Occupation and Economic activity coding using natural language processing

Organisation: Instituto Nacional de Estadística y Geografía (INEGI)

Authors: José Alejandro Ruiz Sánchez (INEGI), Jael Pérez Sánchez (INEGI), Adrián Pastor (CIMAT)

Date: 02.28.2020

Version: 1.0

1. General background, why and how this study began

Most projects related to the production of statistics (surveys, census or administrative records) contain items with textual answers for where it is necessary to assign numerical codes in order to present and analyze data. This process is known as *Coding*.

The National Institute of Statistics and Geography (INEGI, by its acronym in Spanish) has developed a codification system for sociodemographic projects using deterministic rules to assign a valid code to each description. From now on, we will refer to it as "deterministic coding" (no Machine Learning implied). However, due to the quality of some descriptions, deterministic coding becomes impossible for each record. For those descriptions not coded, a group of experts assigns a code through a codification system interface. This process has been named as "assisted coding".

To this day, the current process (deterministic coding & assisted coding) codes all the records with quality levels over 95% for most of the variables. This has not been the case for the Occupation and Economic activity variables in surveys like the National Occupation and Employment Survey (ENOE, for its acronym in Spanish) and the National Household Income and Expenditure Survey (ENIGH, for its acronym in Spanish). These are among the most important projects in the Institute where the quality level is around 90%, although the percentage of deterministic coding has not surpassed 76% of the records and human experts do the rest.

The National Household Income and Expenditure Survey (ENIGH) was chosen to begin the exploration of Machine Learning (ML) models to improve the current coding process for Occupation and Economic activity variables. The project envisages three stages. The first stage seeks to explore ML algorithms and evaluate their quality so one or two can be selected.

In the second stage, the performance of the chosen ML model(s) with the performance of the current codification scheme (for that we will use the upcoming 2020 ENIGH) will be compared. A group of thematic experts will verify the codes of those cases where ML and the current process differ and we will use this new information as an additional precision metric for the ML model performance.

The incorporation of the ML model to the current productive process is the main goal of the third stage, keeping in mind the preservation of minimal acceptable quality achieved by the current codification scheme.

The standardization of Classifications and Codification Strategies Department designs and implements the coding process of all sociodemographic projects inside INEGI. When this necessity comes out, a collaboration with the Research Department emerged with the task of developing a more efficient process where ML could play a role. This project was started with the approval of the Director General of Sociodemographic Statistics and the General Deputy Research Director.

This project meets the prevailing need to research and innovate statistics generation processes, aiming to improve the quality and timeliness of results. The contributors have vast knowledge in ML techniques and postgraduate degrees in Statistics, Economics and Computer Sciences.

2. Data

2.1 Input Data

We chose the 2018 National Household Income and Expenditure Survey (ENIGH) to apply Natural Language Process (NLP) and ML techniques to code Occupation and Economic activity variables. This survey collects information every two years about sociodemographic, economic and labor characteristics of the Mexican population. In 2018 the survey was conducted on over 74 600 households, on which information about the economic activity and occupation, among other variables, of 158 568 individuals was gathered.

Currently, the department in charge of the coding process relies on primary variables (textual responses) and auxiliary variables (categorical responses). As the name implies, primary variables are those that have the most relevance in the process. These same variables make up the main input for the ML algorithm.

Table 2.1 Features used for classification

Main variables (text)	Auxiliary variables (categorical)
Name of occupation Tasks or roles Economic activity of the company or institution Name of the company	Place where the person carried out the activities Classification of the company Whether the person has personnel under their command Academic level Academic grade Approved academic level Approved academic grade Company size Whether the person works inside the country Additional questions about the company

2.2 Data preparation

The main variables in Table 2.1 are submitted to a normalization and cleaning process. This process is the same as the one in the current codification scheme consisting in article suppression, stemming and lemmatization. These processes are implemented with computational routines whose goal is the homogenization of texts before the deterministic coding phase. It is worth mentioning that the computational routines were programmed inside INEGI and, to this day, we do not use third party libraries.

