An Empirical Comparison of Portuguese and Multilingual BERT Models for Auto-Classification of NCM Codes in International Trade

Roberta Rodrigues de Lima[1]*, Anita Maria da Rocha Fernandes[1] *, James Roberto Bombasar[2], Bruno Alves da Silva[1], Paul Croker [3] and Valderi Reis Quietinho Leithardt [4], [5]*

[1] Laboratory of Applied Intelligence, School of the Sea Science and Technology, Itajaí, 88302-901, Brazil; robertalima@edu.univali.br (R.R.L); anita.fernandes@univali.br (A.M.R.F); silvabruno@edu.univali.br (B.A.S).
[2] Centro Universitário Avantis. Balneário Camboriú 88339-125, Brazil; james.bombasar@uniavan.edu.br (J.R.B)
[2] Departamento de Informática, Universidade da Beira Interior, Instituto de Telecomunicações, Delegação da Covilhã, 6201-601 Covilhã, Portugal; crocker@di.ubi.pt (P.C)
[4] COPELABS, Lusófona University of Humanities and Technologies, Campo Grande 376, 1749-024 Lisboa, Portugal;
[5] VALORIZA, Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal; valderi@ipportalegre.pt (V.R.Q.L)

* anita.fernandes@univali.br

Abstract: The classification of goods involved in international trade in Brazil is based on the Mercosur Common Nomenclature (NCM). The classification of these goods represents a real challenge due to the complexity involved in assigning the correct category codes especially considering the legal and fiscal implications of misclassification. This work focuses on the training of a classifier based on Bidirectional En-coder Representations from Transformers (BERT) for tax classification of goods with NCM codes. In particular this article present results from using a specific Portuguese Language tuned BERT model as well results from using a Multilingual BERT. Experimental results justify the use of these models in the classification process and also that the language specific model has a slightly better performance.

Keywords: NCM classification; Natural language processing; Multilingual BERT; Portuguese BERT; Transformers; NLP; BERT.

1. Introduction

The Mercosur Common Nomenclature (NCM) is a system used by the South American trade bloc Mercosur to categorize goods in international trade and to facilitate customs control according to [16]. The NCM is divided into 96 Chapters which contain more than 10,000 unique NCM codes. An NCM code is an 8-digit numeric code than represents the goods and is required in the process of importing products in Brazil. Although it's a necessity the process of classifying goods can constitute a real challenge due to the complexity involved in assigning the right code to each imported good giving the substantial number of codes and their technical details. During the import process one of the first documents required by Brazil is the Import Declaration in which the NCM code must be assigned to the product. In the case of a missing document or a misclassification of the NCM Code, the fine can achieve up to 75% of product price according to Brazilian Law 9430/1996 Art. 44. This makes classification a key challenge. Since the proposition of the Transformers in [22] the Natural Language Processing (NLP) area was hugely impacted by a model that didn't need the recurrences layers and was based only on attention mechanisms. BERT was proposed two years later by [11] stacking only encoder layers from the

Transformers and achieving state of the arts results in eleven Natural Language Processing tasks in the GLUE Benchmark and thus allowing many NLP tasks to take advantage of this approach. Regarding the international trade data, the Brazilian Revenue Service maintains a system called Siscori that currently contains all data relative to Brazilian imports and exports, including a detailed description of the goods and their respective NCM Code. The focus of this work is to use this international trade data to fine-tune a classifier using the BERT model. The Multilingual BERT model proposed in [11] and the Portuguese BERT (BERTimbau) proposed in [21] will be used an empirical comparison between the performance of the two models will be given.

2. Materials and Methods

2.1 Mercosur Common Nomenclature

The Mercosur Common Nomenclature (NCM) was created with the aim of standardizing the classification of goods and to facilitate customs control among the countries that belong to the Mercosur: Argentina, Brazil, Paraguay, Uruguay, and Venezuela [16]. The NCM is based on the Harmonized System (HS) which is maintained by the World Customs Organization (WCO) [19]. The 8-digit code is composed by the Harmonized System 6-digit code extended by two additional codes as illustrated in Figure 1.

