Applying Named Entity Recognition models on Brazilian Legal Documents

Alice da Silva de Lima
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
alice.lima@aluno.unb.br

***Abstract*—Search and information retrieval on legal content documents is an important task for governmental purposes. Once these documents are usually large and unstructured, the task of identifying the main entities of the text can take time when executed by humans. Thus, to automatize the task, Named Entity Recognition, a Natural Language Processing task, can extract these entities from a large number of documents. In this context, the present report presents the use of a legal information documents corpus to evaluate Deep Learning models performance on Named Entity Recognition task. Search and information retrieval on legal content document is an important task for governmental purposes. These documents are usually large and unstructured, thus identifying entities of the text can take time when executed by humans. To automatize the task, Named Entity Recognition, a Natural Language Processing task, can extract these entities from a large number of documents. In this context, the present report presents the use corpus of legal information documents to evaluate Deep Learning models performance on Named Entity Recognition task.**

***Index Terms*—Natural Language Processing, Named Entity Recognition, Portuguese Corpora, Deep Learning**

I. Introduction

Most of Brazilian legal content documents are available on unstructured documents such as PDF files or on text fields at websites. Because of that, legal data are often not organized, which makes search and information retrieval difficult tasks for humans because of the large amount of data [1]. Such tasks are important to increase the government transparency, for instance [2].

To overcome human limitations, the use of Machine Learning techniques is an alternative to organize legal data. Specifically, Natural Language Processing (NLP), a subarea of Machine Learning, consists of making machines interpret human languages. As a part of NLP, Named Entity Recognition (NER) is the activity of extraction and classification of named entities contained in a natural language text using predefined entity categories. Specifically, the task aims to assign to each token of the text a single label that represents it's category [3]. There are countless applications of this task, such as E-Commerce Search Queries [4].

In this context, the use of NER for Legal Documents is convenient because of it's automation, that makes possible extracting main information of acts such as contracts and acquisitions, which is often requested by legal entities for analysis and decision making. The human process of reading the whole document to identify the main information of text can take time, so NER application turns the process faster [5].

The present report aims to use a Brazilian corpus of legal information documents annotated and reviewed by humans to evaluate Deep Learning models performance on Named Entity Recognition task. The models utilized were: LSTM with pre-trained Word2Vec word embedding (Word2Vec-LSTM), CNN-LSTM and CNN-BiLSTM. These models implementations were made utilizing the libraries Keras and Gensim. The metrics used to evaluate performance were F1-score, precision and recall. Moreover, accuracy and loss were also observed at the training process.

The rest of this paper is divided into the following sections: Section 2 presents the related works; Section 3 presents the proposed method; Section 4 addresses the experimental results; and Section 5 concludes this report.

II. Related works

The present section addresses works found in literature focused on Named Entity Recognition that specifically used legal documents.

C. Mota et al. [1] presents an evaluation of three neural network architectures for recognizing entities named in initial requests. Due to the lack of works that address the detection of legal entities from texts written in Portuguese using models based on neural networks available in NLP libraries, the work aims to investigate the performance of three implementations considering precision, recall and F1-score. Through the extraction of initial petitions from Brazilian justice processes, a database was built, which was used to analyze the performance of the models along with the LeNER-Br database [6]. The models evaluated were BiLSTM-CRF model from Flair[1] framework, *Convolutional Neural Network (CNN)* from the Spacy[2] library and the LSTM-CR model proposed in [7]. As a result, the Flair framework model showed superior results in precision and F1-score, the best result for recall was the LSTM-CRF model, however the results of the two models did not differ much. The Spacy library model showed the lowest results.

Luz de Araujo et al. [6] presents LeNER-Br, a dataset for named entity recognition which is composed entirely of legal

[1]https://github.com/flairNLP/flair
[2]https://spacy.io/

documents written in Portuguese. In addition, a performance evaluation using the LSTM-CRF model were made on the Paramopama dataset [8] and then the model was retrained on the LeNER-Br dataset. There aren't many manually legal corpora in Portuguese and there are some structure issues with these kind of documents, thus the authors proposed a legal document dataset for Name Entity Recognition manually annotated. The new dataset is composed by legal documents from Brazilian Courts and also by legislation documents; documents were pre-processed and then the WebAnno[3] tool was used to manually annotation. To evaluate the performance in both models, F1-Score was used. LeNER-Br achieved better performance in legal cases and legislation recognition than Paramopama. Moreover they both achieved similar results to person, time and organization entities, with scores above 80%.

