": "O"}, {"id": 7, "orth": "União", "head": 0, "dep": "", "ner": "O"}, [...] {"id": 50, "orth": "Resolução", "head": 0, "dep": "", "ner": "B-NORMA"}, {"id": 51, "orth": "nº", "head": 0, "dep": "", "ner": "I-NORMA"}, {"id": 52, "orth": "246/2011", "head": 0, "dep": "", "ner": "L-NORMA"}]

Fonte: Elaborado pelo autor (2019).

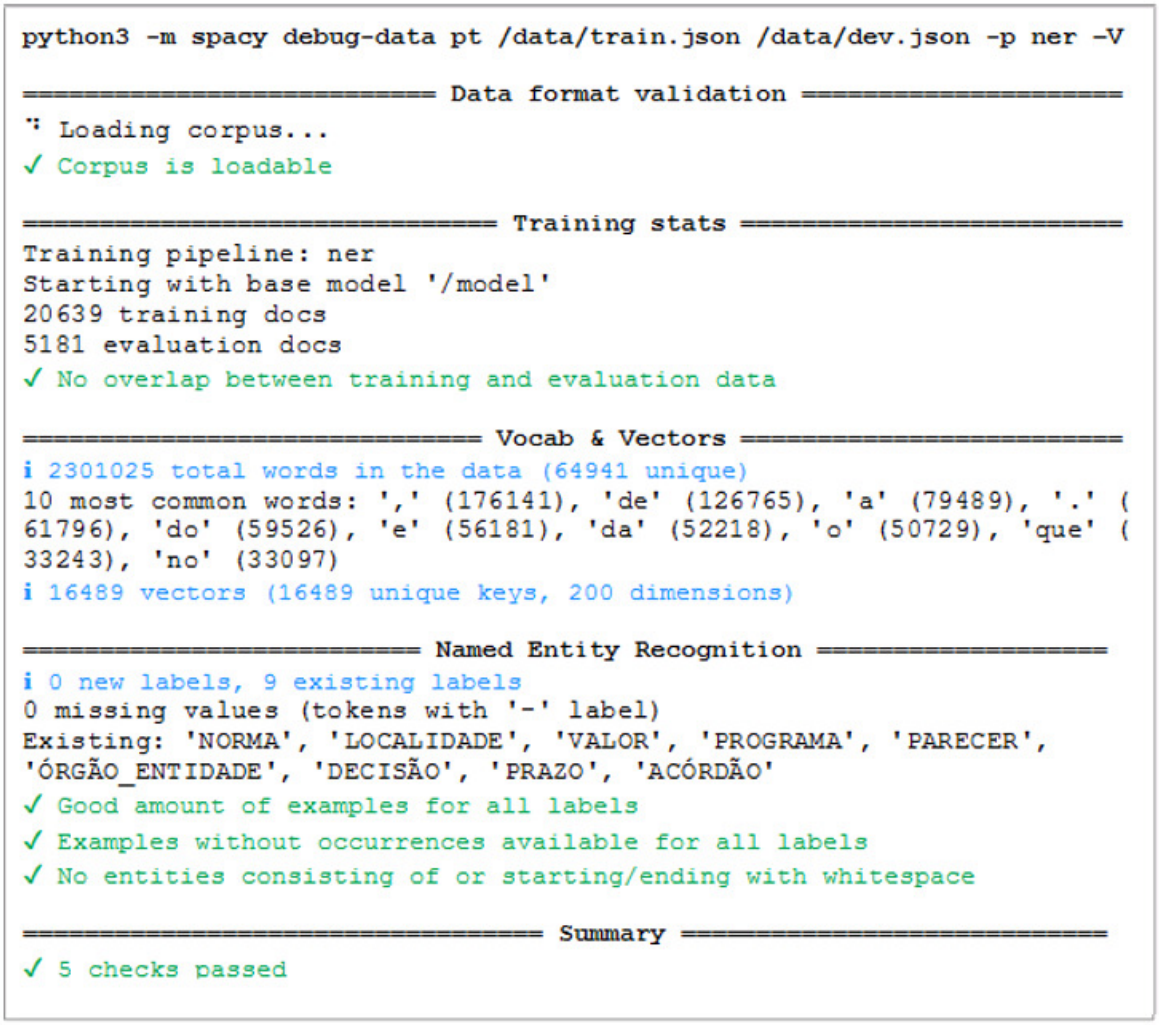
Portanto, fez-se necessário a conversão do arquivo de saída gerado na etapa prévia para o formato em questão.

Por fim, como última atividade de preparação dos dados, foi executado o comando *debug_data*, que analisa, depura e valida os dados de treinamento e desenvolvimento, gerando estatísticas úteis e identificando problemas tais como anotações inválidas, baixo volume de anotações e outros mais.

Ao fim da etapa de preparação, foram gerados 25.820 documentos anotados. Deste total, 80% foram reservados para treinamento e 20% para validação (ou desenvolvimento, segundo a denominação usada pelo spaCy).

O código-fonte utilizado para ambas as operações está disponível no APÊNDICE B – Códigos-fonte e os resultados estão apresentados na Figura 18.

Figura 18 – Resultados da verificação dos dados.



Fonte: (SPACY, 2017).

4.4 MODELAGEM

Na fase de modelagem, uma técnica é escolhida e um algoritmo que a implemente é submetido a um processo de aprendizagem com base nos dados previamente preparados. Se necessário, diversas iterações são realizadas para ajuste dos parâmetros ou dos dados de entrada a fim de se obter o modelo mais otimizado.

Na sequência, são apresentadas as três etapas executadas no aprendizado do modelo.

4.4.1 Inicialização do modelo

A primeira etapa da modelagem foi a inicialização do modelo, realizada por meio do comando *init-model*, e que teve por objetivo gerar um modelo contendo os vetores de palavras gerados previamente.

Outra operação necessária, executada via programação, foi a inclusão das entidades próprias do modelo ao componente NER do *pipeline* de processamento do spaCy.

O código-fonte utilizado para ambas as operações está disponível no APÊNDICE B – Códigos-fonte e os resultados estão apresentados na Figura 19.

Figura 19 – Resultado da inicialização do modelo.

```
python3 -m spacy init-model pt /model --vectors-loc /data/word2vec.txt

⠋ Creating model...
✓ Successfully created model

⠋ Reading vectors from /data/word2vec.txt
✓ Loaded vectors from /data/word2vec.txt
✓ Sucessfully compiled vocab
16688 entries, 16489 vectors
Created blank 'pt' model
```

Fonte: (SPACY, 2017).

4.4.2 Pré-treinamento

Outro recurso interessante do spaCy, incorporado na versão 2.1 e citado por HONNIBAL e MONTANI (2019) como uma das grandes novidades na pesquisa em PLN em 2018, é o pré-treinamento, executado por meio do comando *pretrain*. O comando treina um modelo de linguagem para prever o vetor de palavras com base nas palavras ao redor. Um modelo de linguagem atribui probabilidades a sequências de símbolos arbitrários, de modo que quanto mais provável uma sequência (w_1, w_2, ..., w_n) existe nessa linguagem, maior a probabilidade (HUYEN, 2019). O comando carrega vetores de palavras pré-treinados e treina um componente como CNN, BiLSTM, etc. de forma a projetar vetores que correspondem aos valores pré-treinados. Os pesos são salvos após cada época e o conjunto com melhor resultado pode então ser

passado como parâmetro ao comando *train* para o treinamento propriamente dito. Assim, em vez de inicializar as camadas da rede neural convolucional do spaCy com pesos aleatórios, o modelo utiliza então pesos pré-treinados.

