Natural Language Processing for Identification of Tax-Related Doubts

João Victor M. de Macedo
Dept. of System and Computing Engineering
State University of Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
joaovicmonteiro.m@gmail.com

Leonardo Andrade
Getúlio Vargas Foundation (FGV)
Secretary for Finance of the State of Rio de Janeiro (SEFAZ-RJ)
Rio de Janeiro, Brazil
leonardo.andrade@sefaz-rj.gov.br

Karla Figueiredo
Department of Informatics and Computer Science
Institute of Mathematics and Statistics
State University of Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
karlafigueiredo@ime.uerj.br

Abstract—This work aimed to investigate Natural Language Processing (NLP) algorithms to automate the "Fale Conosco" (Contact Us) channel of SEFAZ-RJ, used to clarify taxpayers' doubts sent via email. Due to the social distancing situation caused by the COVID-19 pandemic, the channel has become a consolidated means to address tax-related inquiries. Thus, employing Machine Learning/Deep Learning techniques, taxpayers' doubts were classified with the objective of automating the response process. The results with the BERT-based model achieved an accuracy of 96.6%, contributing to a proposal for reformulating the taxpayers' inquiry form, as well as indicating more promising techniques to initiate the automation process of the "Fale Conosco" channel at SEFAZ-RJ.

Keywords—LSTM, BERT, Machine Learning, Deep Learning, Tax Law, Natural Language Processing

I. Introduction

The organizational model of the Brazilian National Tax System is characterized by the autonomy of the Union, States, and Municipalities in the elaboration of their respective tax laws. However, such decentralization can result in conflicts between federal entities regarding tax legislation, leading to legal disadvantages for businesses. In this regard, the literature indicates that structuring the Brazilian tax system in such a way imposes a high cost on companies to maintain tax compliance [1].

Considering this reality, the State Department of Finance and Planning of Rio de Janeiro (SEFAZ-RJ) provides a taxpayer assistance service through an electronic messaging channel aimed at clarifying doubts related to state tax legislation. However, this method has some limitations, such as: i) the complexity of interpreting the legislation, which can increase the workload for the legal and technical staff of SEFAZ; ii) the high cost of time and human resources to perform this task manually; iii) the repetitiveness of questions, where many taxpayers have doubts about the same context, which can result in inaccurate answers due to different interpretations by auditors; and iv) with the COVID-19 pandemic and the need for social distancing, the number of messages received has significantly increased, causing delays and financial losses for the state, particularly regarding the fiscal recovery of the State of Rio de Janeiro. In general, the manual response time for questions was up to two business days. However, due to this increased volume of messages received, it was necessary to extend these deadlines. On the other hand, the increased usage has solidified the communication channel between taxpayers and SEFAZ-RJ.

Thus, due to the large volume of processes and often the inefficiency in their progress, the legal field in all its branches has shown great interest in Natural Language Processing studies, aiming to simplify certain procedures, such as the classification of Supreme Court legal documents [2], [3], [4], and [5].

Tax law is considered one of the most complex areas because it involves textual and numerical information, such as rates, percentages, and values. Unfortunately, natural language processing is rarely used in this field, except for some cases, such as the study by Ash and Marian (2019) [6], which only explored linguistic similarity between pairs of treaties in force during a specific year, in an empirical manner.

Consequently, continuing the process of automating responses to taxpayers' inquiries in Portuguese, solutions were investigated using Long-Short Term Memory (LSTM) networks [7] and the BERT architecture [15], based on Transformers, aiming to embed term vectorization in the learning process for the classification of tax-related doubts. This choice was motivated by the small available dataset and the results obtained in previous work. This study aims to investigate methodologies for natural language processing of texts in Brazilian Portuguese in the context of tax law, using different solutions based on recurrent neural network (RNN) architectures and word embedding models.

The remainder of the article is organized into five additional sections: Section II briefly introduces the technical foundations necessary to understand better the methods and models developed in this work. The methodology used to solve the proposed problem, as well as its applicability to the problem domain, is described in Section III. Case studies are presented in Section IV. Section V presents the obtained results and their discussions, and the last section presents the conclusions and perspectives for future work.