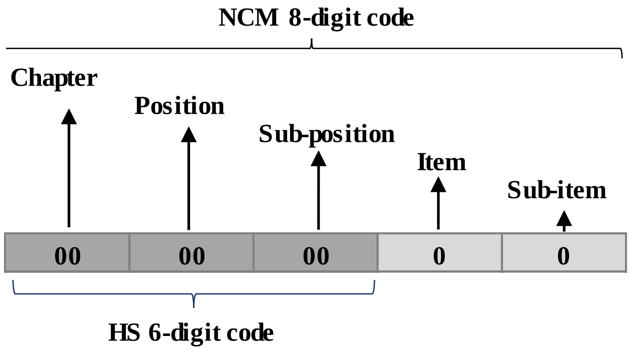


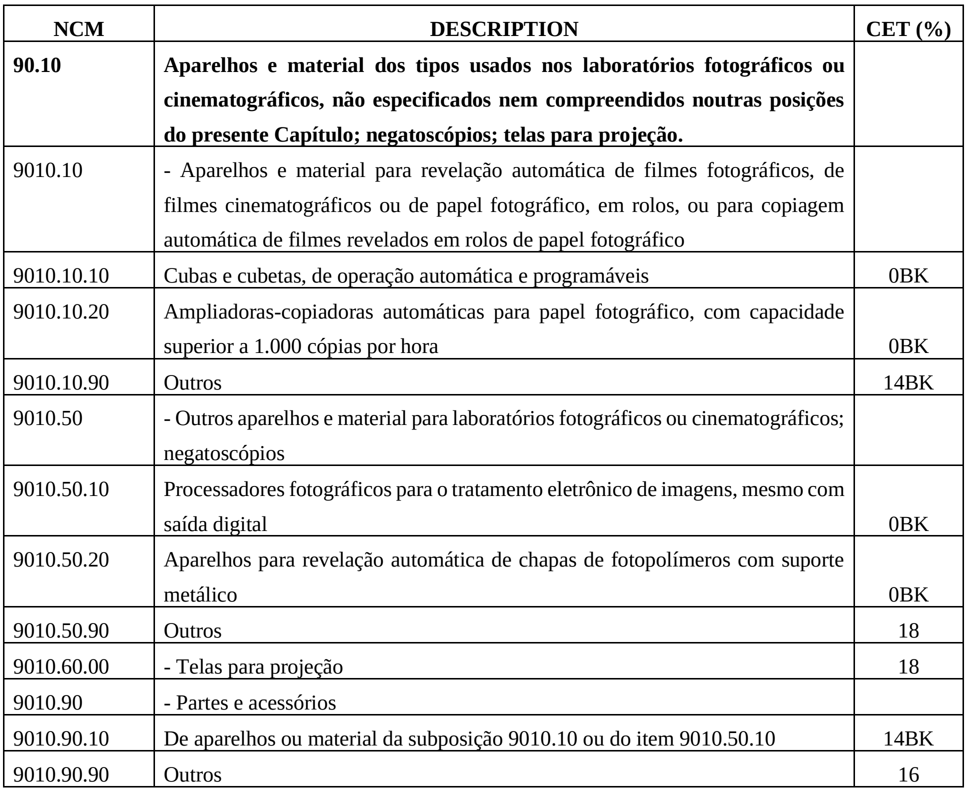
Figure 1: Composition of the NCM Code

The composition of the NCM code starts with the first two digits from the HS code that specify the Chapter followed by two digits that refer to the Position. The next two codes are related to the Sub-position and the last two digits represents the Item and the Sub-item. According to [16] the addition of the last two digits allows for a more detailed specification of the goods based on each countries specific needs.

2.2 Tax Classification of Goods

The main use for the NCM codes in Brazil is for the tax classification of goods. Fees and taxes will be applied in the import process according to the code assigned to the good. According to the Brazilian Revenue Service, the NCM is also used in customs valuation, in statistical data involving import and export data and in import licenses, for special customs regimes such as goods identification, amongst other use cases.

The classification of goods, which is the focus of this work, is the process of assigning an NCM code to the good according to its technical features and characteristics. The NCM codes are maintained in an NCM table provided by the Brazilian Revenue Service which aggregates more than 10,000 codes distributed in 21 sections and 96 chapters. The chapters start with a chapter for live animals and end with a chapter for art works like paintings [19]. An example is shown in Figure 2 which shows chapter 90 which is related to photographic and cineographic products.



NCM	DESCRIPTION	CET (%)
90.10	**Aparelhos e material dos tipos usados nos laboratórios fotográficos ou cinematográficos, não especificados nem compreendidos noutras posições do presente Capítulo; negatoscópios; telas para projeção.**	
9010.10	- Aparelhos e material para revelação automática de filmes fotográficos, de filmes cinematográficos ou de papel fotográfico, em rolos, ou para copiagem automática de filmes revelados em rolos de papel fotográfico	
9010.10.10	Cubas e cubetas, de operação automática e programáveis	0BK
9010.10.20	Ampliadoras-copiadoras automáticas para papel fotográfico, com capacidade superior a 1.000 cópias por hora	0BK
9010.10.90	Outros	14BK
9010.50	- Outros aparelhos e material para laboratórios fotográficos ou cinematográficos; negatoscópios	
9010.50.10	Processadores fotográficos para o tratamento eletrônico de imagens, mesmo com saída digital	0BK
9010.50.20	Aparelhos para revelação automática de chapas de fotopolímeros com suporte metálico	0BK
9010.50.90	Outros	18
9010.60.00	- Telas para projeção	18
9010.90	- Partes e acessórios	
9010.90.10	De aparelhos ou material da subposição 9010.10 ou do item 9010.50.10	14BK
9010.90.90	Outros	16

Figure 2: Excerpt from Chapter 90 in NCM table

[7] states that the main purpose of NCM classification is the collection of the Import Tax, which is based on the Mercosur's Common External Tariff (CET), as well as the establishment of commercial defense rights, such as in the case of anti-dumping. Thus, the correct identification of the NCM code in the Import Declaration (DI) is necessary to guarantee the correct collection of taxes, as well as the surcharges that guarantee commercial defense.

As stated, the NCM code is necessary in the registration of the import license, a mandatory document in the import process in Brazil. According to Brazilian Law 6.759/2009 Art. 706, which regulates the administration of customs activities and the inspection, control, and taxation of foreign trade operations, a 30% fine on the imported good price can be applied if the license is required and not presented during the import process.

An incorrect classification of the goods in the NCM code can also lead to the application of fines. In accordance with Brazilian Law 6.759/2009 Art. 711, a 1% fine is applied on the customs value of the goods if there is an incorrect classification of the NCM code. In addition, Brazilian Law 9,430/1996 Art. 44 states that, in cases of issue of the Official Letter, fines of 75% will be applied on the total or difference of tax or contribution in cases of lack of payment or tax collection, lack of declaration and in cases of inaccurate declaration.

