Procesamiento del Lenguaje Natural, Revista nº 69, septiembre de 2022, pp. 73-82 recibido 31-03-2022 revisado 18-05-2022 aceptado 20-05-2022

Multi-label Text Classification for Public Procurement in Spanish

Clasificación multi-etiqueta de textos de licitaciones públicas en español

María Navas-Loro, Daniel Garijo, Oscar Corcho
Ontology Engineering Group, AI.nnovation Space, Universidad Politécnica de Madrid
mnavas@fi.upm.es, daniel.garijo@upm.es, ocorcho@fi.upm.es

Abstract: Public procurement accounts for a 14% of the annual budget of the different governments of the European Union. In Europe, contracting processes are classified using Common Procurement Vocabulary codes (CPVs), a taxonomy designed to facilitate statistical reporting, search and the creation of alerts that can be used by potential bidders. CPVs are commonly assigned manually by public employees in charge of contracting processes. However, CPV classification is not a trivial task, as there are more than 9,000 different CPV categories, which are often assigned following heterogeneous criteria. In this paper we have created a CPV classifier that uses as an input the textual description of the contracting process, and assigns CPVs from the 45 top-level CPV categories. We work only with texts in Spanish, although our approach may be easily extended to other languages. Our results improve the state of the art (10% F1-score improvement) and are available online.
Keywords: CPV, Multi-label Classification, Public Procurement, Hierarchical Classification.

Resumen: Las licitaciones públicas suponen el 14% del presupuesto anual de la Unión Europea. En Europa, los procesos de contratación se clasifican usando la taxonomía Common Procurement Vocabulary (CPVs), diseñada para facilitar la generación de estadísticas, las búsquedas y la creación de alertas que puedan utilizar los posibles licitadores. Los códigos CPV suelen ser asignados manualmente por los empleados públicos encargados del proceso de contratación. Sin embargo, la clasificación de textos de acuerdo con estos códigos no es trivial, pues existen más de 9000 CPVs y no siempre se siguen los mismos criterios para su asignación. En este artículo se propone un clasificador que utiliza como entrada la descripción textual del proceso de contratación, y produce códigos de entre las 45 categorías de CPV más generales de la jerarquía. Trabajamos sólo con textos en español, aunque nuestro enfoque puede extenderse fácilmente a otros idiomas. Los resultados obtenidos superan el estado del arte (10% de mejora en F1), y se encuentran disponibles online.
Palabras clave: CPV, Clasificación Multi-etiqueta, Licitaciones Públicas, Clasificación Jerárquica.

1 Introduction

Public authorities in the European Union spend around 14% of the yearly Gross Domestic Product (around 2 trillion euros) purchasing services, utilities and supplies.[1] Access to this data is crucial for enabling a single digital market in Europe, as well as for accountability and transparency. Hence many governments provide this data in their open data portals as well as in `data.europa.eu`, and a number of platforms have been developed to improve both the efficiency and transparency in public procurement[2] (Soylu et al., 2022).

Common Procurement Vocabulary codes (CPVs)[3] help classify public procurement processes in the European Union across dif-

[1] `https://ec.europa.eu/growth/single-market/public-procurement_en`

[2] `https://opentender.eu/es/about/about-opentender`

[3] `https://simap.ted.europa.eu/web/simap/cpv`

ISSN 1135-5948. DOI 10.26342/2022-69-6

ferent languages. Thanks to CPVs, decision makers can easily explore contracting processes across Europe, and companies from different countries may use them to detect procurement processes of interest, independently of the country of origin.

Each public procurement process must be classified with at least one CPV. However, manual CPV classification presents three main challenges. First, there are thousands of possible codes (more than 9000), some of them with similar purposes, making it difficult for those assigning or curating them to decide which codes better suit a specific process. Second, countries with different official languages and countries with more than one official language, such as Spain or Belgium, often have offers in different languages (e.g., Catalan, Basque, Castilian, etc.). Offices from different regions therefore follow different classification guidelines. Third, CPVs are organized in a hierarchy, and thus annotated at different levels of granularity according to the annotator's or department's criteria. For example, the CPV "Pharmaceutical products" (3360000) shown in Figure 1 is often overgeneralized, instead of using more specific codes that shed more light in the type of purchase. This issue is in fact reflected in the European Union Policy Handbook, where the need of suggesting users to select more specific CPV codes is stressed (European Commission, 2020).

