A Four-Step Cascade Methodology to Classify MCN Codes Using NLP Techniques

Pedro Pinheiro, Luan Siqueira, Marcos Amaris

[1]Universidade Federal do Pará
Faculdade de Engenharia de Computação, Tucuruí - Pará

pedrobraga85@gmail.com, luanrsiqueira18@gmail.com, amaris@ufpa.br

Abstract. *The MCN is a regional nomenclature for categorizing goods adopted by Mercosur countries. This nomenclature divides products using 8 digits, separated into 4 parts, Chapter, Heading, Subheading and Item/Subitem. There are indications that about 30% of the goods shipped globally have the wrong code because it is a manual process. This work aims to develop a process to classify the textual descriptions of the products present in the Electronic Invoices (NF-e). The classification was done using Natural Language Processing (NLP) techniques and tested using 2 different machine learning algorithms, Support Vector Machine (SVM) and Naive Bayes. A database of 340,000 distinct products was used for the experiments. We divided the process into 4 classification models, made to classify the 4 parts of the MCN. The data was divided into 80% training and 20% testing, and we obtained an accuracy of 89% for a total of 98 classes of the first 2 digits, and 76% using a cascade technique to classify the 8 digits.*
Keywords*: Natural Language Processing, Machine Learning, Text Classification and Mercosul Common Nomenclature.*

Resumo. *A NCM é uma Nomenclatura regional para categorização de mercadorias adotada por países do Mercosul. Essa nomenclatura divide produtos usando 8 dígitos, separados em 4 partes, Capítulo, Posição, Subposição e item/Subitem. Há indícios que cerca de 30% das mercadorias enviadas globalmente estão com seu código errado por ser um processo manual. Esse trabalho tem como objetivo desenvolver um processo para classificar as descrições textuais dos produtos presentes nas Notas Ficais eletrônicas (NF-e). A classificação foi feita utilizando as técnicas de Processamento de Linguagem Natural (PLN) e testada usando 2 diferentes algoritmos de aprendizado de máquina, Máquina de Vetores de Suporte (SVM) e Naive Bayes. Para os experimentos foi usada uma base de dados de 340.000 produtos distintos. Dividimos o processo em 4 modelos de classificação, feitos para classificar as 4 partes da NCM. Os dados foram divididos em 80% treinamento e 20% teste e Obteve-se um acurácia de 89% para um total de 98 classes dos 2 primeiros dígitos, e 76% de utilizando uma técnica de cascata para classificar os 8 dígitos.*
Palavras–chave*: Processamento de Linguagem Natural; Aprendizagem de máquina; Classificação de Texto; Nomenclatura Comum do Mercosul;*

1. INTRODUCTION

With globalization, the import and export of manufactured goods have increased to large dimensions, this brought a major obstacle, which challenges customs and vendors, this

challenge was the categorization of goods. In 1988 the World Customs Organization (WCO) created a nomenclature called Harmonized System (HS). This nomenclature served as a basis for the creation of the Mercosur Common Nomenclature (MCN), which is an ordered system that allows the utilization of its own rules and procedures, to determine a single numerical code for a specific commodity. The MCN is a regional Nomenclature for categorizing goods adopted by Brazil, Argentina, Paraguay, and Uruguay since 1995, and is used in all foreign trade operations in Mercosur countries [Roberto Scalco et al. 2015].

Each product description is related to a specific MCN code. The commercial or customs establishment has the responsibility to correctly inform the MCN of each product. However, this process is susceptible to human error due to the large diversity of products and lack of knowledge of the Nomenclature or misinterpretation of the MCN rules. In the city of Paraíba, a case happened about the misinformation of these codes in commercial establishments, either by mistake or dishonesty of the taxpayer. the State is directly affected, not collecting the tax on the goods or collecting from a good that should be exempt from tax thus harming the commercial stablishment itself, because each MCN code has linked different corresponding taxes, such as aliquot value, PIS (Program of Social Integration) and COFINS (Contribution for the Financing of Social Security), ICMS (Tax on the Circulation of Merchandise and Services), and TIPI (Tax on Industrialized Products).