II. Theoretical Framework

A. Long Short-Term Memory [LSTM]

The Long-Short Term Memory (LSTM) algorithm [7] is a type of recurrent artificial neural network architecture used in natural language processing (NLP) to address the issue of long-term dependencies in conventional recurrent neural networks (RNNs).

This algorithm is composed of cells, with four internal components that interact with information in a differentiated manner. Due to the temporal nature of word distribution in a sentence or text, Recurrent Neural Networks (RNNs) have been widely applied in natural language processing (NLP) [8].

LSTMs feature an additional state known as the cell state, allowing the model to carefully remove or add information, regulated by structures called gates, such as the input gate, forget gate, and output gate. These gates control whether information should be retained or discarded during the processing of the neural network.

The first gate, the forget gate, decides which information will be discarded from the cell state. The second or input gate determines which information will be added to the cell state. This process occurs in two steps: the first identifies the values to be updated, and the second generates a list of candidate values to be included. The outputs of these two steps are then combined, and the cell state is updated. The third gate is the output gate, which determines which information will be presented at the current time step by the LSTM network [7].

B. Vectorization

During vectorization, a crucial step in data preprocessing, words are converted into numerical representations before being used by machine learning models. In this context, a popular vectorization technique deserves special mention: Word2Vec [9], and consequently, the use of its concept in Keras Embedding [10]

1) Word2Vec: Word2Vec is a widely used algorithm for creating continuous vector representations of words. It is based on neural networks that map words to dense vectors, where semantically similar words are located close to each other. There are two main approaches for implementing Word2Vec: the skip-gram model [11] and the Continuous Bag-of-Words (CBOW) model [12]. The skip-gram model aims to predict context words given a target word, while the CBOW model aims to predict the target word based on its context. By training on a large corpus of unlabeled text (also developed in the project's context), Word2Vec learns to capture semantic and contextual relationships between words, producing word vectors that can be used as input in machine learning models such as LSTM networks.

2) Keras Embedding: Keras Embedding is a neural network layer available in the Keras library [13], which enables learning vector representations of words during model training. This layer maps words to dense vectors of fixed size and updates these vectors based on the task at hand, optimizing the model's performance. Using Keras Embedding in conjunction with other architectures, such as LSTM networks, makes capturing contextual and semantic information of words in text sequences possible. Overall, this approach helps improve the accuracy and performance of text classification models.

Both vectorization methods were adopted in this work within the training context using LSTMs.

C. Bidirectional Encoder Representations from Transformers (BERT)

This is a pre-trained language model based on neural networks, developed by Google AI Language. It utilizes the Transformers architecture [14] to capture contextual information of words and sentences. During the pre-training process, BERT [15] learns to predict masked words and the next sentence in a large corpus of unlabeled text. This allows the model to acquire a general representation of linguistic context, surpassing the limitations of previous models. BERT undergoes fine-tuning on specific NLP tasks, adapting to different applications. It has achieved impressive results in various NLP tasks, improving performance and accuracy.

Transformers [14] are neural models used in natural language processing (NLP). Their architecture is based on attention mechanisms, allowing the model to capture relationships between words. Transformers process the input sequence in parallel, capturing long-range dependencies. They have achieved advanced results in various NLP tasks and are widely applied in automatic translation, text summarization, and sentiment analysis, among others. Their ability to capture complex contextual relationships makes them efficient and effective for practical applications.

D. “Fale Conosco” of SEFAZ-RJ

The data used in this study are derived from the *"Fale Conosco"* (Contact Us*)* system of SEFAZ-RJ, which is a customer service channel for taxpayers in the State of Rio de Janeiro. These data are confidential and cannot be shared. In the system, taxpayers can submit their questions, identifying themselves through email, name, and CPF, which SEFAZ-RJ has removed. Additionally, they are required to fill in information fields, indicating the tax or fee related to the question and the subject from a predefined set of options. However, often the information in these fields does not correspond to the content of the messages, making it impossible to send automated responses based solely on the identification of the tax and subject.