The whole data preparation process consists of two phases applied to both main and auxiliary variables: “basic preparation” and “synonyms replacement”. Basic preparation consists of changing the descriptions in each record to uppercase, eliminating accent marks, double spaces and blank spaces at the beginning, middle or end of the description, eliminating some specifics words (for instance, “ETCETERA”) and phrases (for instance, “I do not remember my company’s name”), as well as unique characters (symbols or signs) when they come without more text. In the same phase, we do some stemming process aiming to transform plurals into singulars.[1] Synonyms replacement, which can be thought similar to lemmatization, aims to replace a set of alike words by their root. To accomplish the task, the coding department at INEGI has being developing a word dictionary through the years to match a set of alike words to the appropriate root word.[2]

We carried out the exact same cleaning and normalization procedure before the ML process to code.

2.3 Feature selection

Feature selection was not used for the ML codification. The variables in Table 2.1 are the ones used for the text vectorization process.

2.4 Output data

Our objective is to use traditional ML methods (mainly because we have some experience with SVM and because we want to have a strong benchmark before using *Deep Learning*) to classify two of the variables that present the biggest challenges for the production processes in INEGI: Occupation and Economic activity. The variable Occupation is classified according to the National Classification System for Occupations (SINCO), while the variable Economic activity is classified based on the North American Industrial Classification System (SCIAN) for households.

There are 461 classes for the variable Occupation and 157 classes for Economic activity in the 2018 ENIGH database. Out of the 158 568 records in the database, 58 belonged to classes with less than four records each and were excluded.

3. Machine Learning Results

3.1 Models tested

The results of the different exercises shown here have the purpose of informing about methodological variations favorable for our classification problems. This will help similar projects to find a more precise-methodological structure. We applied all exercises to the same subset of records (75% for training and 25% for testing) and we took as our ground truth the data base already coded by the current processes (human assisted coding plus decision rules coding).

Unless stated otherwise, we use features in Table 2.1 as input and the Scikit-learn Python library. The first set of exercises has the main goal to see how useful it would be to split words, combine word segments, or use complete words as text variables; this is, to experiment with different combinations of n-grams, either in length as in joins between them. In the first exercise each text entry is divided in 6-grams (6-characters). We tried different lengths in previous exercises, included those with the more common word either as unigrams and bigrams. These 6-grams are used as

[1] See the Annex 1 for more details

[2] See the Annex 2 for more details

inputs to generate a TF-IDF matrix with 30 000 dimensions, which would later be used in a SVM algorithm to make the classification. The second exercise uses 6-grams as well as 10-grams; for each of these sets we generated a TF-IDF matrix with 30 000 dimensions, which makes a 60 000 dimension vectorial representation for every entry in the database when joined with each other. The third exercise generates a 6-gram TF-IDF matrix and a TF-IDF matrix made by joining two complete words; this is, instead of dividing a word, we generate word pairs. As is shown in Table 3.1, the last one gets the best results in terms of Accuracy (number of records were the ML algorithm assigned the same code as the one assigned by the current process divided by the total number of records, 0.8793).

Table 3.1 Accuracy for different text representations. SVM classifier

	Economic activity	Occupation
6-grams	0.8782	0.8204
6-grams, 10-grams	0.8781	0.8189
6-grams, 2-words	0.8793	0.8188

The goal of the second set of exercises is to value the use of different classification met**h**ods or the combination of them. In our case, we found that an ensemble of the following methods works the best: SVM, Logistic regression, Random Forest, Neural Networks, XGBoost, K-NN, Naïve Bayes and Decision trees. We also found that the improvement is significant when we use complete words together with 6-grams (see Table 3.2)

Table 3.2 Accuracy for different text representations. Assembly classifier

	Economic activity	Occupation
6-grams	0.8849	0.8474
6-grams, 10-grams	0.8825	0.8467
6-grams, 2-words	0.8905	0.8505

As expected, results vary significantly among classification algorithms, and it can be convenient to assign different weights to each one (resembling a voting process). Among the different exercises we carried out for the Economic activity variable, we found an improvement when assigning the following weights in the voting process (each weight can be seen as the number of votes a specific ML algorithm has in the final call to determine the code to be assigned to a record): 4 to SVM, 2 to logistic regression, 1 to Random forest, 3 to Neural networks, 2 to XGBoost, 1 to K-NN, 1 to Naïve Bayes and 3 to Decision trees. The improvements are noteworthy in metrics such as Recall and F1 (see Table 3.3)

Table 3g.3 Results using 6-grams and 2-words. Economic activity

	Accuracy	Precision	Recall	F1
Ensemble with equal weights	0.8905	0.6925	0.6149	0.6365

Ensemble with different weights	**0.8921**	**0.6767**	**0.6420**	**0.6512**

For the Occupation variable, the weights that generated the best results were as follows: 4 to SVM, 2 to Logistic regression, 2 to Random Forest, 4 to Neural Networks, 3 to XGBoost, 1 to K-NN, 1 to Naïve Bayes, 3 to Decision trees (see Table 3.4).