The creation of the Electronic Invoice in 2004 aims to implement a standard of a national model of electronic fiscal documents. This standard will replace the current system on paper, with legal validity guaranteed by the digital signature of the sender, reducing tax evasion and increasing tax collection and reliability of the Invoice [SEFAZ 2021]. With this, the State Department of Finance gets to verify the MCN codes informed in the Invoices, according to [Ding et al. 2015], about 30% of goods sent globally are with the wrong code, this draws the attention of the whole world in search of solutions [Yu et al. 2012].

In recent years, it was significant the integration of machine learning and Natural Language Processing (NLP) techniques, which is an area of Artificial Intelligence that studies the automatic generation and understanding of natural human languages both in text and speech. This technology is used to develop applications such as translations between languages, chatbots, text summarization, sentiment analysis, and many others [Sebastiani 2002]. Within this universe, there are works in the areas of short text classification [Wang et al. 2017] and product classification.

In this project, we used a large amount of data about commercial products that were generated with Electronic Invoices. A routine was developed to extract text descriptions from the invoices with the MCN of each product. We collected approximately 340,000 distinct textual descriptions. Thus, textual descriptions were transformed to structured data using a tokenization process of NLP and other techniques as stop-words, stemming and frequency–inverse document frequency. Different process of data Balance were perform to build machine learning models. we used this data to exploit machine learning techniques in conjunction with NLP to create predictive models to assist in the MCN code classification. We evaluated 2 supervised learning algorithms. They were a Support Vector Machine and a Naive Bayes classifier.

This paper is structured as follows: Section 2 describes fundamental concepts for a

good understanding of this work. Section 3 shows some related works. Section4 presents the methodology used in this research, then the results are presented in Section 5 and finally the conclusions in Section 6.

2. CONCEPTS AND THEORETICAL BACKGROUND

2.1. Electronic Invoice

[Brasil 2003] provides for ancillary obligations, namely, the transmission of taxpayers' tax and economic information to the tax authorities for inspection purposes, as well as tax registration and tax data. The exchange of tax information supports each Treasury Department (SEFAZ in Brazil) to combat tax evasion and reduce delinquency through data crossing and electronic verification mechanisms. This facilitates the identification of taxpayers in irregular situations [Sousa 2010].

The implementation of the Electronic Invoice (NF-e) aims to simplify the guarantee obligations and save on the storage and the environment of paper documents while helping to combat tax evasion and avoidance. The Public Digital Bookkeeping System (SPED) in Brazil creates an environment in which the Treasury and the federal tax authorities can combine accounting and tax information, identify fraud and tax evasion, and cover the entire production chain. It also defines new control and management processes, reliability of information, synchronization of records, consistency and integration between corporate and tax systems [Bonfim et al. 2012].

DANFe is an acronym in Portuguesse for "Electronic Invoice Auxiliary Document". This document is a readable and simplified presentation of the Invoice. In a nutshell, the DANFe is a printed document that contains the main information of an Electronic Invoice (NF-e). For the present work we extracted from the NF-e the product descriptions.

2.1.1. Mercosur Common Nomenclature (MCN or NCM in Portuguese)

The MCN code is a convention for categorizing goods. It is a Mercosur version of the Harmonized System, which is a multipurpose international product nomenclature developed by the World Customs Organization (WCO). The first six digits of the MCN were inherited from the Harmonized System and added two more digits at the end, which are specific to the Mercosur scope. The first six digits correspond to three pairs of digits that specify the Chapter, the Position, and the Subheading. The seventh and eighth digits are exclusively of the MCN coding, and refer to the Item and Subitem, respectively as shown in Figure 1.

2.2. Natural Language Processing (NLP)

Natural Language Processing has as its fundamental kernel the analyzing and representation of Natural Language at one or more levels of linguistic analysis for a range of tasks or applications.

This process begin with *tokenization*, which is the process of separating words from text, breaking the sentences into blanks and extracting the words, generating a vector of tokens. Next is the removal of the empty words or *Stop-words*, which refers to those