O pré-treinamento permite, portanto, o aprendizado de representações contextuais de palavras, um diferencial dos modelos estatísticos sobre a extração baseada em regras em operações de NERC. Além de prover maior acurácia ao modelo, a representação contextual também favorece a tarefa de desambiguação de entidades.

O *pretrain* foi executado com os parâmetros em seus valores *default*, apresentados no APÊNDICE A – Hiperparâmetros(Figura 29). Conforme já citado, a arquitetura da rede neural usada na biblioteca não está disponível, contudo, a despeito deste fato, o spaCy permite a configuração de 17 hiperparâmetros no pré-treinamento e 31 no treinamento do modelo. As simulações envolvendo tal variedade de opções de otimização, no entanto, requerem tempos de processamento e análise que estão além daquelas disponíveis para o presente estudo. Portanto, tai refinamentos foram deixados para trabalhos futuros, limitando-se a presente análise aos valores *default*.

Os experimentos foram realizados em um ambiente virtualizado VMware vSphere 6.7, no cluster Corporate Gold que contém exatamente 20 hosts ESXi (Huawei CH242 V3 DDR4) totalizando 960 processadores do tipo Intel (R) Xeon (R) CPU E7-4830 v3 @ 2.10GHz, onde cada host possui a seguintes especificações: 96 processadores lógicos, 48 CPUs virtuais e 1TB de memória RAM. A máquina utilizada possui 28 CPUs virtuais, 300GB de memória RAM, 512GB de armazenamento, sistema operacional CentOS 7 (64-bit) e foi utilizada de maneira compartilhada com outros usuários.

Foram executadas um total de 100 épocas, com uma duração de 10 minutos por iteração. Os resultados são mostrados a seguir.

Figura 20 – Resultado do pré-treinamento do modelo (tabular).

python3.6 -m spacy pretrain ../data/full-origin-pretrain.jsonl ../model/initial ../model/pretrain -i 100

ℹ Not using GPU

✓ Created output directory

✓ Saved settings to config.json

Loading input texts...

Loading input texts...

✓ Loaded input texts

Loading model '../model/initial'...

✓ Loaded model '../model/initial'

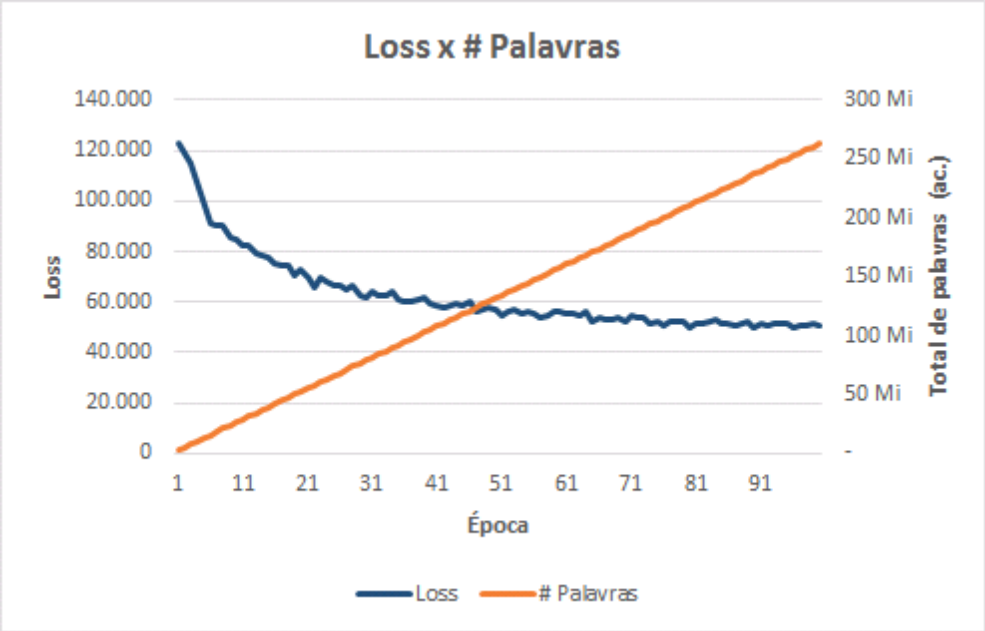
========== Pre-training tok2vec layer - starting at epoch 0 ==========

#	# Words	Total Loss	Loss	w/s
0	302861	300939.375	300939	3463
0	607127	595155.531	294216	3501
0	916996	886748.125	291592	3489
0	1219218	1157149.50	270401	3488
0	1532243	1421349.62	264200	3428
0	1836565	1661081.48	239731	3345
0	2147102	1888197.58	227116	3305
0	2452850	2102023.25	213825	3497
0	2633765	2224844.08	122820	3433
...				
99	261050657	89673126	83446	3536
99	261356853	89756086	82960	3531
99	261664289	89839415	83328	3553
99	261965972	89921099	81683	3507
99	262270601	90003892	82792	3538
99	262578880	90088092	84199	3538
99	262887683	90172102	84010	3521
99	263191543	90254841	82739	3521
99	263376500	90305470	50628	3493

✓ Successfully finished pretrain

Fonte: (SPACY, 2017).

Figura 21 – Resultado do pré-treinamento do modelo (gráfico).



Fonte: Elaborado pelo autor (2019).

4.4.3 Treinamento

O treinamento foi executado por meio do comando *train*, tendo como parâmetro o modelo pré-treinado previamente. Assim como ocorreu no pré-treinamento, foram utilizados os valores *default* dos hiperparâmetros (APÊNDICE A – Hiperparâmetros– Figura 30) e executadas 100 épocas, com uma duração de 10 minutos por iteração.

No APÊNDICE B – Códigos-fonte, encontra-se o código-fonte utilizado. Os resultados são apresentados adiante.

Figura 22 – Resultado do treinamento do modelo (tabular).

python3.6 -m spacy train pt ../model/train ../data/train.json ../data/dev.json -p ner -t2v ../model/pretrain/model99.bin -n 100

Training pipeline: ['ner']

Starting with blank model 'pt'

Counting training words (limit=0)

Loaded pretrained tok2vec for: ['ner']

Itn	NER Loss	NER P	NER R	NER F	Token %	CPU WPS
1	71890.882	91.470	93.379	92.414	100.000	19506
2	36937.499	92.317	93.821	93.063	100.000	19114
3	30505.960	92.370	94.033	93.194	100.000	18244
4	27141.864	92.263	94.181	93.212	100.000	17956
5	25073.377	92.314	94.279	93.286	100.000	19089
...						
95	6008.285	93.112	94.819	93.958	100.000	19699
96	5762.671	93.106	94.835	93.963	100.000	19621
97	6032.785	93.139	94.885	94.004	100.000	20160
98	6112.024	93.147	94.893	94.012	100.000	19510
99	6102.180	93.149	94.917	94.025	100.000	18902
100	5949.375	93.106	94.958	94.023	100.000	19981