2.3 Data Classification

The process of Data Classification involves assigning categories to each of the objects or entities involved, also known as classes. [1] defines the data classification process mathematically:

For a given matrix D of training of size n x d and classes with values between $1 \ldots k$, associate each of the n rows in D, create a training model M that can be used to predict the class of a dimension record d where the record $\gamma \notin D$. According to [8], classification problems are one of the most common applications in data mining and these tasks are also quite frequent in everyday life.

Classification problems are said to be supervised when the relationship of training data with the class itself is learned [1]. Thus, after the training, a classifier will serve to classify new records in those predefined classes, whose learning took place based on the labelled training data. Finally, the new records that will be provided to the classifier for class determination are called test data and are used to measure the classifier's performance on unknown records.

2.4 BERT

The BERT (Bidirectional Encoder Representations from Transformers) model proposed by [11] is based on a bidirectional multilayer Transformer encoder from the original implementation of Transformers proposed by [22]. BERT works by pre-training deep bidirectional representations from unlabeled data in both context directions. Thus, once pretrained, a fine-tuning procedure can be performed on top of the model by adding a layer thereby allowing its use in a wide variety of tasks [11].

According to [11] the procedure with BERT is composed of two steps: a pre-training step followed by a fine-tuning step. In the pre-training stage, the model is trained in unlabeled databases in two large tasks called Masked Language Model (MLM) and Next Sentence Prediction (NSP). At the end, in the fine-tuning step, the model is first initialized with the pre-training parameters, and then later has its weights updated based on the training using labelled data for the specific task.

The fine-tuning step tends to be much faster and straightforward than the training process: if it's a classification task, for example, there’s simply a need to add a classification layer to the pre-trained model, which will result in all parameters being adjusted for the new task. In this case, these tasks are called downstream, as they present themselves as supervised learning tasks that use a pre-trained model or component. Figure 3 adapted from [2] illustrates the fine-tuning process on a BERT model for the classification of NCM codes from product descriptions, which is the focus of this work.

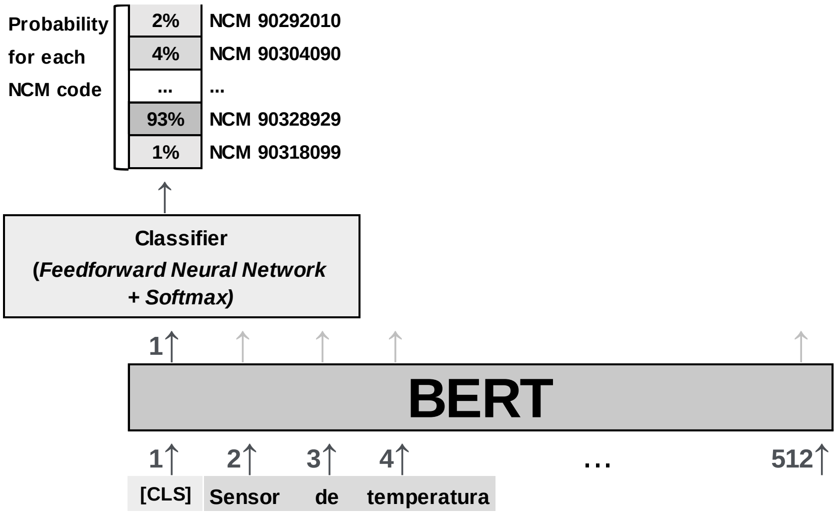


Figure 3: Fine-tuning process

In summary, after the pre-training phase, the model will have the idea of language and context, and after fine tuning it will be provided with the means to solve the specific problem e.g., a problem of classification. This approach is referred in [11] as knowledge transfer and its relevance has already been proved in the computer vision field and has also brought state-of-the-art results in a series of NLP tasks. It has helped in the dissemination of models that use Transformers for a series of applications, as it enables the achievement of results with reduced training time, since it already starts from a pre-trained model as a basis.

2.4.1 Multilingual BERT

Combined with the BERT model originally trained in English, [11] also make available the Multilingual BERT uncased, a pre-trained model in 102 different languages, which can be used in the same way as the single language model. According to the authors, this model exists allows the use of several languages, since it is nor practically feasible to maintain so many isolated language models. However, a disadvantage of the multi-language model is that, according to its documentation, it can underperform single-language models, particularly in feature-rich languages. In that case, language-specific pre-training is indicated for increased performance.

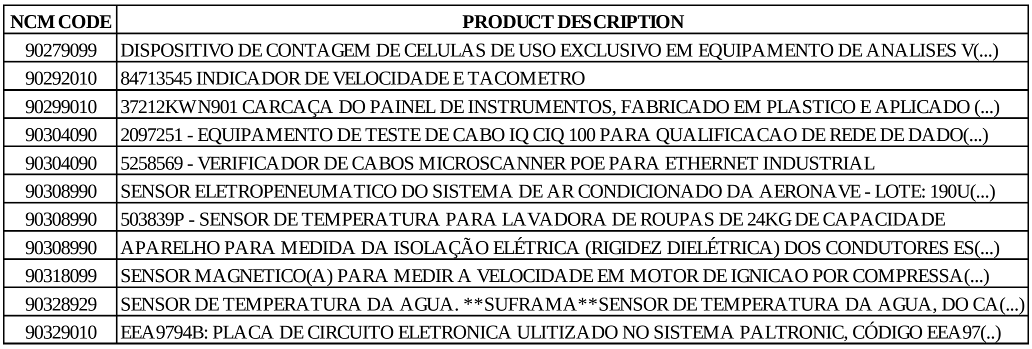
2.4.2 Portuguese BERT

[21] presented a BERT model trained entirely in Brazilian Portuguese, which they nicknamed BERTimbau. According to the authors, the Portuguese model replicates the pre-training procedures with just a few changes and is available in two sizes with the same number of layers and parameters as the original BERT: Base and Large. According to [21], the training was based on the brWaC Corpus, web as corpus in Brazilian Portuguese and the model was evaluated in three distinct downstream tasks: Sentence Textual Similarity, Recognizing Textual Entailment and Named Entity Recognition, outperforming multilingual model in both sizes results.