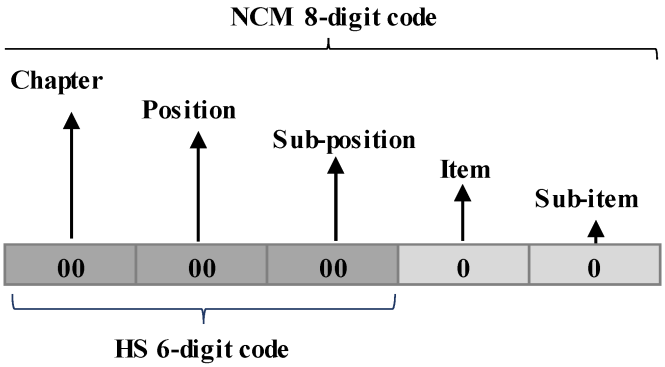

Figure 1. Composition of the MCN (Mercosur Common Nomenclature)

words that are more common in the language or connection words or those with low significance for the sentence and for the process classification. Finally, the process of stemming or stemization is done [Orengo and Huyck 2001], which is the name given to the process of combining variations of a word into a common root word. It must be used a statistical metric related to the frequency of the final words. Below, better details of each one of the NLP phases that we followed in this project.

2.2.1. Tokenization, Stop-Words, Stemming, and Term Frequency

Tokenization aims to transform texts into logical data structures. It is a process to separate a piece of text into smaller units called tokens. The literature typically classify the tokenization into 3 types – word, character, and subword. In this work, we performed the tokenization by words. This is done dividing the descriptions of the e-NF in words; thus, each division is called a token.

In documents, higher-frequency words are more important to represent the content than lower-frequency words. For this reason, in computer sciences commonly **stop-words** are terms that needs to be ignored in a textual process from a source. Some high-frequency words would be the prepositions ”the”, ”for”, ”in” with low content about the main information of the documents. These words may mislead the results. Commonly, each language has a list with the term of stop-word that needs to be deleted e.g. determiners, coordinating conjunctions, prepositions, among others. Therefore, we need to ignore them reducing the data dimension and increasing the relevance between words and documents or categories.

Stemming is the process of producing morphological variants of a root/base word. The root is the element that contains the fundamental meaning of a word. For example, the words in Portuguese *vidro*, *vidraça*, *vidraceiro*, have the same root, ”vidr”. Stemming allows querying using any of these terms to return documents that have the same root. Similar to the *Stop-Words* process, *Stemming* reduces the number of dimensions and increases the relevance between words and documents or categories.

2.2.2. Term Frequency

In machine learning models a vector representation is required for text analysis or any problem. In text analysis, the importance or relevance of each word is used and normally it is given as a term frequency. In this work, we used the Term Frequency-Inverse Document Frequency (TF-IDF). [Neto et al. 2000] explained, basically, the term frequency of a word w in document d, denoted by $TF(w, d)$, is the number of times the word w appears in document d. The document frequency of a word, denoted by $DF(w)$, is the number of documents in which w appears. Thus, exists too the inverse document frequency of a word w, denoted by $IDF(w)$, see equation 1. Therefore, a word w will have low relevance if the word appears in more than one document, indicating that the word has little distinguishing power between documents. On the other hand, the inverse document frequency of a word w is high if the word appears in the documents, indicating that the word has high discriminative power for documents.

$$IDF(w) = 1 + log(|D|/DF(w)) \tag{1}$$

To find the highest ranked words. We can express this in a single formula, see equation 2.

$$TF - IDF(w, d) = TF(w, d) * IDF(w) \tag{2}$$

After pre-processing the descriptions and transforming them into a vector representation, it is ready to be input to the classifier. below, we explain the two different simple machine learning models that we evaluated as classifiers.

Supervised machine learning is one of the most used techniques in text classification. This task aims identifying, distinguishing, and understanding objects from any source of data. In classification problems the objects are grouped into categories, usually for a specific purpose. Text classification is becoming an essential technology for processing and organizing large volumes of documents. Automatic text classification is treated as a supervised machine learning technique. The goal of this technique is to determine whether a given document d belongs to a certain category by looking at the words w in that category [Kadhim 2019].

Below, we define briefly the two supervised machine learning techniques that we used in this work. they are Support vector machine (SVM) and Naïve Bayes (NB) classifiers

2.2.3. Support Vector Machine (SVM)

Initiated studies were done by Vapnik in 1995, SVM is a widely used technique for classification and regression problems. SVM uses partitioning-based supervised classification methods, whose goal is to build hyperplanes with the best possible separation between classes. This technique has been used successfully in a number of pattern recognition applications, such as: text classification, SPAM classification, handwriting recognition, text analysis, gene expression, etc. SVM belongs to the general category of kernel methods,

which are algorithms that depend on the data only through dot-products. The dot product can be replaced by a kernel function which computes a dot product in some possibly high dimensional feature space Z. It maps the input vector x into the feature space Z, allowing calculations to be performed in this higher dimensional space. SVM uses different kernel functions to separate data in hyper-planes.