✓ Saved model to output directory

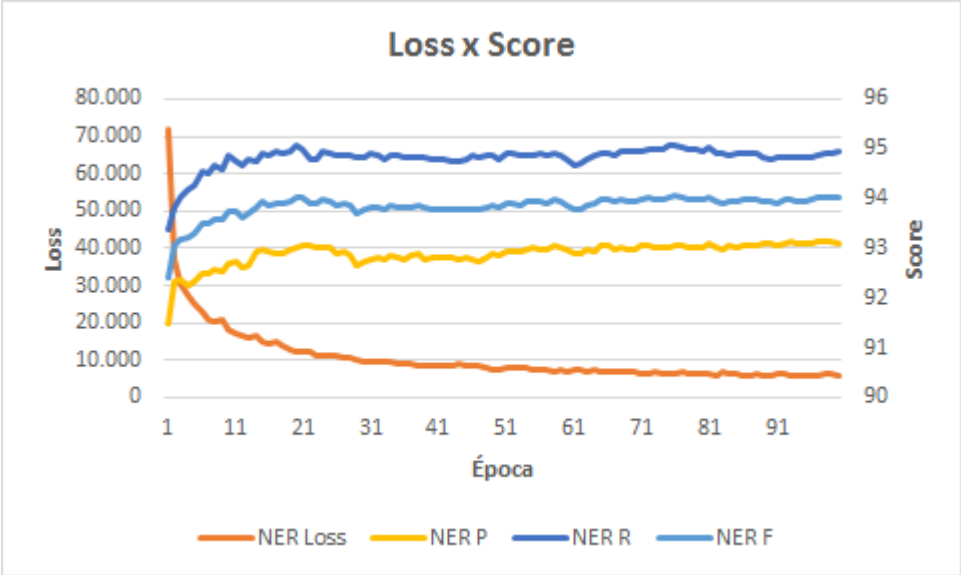
../model/train/model-final

✓ Created best model

../model/train/model-best

Fonte: (SPACY, 2017).

Figura 23 – Resultado do treinamento do modelo (gráfico).



Fonte: (SPACY, 2017).

Os valores de precisão (NER P), revocação (NER R) e F1-score (NER F) apresentados durante o treinamento foram de 93%, 95% e 94%, respectivamente. Como pode ser visto na Figura 23, nota-se que o comportamento exponencial da função de custo: seus valores decrescem rapidamente até a décima iteração, a partir do qual a curva passa a ter uma derivada quase nula. Como efeito, o mesmo comportamento ocorre, em sentido inverso, nas medidas de desempenho.

4.5 AVALIAÇÃO

Para a avaliação do modelo, foi gerada uma base de testes da mesma fonte utilizada no treinamento, porém com acórdãos mais recentes, compreendendo o período de 01/08/2019 a 31/01/2020. No total, foram anotadas 2.298 entidades em 386 acórdãos e 752 deliberações.

4.5.1 Teste do modelo disponível

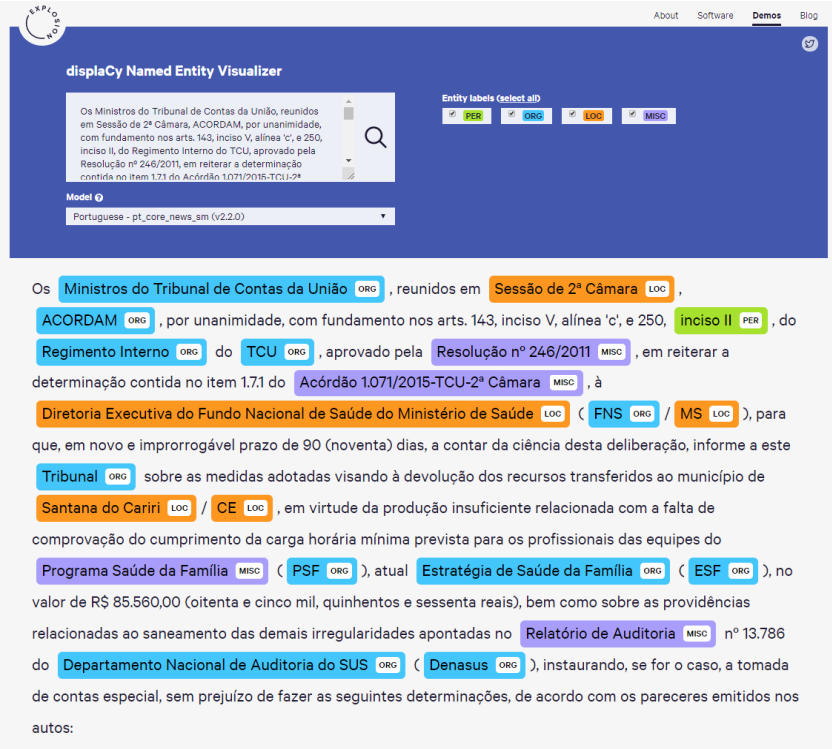
O spaCy suporta uma variedade de idiomas. Atualmente, estão disponíveis na biblioteca modelos estatísticos pré-treinados em 11 línguas, incluindo o português.

Assim, antes de se proceder com o treinamento de um modelo customizado, o modelo disponibilizado foi submetido a uma avaliação quanto a sua eficácia sobre o corpus de teste, para fins comparativos.

O modelo em língua portuguesa acessível no spaCy é o "pt_core_news_sm", versão 2.2.0. Ele foi treinado no WikiNER (NOTHMAN et al., 2013), corpus formado por artigos da Wikipedia, e no Bosque (AFONSO et al., 2002), *treebank*[6] português pertencente ao projeto *Universal Dependencies* (UD)[7]. Construído sobre uma rede neural convolucional, o modelo oferece suporte à identificação das classes PER, LOC, ORG e MISC.

Para o teste do modelo, utilizou-se, preliminarmente, o exemplo de acórdão já citado neste trabalho e a ferramenta online "displaCy Named Entity Visualizer"[8], oferecida pela mesma empresa desenvolvedora do spaCy. A ferramenta realiza a extração de entidades identificadas em um texto qualquer fornecido pelo usuário, a partir do modelo selecionado por ele. O resultado está apresentado na Figura 24.

Figura 24 – Exemplo de extração com uso de modelo pré-treinado em língua portuguesa.



Fonte: *displaCy Named Entity Visualizer* (Explosion)

[6] Corpus que contém anotações referentes à estrutura sintática e semântica dos termos.
[7] Para maiores detalhes, consultar https://universaldependencies.org/.
[8] Disponível em https://explosion.ai/demos/displacy-ent.

Concernente apenas às entidades comuns ao presente trabalho – ORG e LOC, viu-se que o modelo não as identificou adequadamente. No tocante à primeira, a entidade "Diretoria Executiva do Fundo Nacional de Saúde do Ministério da Saúde" foi identificada como LOC. Além disso, foram também classificadas nesta categoria, erroneamente, os termos "ACORDAM", "Regimento Interno", "Estratégia de Saúde da Família" e "ESF". Nas entidades da classe LOC do exemplo avaliado, o modelo se saiu melhor, embora tenha identificado "Sessão de 2ª Câmara" incorretamente como tal.

Passando-se ao teste sobre toda a base, obteve-se o resultado apresentado na Tabela 7.

Tabela 7 – Resultado da avaliação do modelo disponível.

Classe da entidade	**Precisão (%)**	**Revocação (%)**	**F1-Score (%)**
ORG	8,86	21,02	12,47
LOC	0,58	16,94	1,12

Fonte: Elaborado pelo autor (2019).

Como foi possível perceber, a performance obtida foi desprezível. Tal fato não causa surpresa, visto que os corpora nos quais o modelo foi treinado difere de maneira substancial do corpus usado na presente pesquisa.