In order to address these issues and ease the assignment of CPV codes to procurement processes, this paper presents an approach to automatically assign high-level CPV codes (i.e., the 45 most general categories) to a procurement process. In this paper, we assume that we have the textual description of the process and that the text is in Spanish. Different methods have been tested to this end, outperforming the previous available results for the Spanish language. We expect this research line will help public procurement practitioners in assigning CPV codes in a more homogeneous manner by providing suggestions that humans can use in their decision process.

The rest of the paper is organized as follows. Section 2 introduces the CPV classification problem in detail, explaining the rationale behind each part of the codes. Section 3 summarizes the related work done in the context of multi-label text classification, as well as existing approaches for CPV classification in Spanish. Section 4 describes how the corpus used to train our classifier was developed, while in Section 5 we outline our approach. Finally, Section 6 details the results obtained by the different classification techniques used, and Section 7 concludes our work.

2 Background

The Common Procurement Vocabulary (CPV) allows classifying public procurement processes with a homogeneous code that represents the need and main object of the requested contract. Several CPV codes may be used to describe a single offer. The format of these CPV codes follows a five-level tree structure comprising the following digits:

- The first two digits identify the divisions (XX000000)
- The first three digits identify the groups (XXX00000)
- The first four digits identify the classes (XXXX0000)
- The first five digits identify the categories (XXXXX000)
- The following three digits give a greater degree of precision within each category (00000XXX)

A ninth check digit serves to verify the previous digits, and has no meaning by itself (00000000-Y).

Therefore, the task of automatically classifying CPVs increases in complexity the more digits we aim to predict. The current official list of CPVs has 9454 possible codes, grouped into 45 different divisions, 317 groups, 1321 classes and 3704 categories. In this paper we focus in classifying CPVs at the division level.

3 Related Work

While text classification has been widely explored in the literature (Aggarwal and Zhai, 2012; Minaee et al., 2021), multi-label classification for the Spanish language has received less attention so far. The main difference between the multi-label text classification case presented in this paper and other popular problems like sentiment analysis is the amount of possible labels. Sentiment analysis labels correspond to certain degrees

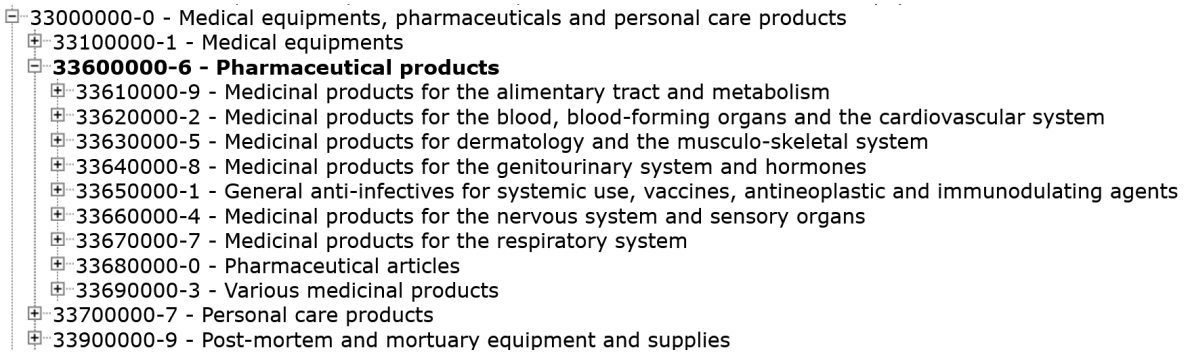

Figure 1: Excerpt of the tree-structure of CPV code 33600000, "Pharmaceutical products", extracted from `http://www.cpv.enem.pl/en/33600000-6`.

of positive and negative emotions, or to a taxonomy of emotions, whilst CPV labels may contain up to thousands of possible options. In order to target this kind of problems, a new subtask has been defined inside multi-label text classification: *extreme multi-label text classification* (XMTC) (Liu et al., 2017).