2.2.4. Naive Bayes Classifiers

Naive Bayes classifiers are the most popular technique in machine learning literature for applying probability concepts. The *Naive Bayes* model is a set of probabilistic classifiers of observations based on the application of the *Bayes* theorem, which defines that the conditional probability of an event is the probability obtained by the additional information of another event that has already occurred [Andre Dieb Martins 2013].

Although it is a simple model, it gives good results and is very useful for large volumes of data. In summary, the term *"Naive"* refers to the way the algorithm analyzes the features in the dataset, assuming that the features are independent of each other [Russell and Norvig 2003].

3. RELATED WORKS

There are few papers that discuss the problem of predicting HS or MCN codes, we realized a fast search in Google Scholar and IEEExplore and we found out 79 works and 1 paper, respectively, using the next search string in both electronic libraries (*"supervised learning"AND classification AND ("Harmonized System"OR "Mercosur Common Nomenclature")*). Below, we mentioned the work that we found more relevant for this research.

The Work of [de Abreu Batista et al. 2018] consists of the development of a classifier for the automatic categorization of product descriptions in their MCN codes, the goal is to extract data from the Electronic Invoice to Consumer (NFC-e), to perform a supervised learning using the *Naïve Bayes* algorithm, the results showed an average accuracy of 86.5% of 2 classes only.

[Luppes et al. 2019] Proposed a Convolutional Neural Network (CNN) architecture to label descriptions based on short text descriptions, they used *embeddings word* techniques with various onlines databases such as **DBmedia**, they obtained results of 92% for the initial 2 digits, also chaptered from **HS-2**.

[Ding et al. 2015] Uses a vector space approach. Its dataset was obtained from Singapore Customs. It obtained great performance in one of the chapters, with 98% accuracy, the main part of their findings was that the smaller the descriptions available, the more difficult it became to classify them in a chapter: when classifying a description with up to three tokens, they can only achieve an accuracy of only 15

[Li and Li 2019] approaches the problem differently. They built two simple CNNs, one for text and one for images, to classify shoe descriptions into four classes based on six types of HS codes. They achieved an accuracy of 93%. Their dataset includes 10,000 images of the shoes along with a text description.

Recently, in [de Lima et al. 2022] the authors made use of the BERT (Bidirectional Encoder Representations from Transformers) model to train only one classifier that aims to classify descriptions on its respective MCN chapter code. The authors divided their dataset into 96 chapters and focused the classification only inside a single chapter. Their classifier is used to predict only products in Chapter 90. Their best combination of hyper-parameter resulted on an accuracy of 0.837%.

The works of [de Abreu Batista et al. 2018] and [de Lima et al. 2022] presented a classification of the MCN, considering only the two fist digits, specific chapter only. [Luppes et al. 2019] shown a classification considering only the first 4 digits. The present work differ of others that we proposed to classify 98 different chapters classes and the complete MCN code, using a cascate methodology not found in the related works. We evaluated 2 different classification models, Support Vector Machine and naive bayes, different from the works of [Ding et al. 2015] and [Li and Li 2019] that use Neural Networks.

4. METHODS AND MATERIALS

The objective of this section is to present the methods and materials used to classify the item descriptions contained in the NF-e into corresponding MCN codes. These methods involved converting the text data into logical data structures based on the frequency of the terms. Data was labeled and organized depending on the MCN code of each product. We performed a cleaning data process based on Natural Language Processing. We executed oversampling and undersampling procedures to balance data. Subsequently, we train different classification algorithms based on supervised learning techniques to finally evaluate their performances.

To do this, we built a methodology divided into four steps, they are (1) statistical analysis and data balancing; (2) pre-processing of the product descriptions; (3) training and validation of the classifiers; (4) analysis and interpretation of the results, see Figure 2.