Em rápido escrutínio no conjunto de textos utilizados pelo modelo pré-definido, foi possível identificar as discrepâncias que justificam seu baixo desempenho quando aplicado ao corpus formado pelas deliberações do Tribunal. Enquanto neste último, ÓRGÃO OU ENTIDADE refere-se a entidades públicas da administração pública brasileira que, grosso modo, não possuem ampla menção na Wikipedia, naquele, a semântica é abrangente, onde ORG remete a um vasto leque de possibilidades, tais como organismos internacionais, empresas, partidos políticos, universidades e igrejas.

O mesmo ocorre com a entidade LOC, que abrange desde trópicos até rios, bem diferente dos municípios brasileiros que foram o objeto da entidade LOCALIDADE no presente trabalho.

Ainda, os resultados também dependem de premissas adotadas na anotação das entidades. No exemplo citado, notou-se que o nome e a sigla do órgão, ainda que grafados na sequência, foram identificados como entidades separadas, a despeito do fato de representarem um

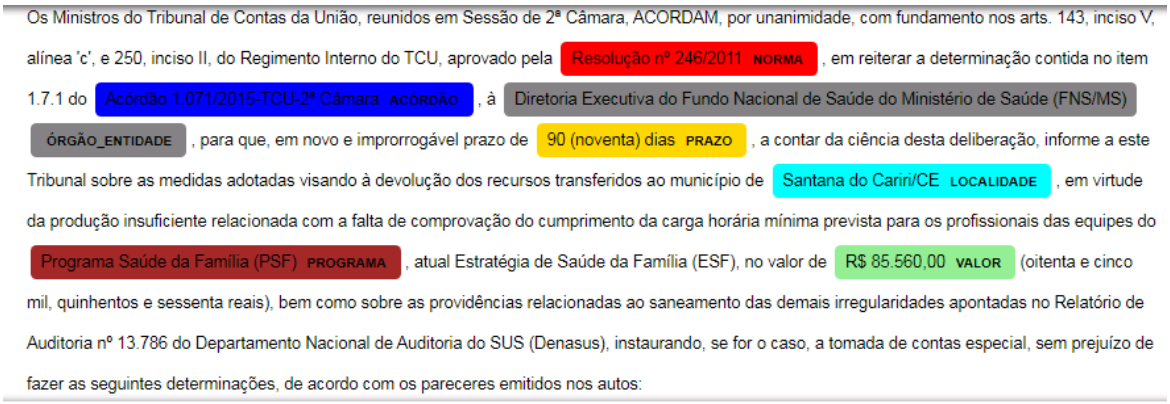
único elemento do ponto de vista semântico. De forma análoga, o nome e a unidade da federação do munícipio receberam marcações individualizadas de localidade. No caso ora em estudo, por outro lado, os casos citados receberam uma única anotação.

Portanto, dado que modelos estatísticos fazem previsões com base nos exemplos em que foram treinados e que a precisão depende do domínio de interesse, bem como dos critérios adotados durante as anotações, é natural que modelos customizados tenham desempenhos superiores a aqueles de espectro mais genérico, como foi caso observado no presente estudo.

4.5.2 Teste do modelo treinado

Inicialmente, aplicou-se o modelo treinado ao exemplo usado neste trabalho, o que produziu a saída adiante, que é exatamente a mesma da Figura 14.

Figura 25 – Exemplo de acórdão com as entidades previstas em destaque.



Fonte: Elaborado pelo autor (2019).

Em seguida, o modelo foi confrontado com a base de testes, por meio do comando *evaluate*, cujo código se encontra transcrito no APÊNDICE B – Códigos-fonte.

Os resultados são apresentados na Tabela 8 adiante.

Tabela 8 – Resultado da avaliação do modelo treinado.

Classe da entidade	Total de entidades[9]	Precisão (%)	Revocação (%)	F1-Score (%)
ÓRGÃO_ENTIDADE	844	90,99	86,58	**88,73**
NORMA	783	93,74	86,04	**89,73**
PRAZO	295	95,93	79,49	**86,94**
ACÓRDÃO	145	91,03	85,71	**88,29**
LOCALIDADE	50	90	76,27	**82,56**
VALOR	39	100	100	**100**
PROGRAMA	39	82,05	50,79	**62,74**
PARECER	3	100	75	**85,71**
DECISÃO	1	100	100	**100**

Fonte: Elaborado pelo autor (2019).

A análise dos resultados demonstrou que:

- ÓRGÃO OU ENTIDADE, NORMATIVOS LEGAIS (“NORMA”), PRAZO DE VENCIMENTO (“PRAZO”) e ACÓRDÃO obtiveram bons desempenhos, em especial os dois primeiros, dada a alta complexidade inerente à formação dessas classes, conforme relatado previamente;

- LOCALIDADE apresentou desempenho razoável; no entanto, a baixa quantidade de ocorrências na base de teste não permitiu se extrair maiores conclusões a respeito;

- VALOR MONETÁRIO (“VALOR”), PARECER e DECISÃO NORMATIVA (“DECISÃO”), embora tenham obtido níveis satisfatórios de acurácia, também carecem de uma avaliação em bases mais amplas; e

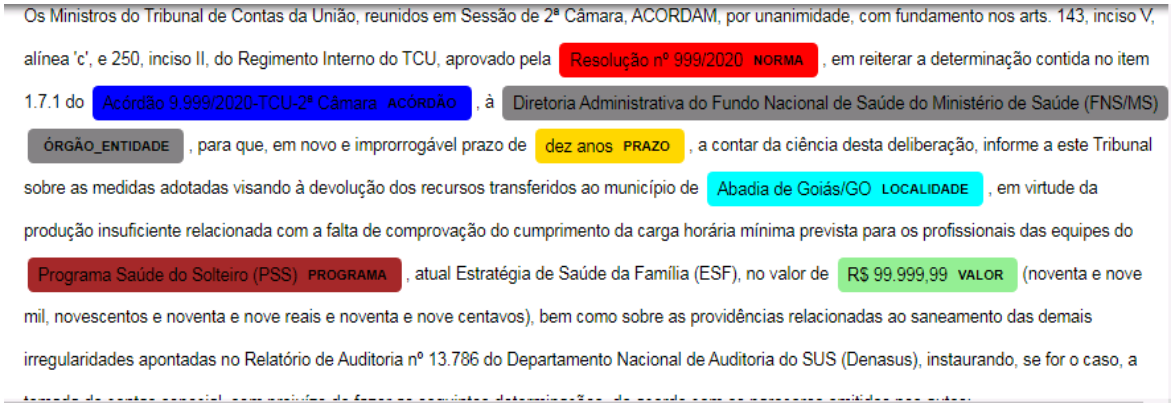
- PROGRAMA DE GOVERNO (“PROGRAMA”) igualmente não foi submetido a uma avaliação com uma grande massa de testes; aqui, porém seu baixo *score* chama a atenção; atribui-se o fato a um menor número de incidências desta classe nos acórdãos, o que implica um menor volume de entidades treinadas; outro elemento contribuinte, embora em menor grau,

[9] Total de entidades previstas pelo modelo, incluindo as repetições.

é o efeito das ocorrências sem qualificador, já relatado, quando a entidade é grafada abreviada ou por meio de sigla, tais como “Procrofe”, “Pnae” e “PDDE”, o que dificulta a descoberta do contexto em que ela está inserida;

Foram executados ainda dois testes pontuais. No primeiro, verificou-se a capacidade do modelo em identificar entidades não presentes na base usada para treinamento. Para tanto, substitui-se as entidades reais por fictícias, inexistentes na base de treinamento, e submeteu-se a “nova” deliberação ao modelo. Como pode ser visto na Figura 26, todas as entidades foram corretamente identificadas.