XMTC addresses the problem of assigning to a document its most relevant subset of class labels from an extremely large label collection (Liu et al., 2017). The work by Gargiulo et al. (2019) analyzes the impact of using different word embedding models in Deep Learning targeting extreme multi-label classification. Their approach uses Convolutional Neural Networks (CNN) to classify 27,775 hierarchical labels in the biomedical domain. Similarly, Liu et al. (2017) compared CNN to other approaches in XMTC, such as KNN-based approaches like SLEEC (Bhatia et al., 2015) or tree-based methods like FastXML (Prabhu and Varma, 2014). Finally, Chang et al. (2020) proposed a scalable framework to fine-tune Deep Transformer models that performed well in different XMTC datasets.

Regarding specific previous work on CPV classification, one of the main results was the multilingual model built by Kaan Görgün.[4] This model categorizes public procurement descriptions in multiple languages among 45 different division labels, with an F1 Score of 0.68. Industrial approaches have also targeted the CPV code classification problem, such as the solution developed by the data science consultancy uData (Deloitte, 2020), using a hierarchical nested approach consisting of one model to predict the first two digits of the CPV code, 50 models to predict the third code (depending on the first model results) and 250 additional models to predict the fourth digit. Other approaches in the literature include a deep learning sequence-processing regression algorithm (also containing several classifiers, considering different aspects of CPVs) (Suta, 2019), or the approach by Ahmia (2020), who used Linear SVMs in order to predict the first two digits of the CPV codes. SVMs were also used in Kayte and Schneider-Kamp (2019). Since the only model available for reuse and evaluation for the Spanish language is the one from Kaan Görgün, we use it as a baseline for comparison against our approach, making both training data and model results available to the community.

4 Creating a Spanish CPV Corpus

We created our training corpus with open data from historical public procurement from the Spanish Treasury's website (Hacienda[5]). We decided to use data from 2019, in order to avoid including later data that may have been influenced by public procurement related to COVID19 pandemics. Procurement processes' metadata were processed from their original format (Atom Syndication Format[6]) using different scripts available in our paper repository (Navas-Loro, Garijo, and Corcho, 2022).[7] Document pre-processing included the following stages:

1. **Information extraction** from all the

[4] `https://huggingface.co/MKaan/multilingual-cpv-sector-classifier`

[5] `https://www.hacienda.gob.es/es-ES/GobiernoAbierto/Datos%20Abiertos/Paginas/LicitacionesContratante.aspx`

[6] `https://www.w3.org/2005/Atom`

[7] `https://github.com/oeg-upm/cpv-classifier`

information contained in the Atom documents. We only retrieved the textual description of the offers and the different CPV codes assigned to them. This is represented as a CSV file in order to ease its further processing.

2. **Duplicate deletion** and trim of the descriptions. Additionally, we only keep texts in Spanish (to this aim we used fastText's language identification functionality[8]).
3. **Train/test dataset division**, in order to make the dataset more manageable, we split it into train and test sets (70/30) before uploading it to our public code repository.
4. **In-code preprocessing**. An additional set of scripts were used to remove rows with no CPV code assigned and generalize CPV codes to the division level, which is the one we use in our experiments.

The result of the first two steps are two csv files, available in our repository. The code used for all processing scripts can also be found in the same location. Figure 2 shows the distribution for each of the 45 division labels, which are clearly unbalanced. The most frequent label ('45', that represents the division 'works') is present in 16128 instances of the the training set, while label '76' is only present in 13 instances.

5 Approach

We addressed CPV classification in a hierarchical manner: instead of creating a classifier for nine thousand labels, we took advantage of the hierarchical structure of the CPVs and created a classifier for the 45 available divisions (first two digits). We believe this to be a good first step due to the training data available for most categories.