Table 3.4 Results using 6-grams and 10-grams. Occupation

	Accuracy	Precision	Recall	F1
Assembly with equal weights	0.8447	0.6441	0.5384	0.5639
Assembly with different weights	**0.8505**	**0.6437**	**0.5637**	**0.5831**

These previous results are obtained considering only one phase in the classification process; this is, no matter the number of classes, each record is classified in one class and the whole process is done at once. However, a main characteristic of the Economic activity and Occupation codes is that they are hierarchical: every code is formed by 4 digits where the first two belong to sector code, in a way that many codes (classes) can belong to the same sector. The variable Economic activity is comprised of 157 classes belonging to 57 sectors; the variable Occupation has 461 classes grouped in 52 sectors. This structure in the codes can be used to try to increase the predictive power of the algorithm. The idea is to help to code a record by adding new features, each one of these new features representing a possible sector (two-digit code) to which a record belongs to. The whole process is like telling a record to what sector it may belongs to, hence reducing the possible options to assign a four-digit code.

To test this idea we make the following experiment: we first partition the whole data base in two, one is going to be the training set and the other one is going to be the test set. Then, and only for the training set, we add a group of dichotomic features, each one of these variables has a value of 1 if the record belongs to that sector (two-digit code), and 0 otherwise. This is possible because we only want to get the parameters of the fitted model (think of the coefficients of a SVM algorithm). This new group of additional variables, together with the variables in Table 1, are used to train and obtain an integral classification algorithm ("class model") based on the records of the training set.

Then, and independently of the process described above, we developed a SVM algorithm to predict the sector (two-digit code). For that we take the training set and based on the features of the Table 1 we are able to get the parameters to predict a two-digit code (dependent variable) for a record. That way we come up with a model to predict a two-digit code for the test set, in which we are supposed not to know the true two-digits code but we do know the features on Table 1. Now we have a way to predict a two-digit code for the test set which in turn will become the dichotomic features needed in what we called "class model" above. Together, those new and predicted dichotomic features for the test set plus the features in Table 1 are what we need to apply the class model parameterized using the training set. We ended up with an integral model to assign a record to a specific four-digit code.

Results of this exercise are shown in tables 3.5 and 3.6. For comparison, in the first row we replicate the results shown in table 3.1, where we use SVM and TF-IDF matrices with 6-grams joined to fullword pairs. By using SVM and the same TF-IDF matrix architecture, we imply that the two-stage exercise (sector first, and then class) for the variable Economic activity improves only in the metric Precision; however, for the variable Occupation there seems to be a significant improvement in Accuracy and Precision.

Table 3.5 Hierarchical classification model using SVM. Economic activity

	Accuracy	Precision	Recall	F1
One stage (6-grams, 2-words)	0.8793	0.6372	0.6760	0.6511
Two stages (6-grams, 2-words)	0.8774	0.6600	0.6452	0.6443

Table 3.6 Hierarchical classification model using SVM. Occupation

	Accuracy	Precision	Recall	F1
One stage (6-grams, 2-words)	0.8188	0.5353	0.5918	0.5531
Two stages (6-grams, 2-words)	0.8312	0.5786	0.5730	0.5648

As we can observe from tables 3.5 and 3.6, the two-stage hierarchical classification managed to improve the coding for the Occupation variable using SVM. One of the tests left to do is to use the assembly method in table 3.4 and combine it with the two-stage hierarchical classification method.

3.2 Final model(s) selected and criteria

Although results in the previous section are encouraging, they do not constitute a definitive model; however, the main metric we will use to take decisions will be Accuracy. Up until now, assembled models have provided the best results; however, for the vectorization processes we used TF-IDF, which can be complemented or substituted with state-of-the-art methods.

3.3 Hardware and software used

A workstation with the following characteristics is used for all processes:

- 20 physical cores
- 256 GB RAM
- 4TB storage
- 8GB AMD graphic card
- Ubuntu 18.04.3 LTS
- ROCM 3.0
- Python 3.6
- Scikit-learn 0.22.1
- Keras 2.3

3.4 Runtime to train the model

Model	Time

SVM	21 min
Logistic regression	3.6 hours
Random forest	12 min
Neural network	4 hours
XGBoost	6 hours
KNN	0.001 min
Gaussian Naïve Bayes	0.60 min
Decision trees	13.7 min
Extra trees	15.5 min

4. Code/programming language

We have attached to this document the Python code.