III. Proposed Method

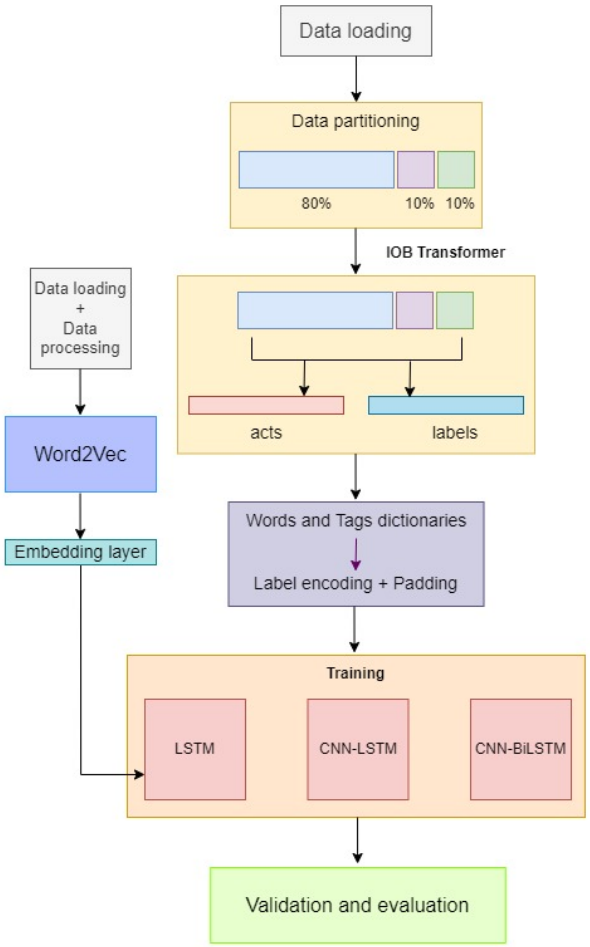

Fig. 1. Methodology workflow.

The figure above illustrates all the methodology steps. This section details the methodology adopted. Google Colaboratory was used to develop the experiments.

[3]https://webanno.github.io/webanno/

A. Corpus, data loading and data processing

The corpus is composed by legal information documents, also known as acts, extracted from *Diário Oficial do Distrito Federal*[4] (Official Gazette of the Federal District). There are six classes of acts: Extrato de Contrato (Contract Extract), Aditamento Contratual (Amendment), Aviso de Licitação (Acquisition Process Notice), Aviso de Revogação/Anulação de Licitação (Bid Revocation Notice), Aviso de Suspensão de Licitação (Suspension of Bidding Notice) e Extrato de Convênio (Covenant Extract). Moreover, the corpus was annotated and reviewed by students from Universidade de Brasília.

Table 1 shows the amount of each class of act, as well as the total amount of documents:

TABLE I
Number of Labeled Acts

Class	Quantity
Aviso de Licitação	638
Aviso de Revogação/Anulação de Licitação	46
Aviso de Suspensão de Licitação	68
Extrato de Aditamento Contratual	1537
Extrato de Contrato	1537
Extrato de Convênio	24
Total	3850

Data partitioning

To split data into training, testing and validation first it was necessary to split by classes because the number of documents per class was not balanced, as Table 1 indicates. Thus, to each class the acts were divided in the following manner: 80% for training, 10% for validation and 10% for test. After that, the lists of training for each class were concatenated, the same goes for test and validation lists.

IOB tags

In this experiment the IOB (Inside, Outside, Beginning) tagging format was adopted. As the name suggests, the "**B**" prefix indicates that the tag is the beginning of a entity, "**I**" prefix indicates it is inside a entity and "**O**" is for tokens that don't belong to a entity [9].

Once the original Dataset didn't have the IOB tags, it was necessary to iterate over the acts and tag each token of the text. For this step, an IOB Transformer Algorithm[5] was used.

Words and Tags dictionaries

Next step was creating dictionaries for both vocabulary and tags. These dictionaries were necessary to convert words from the corpus and labels (IOB tags) into numeric values. Also, the size of the dictionaries were used as some arguments to build the models.