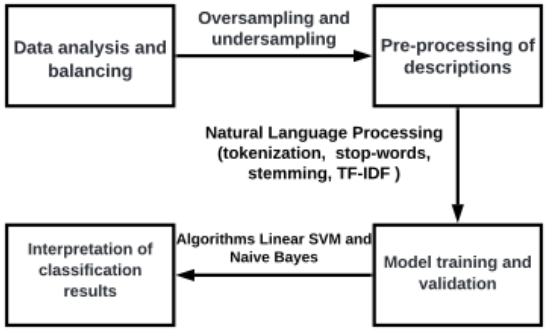

Figure 2. Natural Language Processing Workflow done in this work

4.1. Data analysis and balancing

An accounting office of a unknown city (omitted for double-blind revision) provided data that we used in this research. The accounting office stored data locally in the Database Management Systems (DBMS) **PostgreSQL**. The information in the DBMS are product descriptions and their respective MCN codes, as Table 1 shows. That information

was extracted from the Electronic Invoices (NF-e), resulting in 340,000 distinct descriptions. The work [Luppes et al. 2019] proposed that MCN code can be divided into 4 parts: chapter, position, sub-position and item/subitem, as shown in Table 1.

Table 1. Segregated MCN Codes

PRODUCT DESCRIPTIONS	MCN CODE	MCN-2	MCN-4	MCN-6	MCN-8
BUSCOFEM 400MG 10 CAPS GEL	30049029	30	04	90	29
ACHOCOLATADO EM PO TODDY ORIGINAL 200G	90308490	90	30	84	90
TEMPERO SAZON CARNES 60G	21039021	21	03	90	21
VERDURAS DE REAPROVEITAMENTOKG	07020000	07	02	00	00
OLEO DE SOJA COMIGO 900ML	15079011	15	07	90	11
FEIJAO IMPERIAL CARIOCA 1KG	07133399	07	13	33	99
OLEO DE SOJA COMIGO 900ML	15079011	15	07	90	11
SOUTIEN C:41T:EG	62121000	62	12	10	00
DESINFETANTE PINHO BRIL SILVESTRE 1LT	38089419	38	08	94	19
COXA SOBRECOXA FRANGO KG	02071400	01	07	14	00

Due to the wide variety of MCN codes and diversity of products pro code, some codes tended to appear more and others less, generating an unbalance in the data as shown in Figure 3

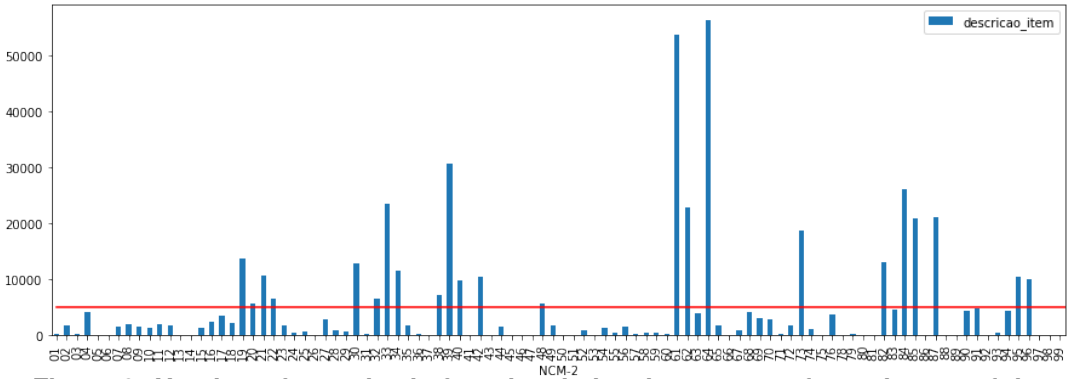

Figure 3. Number of samples before data balancing process from classes of the first two digits of the MCN (chapter)

We used two different techniques for data balancing, they are under-sampling and over-sampling and were proposed by [Prati 2006]. In the replication process of the minority classes **over-sampling**), the texts were translated to English and retranslated to Portuguese, and inserted again into the dataset. The classes that had less than 1000 samples were removed as shown in Table 2. The classes with larger samples number, we removed random samples executing the second technique for data balancing, **under-sampling**, limiting to 5000 samples. Figure 3 shows the threshold used to deleted the samples. Before the data balancing process, we had 340 thousand samples approximately, and after data balancing, we obtained 250 thousand samples. Figure 4 shows the result after balancing with the MCN chapter experiment with the first two digits. For reason of space and for question of reproducibility other images, scripts of the experiments and data can be found in this repository `https://github.com/`.