III. Methodology

The main approach consisted of classifying the different topics associated with ICMS (Tax on Circulation of Goods and Services, including interstate and inter-municipal transportation and communication). In this way, a classifier model, as indicated in Figure 1, was built to categorize inquiries into different subjects within the context of ICMS. For this approach, two models were investigated: one based on LSTM and the other on BERT.

The LSTM-based model was evaluated using two vectorization approaches: Word2Vec and Keras Embedding. corpus was created using the SEFAZ-RJ database to achieve vectorization with Word2Vec efficiently.

The databases were split in a ratio of 70% for training, 20% for validation, and 10% for testing.

In the case of BERT, the same data split ratio was followed. The significant difference provided by using the BERT model is that vectorization was not required as in the LSTM-based approach. BERT is capable of learning vectorization while simultaneously learning text classification, resulting in a more robust contextual representation of words. This eliminates the need for a pre-processing step of term vectorization.

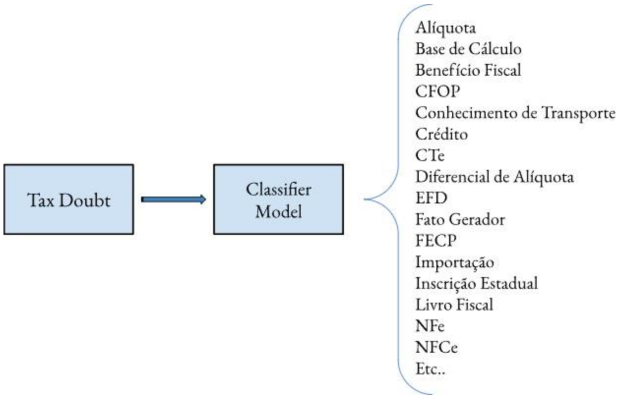


Fig. 1. ICMS Subject Classification Modeling

IV. Case Study

A. SEFAZ-RJ Database

In this stage, work focused on the inquiries submitted by taxpayers, collected through the SEFAZ-RJ's taxpayer support system "*Fale Conosco*" (Contact Us). These inquiries were limited to 2018 due to the constant changes in tax legislation. Another reason to restrict the volume of data was the need for manual validation of the labels (taxes and subjects) provided by the taxpayers during the submission of inquiries, by auditors and analysts.

The database consists of several attributes, including the requested date, subject, tax, attendant, reviewer, protocol, question, and answer. Originally, these records are divided among three taxes and fees, with a total of 56 corresponding subjects.

The total number of documents (i.e., questions) per tax/fee collected in 2018 is as follows: IPVA → 281, ICMS → 10.595, ITD → 170, and Fees → 36. Considering ICMS as the most important tax due to its revenue volume, it also has the highest number of questions, making it the focus of interest in this work. Therefore, due to the small number of questions that could be manually validated by the team of auditors at SEFAZ-RJ, the questions related to the Inheritance and Gift Tax (ITD), Property Tax on Motor Vehicles (IPVA), and State Fees could not be used in this initial stage.

Analyzing the questions per ICMS subject, it was also found that some of them were not feasible to be properly learned. Thus, the model could be developed for a total of 24 subjects. The ICMS subjects are enumerated below, along with their respective quantities indicated in Figure 2: Alíquota, Base de Cálculo, Benefício Fiscal, CFOP, Conhecimento de Transporte, Crédito, CTe, Diferencial de Alíquota, EFD, Fato Gerador, FECP, Importação, Inscrição Estadual, Livro Fiscal, NFe, NFCe, Nota Fiscal, Parcelamento, Pagamento, Restituição, Saldo Credor, Simples Nacional, SPED, Substituição Tributária.