3. Procedures

3.1 Data Selection

The data used in this work was obtained from "Siscori" which is a website provided by the Brazilian Revenue Service that currently contains all data on the import and export of goods from 2016 to 2021. The data records include a detailed description on the good (with a maximum of 250 characters), country of origin, country of destination, related costs and the NCM code which the good was classified as well as other information that was not relevant to this work.



NCM CODE	PRODUCT DESCRIPTION
90279099	DISPOSITIVO DE CONTAGEM DE CELULAS DE USO EXCLUSIVO EM EQUIPAMENTO DE ANALISES V(...)
90292010	84713545 INDICADOR DE VELOCIDADE E TACOMETRO
90299010	37212KWN901 CARCAÇA DO PAINEL DE INSTRUMENTOS, FABRICADO EM PLASTICO E APLICADO (...)
90304090	2097251 - EQUIPAMENTO DE TESTE DE CABO IQ CIQ 100 PARA QUALIFICACAO DE REDE DE DADO(...)
90304090	5258569 - VERIFICADOR DE CABOS MICROSCANNER POE PARA ETHERNET INDUSTRIAL
90308990	SENSOR ELETROPENEUMATICO DO SISTEMA DE AR CONDICIONADO DA AERONAVE - LOTE: 190U(...)
90308990	503839P - SENSOR DE TEMPERATURA PARA LAVADORA DE ROUPAS DE 24KG DE CAPACIDADE
90308990	APARELHO PARA MEDIDA DA ISOLAÇÃO ELÉTRICA (RIGIDEZ DIELÉTRICA) DOS CONDUTORES ES(...)
90318099	SENSOR MAGNETICO(A) PARA MEDIR A VELOCIDADE EM MOTOR DE IGNICAO POR COMPRESSA(...)
90328929	SENSOR DE TEMPERATURA DA AGUA. **SUFRAMA**SENSOR DE TEMPERATURA DA AGUA, DO CA(...)
90329010	EEA9794B: PLACA DE CIRCUITO ELETRONICA ULITIZADO NO SISTEMA PALTRONIC, CÓDIGO EEA97(..)

Figure 4. Chapter 90 import data sample.

Since this paper makes use of the BERT model to train a classifier that aims to classify descriptions on its respective NCM code, only the good/product description and NCM code columns were selected for this work. Whereas the amount of NCM codes is over 10,000 and it's currently divided into 96 chapters, the focus of the classifier developed in this paper will be to predict NCM codes inside a single chapter.

Regarding the choice of the chapter, [12] present in their work a classifier to the Harmonized System (HS) codes using Background Nets with a focus on Chapters 22 and 90 since both these chapters were more prone to classification errors in daily classification tasks made by international business analysts than any other chapter. [4] present a classifier for NCM codes using Naïve Bayes algorithm and focuses on the chapters 22 and 90 referred by [12] as the most likely to practical errors.

Given that the difference in the amount of NCM codes in Chapter 90 is more than ten times the amount of NCM codes in Chapter 22, this paper will focus only on Chapter 90 since it tends to have more difficulties involved. As mentioned, the Siscori website provides Brazilian import data from 2016 to 2021, and the import data from January 2019 through August 2021 was selected in this experiment, making a total of 7,645,573 records.

3.2 Data Pre-processing

Since the data was obtained from the official imports and exports Siscori website, there was some noise that needed to be removed in a cleaning process before beginning to train the model. At first, as shown in Figure 4, most of the goods description are in upper case and only a few of them come cased as regular sentences, so initially all records were converted to lower case.

Using Regular Expressions, production and manufacturing dates were removed, as well as product batches and their abbreviations and numbers, since these data are not relevant to the distinction between classes and can interfere in the training. Besides that, codes and terms related to billing and part number (PN) were also removed, since they are specific codes for each company. In addition, extra spaces and some special characters present in the sentences that would not contribute to the learning process and these were also removed.

After the data cleaning phase records that had duplicate descriptions, this is data which most likely come from the same importing companies. After the cleaning and removal of duplicate records, the database was reduced from 7,645,573 records to a total of 3,481,090.

Table 1. Data Records

	No. of Records	**No. of Classes**
Original import data.	7,645,573	325
After cleaning and duplicate removal.	3,481,090	325
After undersampling.	265,818	325

Since the processing of this amount of data would still be significant an undersampling technique was carried out with Imblearn's Random Undersampler. Giving that the dataset was unbalanced, the undersampling process for this research kept a ratio of 1:300 samples between the minority and majority classes. The minority class presented only 4 records so the majority class would present 1200 records giving the chosen ratio with the samples being selected randomly. After the undersampling process, the database was reduced from 3,481,090 to 265,818 records keeping the original number of classes of 325. All the pre-processing steps are summarized in Table 1.

3.3 Data Transformation

This paper code was written in Python and the main library used to fine-tune the BERT model was Simple Transformers, by Thilina Rajapakse. For multiclass classification, the format of training and testing data required by the library states that classes must be provided as integers starting from 0 to n. Therefore, this simple transformation of the NCM codes and adaptation of the column names according to the documentation was performed. The NCM codes and their respective indexes were also stored in a dictionary, for later referral after the prediction process.