The only model openly available to perform this task is the model from Kaan Görgün (from now, MKaan) mentioned in the Related Work section. This model also targeted just the first two digits of the CPV code, so we use it as a baseline to compare the different approaches we have tested.

In order to perform multi-label classification, several approaches can be used. We can use algorithms adapted to the task, such as decision trees or random forests, or we can also use binary classifiers like Naïve Bayes or SVM and then apply different strategies so that they serve for multi-label classification. Another option is to fine-tune existing transformers, as done in the approach by MKaan. We briefly present below the different approaches we tested.

5.1 Classical Techniques

We tried the following classifiers:

Naïve Bayes (Minsky, 1961) has been widely used for text classification (İşgüder-Şahin, Zafer, and Adah, 2014), specially for sentiment analysis and SPAM classification. Although this algorithm relies on probability independence, it works very well even when this assumption is not met.

SVM Support Vector Machines (SVM) (Boser, Guyon, and Vapnik, 1992) are linear classifiers that define an hyperplane in order to discriminate among classes. SVM have been frequently used for multiclass classification tasks.

SVM with RBF kernel Besides testing the linear version of SVM, we also evaluated the performance of an SVM with the Radial Basis Function as kernel, that is:

$$rbf_\gamma = e^{-\gamma\|x-x'\|^2} \quad (1)$$

with parameter $\gamma \geq 0$.

Decision Trees (Quinlan, 1986) are an intuitive way to classify instances. In our implementation we used the sklearn optimized version of the CART algorithm.[9]

Random Forests (Breiman, 2001) are a tree-based ensemble approach to classification that overcomes most of the problems with decision trees, such as high variance. Due to this robustness they have been frequently used for Extreme Multi-label Classification (Siblini, Kuntz, and Meyer, 2018).

K-Nearest Neighbours (K-NN) (Hand, 2007) is widely used for multi-label classification (Zhang and Zhou, 2007). The idea behind K-NN is to check the K labeled instances that are the closest to the new instance and classify it with the most common label from these neighbours.