Figura 26 – Exemplo de acórdão com entidades não treinadas previstas em destaque.

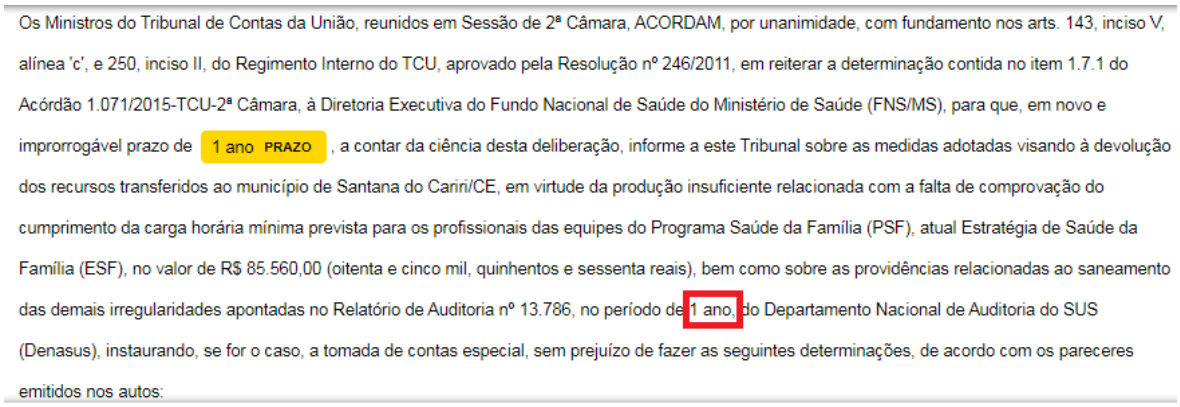


Fonte: Elaborado pelo autor (2019).

No segundo teste, desejava-se saber se o modelo tinha a habilidade de promover algum nível de desambiguação. Usando o mesmo exemplo base, foi incluída a expressão “1 ano” em dois locais do texto, um em referência ao prazo de vencimento da determinação e outro relativo à duração das irregularidades apontadas. Vê-se na Figura 27[10] que, acertadamente, apenas a primeira ocorrência foi marcada como entidade da classe PRAZO, enquanto nada foi apontado na segunda (destaque do autor).

[10] As demais entidades foram omitidas para melhor destaque do caso em questão.

Figura 27 – Exemplo de acórdão com caso de desambiguação.



Fonte: Elaborado pelo autor (2019).

Ambos os testes parecem supor que tanto a descoberta de novas entidades quanto a desambiguação parecem funcionar em alguma medida no modelo treinado. Contudo, a confirmação de tais propriedades ainda carece de testes mais abrangentes.

Por último, verificou-se a ocorrência de 35 casos previstos corretamente, porém não anotados ou anotados indevidamente. Dentre eles, destacaram-se:

- Os acórdãos indicados na expressão “acórdãos 1884/2010 - Relator Ministro Benjamin Zymler; 307/2011 - Relator Ministro-Substituto Augusto Sherman; 2962/2012 - Relator Ministro José Múcio Monteiro; 3400/2012” foram anotados separadamente; no entanto, o modelo indicou tratar-se de uma única entidade, o que se coaduna com as premissas adotadas neste estudo, no sentido de se registrar em conjunto entes que não subsistem isolados do ponto de vista semântico, quando presente apenas um único termo qualificador;

- As entidades “Concórdia do Pará/PA” e “Maurilândia do Tocantins/TO” foram anotadas erroneamente como pertencentes à classe PRAZO, porém classificadas adequadamente como LOCALIDADES pelo modelo;

- A entidade “Recomendação Conjunta 66/2014 do Ministério Público Federal e da Controladoria-Geral da União” foi anotada apenas como “Recomendação Conjunta 66/2014”, contudo reconhecida acertadamente pelo modelo em toda a sua extensão descritiva; e

- Os órgãos indicados no trecho “Determinar ao Ministério da Cidadania e ao Dnit-SRE/MT, que...” foram incorretamente anotados como uma única entidade, porém perfeitamente capturados em separado pelo modelo.

4.6 IMPLANTAÇÃO

A última etapa de um projeto de mineração é a disponibilização do produto desenvolvido no ambiente organizacional para uso em regime de produção pelos usuários-alvo com vistas à melhoria esperada dos processos de negócio. Envolve o empacotamento e a documentação da solução, bem como a definição da estratégia de manutenção e evolução.

No presente estudo, não foi possível alcançar este estágio, visto que algumas tarefas ainda são necessárias para a produção de uma versão final e operacional, tais como a otimização do modelo e a normalização das entidades.

Ademais, o objetivo primeiro do trabalho foi avaliar a aplicabilidade de utilização de modelo de NERC para a extração de entidades na base textual de deliberações, com uso da biblioteca spaCy. A partir daí, confirmada a proficuidade da solução, abre-se espaço para as etapas subsequentes com vistas a sua complementação, otimização e empacotamento como produto final e acabado. Ademais, haverá de se avaliar a necessidade de proposição efetiva de ação formal, em linha com o planejamento institucional, de forma a legitimar a alocação dos recursos necessários, assim como oficializar o envolvimento das unidades participantes.

5 CONSIDERAÇÕES FINAIS

Este capítulo trata das conclusões e apresenta os trabalhos futuros sugeridos a partir dos resultados obtidos neste trabalho.

5.1 CONCLUSÕES

A aplicação da técnica de extração e classificação de entidades na base textual de acórdãos do TCU, por meio de modelo estatístico disponível na biblioteca spaCy, se mostrou muito promissor para a maioria das classes pesquisadas. Os resultados demonstraram acurácia acima de 85% em 7 das 9 classes de entidades avaliadas. Em especial, a identificação de órgãos e entidades e de normativos legais em seus mais variados instrumentos apresentou, mesmo em um modelo não otimizado, resultado satisfatório o suficiente para embasar a continuidade das pesquisas.

No entanto, algumas classes, como locais e programas de governo, requerem uma avaliação mais cuidadosa e, possivelmente, uma solução mais abrangente. Nestes casos, onde o domínio da informação é finito e conhecido, uma tarefa complementar de vinculação com bases de referências pode ser necessária para assegurar maior cobertura na extração. Ainda assim, no caso de LOCALIDADE, foi possível obter uma acurácia acima de 80%, o que já permite extrair uma razoável noção quanto ao alcance da atuação do Tribunal no âmbito nacional, como mostra a figura seguinte contendo a indicação dos municípios citados nas deliberações exaradas durantes os últimos dez anos.

Figura 28 – Municípios citados nas deliberações nos últimos dez anos.

Fonte: Elaborado pelo autor (2019).

Por fim, o emprego de modelos estatísticos para a captura de entidades bem formadas, tais como prazos e valores monetários, cujas notações são relativamente padronizadas, parece não ser a melhor abordagem. Os resultados obtidos sugerem que a extração baseada em regras oferece uma melhor alternativa de custo x benefício.

Em suma, a evolução dos algoritmos de aprendizado de máquina tem permitido alavancar as técnicas de mineração textual como a NERC, especialmente quando aplicada sobre um modelo customizado e treinado em corpus específico. No entanto, os estudos sugerem que as análises devem ser realizadas classe a classe, visto que o desempenho do modelo é afetado não apenas pela extensão do treinamento, como também pelas propriedades de cada entidade, tais como seu padrão de ocorrência e seu grau de variabilidade. No caso presente, o estudo apontou a viabilidade para a extração semiautomática de órgãos/entidades, normativos legais e acórdãos por meio de modelos puramente estatísticos, de programas de governo e municípios por meio de modelos estatísticos complementado com métodos de pesquisa, e prazos de vencimento, valores monetários, pareceres e decisões normativas por meio regras.