For this work, the selected Multilingual BERT model proposed by [11] was the uncased version on the base form (12 layers). Regarding Portuguese BERT (BERTimbau), since only the cased version was available it was chosen also on its base form. In this case, it is important to emphasize that for the BERT model to perform tasks called downstream¸ such as classification, the Simple Transformer library provides models with a classification layer already implemented on top of it. Besides that, Simple Transformers allow the model to be supplied with a list of weights to be assigned to each class in the case of unbalanced datasets. As the database contains real Brazilian import data and presents a considerable difference in the amount of some goods in relation to others with respect to NCM codes, it's clearly a highly unbalanced base. To make use of this configuration allowed by Simple Transformers, the weights were calculated inversely proportional to the number of records per class and provided to the model as a parameter, to be used in the loss calculation during training and evaluation, thus minimizing the effects of the unbalanced dataset.

3.4 Hyperparameter Tuning

Having defined the parameters referred to the model itself, in addition to parameters such as seed to allow the replication of results later, some hyperparameters used in the BERT model were defined as well. [11] specify that most hyperparameters used in fine tuning are the same as in pre-training, except for batch size, learning rate and number of training epochs. Along these lines, the authors suggest some values that, based on the experiments, have shown to work well for different types of tasks.

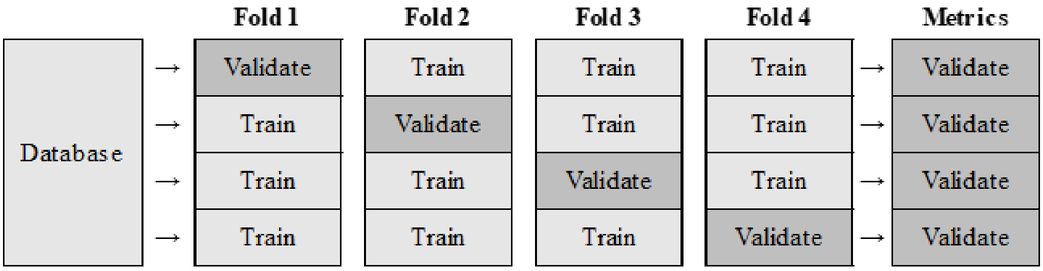
For this, [11] suggests values for batch size as 16 or 32, for learning rate as 5e-5, 3e-5, 2e-5, as well as training epochs of 2, 3 and 4. The authors also reiterate that during their experiments, it was noticed that for large databases (+100,000 labelled data), the model was far less sensitive to the change on these parameters than on smaller databases. Thus, [11] also amend that because the fine-tuning process is relatively fast, it is suggested to carry out an exhaustive search among these parameters to find those that best fit the model. Considering the suggestion, a grid search was performed on the training of the classifier of Chapter 90 comprising all 18 scenarios for each BERT model (Multilingual and Portuguese).

3.5 Hardware and Platform

Since Transformers allow parallelization and thus their use with the proper hardware is very important. For the execution of this work, the code developed in the Python language was executed in a notebook on Google Colab, an interactive environment that is easy to configure and share. For the execution of the experiments used in this research, Google Colab in its Pro version was used due to the availability of fast GPUs and large amounts of RAM. In this case, all 36 scenarios in this experiment were run on Google Colab Pro and a Tesla P100 PCIe 16GB GPU was assigned for the notebook.

3.6 Model Training and Validation

In order to evaluate the classifier, the training and validation processes were performed using k-fold cross-validation. This is illustrated in Figure 5 in a 4-fold Cross-validation which was the number of folds used in this paper's experiments. In [8] defines that in k-fold cross-validation, if the database comprises N instances, these are divided into k equal parts, where k is usually a small number, such as 5 or 10. With the database separated into parts a series of k executions is performed using one of the k parts as a test base and the other (k-1) parts as a training base.



	Fold 1	Fold 2	Fold 3	Fold 4	Metrics
Database	Validate	Train	Train	Train	Validate
	Train	Validate	Train	Train	Validate
	Train	Train	Validate	Train	Validate
	Train	Train	Train	Validate	Validate

Figure 5. 4-fold Cross-validation.

Thus, after k executions, [8] defines that the total number of correctly classified instances is divided by the total number of instances N, resulting in an overall accuracy of the model. One of the variations of cross-validation is the stratified cross-validation, which, according to [1], uses a representation proportional to each class in the different folds and generally provides less pessimistic results. As the database used is quite unbalanced, the stratified cross-validation was run in all experiments with the different hyperparameters, to ensure more consistent results. There are other scenarios and applications as described in [25].

3.7 Metrics Selection

To analyze the performance of the classifier, three different metrics were used, as some are more adequate to the characteristics of the database used in this work: accuracy, cross-entropy loss, and Mattews Correlation Coefficient (MCC).

3.7.1 Accuracy

Accuracy represents the number of instances correctly classified in relation to the total number of instances evaluated. According to [15], although accuracy has been shown to be the simplest and most widespread measure in the scientific literature, it presents some problems in cases in which the performance of unbalanced databases is evaluated. According to the authors, there is an accuracy problem in not being able to distinguish well between different distributions of wrong classifications.

3.7.2 Cross-Entropy Loss

The Cross-Entropy Loss configures a loss function commonly used in classification tasks, being a performance measure of models, whose output involves probability values. In this case, the Cross Entropy Loss increases as the predicted probability diverges from the real one, and therefore the objective is for its value to be as close to zero as possible. To calculate the Cross Entropy Loss in the multiclass case, according to [24] the expression can be written for each instance, where $y(k)$ has the value 0 or 1, indicating whether the class k is the correct classification for the prediction $\hat{y}(k)$, in the form:

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log\, \hat{y}^{(k)}$$

[10] define that the Class-Balanced Loss introduces a weight factor that is inversely proportional to the number of instances of a class and is used precisely to address the problem of unbalanced databases. Thus, as this work deals with this case of unbalance and the Simple Transformers library allows the supply of a vector of weights for the model, these weights are used as multipliers in the Cross Entropy Loss function.