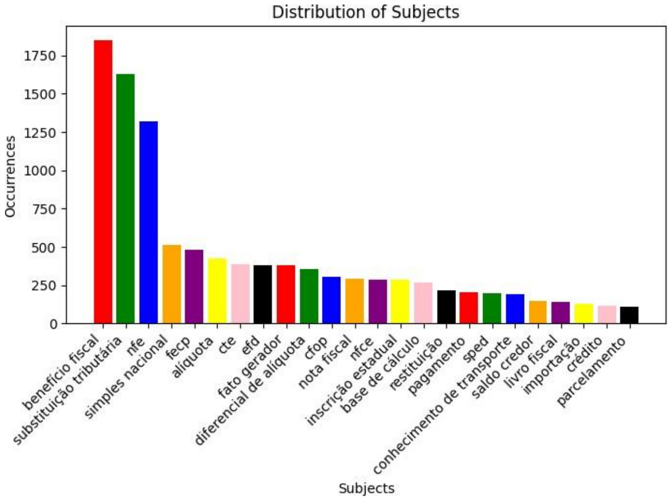


Fig. 2. Distribution of ICMS subjects

B. Data Pre-Processing

The data processing steps were carried out as follows:

Firstly, a *corpus* was created to create *word embeddings*. The ICMS-related text underwent several data preprocessing steps, including converting to lowercase (case-folding), removing accents, removing punctuation, and finally, removing *stopwords* (using the Portuguese *stopwords* provided by the NLTK library [16]).

Next, the Word2Vec algorithm was used to generate *embeddings* of N dimensions with different window sizes, which will be discussed later.

It is worth noting that this preprocessing step was only necessary for the LSTM model and not required for the BERT model.

After creating the *corpus* and *embeddings*, the dataset needed to be treated for training purposes. It was necessary to balance the ICMS dataset, using the "Parcelamento" topic as a reference, which had only 108 occurrences. Therefore, an *under-sampling*, process was performed, randomly selecting 108 questions for each of the 23 topics, in addition to "Parcelamento".

The same treatment used for the corpus was applied with the dataset adequately balanced. Thus, the dataset was divided into 70% for training, 20% for validation, and 10% for testing. This resulted in a training set with 1825 samples, a validation set with 529 samples, and a test set with 265 samples, with the 24 classes stratified.

As the text was written by users who expressed their doubts following the standard language or without spelling mistakes, relying on a Brazilian Portuguese spell checker was unnecessary.

V. Results

This section presents the results of searching for the best modeling approaches for natural language processing solutions (LSTM and BERT) to identify the optimal architecture. The motivation behind this search lies in the main challenge of the problem: a small dataset with many classes often containing similar subjects.

The initial approach of the project involved implementing multiclass classification using recurrent neural networks (LSTMs). To achieve this, the dataset needed to be tokenized, one question at a time, followed by *padding* methodology to ensure that all input data sequences had the same length.

The modeling investigation included determining the appropriate number of LSTM units, defining the dimension of the *embedding*, and evaluating which type of *embedding* was more effective, whether it was Word2Vec (with a window size of 7, after preliminary evaluations) or *Keras Embedding*.

After 330 different exhaustive combinations of values, adjusting the LSTM units from 50 to 225 in increments of 25 units, and varying the *embedding* dimension from 10 to 325 (using both Word2Vec and *Keras Embedding* for the same dimensions), a result was obtained indicating that *Keras Embedding* outperformed Word2Vec. This is evidenced by Table 1, where the choice of *Keras Embedding* ("keras_emb") is among the top eight models with the highest validation accuracy.

The LSTM model was built entirely using the Keras library [13], parameterized with "categorical_crossentropy" as the *loss function*, "adam" as the *optimization algorithm*, *early stopping* using validation loss with the patience of 6, a batch size of 64, "tanh" as the *activation function*, a *dropout rate* of 0.2, and "softmax" as the *activation function* for the output layer. All models were evaluated with the option to train for up to 250 epochs.

Figures 3 and 4 depict the performance of the best model identified in Table 1, using 100 cells in the LSTM model and an *embedding* vector with a dimension of 175.