[8] https://fasttext.cc/docs/en/language-identification.html

[9] https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

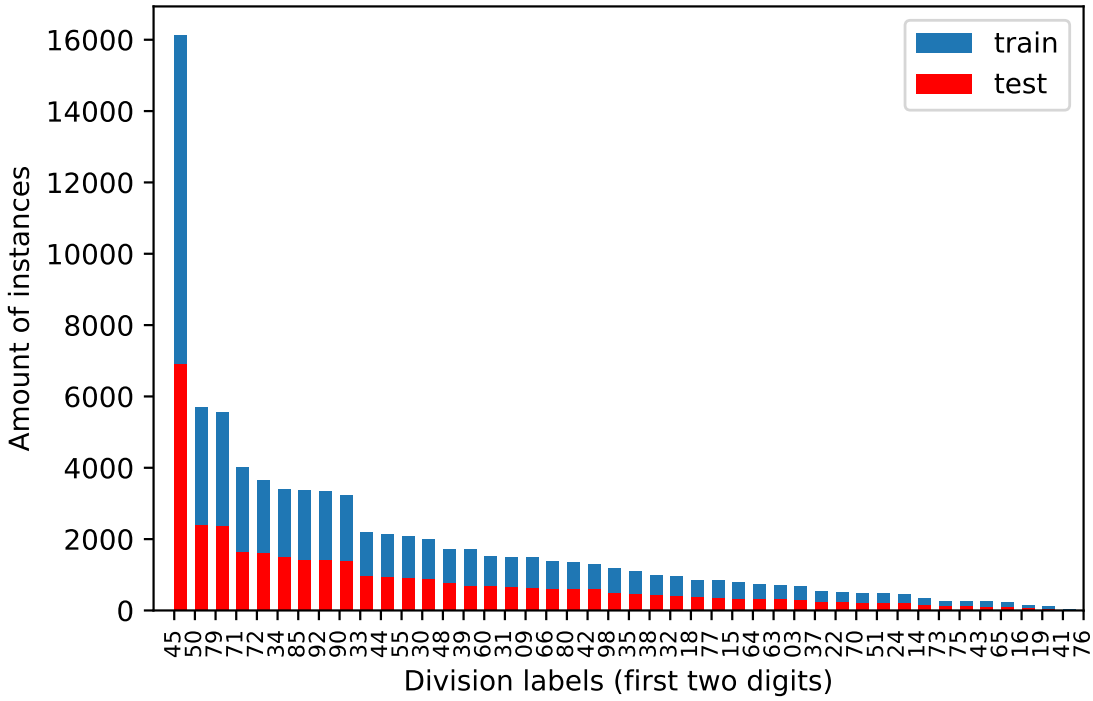

Figure 2: Bars (y axis) represent the amount of instances per division label (x axis). Blue bars represents the amount of labels in the training set, while red bars represent the number of instances in the evaluation set.

AdaBoost (Freund and Schapire, 1997) is a meta-estimator that fits different versions of models using boosting (i.e., different versions of the training dataset). We used the implementation defined in Hastie et al. (2009): AdaBoost-SAMME.

For all these approaches we used the Term Frequency - Inverse Document Frequency (TF-IDF) technique for vectorization, allowing n-grams with $n = 3$. For those algorithms that do not support multi-label classification, we decided to use the One-vs-the-rest (OvR) or One-vs-all strategy, frequently used for multiclass classification, where one binary classifier per label is built in order to decide if an instance should be classified with that label or not.

5.2 RoBERTa fine-tuned approach

In addition to the aforementioned classical approaches, we also decided to fine-tune a transformed-based model for the Spanish language, namely *RoBERTa-base-bne* (Gutiérrez-Fandiño et al., 2021), on a dataset derived from Spanish Public Procurement documents from 2019.

RoBERTa-base-bne is a transformer-based masked language model based on the RoBERTa model and pre-trained using the largest Spanish corpus known to date (570GB), compiled from the annual web crawlings performed by the National Library of Spain (Biblioteca Nacional de España) from 2009 to 2019.[10]

Table 1 summarizes the hyperparameters used in the fine-tuning process, performed using the HuggingFace transformers library. The whole training process can be reproduced using the notebook '`fine-tuned-roberta-for-spanish-cpv-codes.ipynb`' in our code repository.

6 *Evaluation*

This section describes how we evaluated the results obtained with the different approaches, and discusses them.

6.1 Metrics

We use two sets of metrics in our evaluation. First, we use *general* metrics such as the Area Under the ROC Curve (ROC AUC), F1-score and accuracy. Second, we use multi-label specific metrics, i.e., coverage error and

[10] `https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne`

Parameter	Value
learning rate	$2 * 10^{-5}$
train batch size	8
eval batch size	8
seed	42
optimizer	adam
epochs	10

Table 1: Summary of the hyperparameters used for training the RoBERTa fine-tuned model used in our analysis.

Label Ranking Average Precision. We briefly describe all these metrics below.

6.1.1 General Metrics

The metrics used that are not specific to multi-label classification are the following:

Area Under the ROC Curve (AUC): measures the capability of a classifier to distinguish between classes. The higher the AUC, the better the model can make the distinction among classes.

F1-score: harmonic mean between precision and recall, widely adopted to monitor both metrics at the same time.

Accuracy: fraction of predictions that the model classified correctly.

6.1.2 Coverage Error

The coverage error computes the average number of labels that have to be included in the final prediction such that all true labels are predicted. That is, the average amount of ranked labels to take into account to miss no true label.

$$coverage(y, \hat{f}) = \frac{1}{n_s} \sum_{i=0}^{n_s-1} \max_{j:y_{ij}=1} rank_{ij} \quad (2)$$

with n_l being the amount of labels, n_s being the amount of samples, $\hat{f} \in R^{n_s \times n_l}$ the score associated with each label, $y \in \{0,1\}^{n_s \times n_l}$ the ground truth labels, $rank_{ij} = \left\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\right\}$.