3.7.3 Mattews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) is calculated directly from the confusion matrix, and its values range between -1 and +1. A +1 coefficient represents a perfect prediction, 0 an average prediction, and -1 an inverse prediction. [13] presents its generalized form for the multiclass case, in which C_kl are the elements of the confusion matrix:

$$MCC = \frac{\sum_{klm} C_{kk}C_{lm} - C_{kl}C_{mk}}{\sqrt{\sum_k (\sum_l C_{kl}) \left(\sum_{\substack{l' \\ k' \neq k}} C_{k'l'}\right)} \sqrt{\sum_k (\sum_l C_{lk}) \left(\sum_{\substack{l' \\ k' \neq k}} C_{l'k'}\right)}}$$

[15] refer to the MCC in their work as a performance measure of a multiclass classifier, and point out that, in the most general case, the MCC presents a good harmonization between discrimination, consistency and consistent behaviors with a varied number of classes and unbalanced databases. In addition,[15] show that the behavior of the MCC remains consistent both in cases of binary and multiclass classifiers.

In their work on binary classifications, [9] point out that due to its mathematical properties, the MCC incorporates database unbalance. At this point, [9] reiterate the fact that the MCC criterion is quite direct and intuitive regarding its score: for a high score it is necessary that the classifier correctly predicts most negative classes and most positive classes, regardless of their proportions to the database.

Thus, despite the work of [9] focusing on binary classification, the main differential of the MCC becomes clear: it benefits and penalizes for each class, since a good performance of the classifier will occur when it has a good predictive power for each of the classes together. In this sense, regardless of the number of instances of a class being much lower than another (in the case of unbalanced databases), the MCC maintains its performance evaluation power consistently, while classification errors in classes are not significant in relation to the quantity total instances could go unnoticed in other metrics. In [26] metrics and applications for different scenarios are defined.

4. Experimental Results

Both Multilingual BERT and Portuguese BERT experiments were carried out on a grid search comprising all 18 scenarios that encompasses the combinations of parameters for batch size, epochs and learning rate suggested in [12]. A 4-fold stratified cross-validation was performed and the results presented on Table 2 and Table 3 for cross-entropy loss, accuracy, and Mattews Correlation Coefficient are averaged among each fold. Both Table 2 and Table 3 are sorted by highest MCC result first, since it's the most suitable metric for the unbalanced database.

The experiments regarding Multilingual BERT demonstrated that the best result regarding MCC metric was the one presented where the batch size hyperparameter was set to 16, learning rate to 5.00E-05 and total epochs of 4. This best combination of hyperparameter resulted on a MCC of 0.8362, an accuracy result of 0.8369 and a cross-entropy value of 0.7326 as shown on Table 2.

Table 2. Multilingual BERT Results for Chapter 90

Batch Size	Epochs	Learning Rate	Cross-entropy Loss	Accuracy	MCC
16	4	5.00E-05	0.7326	0.8369	0.8362
16	4	3.00E-05	0.7398	0.8341	0.8334
32	4	5.00E-05	0.7339	0.8314	0.8307
16	3	5.00E-05	0.7580	0.8249	0.8242
16	4	2.00E-05	0.7716	0.8246	0.8239
32	4	3.00E-05	0.7778	0.8206	0.8199
16	3	3.00E-05	0.7827	0.8192	0.8185
32	3	5.00E-05	0.7853	0.8152	0.8145
16	3	2.00E-05	0.8360	0.8062	0.8055
32	4	2.00E-05	0.8475	0.8040	0.8032
16	2	5.00E-05	0.8390	0.8020	0.8013
32	3	3.00E-05	0.8475	0.8015	0.8007
16	2	3.00E-05	0.8876	0.7917	0.7909
32	2	5.00E-05	0.8995	0.7866	0.7857
32	3	2.00E-05	0.9497	0.7807	0.7798
16	2	2.00E-05	0.9718	0.7750	0.7741
32	2	3.00E-05	1.0014	0.7665	0.7656
32	2	2.00E-05	1.1658	0.7379	0.7369

In the work of [4] for the experiment with Chapter 90 without duplicates which is the most similar to this paper experiment, the classifier presented an average accuracy of 0.8338. In the same scenario, but with terms in English and considering the Harmonized System (HS) classification, the classifier developed by [12], obtained an average accuracy of 0.8330. The result from the Multilingual BERT experiment of 0.8362 outperformed both works with a BERT model that was pretrained in 102 languages. Figure 6 shows the hyperparameters optimization process on Multilingual BERT experiment with 18 scenarios using Weight & Biases to illustrate different scenarios.