Table 1. Greater Accuracy (Validation) LSTM

LSTM Units	Model	Accuracy (%)	Loss
50	keras_emb_d300	77,08	1,01
75	keras_emb _d275	84,01	0,81
150	keras_emb _d250	84,47	0,83
125	keras_emb _d250	84,47	0,89
200	keras_emb _d250	84,84	0,81
225	keras_emb _d275	85,03	0,84
175	keras_emb _d325	85,22	0,85
100	keras_emb _d175	85,80	0,82

*keras_emb_dN: N → Dimension of the embedding

Based on the confusion matrix of the LSTM model for the test dataset (Figure 5), it is evident that the results were good, achieving an accuracy of 80.7%. However, the model demonstrated difficulty in handling the subjects related to “Notas Fiscais”. This can be attributed to the existence of different types of invoices, such as “Nota Fiscal”, “NFC-e”, and “NF-e”. Therefore, it is understandable that the model have confused distinguishing between these aspects.

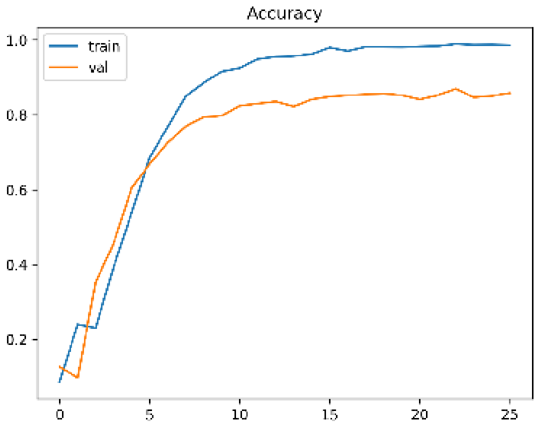


Fig. 3 – Accuracy graphs of training and validation of the best LSTM model with an embedding vector with dimension 175 (last line of Table 1)

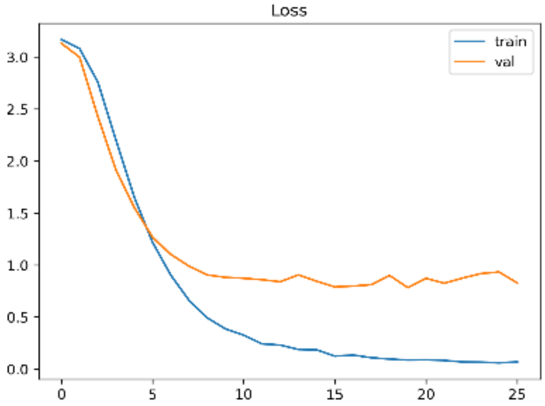
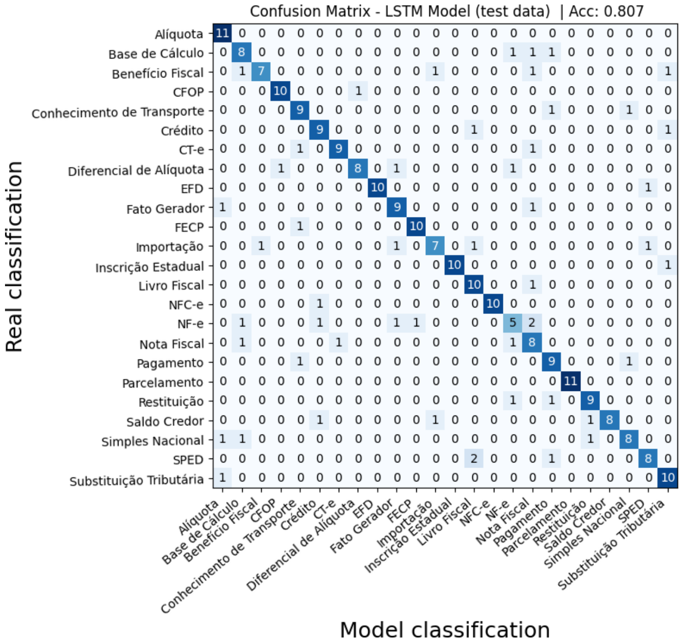


Fig.4. Graphs with training and validation error based con categorical cross-entropy of the best model with an embedding vector with dimension 175 (last row of table 1)

The second approach involved using the BERT architecture [15]. Due to the size of this architecture and the limited amount of data available for training the model, a pre-trained model specific to the Portuguese language context, developed by *Neuralmind* [17], was adopted. Different strategies and preprocessing steps were required when employing BERT compared to LSTMs. Unlike the LSTM approach, BERT does not require similar preprocessing; instead, tokenization needs to be performed according to the model's criteria. In this project, the "bert-base-portuguese-cased" tokenizer from Neuralmind was used. This tokenizer does not perform the same tokenization as the LSTM approach; it returns three tensors:

1. *Input IDs* - represent the sequence of input tokens converted into numerical IDs, where each number represents a specific token in the BERT vocabulary.
2. *Token Type IDs* - indicate the identification of the token type (typically used to distinguish different input sequences in specific tasks, such as questions and answers).