Variable	*Mnemonic*
Economic Activity	SCIAN
Name of occupation	NOM_OCUP
Tasks or roles	TAREA_OCUP
Economic activity of the company or institution	ACT_EMP
Name of the company	NOM_EMP

5. Evolution of this study inside the organization

Since its conception, this project was conducted in collaboration between INEGI´s Research Department, Department of Standardization of Classifications and Codification Strategies Department, and the academia (CIMAT, A.C.). The Research Department at INEGI and CIMAT developed and applied the ML algorithms, while the Department of Standardization provided, cleaned and normalized the input data.

Due to the resources that the codification process currently implies, and the amount of codification tasks done by INEGI, this project can be used as an input to different areas within INEGI. In the following months, we will complete this report on the first stage and share it among other departments.

6. Is this a proof of concept or is it already used in production?

Due to the results obtained during the test stage, this project will continue in a second stage were state of the art methodologies will be made use of, more data will be employed, and it will be expanded to different scenarios. To do this, it is required a computational infrastructure capable of supporting the processes involved, access to categorized data from different sources, training and the constant follow up of researchers specialized in NLP.

Continuing with this project will allow the development of a process where the ML algorithms are incorporated in a workflow similar to the one adopted by various departments in INEGI tasked with the codification of variables.

6.1 What is doable now that was not before?

With the results obtained, we can simulate the advantage of incorporating this ML algorithm inside the current process. We do not consider the project to be sufficiently developed to be implemented in the daily activities at INEGI.

6.2 Is there already a roadmap/service journey available and how to implement this?

Given the recent finalization of the tests stage, we are currently analyzing the results and their usefulness, and we will be working on the planning of the following stages in the coming months.

6.3 Who are the stakeholders?

The main beneficiaries of this project will be the departments in charge of the coding processes, particularly the department in charge of the codification of household surveys.

6.4 Robustness

We have not worked on this stage yet; however, this project is not intended as a substitute of the current processes, but as a supplement, in such a way that the quality assurance process will be linked to the current codification process.

7. Conclusions and lessons learned

The results obtained allow us to identify potential contributions to the current codification processes, such as the reduction of the staff workload, or the increase of the quality of products; however, we have not yet explored the benefits of state-of-the-art methods, therefore the results obtained could be suboptimal.

By developing this project, we have generated base algorithms for the use of NLP, as well as the necessary computational infrastructure. We have seen text vectorization algorithms in depth and moved forward in the knowledge of state-of-the-art methodologies.

8. Has there been a collaboration with other NSIs, universities, etc.?

Yes, the first stage of the project was carried out along a researcher at CIMAT A.C., which is one of the main research and education institutions in Mexico.

9. Next steps

The project will enter a second stage, were we will include state-of-the-art methodologies and evaluate their usefulness. We will compare the performance of the chosen ML model(s) with the performance of the current codification scheme (for that we will use the upcoming 2020 ENIGH). A group of thematic experts will verify the codes of those cases where ML and the current process differ and we will use this new information as an additional precision metric for the ML model performance.

Annex 1. Basic Data Preparation

Uppercase letters, removed accent and singulars

Original examples	*After the process*
á Á À Â Ä ä Ã	A
* + - _ < > ; , . () ( ) @ ? ^	(blank space)
AUXILIAR (CONTABLE)	AUXILIAR CONTABLE
UNIVERSIDAD PUBLICA(UMSNH)	UNIVERSIDAD PUBLICA MSNH
ES MAESTRO, PRIMARIA,	ES MAESTRO PRIMARIA
ING. MECANICO ELECTRICISTA.	ING MECANICO ELECTRICISTA
H.E.B.	HEB
CASAS	CASA

Annex 2. Synonyms Replacement

Original examples	*After the process*
ENEQUEN	ALGODON
HENEQUEN	ALGODON
NENEKEN	ALGODON
SOSQUI	ALGODON
CRIBABA	CRIBAR
CRIBABAN	CRIBAR
CRIBACION	CRIBAR
CRIBADO	CRIBAR

We have attached to this document the complete version of our synonyms dictionary.