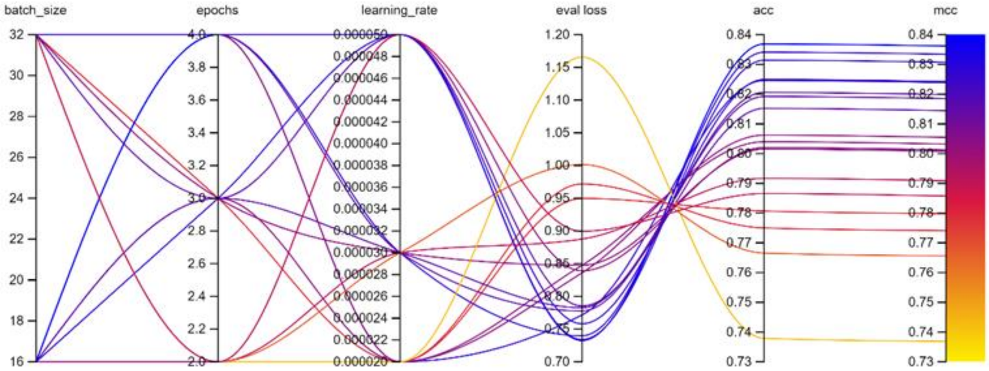


Figure 6. Multilingual BERT Hyperparameter Tuning.

The experiments on Portuguese BERT were also carried out 18 times, foreach possible combination of parameters suggested in [11]. The results showed that the greatest MCC achieved was 0.8491 for a batch size value of 16, 4 epochs and learning rate of 5.00E-05. In this empirical comparison, a lower batch size and higher epochs and learning rates has shown to be the best combination for fine tuning a BERT model using this data for both Multilingual and Portuguese models. For the best-case scenario in Portuguese BERT model experiment, the MCC presented a value of 0.8491, accuracy reached out 0.8497 as cross-entropy loss was 0.6941.

Table 3. Portuguese BERT Results for Chapter 90.

Batch Size	Epochs	Learning Rate	Cross-entropy Loss	Accuracy	MCC
16	4	5.00E-05	0.6941	0.8497	0.8491
16	4	3.00E-05	0.7047	0.8432	0.8426
32	4	5.00E-05	0.6946	0.8424	0.8418
16	3	5.00E-05	0.7054	0.8392	0.8385
16	4	2.00E-05	0.7414	0.832	0.8313
16	3	3.00E-05	0.742	0.8291	0.8284
32	4	3.00E-05	0.7434	0.8288	0.8281
32	3	5.00E-05	0.7358	0.8281	0.8274
16	2	5.00E-05	0.7707	0.8184	0.8176
16	3	2.00E-05	0.8001	0.815	0.8142
32	3	3.00E-05	0.8079	0.8114	0.8106
32	4	2.00E-05	0.8165	0.8106	0.8098
16	2	3.00E-05	0.8331	0.8046	0.8038
32	2	5.00E-05	0.8308	0.8039	0.8032
32	3	2.00E-05	0.9161	0.7877	0.7869
16	2	2.00E-05	0.9309	0.7847	0.7839
32	2	3.00E-05	0.9481	0.779	0.7782
32	2	2.00E-05	1.1177	0.7459	0.7449

1 Tables may have a footer.

Portuguese BERT results outperformed Multilingual BERT MCC results in all 18 scenarios shown. In this case, the model also improves results when comparing to [4] accuracy results for Chapter 90 with no duplicates and [12] results for the same chapter and scenario 90 but for Harmonized System (HS). Figure 7 also presents the hyperparameter tuning process illustrated using Weight & Biases for the Portuguese BERT model experiment.

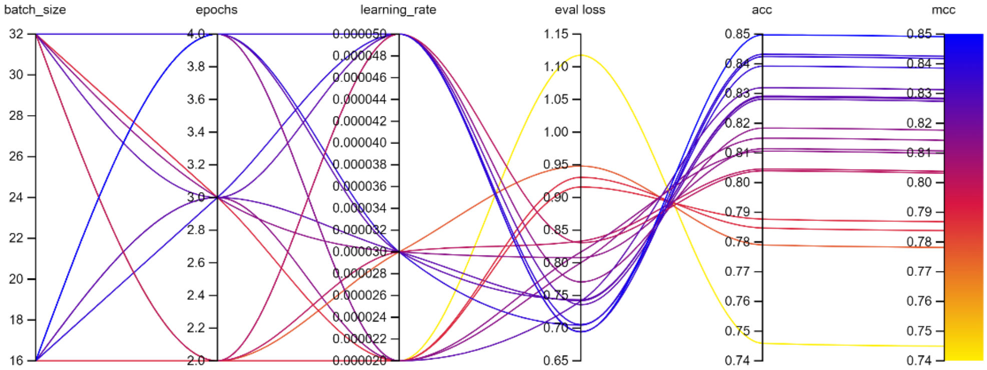


Figure 7. Portuguese BERT Hyperparameter Tuning.

Having presented the results, mainly about the Matthews correlation coefficient (MCC), it was noticed that with little computational cost it became possible to train a relevant performance classifier, mainly due to the knowledge transfer process that the BERT model allows. Thus, with the process of fine-tuning for the classification of product descriptions in their respective NCM codes, it was successfully executed leading to results such as 0.8491 using the Portuguese pre-trained model for the MCC in a task with 325 distinct classes.

6. Conclusion

Given the difficulties in the classification of goods in Brazil and the expensive fines for classification errors, the use of the developed classifier as a suggestion or starting point in this classification process is obviously an important and relevant result. Considering that Chapter 90 type goods have one of the greatest errors in the literature related to foreign trade, the classifier could be useful for supporting foreign trade analysts' decisions when classifying goods, mainly in chapters that require more technical knowledge and details.

Regarding the model comparisons, as BERT Multilingual documentation stated, specific-language methods tend to have better results particularly in feature-rich languages like Portuguese, which is proven in the empirical comparison made in this paper. The Portuguese BERT outperformed Multilingual BERT, meanwhile the results with Multilingual BERT are encouraging considering the training in 102 languages and even outperformed related work on same scenarios.