3. *Attention Masks* –is an attention mask that indicates which elements of the sequence are actual tokens (1) and which are padding or filler tokens (0).



	Alíquota	Base de Cálculo	Benefício Fiscal	CFOP	Conhecimento de Transporte	Crédito	CT-e	Diferencial de Alíquota	EFD	Fato Gerador	FECP	Importação	Inscrição Estadual	Livro Fiscal	NFC-e	NF-e	Nota Fiscal	Pagamento	Parcelamento	Restituição	Saldo Credor	Simples Nacional	SPED	Substituição Tributária
Alíquota	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Base de Cálculo	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
Benefício Fiscal	0	1	7	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1
CFOP	0	0	0	10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Conhecimento de Transporte	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
Crédito	0	0	0	0	0	9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
CT-e	0	0	0	0	1	0	9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Diferencial de Alíquota	0	0	0	1	0	0	0	8	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
EFD	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Fato Gerador	1	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	1	0	0	0	0	0	0	0
FECP	0	0	0	0	1	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
Importação	0	0	1	0	0	0	0	0	0	1	0	7	0	1	0	0	0	0	0	0	0	0	1	0
Inscrição Estadual	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	1
Livro Fiscal	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	1	0	0	0	0	0	0	0
NFC-e	0	0	0	0	0	1	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
NF-e	0	1	0	0	0	1	0	0	0	1	1	0	0	0	0	5	2	0	0	0	0	0	0	0
Nota Fiscal	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	8	0	0	0	0	0	0	0
Pagamento	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	1	0	0
Parcelamento	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0
Restituição	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	9	0	0	0	0
Saldo Credor	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	8	0	0	0
Simples Nacional	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8	0	0
SPED	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	8	0
Substituição Tributária	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10

Fig. 5. Confusion Matrix - LSTM Model (test data)

The model construction relied entirely on the PyTorch [18] and Transformers [14] libraries. The other hyperparameters for training were set as follows: *loss function* = "categorical_crossentropy", *optimization algorithm* = "adam", *learning rate* = 10^{-6}, *early stopping* based on validation loss, patience = 6, *batch size* = 8, *activation function* = relu (Rectified Linear Unit), and *dropout* = 0.2. The models were initially programmed to adjust the weights for 35 epochs.

During the training process of the BERT model, variations were made in the hyperparameters, specifically in the Learning Rate, with values of 10^{-6}, 10^{-5} and 10^{-4}, as well as in the number of epochs, with values of 25, 30, 35, 40, and 45. After analyzing the results obtained, it was found that the best combination of hyperparameters is the one described above, with a Learning Rate of 10^{-6} and 35 epochs, remaining faithful to the initially programmed configuration. It is important to note that all the experimented variations showed similar results among them.

Table 2 presents the loss and accuracy obtained with the validation set, and Figures 6 and 7 show the training and validation accuracy and loss for the BERT model, respectively.