4.2. Pre-processing of textual descriptions

This step describes the pre-processing performed on the textual descriptions of the products contained in the NF-e, and aims to extract the feature vector of each of the input

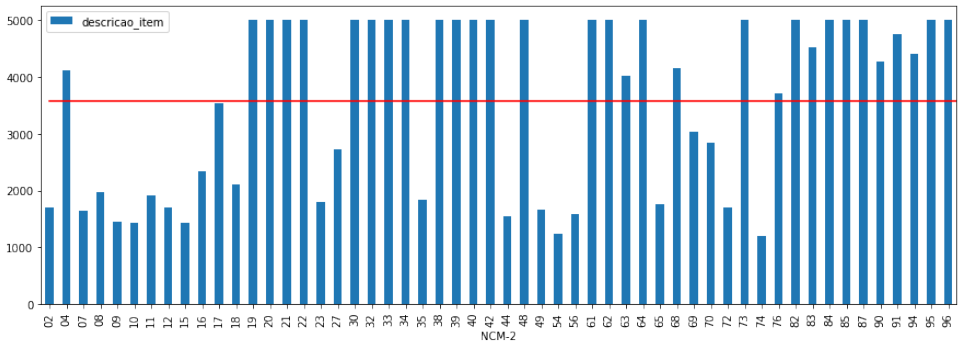

Figure 4. Number of samples in Chapter MCN-2 after balancing

Table 2. Number of classes before and after the data balancing

NCM Digits	Unbalanced	Balanced
2 digits	99	98
4 digits	85	81
6 digits	83	82
8 digits	90	88

documents as shown in Figure 2. The algorithm takes a series of descriptions as input and starts by tokenizing them, breaking the text, and extracting a vector of **tokens**. The next stage of the algorithm is to remove the *stopwords* and move to apply *stemming* on the remaining *tokens*, as mentioned in Subsection 2.2.

With the conclusion of the previous steps, the textual descriptions tend to reduce their dimensions due to the removal of *stopwords* and *stemming*. The average dimension (number of words) of the descriptions before pre-processing was 36 characters and after the pre-processing, this average falls to 29 words, as Figure 5 shows. The box plots of Figure 5 shows the statistical information about the number of words per sample in the dataset, thus this figure shows the median of words per samples, the upper and lower first quartiles and outliers are marked as individual points.

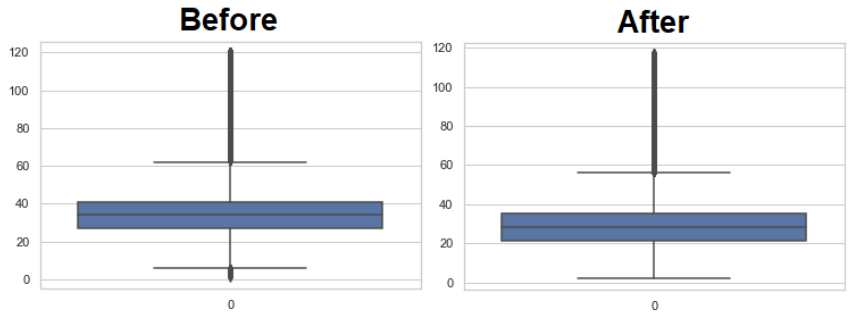

Figure 5. Statistical information about number of words per sample in the dataset

The next step is to consider the term frequency of the words. We calculated the TF and IDF metrics (equation 1), which will be the basis for the TF-IDF equation 2, as explained in Subsection 2.2. the result of the TF-IDF operation will produce the weights of the feature vectors used for training and testing our classifier.

4.3. Classifier training and testing

As we said above, in order to classify all the MCN code we divided it into four parts. Figure 6 presents the classification process performed in this work. In this process, for each output of the model, the predicted MCN (output) is used as input together with the tokenized textual description to predict the subsequent 2 digits of the MCN. In the final, we have all the MCN complete using 4 four steps and different machine learning models. The flowchart of the Figure 6 explains visually this cascade methodology.