The knowledge transfer process in models such as BERT combined with the availability of import and export records by the Brazilian Revenue Service has made the development of classifiers like this possible. The possibility of fine-tuning the model, as well as its parallel nature allow for a reduction in training time and make it feasible to run it on local machines or notebooks running in the cloud as shown in this research. In addition to this, the availability of open-source libraries and models allows the sharing of knowledge and the implementation of solutions using state-of-the-art technologies by developers worldwide, as is the case of the Multilingual BERT proposed in [11] and Portuguese BERT proposed by [21]. In future works, we intend to expand the tests on large-scale shared data, one option is the use of algorithms and BIGDATA. Simulations with different data structures and metrics is also intended to broaden the spectrum of research and contributions.

References

1. Aggarwal, C. C. (2015) Data Mining: the textbook. Springer, Switzerland.
2. Alammar, J. (2018) The Illustrated BERT, ELMo, and co.: How NLP Cracked Transfer Learning. http://jalammar.github.io/illustrated-bert/. (Accessed 7 Jul 2021).
3. Alammar, J. (2018) The Illustrated Transformer. 1https://jalammar.github.io/illustrated-transformer/. (Accessed 11 May 2021).
4. Batista, R. A., Bagatini, D. D. S., Frozza, R. (2018) Classificação Automática de Códigos NCM Utilizando o Algoritmo Naïve Bayes. Isys - Brazilian Journal of Information Systems, [S.L.], v. 11, n. 2, p. 4-29. Sociedade Brasileira de Computação. http://dx.doi.org/10.5753/isys.2018.361.
5.
6. Biewald, L. (2021) Experiment Tracking with Weights and Biases. Weight and Biases. http://wandb.com/. (Accessed 9 Jun 2021).
7. Bizelli, J. S. (2003) Classificação Fiscal de Mercadorias. Aduaneiras, São Paulo.
8. Bramer, M. (2016) Principles of Data Mining. 3rd ed. Springer, London.
9. Chicco, D; Jurman, G. (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. Bmc Genomics, [S.L.], v. 21, n. 1, p. 1-13. Springer Science and Business Media LLC. http://dx.doi.org/10.1186/s12864-019-6413-7.
10. Cui, Y.; Jia, M.; Lin, T.; Song, Y.; Belongie, S. (2019) Class-Balanced Loss Based on Effective Number of Samples. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9260-9269, doi: 10.1109/CVPR.2019.00949.
11. Devlin, J. et al. (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, p. 4171–4186.

12. Ding, L.; Fan, Z.; Chen, D. (2015) Auto-Categorization of HS Code Using Background Net Approach. Procedia Computer Science, [S.L.], v. 60, p. 1462-1471. Elsevier BV. http://dx.doi.org/10.1016/j.procs.2015.08.224.
13. Gorodkin, J. (2004) Comparing two K-category assignments by a K-category correlation coefficient. Computational Biology And Chemistry, [S.L.], v. 28, n. 5-6, p. 367-374. Elsevier BV. http://dx.doi.org/10.1016/j.compbiolchem.2004.09.006.
14. Gorodkin, J. (2006) The Rk Page. https://rth.dk/resources/rk/introduction/index.html. (Accessed 8 August 2021).
15. Jurman, G.; Riccadonna, S.; Furlanello, C. (2012) A Comparison of MCC and CEN Error Measures in Multi-Class Prediction. Plos One, [S.L.], v. 7, n. 8, e41882. Public Library of Science (PLoS). http://dx.doi.org/10.1371/journal.pone.0041882.
16. Keedi, S. (2011) ABC do Comércio Exterior: abrindo as primeiras páginas. 4th ed. Aduaneiras, São Paulo.
17. Rajapakse, T. (2021) Simple Transformers. https://simpletransformers.ai/. (Accessed 2 May 2021).
18. Receita Federal do Brasil. (2020) NCM: Nomenclatura Comum do Mercosul. https://receita.economia.gov.br/orientacao/aduaneira/classificacao-fiscal-de-mercadorias/ncm. (Accessed 21 July 2021).
19. Receita Federal do Brasil. (2021) Sistema Apoio Siscori. https://siscori.receita.fazenda.gov.br/apoiosiscori/consulta.jsf. (Accessed 25 July 2021).
20. Scikit-Learn. (2021) Metrics and scoring: quantifying the quality of predictions. Disponível em: https://scikit-learn.org/stable/modules/model_evaluation.html. (Accessed 20 July 2021).
21. Souza, F.; Nogueira, R.; Lotufo, R. (2020) BERTimbau: pretrained BERT models for brazilian portuguese. Intelligent Systems, [S.L.], p. 403-417. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-61377-8_28.
22. Vaswani, A. et al. (2017) Attention is all you need. In: Advances in neural information processing systems. [S.l.: s.n.], p. 5998–6008.
23. Wu, Y. et al. (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.
24. Zafar, I. et al. (2018) Hands-On Convolutional Neural Networks with TensorFlow: solve computer vision problems with modeling in tensorflow and python. Packt Publishing, Birmingham.
25. Fava, L.P.; Furtado, J.C.; Helfer, G.A.; Barbosa, J.L.V.; Beko, M.; Correia, S.D.; Leithardt, V.R.Q. A Multi-Start Algorithm for Solving the Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints. Symmetry 2021, 13, 1697. https://doi.org/10.3390/sym13091697
26. Verri Lucca, A.; Mariano Sborz, G.A.; Leithardt, V.R.Q.; Beko, M.; Albenes Zeferino, C.; Parreira, W.D. A Review of Techniques for Implementing Elliptic Curve Point Multiplication on Hardware. J. Sens. Actuator Netw. 2021, 10, 3. https://doi.org/10.3390/jsan10010003