[4]https://www.dodf.df.gov.br/

[5]https://github.com/mstauffer/tcdf_text_classification/blob/main/iob_transformer.py

Padding

After transforming data to numerical values, the next step was to make inputs with the same size with padding. To evaluate the best value, an analysis of the acts size was made. Figure 2 shows the results.

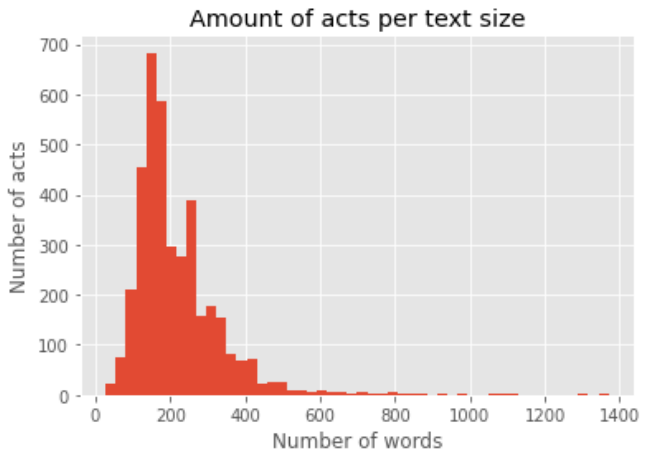

Fig. 2. Amount of acts per text size

The padding size was set to 450 because the size of most acts are below this value. With all inputs and outputs padded, the models were trained.

B. Models

With data prepared, the models were built and trained. All models were made using the resources from Keras library[6] (Layers, Optimizers and Callbacks). To evaluate the models *seqeval*[7] framework was used.

The first model has a pre-trained word embedding built with Word2vec method. To implement the technique, library Gensim[8] was utilized.

All models have a LSTM (Long Short Term Memory) layer, which is a kind of Recurrent Neural Network able to learn long-term dependencies, in other words, it's main characteristic is remembering information for long periods. The architecture allows to remove and add information to a cell state and it is regulated by the gates. There are three gates: Forget gate, Input gate and Output gate. In addition, sigmoid function is utilized in this neural network in order to filter values to be remembered. Long Short Term Memory networks were introduced by Hochreiter and Schmidhuber in 1997 [10].

Word2Vec-LSTM

The first model has a pre-trained word embedding. Word embeddings consists in transform words into numerical representation where each word is mapped to one vector. The vectors are learned similarly to a neural network. The vector representation try to capture characteristics of the word based on the full text. Word2Vec, a word embedding technique, can strongly estimate words' meanings based on their occurrences in the text [11]. Another corpus of official documents was utilized, the Dataset has personnel related acts and it's structure is similar to to main corpus used in this experiment. The model was build with Word2Vec module from Gensim.

An embedding layer was built utilizing the embedding matrix from the model. Next step was to create a sequential model with Keras, adding LSTM layer and others shown at Figure 3. We can observe in output shape the value 76, which is the number of existing tags (labels).