5.2 TRABALHOS FUTUROS

Dada a limitação de tempo, não foi possível neste trabalho a execução de simulações com variações na parametrização do modelo. Conforme já citado, o spaCy oferece a possibilidade de refinamento de quase 50 hiperparâmetros nas rotinas de pré-treinamento e treinamento. Também é possível ajustar a dimensionalidade e o tamanho da janela deslizante usado pelo algoritmo de geração do vetor de palavras. Visto que tais operações consomem considerável tempo de processamento, elas devem somente ser realizadas após a compreensão de cada parâmetro e seu possível efeito no contexto da solução, de forma a se identificar os candidatos mais promissores. Por outro lado, a otimização de qualquer modelo é essencial para a maximização de sua precisão e deve ser realizada na sequência do presente estudo.

Outra tarefa necessária é a resolução das entidades, conhecida por Vinculação de Entidade Nomeada (*Named-Entity Linking* – NEL)[11], a fim de atribuir identidade única às entidades por meio de correspondência dos termos a bases externas de referência (de municípios, de órgãos ou de programas de governo, por exemplo) . Trata-se de etapa imprescindível, pois além de erros ortográficos, uma mesma entidade pode ser grafada de diversas formas. É o caso de ÓRGÃO OU ENTIDADE, cujo nome ora é escrito por extenso, ora abreviado, ora com sufixo indicativo de sua natureza comercial (p. ex. “S. A.”). À título de exemplo, foram encontradas 21 anotações diferentes para a entidade “Petrobras”. Portanto, o desafio de converter informação não-estruturada em estrutura passa inevitavelmente pela regularização dos dados.

Por fim, porém igualmente relevante é a tarefa de resolução de coreferências (*coreference resolution*), que consiste em identificar os diferentes termos vinculados a mesma referência semântica. No exemplo “recomendar à Mesa do Congresso Nacional e à sua Comissão Mista de Planos, Orçamentos Públicos e Fiscalização que...”, o modelo identificou duas entidades distintas da classe ÓRGÃO OU ENTIDADE – Mesa do Congresso Nacional e Comissão Mista de Planos, Orçamentos Públicos e Fiscalização. A segunda, no entanto, não possui acepção própria (comissão de onde ?), carecendo, portanto, de tratamento complementar para que sua referência (Congresso Nacional) seja identificada.

[11] Também denominada de Desambiguação de Entidade Nomeada (*Named-Entity Disambiguation* – NED), Normalização de Entidade Nomeada (*Named-Entity Normalization* – NEN), ou simplesmente NERD.

REFERÊNCIAS

AFONSO S.; BICK, E.; Haber, E.; SANTOS, D. Floresta sintá(c)tica: a treebank for Portuguese. In: Proceedings of the Third International Conference on Language Resources and Evaluation. **LREC**. Las Palmas de Gran Canaria, Spain. 2002. p. 1698-1703.

ANDRADE, Patrícia Helena Maia Alves de. **Aplicação de técnicas de mineração de textos para classificação de documentos**: um estudo da automatização da triagem de denúncias na CGU. 2015. xi, 54 f., il. Dissertação (Mestrado Profissional em Computação Aplicada). Universidade de Brasília, Brasília, 2015.

ARAUJO, P. H. L. de; CAMPOS, T. E. de; OLIVEIRA, R. R. R. de; STAUFFER, M.; COUTO, S.; BERMEJO, P. LeNER-Br: A Dataset for Named Entity Recognition in Brazilian Legal Text. In Proceedings of the International Conference on the Computational Processing of Portuguese. Canela, Rio Grande do Sul. 2018. p. 313–323.

BATISTA, David S. **Named-Entity evaluation metrics based on entity-level**. 2018. Disponível em: <http://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation>. Acesso em: 6 mar. 2020.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing with Python**. O'Reilly Media. 2009.

BROWNLEE, Jason. **What Are Word Embeddings for Text?** Machine Larning Mastery. 2017. Atualizado em 2019. Disponível em: <https://machinelearningmastery.com/what-are-word-embeddings>. Acesso em: 6 mar. 2020.

DALE, R.; MOISL, H.; SOMERS, H.M. **Handbook of Natural Language Processing. Computational Linguistics**, vol. 27, 2000. p. 602–603.

DOZIER, C.; KONDADADI, R.; LIGHT M.; Named Entity Recognition and Resolution in Legal Text. In Semantic Processing of Legal Texts. Springer, Heidelberg. 2010. p. 27–43.

DUTRA E SILVA, Luis André. **Use of Deep Learning in Oversight**. 2016.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery: An overview. In Advances in knowledge discovery and data mining. AAAI Press. Menlo Park, CA. 1996. p. 1–34.

FELDMAN, Ronen; SANGER, James. **The Text Mining Handbook**: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press. Cambridge, Inglaterra. 2006.

FREITAG, Dayne. **Machine learning for information extraction in informal domains.** Ph.D. thesis, Carnegie Mellon University. Pittsburgh, EUA. 1998.

GOLDBERG, Yoav. Neural network methods for natural language processing. Synthesis Lectures on Human Language Technologies. Graeme Hirst, University of Toronto. 2017.

GRISHMAN, Ralph; SUNDHEIM, B. Message Understanding Conference - 6: A Brief History. In Proc. International Conference on Computational Linguistics, pp. 466–471. **COLING**. 1996.

HAN, L.; XIAODONG, Z.; DEREK, F. W.; LIDIA, S. C. 2015. Chinese named entity recognition with Graph-based semi-supervised learning model. In Proceedings of the 8th SIGHAN Workshop on Chinese Language Processing (SIGHAN-8), pp. 15–20. Association for Computational Linguistics & Asian Federation of Natural Language Processing. 2015.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. **Long short-term memory. Neural computation.** vol. 9, n. 8, pp. 1735-1780, 1997.

HONNIBAL, M. e MONTANI, I. **Introducing spaCy v2.1**. 2019. Disponível em: < https://explosion.ai/blog/spacy-v2-1 >. Acesso em 6 mar. 2020.

HUYEN, C. Evaluation Metrics for Language Modeling. The Gradient. 2019. Disponível em: < https://thegradient.pub/understanding-evaluation-metrics-for-language-models >. Acesso em 6 mar. 2020.

IBM. IBM® SPSS® Modeler **CRISP-DM (Cross-Industry Standard Process for Data Mining) Guide**. [S.l: s.n.]. 2014. Disponível em: <https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm>. Acesso em: 6 mar. 2020.

LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning (ICML). ACM, 2001.

LAMPLE, G.; BALLESTEROS, M.; SUBRAMANIAN, S.; KAWAKAMI, K.; DYER, C. Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT. Association for Computational Linguistics (ACL), San Diego, California, 2016. p. 260–270.

MIKOLOV, T.; YIH, W; ZWEIG, G. Linguistic regularities in continuous space word representations. In Proceedings of NAACL-HLT. 2013. p. 746–751.

MISHRA, B. S. P.; DEHURI, S.; KIM, E.; WANG, G. **Techniques and Environments for Big Data Analysis**: Parallel, Cloud, and Grid Computing. Springler. 2016.