TABLE 2. GREATER ACCURACY (VALIDATION) BERT

BERT base	Accuracy	Loss
portuguese-cased	95,7%	0,16

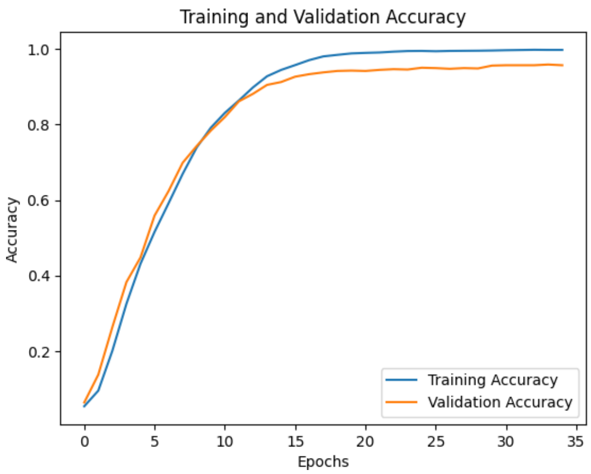


Fig. 6 - Accuracy graphs for training and validation – BERT

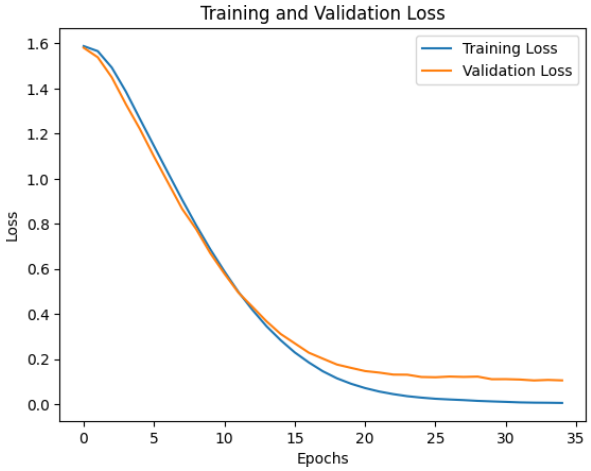


Fig. 7. Graphs with training and validation error based on categorical cross-entropy with BERT with "portuguese-cased" tokenizer"

Based on the confusion matrix results of the BERT model (Figure 8), it was observed that the obtained results were superior to those of the LSTM model, achieving an accuracy of 96.6%.

In contrast, the BERT model demonstrated better capability in handling topics related to invoices: Notas Fiscais, NFC-e, and NF-e. However, it is important to note that there are still cases where the model struggles to classify accurately, such as Conhecimento de Transporte. This is due to the specific nuances of Conhecimento de Transporte and other topics where the model does not perform as well in classification, as it fails to capture the specific intricacies adequately.

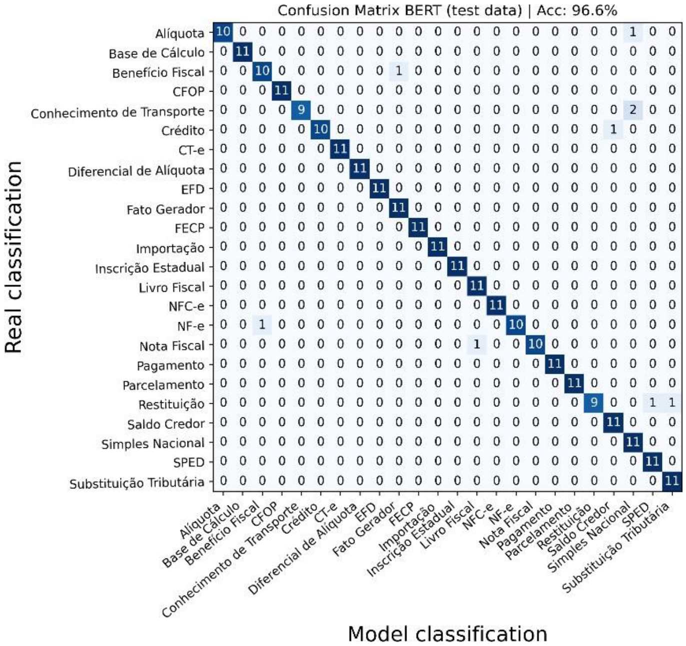


Fig. 8. Confusion Matrix - BERT Model (test data)

VI. Conclusion

This study investigated two classification approaches for Portuguese texts that contain tax-related inquiries regarding ICMS in the State of Rio de Janeiro, collected explicitly through the "Fale Conosco" channel of SEFAZ-RJ.

Due to the low number of inquiries in some subjects, it was necessary to reduce the number of classes to be classified, and even so, the remaining classes had 108 samples per class. Thus, there was doubt about the ability of a model with a larger architecture (such as BERT) to discriminate among 24 classes.