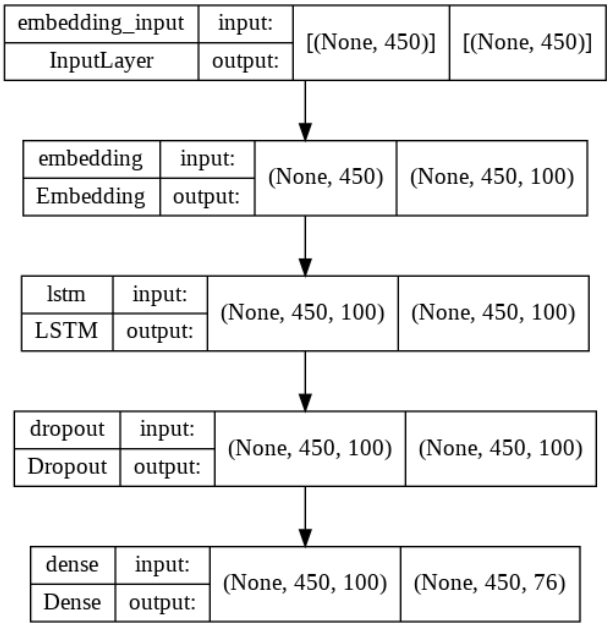

Fig. 3. Word2Vec-LSTM Architecture

CNN-LSTM

The second model consists in a combination of Convolutional Neural Network and a Long Short Term Memory neural network. This algorithm is usually used in tasks with images, such as Image Analysis/Classification and Media Recreation. But it can also be implemented to NLP tasks. In the CNN model a feature matrix passes through convolution layers with Kernels (filters), activation function layer (usually RELU function), pooling layers, fully connected layers and then a activation function is applied to classify an object [13]. Text can be represented as list of lists (matrix), so it is possible to use a Conv1D layer[9] to implement the CNN-LSTM model.

CNN-BiLSTM

The third and last model architecture is similar to the second one. They differ at the LSTM layer. A Bidirectional LSTM layer was implemented instead. That network allows input to flow in both directions and information can be utilized from both sides. Figure 5 illustrates its architecture.

[6]https://keras.io/
[7]https://github.com/chakki-works/seqeval
[8]https://radimrehurek.com/gensim/
[9]https://keras.io/api/layers/convolution_layers/convolution1d/

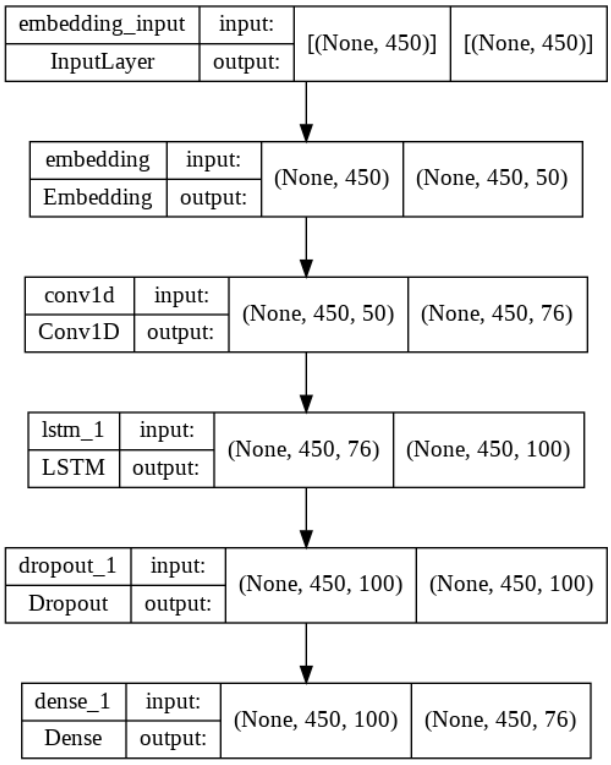

Fig. 4. CNN-LSTM Architecture

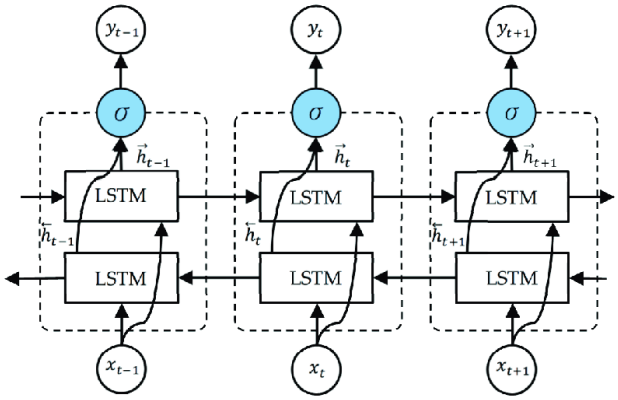

Fig. 5. Architecture of Bidirectional LSTM with three consecutive steps [12]

IV. Experimental Results

TABLE II
F1-Scores for each model

Model	F1-Score
Word2Vec-LSTM	0.60
CNN-LSTM	0.56
CNN-BiLSTM	0.66

The evaluating indicates similar F1-Scores, where CNN-BiLSTM has the higher score and CNN-LSTM the lowest one. In general, it can be observed that the scores surpassed 0.5 (the metric ranges from 0 to 1) and although they are above