NADEAU, David; SEKINE Satoshi. **A survey of named entity recognition and classification.** Linguisticae Investigationes, vol. 30, no. 1, 2007. p. 3–26.

NANCY, Chinchor; SUNDHEIM, Beth. MUC-5 Evaluation Metrics. In Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held. Baltimore, Maryland. 1993. p. 69–78.

NOTHMAN, J.; RINGLAND, N.; RADFORD, W.; MURPHY, T.; CURRAN, J. R. **Learning multilingual named entity recognition from wikipedia**. Artificial Intelligence, 194. 2013. p. 151–175.

REHUREK, Radim; SOJKA, Petr. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Eelra. Valletta, Malta. 2010. p. 45–50.

SAHLGREN, M. **A brief history of word embeddings (and some clarifications).** 2015. Disponível em: <https://www.linkedin.com/pulse/brief-history-word-embeddings-some-clarifications-magnus-sahlgren>. Acesso em: 6 mar. 2020.

SEPROC. **Manual de Procedimento.** Minuta. Secinf. 2019.

SPACY. HONNIBAL, Matthew; MONTANI, Ines. **spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing**. 2017. Disponível em: < https://spacy.io/>. Acesso em: 6 mar. 2020.

TAN, A.-H; MUI, H.; TERRACE, K. **Text Mining**: The state of the art and the challenges. In Proceedings, PAKDD'99 workshop on Knowledge Discovery from Advanced Databases. Beijing. 1999.

TCU. **O TCU e o Desenvolvimento Nacional.** Contribuições para a Administração Pública. Tribunal de Contas da União. Brasília. 2ª edição. TCU, Segecex, 2018b. Disponível em: <https://portal.tcu.gov.br/biblioteca-digital/o-tcu-e-o-desenvolvimento-nacional.htm>. Acesso em: 6 mar. 2020.

TCU. **Relatório da Secretaria-Geral de Controle Externo 2017/2018.** Tribunal de Contas da União. Brasília: TCU, Segecex, 2018a. Disponível em: <https://portal.tcu.gov.br/relatorio-da-secretaria-geral-de-controle-externo.htm>. Acesso em: 6 mar. 2020.

TCU. **Relatório Anual de Atividades do TCU 2018.** Tribunal de Contas da União. Brasília: TCU, 2019a. Disponível em: <https://portal.tcu.gov.br/transparencia/relatorios/relatorios-de-atividades/relatorios-de-atividades.htm>. Acesso em: 6 mar. 2020.

TCU. **Relatório de Políticas e Programas de Governo 2018 (RePP).** Sumário Executivo. Tribunal de Contas da União. Brasília: TCU, 2019b. Disponível em: <https://portal.tcu.gov.br/biblioteca-digital/relatorio-de-politicas-e-programas-de-governo-2018.htm>. Acesso em: 6 mar. 2020.

TORRES, D. **Data science: o que é e como pode ajudar o seu negócio**. Neurotech. 2019. Disponível em: <https://neurotech.com.br/blog/data-science.html>. Acesso em: 6 mar. 2020.

APÊNDICE A – Hiperparâmetros

A seguir são apresentados os valores default utilizados nas rotinas de pré-treinamento e de treinamento.

Pré-treinamento

Figura 29 – Hiperparâmetros de pré-treinamento.

NOME	DESCRIÇÃO	DEFAULT
width	Largura das camadas CNN	96
depth	Profundidade das camadas CNN	4
cnn-window	Tamanho da janela das camadas CNN	1
cnn-pieces	Tamanho *maxout* das camadas CNN	3
use-chars	Incorporação baseada em caracteres	Falso
sa-depth	Profundidade das camadas *self-attention*	0
embed-rows	Número de linhas incorporadas	2000
loss-func	Função *loss* para usar como objetivo (“L2” ou “cosine”)	cosine
dropout	Taxa de *dropout*	0,2
batch-size	Número de palavras por lote de treinamento	3000
max-length	Máximo de palavras por exemplo	500
min_length	Mínimo de palavras por exemplo	5
seed	Semente para geradores de números randômicos	0

Fonte: (SPACY, 2017).

Treinamento

Figura 30 – Hiperparâmetros de treinamento.

NOME	DESCRIÇÃO	DEFAULT
dropout_from	Taxa inicial de dropout	0,2
dropout_to	Taxa final de dropout	0,2
dropout_decay	Taxa de mudança do dropout	0
batch_from	Tamanho inicial do lote	1
batch_to	Tamanho final do lote	64
batch_compund	Taxa de aceleração do tamanho do lote	1,001
token_vector_width	Largura das tabelas de incorporação e das camadas convolucionais	128
embed_size	Número de linhas das tabelas de incorporação	7.500
hidden_width	Tamanho das camadas ocultas de parser e NER	128
learn_rate	Taxa de aprendizado	0,001
optimizer_B1	Adam solver momentum	0,9
optimizer_B2	Adam solver adagrad-momentum	0,999
optimizer_eps	Adam solver epsilon value	1e-08
L2_penalty	Penalidade do L2 regularization	1e-06
grad_norm_clip	Gradiente L2 norm constraint	1

Fonte: (SPACY, 2017).

APÊNDICE B – CÓDIGOS-FONTE

A seguir são apresentados os códigos-fonte utilizados no desenvolvimento do modelo. Os códigos foram escritos em Python e executados na plataforma Jupyter Notebook.

Código-fonte para pré-processamento

```
from gensim.models import Word2Vec
import nltk

import json
import io
import time

eta = time.time()

data_dir="../data"
texts = []

# leitura do corpus
with io.open(data_dir+'/corpus-train.json', encoding='utf8') as f_corpus:
    data_corpus = json.load(f_corpus)

for i, record in enumerate(data_corpus):
    texts.append(record['text'])

## pré-processamento

# remoção de pontuação
sentences = []
for i in range(len(texts)):
    sentences = [re.sub(pattern=r'[\!"#$%&\*+,-./:;<=>?@^_`()|~=]',
                        repl='',
                        string=x
                       ).strip().split(' ') for x in texts[i].split('\n')]
    sentences = [x for x in sentences if x != ['']]
    texts[i] = sentences

# remoção de stop words
nltk.download('stopwords')
nltk.download('punkt')
stopwords = nltk.corpus.stopwords.words('portuguese')

for i in range(len(texts)):
    texts[i][0] = [w for w in texts[i][0] if w not in stopwords]

# concatenação de todas as sentenças em uma única lista
all_sentences = []
for text in texts:
    all_sentences += text

# treinamento do modelo
model = Word2Vec(all_sentences,
                 min_count=5,
                 size=200,
                 workers=2,
                 window=10,
                 iter=30)

# gravação do modelo
model.save(data+'/word2vec')
model.wv.save_word2vec_format(data+'/word2vec.txt', binary=False)

print ("Tempo de processamento: ", int((time.time()-eta)/60),"min")
```