It was found that the LSTM model, using *Keras embedding*, achieved good accuracy but encountered some difficulties in classifying certain subjects. These challenges can be attributed to two factors: the low number of inquiries per class, which hinders the learning of models that inherently require a considerable amount of information, a requirement for deep machine learning; and the high semantic similarity among some subjects, which results in a decrease in classification accuracy for those cases. It is worth noting that the results obtained from the LSTM model were considered better after creating a corpus based on the context of tax law and associating it with *word embedding*, as the results were worse without these steps.

On the other hand, the BERT model proved to be significantly superior to LSTM, even with a relatively small number of samples per class. This disadvantage seems to have been overcome by using a pre-trained model. It is observed that the model performed well for classes related to Notas Fiscais.

Therefore, an area to be explored in the future is the expansion of the dataset of queries, considering that some topics may need to be added or removed due to adjustments in tax legislation. Additionally, it is important to focus efforts on fine-tuning the BERT model, as it appears to be the most efficient despite requiring more computational power during training. In the following steps, the use of question-answering algorithms based on *Transformers*, such as BERT [15] and its advancements, as well as the *Text-To-Text Transfer Transformer (T5) framework* [19], can be mentioned.

Therefore, despite the difficulties encountered in collecting the data, the results obtained are promising in terms of a future process of automating the clarification of taxpayers' doubts.

References

[1] B. Appy, Por que o sistema tributário brasileiro precisa ser reformado. Interesse Nacional, 8(31), pp.65-81, 2015.

[2] P. Casanovas, M. Palmirani, S. Peroni, T. van Engers, and F. Vitali, Semantic web for the legal domain: the next step. Semantic web, 7(3), pp. 213-227, 2016.

[3] R. Dale, Law and Word Order: NLP in Legal Tech, Published online by Cambridge Univ.Press. 2018, DOI: https://doi.org/10.1017/S1351324918000475

[4] L. Robaldo, S. Villata, A. Wyner, et al. Introduction for artificial intelligence and law: special issue "natural language processing for legal texts". Artificial Intelligence. Law 27, pp. 113–115, 2019.

[5] F. Fagan, Natural Language Processing for Lawyers and Judges, Michigan Law Review, Forthcoming. 2020 Available at SSRN: http://dx.doi.org/10.2139/ssrn.3564966

[6] E. Ash and O. Marian, The Making of International Tax Law: Empirical Evidence from Natural Language Processing. UC Irvine School of Law Research Paper No. 2019-02. Available at SSRN: https://ssrn.com/abstract=3314310

[7] S. Hochreiter and J. Schmidhuber, Long short-term memory. Neural computation, 9(8), pp. 1735-1780, 1997.

[8] D. Jurafsky and J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Prentice Hall PTR, 2000.

[9] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space. Available at: https://doi.org/10.48550/arXiv.1301.3781. Code: https://github.com/dav/word2vec

[10] Keras Embedding: https://keras.io/api/layers/core_layers/embedding/

[11] Sanket Doshi: Skip-Gram: NLP Context words prediction algorithm. Available at: https://towardsdatascience.com/skip-gram-nlp-context-words-prediction-algorithm-5bbf34f84e0c

[12] Bhoomika Madukar: The Continuous Bag of Words (CBOW) Model in NLP. Available at: https://analyticsindiamag.com/the-continuous-bag-of-words-cbow-model-in-nlp-hands-on-implementation-with-codes/

[13] Keras Documentation: https://keras.io/api/

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin : Attetion is All You Need. Available at: https://arxiv.org/abs/1706.03762

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available at: https://arxiv.org/abs/1810.04805

[16] NLTK Documentation: https://www.nltk.org/

[17] Neuralmind: BERTimbau Base (aka "bert-base-portuguese-cased"). Available at: https://huggingface.co/neuralmind/bert-base-portuguese-cased

[18] PyTorch Documentation: https://pytorch.org/docs/stable/index.html

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou and Wei Li, Peter J. Liu: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Available at: https://arxiv.org/abs/1910.10683