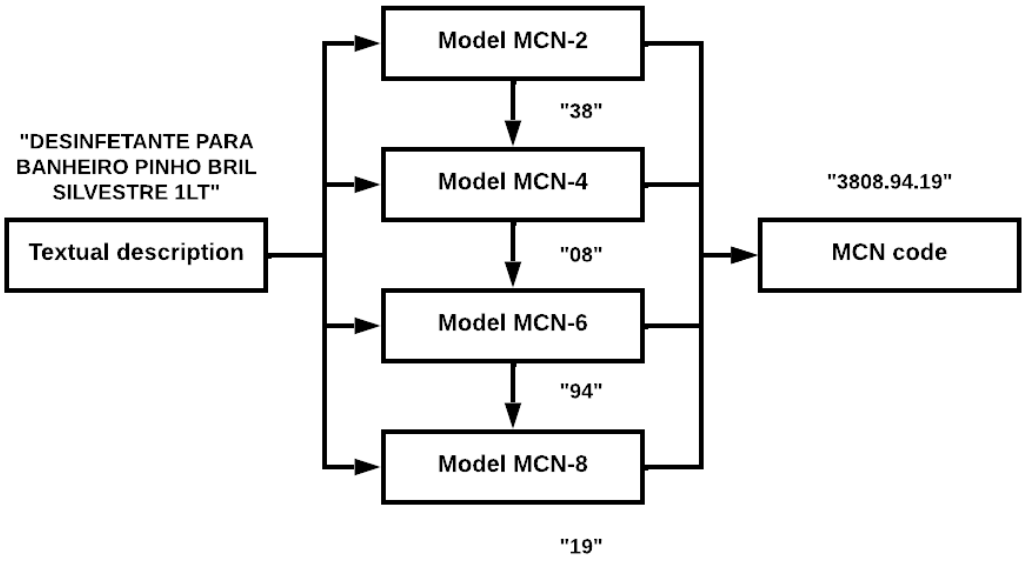

Figure 6. Flowchart of the cascade methodology used in this work to predict all the MCN code

Based on past literature about NLP problems, we have selected 2 simple classification algorithms of supervised learning, which are **Support Vector Machines**, and **Multinomial Naive Bayes**. In future works, we want to evaluate deep learning techniques and concurrent neural networks. Data was divided in 80% training and 20% testing, the input features are the descriptions in vector form and the target variable are the MCN digits or MCN classes.

5. ANALYSIS OF THE EXPERIMENTS AND RESULTS

The main objective of this work was to apply a four-step methodology to predict MCN codes using PNL techniques and machine learning models. Two machine learning models were tested, they were SVM and Naive Bayes classifiers. We followed the same cascade methodology configuration for both classifier models. For SVM model we used a kernel linear assuming linearity in data. As future work we will test no linear kernels.

Figure 7 presents the results of the test of both models done separately. We used the metric of accuracy F1-score to measure them. It can be observed that the results obtained by the Linear SVC in the first two models have a high level of accuracy, considered a good result, while the Naive Bayes only presented a better result in its first model when

the 2 first digits are predicted. Finally, the results of the cascade methodology models are shown in Table 3, it is observed that linear SVC is better than the Naive Bayes algorithm.

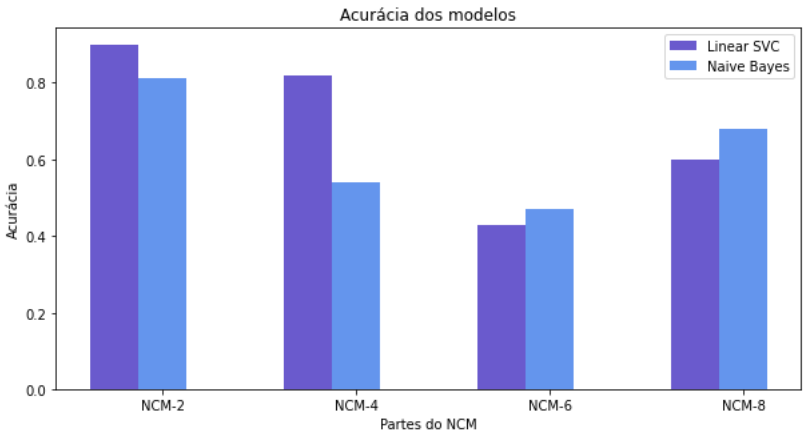

Figure 7. Accuracy of the models in the four-step cascade methodology

Table 3. Accuracy of the model using the methodology in cascade

Metric	Linear SVC	Naive Bayes
F1-score	0.76	0.67

6. CONCLUSIONS AND FUTURES WORKS

In this study, we follow a four-step cascade methodology to predict MCN codes using NLP techniques and machine learning techniques. Text descriptions extracted from the e-NF were used in this project. We developed and evaluated a text mining application using two supervised machine learning techniques: Support Vector Machine; Multinomial Naive-Bayes. We performed data balancing techniques like under-sampling and over-sampling. When we analyzed the results of both classifiers, it was observed that the linear SVC did better.