Código-fonte para conversão de formato e verificação dos dados

```
import subprocess
import json
import jsonlines
import io
import time

eta = time.time()

def gera_doc(nlp, record):
    patterns = [{'label': label, 'pattern': pattern} for pattern, label in record['patter
ns']]
    er = spacy.pipeline.EntityRuler(nlp, patterns=patterns)
    doc = er(nlp(record['text']))
    doc.is_parsed = True
    return doc

def run(command):
    print (command)
    result = subprocess.run(command, shell=True, stdout=subprocess.PIPE, stderr=subproces
s.PIPE, universal_newlines=True)
    print(result.stdout)
    if result.stderr is not None: print(result.stderr)

# define variáveis e entidades
data="../data"
initmodel="../model/initial"

## converte os dados para o formato json serializado e as anotações para o padrão BILOU

# dados de treinamento
with io.open(data+'/train-origin.json', encoding='utf8') as f_train:
    data_train = json.load(f_train)

result_train=[]
for i, record in enumerate(data_train):
    doc = gera_doc(nlp, record)
    result_train.append(spacy.gold.docs_to_json(doc, id=i))

with open(data+'/train.json', 'w') as fh:
    fh.write(json.dumps(result_train))

# dados de validação
with io.open(data+'/dev-origin.json', encoding='utf8') as f_dev:
    data_dev = json.load(f_dev)

result_dev=[]
for i, record in enumerate(data_dev):
    doc = gera_doc(nlp, record)
    result_dev.append(spacy.gold.docs_to_json(doc, id=i))

with open(data+'/dev.json', 'w') as fh:
    fh.write(json.dumps(result_dev))

# dados de teste
with io.open(data+'/test-origin.json', encoding='utf8') as f_test:
    data_test = json.load(f_test)

result_test=[]
for i, record in enumerate(data_test):
    doc = gera_doc(nlp, record)
    result_test.append(spacy.gold.docs_to_json(doc, id=i))

with open(data+'/test.json', 'w') as fh:
    fh.write(json.dumps(result_test))

# executa rotina de verificação dos dados de treinamento e validação
run("python3.6 -m spacy debug-data pt "+data+"/train.json " +data+"/dev.json -b "+initmod
el+" -p ner -V")

print ("Tempo de processamento: ", int((time.time()-eta)/60),"min")
```

Código-fonte para inicialização do modelo

```
import spacy
import subprocess
import json
import jsonlines
import io
import time

eta = time.time()

def gera_doc(nlp, record):
    patterns = [{'label': label, 'pattern': pattern} for pattern, label in record['patter
ns']]
    er = spacy.pipeline.EntityRuler(nlp, patterns=patterns)
    doc = er(nlp(record['text']))
    doc.is_parsed = True
    return doc

def run(command):
    print (command)
    result = subprocess.run(command, shell=True, stdout=subprocess.PIPE, stderr=subproces
s.PIPE, universal_newlines=True)
    print(result.stdout)
    if result.stderr is not None: print(result.stderr)

# define variáveis e entidades
initmodel="../model/initial"
data="../data"
entities = ["NORMA","ACÓRDÃO","DECISÃO","PARECER","ÓRGÃO_ENTIDADE","PROGRAMA","LOCALIDADE
","PRAZO","VALOR"]

# inicializa o modelo
run("rm -r "+initmodel)
run("python3.6 -m spacy init-model pt "+initmodel+" --vectors-loc "+data+"/word2vec.txt")

# carrega o modelo inicializado
nlp = spacy.load(model)

# cria o componente NER
ner = nlp.create_pipe("ner")
nlp.add_pipe(ner, last=True)

# adiciona as entidades customizadas
[ner.add_label(ent) for ent in entities]

# salva o modelo
nlp.to_disk(model)

print ("Tempo de processamento: ", int((time.time()-eta)/60),"min")
```

Código-fonte para o pré-treinamento do modelo

```
import subprocess
import io
import time

eta = time.time()

def run(command):
    print (command)
    result = subprocess.run(command, shell=True, stdout=subprocess.PIPE, stderr=subproces
s.PIPE, universal_newlines=True)
    print(result.stdout)
    if result.stderr is not None: print(result.stderr)

# define variáveis e parâmetros
data="../data"
initmodel="../model/initial" # word vector e componente NER customizado
premodel="../model/pretrain"
arguments="-i 100"

# executa o pré-treinamento
run("rm -r "+premodel)
run("python3.6 -m spacy pretrain "+data+"/pretrain.jsonl "+initmodel+" "+premodel+" "+arg
uments)

print ("Tempo de processamento: ", int((time.time()-eta)/60),"min")
```

Código-fonte para treinamento do modelo

```
import subprocess
import io
import time

def run(command):
    print (command)
    result = subprocess.run(command, shell=True, stdout=subprocess.PIPE, stderr=subproces
s.PIPE, universal_newlines=True)
    print(result.stdout)
    if result.stderr is not None: print(result.stderr)

eta = time.time()

# define variáveis e parâmetros
data="../data"
initmodel="../model/initial"
premodel="../model/pretrain/model99.bin"
model="../model/train"
hiperparameters=""
arguments="-n 100"

# executa o treinamento
run("rm -r "+model)
run(hiperparameters+" python3.6 -m spacy train pt "+model+" "+data+"/train.json "+data+"/
dev.json -p ner -t2v "+premodel+" "+arguments)

print ("Tempo de processamento: ", int((time.time()-eta)/60),"min")
```

Código-fonte para avaliação do modelo

```
import spacy
from spacy.gold import GoldParse
from spacy.scorer import Scorer
from spacy import displacy
import subprocess
import json
import jsonlines
import io
import time

def gera_doc(id, record):
    patterns = [{'label': label, 'pattern': pattern} for pattern, label in record['patterns']]
    nlp_gold = spacy.blank("pt")
    er = spacy.pipeline.EntityRuler(nlp_gold, patterns=patterns)
    nlp_gold.add_pipe(er)
    doc = nlp_gold(record['text'])
    doc.is_parsed = True
    data_test.append(spacy.gold.docs_to_json(doc, id=id))
    return doc

def run(command):
    print (command)
    result = subprocess.run(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE
, universal_newlines=True)
    print(result.stdout)
    if result.stderr is not None: print(result.stderr)

eta = time.time()

# define variáveis
model="../model/train/model-best"
data="../data/"
html="../html/"
entities=["NORMA","ACÓRDÃO","DECISÃO","PARECER","ÓRGÃO_ENTIDADE","PROGRAMA","LOCALIDADE","PRAZO
","VALOR"]
colors=["Red","Blue","Yellow","Green","Gray","Brown","Cyan","Gold","Lightgreen"]

colors_dic = {entities[i]: colors[i] for i in range(len(entities))}
options = {"ents": entities, "colors": colors_dic}

# carrega dados de teste
with io.open(data+"test-origin.json", encoding='utf8') as f:
    data_test_origin = json.load(f)

## avaliação via API com uso de displacy

# carrega o modelo treinado
nlp_pred = spacy.load(model)
data_test=[]
scorer = Scorer()

for id, record in enumerate(data_test_origin):
    pred_value = nlp_pred(record['text'])
    print(*[(ent.label_, ent.text) for ent in pred_value.ents], sep="\n")

for id, record in enumerate(data_test_origin):
    doc = gera_doc(id, record)

    # extrai valor original ("gold")
    gold_text = doc
    gold_ents = ([(ent.start_char, ent.end_char, ent.label_) for ent in doc.ents])
    gold = GoldParse(doc=gold_text, entities=gold_ents)

    # extrai valor previsto pelo modelo ("pred")
    pred_value = nlp_pred(record['text'])

    # calcula score
    scorer.score(pred_value, gold, verbose=False)

    # renderiza o resultado
    displacy.render(pred_value, style="ent", options=options)

print(scorer.scores,"\n")

# avaliação via comando
run("python3 -m spacy evaluate "+model+" "+data+"test.json -dp "+html+" -R")

print ("Tempo de processamento: ", int((time.time()-eta)/60),"min")
```