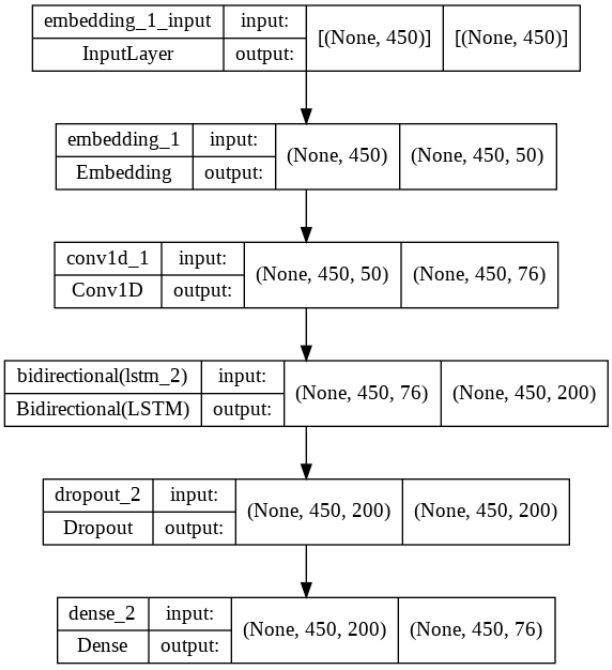

Fig. 6. CNN-BiLSTM Architecture

average, none of them reach 0.8. There are some limitations in the present experiment: the corpus is not large and the lack of some classes of acts affect the models' learning. For future work, it's expected to increase the corpus, as well as use cross validation techniques to compare with the Holdout method presented.

Once there are 76 labels, the evaluation of each one is too large and couldn't be included in the present report. The full classification report is available on the github repository. The script also presents graphics with accuracy and loss of training and validation data.

V. Conclusion

This report presented experiments utilizing legal data corpora to Named Entity Recognition tasks with Deep Learning models. After experiments, the models were validated and evaluated. Although the similar results, CNN-BiLSTM model had the highest performance, while CNN-LSTM had the lowest.

The amount of data affected the experiment, so for future work a larger corpus will be applied, as well as cross validation method. Other LSTM-based models can also be implemented in order to compare their performances.

References

[1] C. Mota, A. Nascimento, P. Miranda, R. Mello, I. Maldonado, and J. Coelho Filho. "Reconhecimento de entidades nomeadas em documentos jurídicos em português utilizando redes neurais", in Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional, Evento Online, 2021, pp. 130-140, doi: https://doi.org/10.5753/eniac.2021.18247.
[2] Knedle. URL http://nido.unb.br/.
[3] Li, Jing, et al. "A survey on deep learning for named entity recognition." IEEE Transactions on Knowledge and Data Engineering 34.1 (2020): 50-70.

[4] Wen, Musen, et al. ”Building large-scale deep learning system for entity recognition in e-commerce search.” Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies. 2019.

[5] Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, Serena Villata. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker . ICAIL-2017 - 16th International Conference on Artificial Intelligence and Law, Jun 2017, Londres, United Kingdom. pp.22. ffhal-01541446f

[6] Luz de Araujo, Pedro Henrique, et al. ”LeNER-Br: a dataset for named entity recognition in Brazilian legal text.” International Conference on Computational Processing of the Portuguese Language. Springer, Cham, 2018.

[7] Lample, Guillaume, et al. ”Neural architectures for named entity recognition.” arXiv preprint arXiv:1603.01360 (2016).

[8] Júnior, C. Mendonça, et al. ”Paramopama: a Brazilian-Portuguese corpus for named entity recognition.” Encontro Nac. de Int. Artificial e Computacional (2015).

[9] Kuriakose, Jeril. ”BIO / IOB Tagged Text to Original Text”. Available in: https://medium.com/analytics-vidhya/bio-tagged-text-to-original-text-99b05da6664

[10] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[11] Vatsal. ”Word2Vec Explained”. Available in: https://towardsdatascience.com/word2vec-explained-49c52b4ccb71

[12] Li, Yunghui & Harfiya, Latifa Nabila & Purwandari, Kartika & Lin, Yue-Der. (2020). Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model. Sensors. 20. 10.3390/s20195606.

[13] Newatia, Rajat. ”How to implement CNN for NLP tasks like Sentence Classification”. Available in: https://medium.com/saarthi-ai/sentence-classification-using-convolutional-neural-networks-ddad72c7048c