As Future Works, we want to use more modern machine learning techniques. Deep learning techniques based on sequence analysis will be tested. A short-term long memory neural network could be a starting point for the next tests.

References

[Andre Dieb Martins 2013] Andre Dieb Martins, Bruno B. Albert, E. C. G. (2013). Classificador de textos otimizado utilizando lei de potencia para palavras raras. *XXXI SIMPOSIO BRASILEIRO DE TELECOMUNICAÇÕES*.

[Bonfim et al. 2012] Bonfim, D. P., Moraes, D., Machado, H., Amorim, M. O., and Raimundini, S. L. (2012). Nota fiscal eletrônica: uma mudança de paradigma sob a perspectiva do fisco estadual. *ConTexto*, 12(21):17–28.

[Brasil 2003] Brasil (2003). Emenda constitucional n. 42.

[de Abreu Batista et al. 2018] de Abreu Batista, R., Bagatini, D. D., and Frozza, R. (2018). Classificação automática de códigos ncm utilizando o algoritmo naïve bayes. *iSys-Revista Brasileira de Sistemas de Informação*, 11(2):4–29.

[de Lima et al. 2022] de Lima, R. R., Fernandes, A. M. R., Bombasar, J. R., da Silva, B. A., Crocker, P., and Leithardt, V. R. Q. (2022). An empirical comparison of portuguese and multilingual bert models for auto-classification of ncm codes in international trade. *Big Data and Cognitive Computing*, 6(1).

[Ding et al. 2015] Ding, L., Fan, Z., and Chen, D. (2015). Auto-categorization of hs code using background net approach. *Procedia Computer Science*, 60:1462–1471.

[Kadhim 2019] Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 52(1):273–292.

[Li and Li 2019] Li, G. and Li, N. (2019). Customs classification for cross-border e-commerce based on text-image adaptive convolutional neural network. *Electronic Commerce Research*, 19(4):779–800.

[Luppes et al. 2019] Luppes, J., de Vries, A. P., and Hasibi, F. (2019). Classifying short text for the harmonized system with convolutional neural networks. *Radboud University*.

[Neto et al. 2000] Neto, J. L., Santos, A. D., Kaestner, C. A., Alexandre, N., Santos, D., A, C. A., Alex, K., Freitas, A. A., and Parana, C. (2000). Document clustering and text summarization.

[Orengo and Huyck 2001] Orengo, V. M. and Huyck, C. R. (2001). A stemming algorithmm for the portuguese language. In *spire*, volume 8, pages 186–193.

[Prati 2006] Prati, R. C. (2006). *Novas abordagens em aprendizado de máquina para a geração de regras, classes desbalanceadas e ordenação de casos*. PhD thesis, Universidade de São Paulo.

[Roberto Scalco et al. 2015] Roberto Scalco, P., Klaold Lippi, M., and de Almeida, M. I. S. (2015). Preço e renda como determinantes da demanda por bens de luxo no brasil: Um estudo econométrico com produtos importados da nomenclatura comum do mercosul. *Brazilian Journal of Management/Revista de Administração da UFSM*, 8(3).

[Russell and Norvig 2003] Russell, S. J. and Norvig, P. (2003). Instructor's solution manual for artificial intelligence: a modern approach.

[Sebastiani 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

[SEFAZ 2021] SEFAZ (2021). Sobre a nf-e.

[Sousa 2010] Sousa, J. P. R. d. (2010). Impactos da utilização da nota fiscal eletrônica nas atividades de monitoramento e fiscalização do icms: um estudo na secretaria da fazenda do estado do ceará. Master's thesis, Universidade Federal do Ceará,.

[Wang et al. 2017] Wang, J., Wang, Z., Zhang, D., and Yan, J. (2017). Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI*, volume 350.

[Yu et al. 2012] Yu, H.-F., Ho, C.-H., Arunachalam, P., Somaiya, M., and Lin, C.-J. (2012). Product title classification versus text classification. *Csie. Ntu. Edu. Tw*, pages 1–25.