6.1.3 Label Ranking Average Precision

Label Ranking Average Precision (LRAP) averages over the ground truth labels assigned to each sample, ranking true labels higher. This metric shows which ratio of higher-ranked labels were true labels.

$$LRAP(y, \hat{f}) = \frac{1}{n_s} \sum_{i=0}^{n_s-1} \frac{1}{||y_i||_0} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{rank_{ij}} \quad (3)$$

with n_l being the amount of labels, n_s being the amount of samples, $\hat{f} \in R^{n_s \times n_l}$ the score associated with each label, $y \in \{0,1\}^{n_s \times n_l}$ the ground truth labels, $rank_{ij} = \left\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\right\}$, $\mathcal{L}_{ij} = \left\{k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\right\}$, $||\cdot||_0$ being the ℓ_0 norm (which computes the amount of nonzero elements in a vector), and $|\cdot|$ representing the cardinality of the set.

6.2 Results and Discussion

We compare our results against the model by MKaan, since it is the only available model that we have been able to find targeting the CPV code assignment problem in Spanish (besides other languages). Since no default threshold or function is provided, we tested different thresholds with the most common functions (softmax and sigmoid). Results are summarized in Table 2 (using only 10% of the dataset), and Table 3 (using the whole dataset).

The results clearly show that the RoBERTa fine-tuned model outperforms the rest of the approaches both when training using just a fraction of the dataset and the full dataset. The model by MKann shows a good performance taking into account its multilingual nature (not specific for the Spanish language). However, MKaan is matched and even outperformed by some of the traditional algorithms in both experiments.

In particular, classical approaches such as SVM, random forests and decision trees, produce remarkably good results (0.69, 0.64 and 0.63 F1 scores respectively on the full dataset). Given that these algorithms are usually less expensive to train, test and use than transformer-based solutions, they are reasonable candidates for assisting in CPV classification at a low cost. One possible explanation for this good performance is that, despite the presence of polysemous words that can be problematic, both the hyperplanes of SVM and the decisions of tree-based methods allow to effectively discriminate each label against all others (that is the strategy usually used to adapt the algorithms

Approach	ROC-AUC	F1	Accuracy	LRAP	Cov. Error
Multinomial NB	0.53	0.11	0.06	0.09	42.32
SVM	0.66	0.47	0.33	0.36	30.19
SVM (rbf)	0.66	0.47	0.33	0.36	30.19
KNN	0.70	0.54	0.41	0.45	26.54
Decision Tree	0.74	0.51	0.49	0.53	22.74
Random Forest	0.68	0.52	0.39	0.41	27.96
AdaBoost	0.75	0.56	0.41	0.49	22.10
RoBERTa fine-tuned (t=0.5)	0.84	**0.74**	**0.68**	**0.73**	14.13
RoBERTa fine-tuned (t=0.6)	0.83	0.73	0.67	0.71	14.86
RoBERTa fine-tuned (t=0.65)	0.82	0.73	0.67	0.70	15.41
RoBERTa fine-tuned (t=0.7)	0.81	0.72	0.64	0.68	16.54
MKaan (sigmoid, t=0.5)	0.80	0.13	0.0	0.07	17.38
MKaan (sigmoid, t=0.7)	0.85	0.19	0.0	0.11	13.31
MKaan (sigmoid, t=0.8)	0.86	0.24	0.0	0.15	12.21
MKaan (sigmoid, t=0.9)	0.87	0.32	0.01	0.23	11.49
MKaan (sigmoid, t=0.95)	0.87	0.42	0.06	0.34	11.64
MKaan (softmax, t=0.01)	**0.88**	0.37	0.25	0.44	**11.05**
MKaan (softmax, t=0.05)	0.86	0.55	0.43	0.59	12.48
MKaan (softmax, t=0.1)	0.85	0.61	0.51	0.64	13.64
MKaan (softmax, t=0.3)	0.81	0.65	0.61	0.66	16.63
MKaan (softmax, t=0.5)	0.79	0.65	0.60	0.63	18.71

Table 2: Results of the different approaches trained and tested on the 10% of the dataset (7243 training samples, 3104 test samples).

Approach	ROC-AUC	F1	Accuracy	LRAP	Cov. Error
Multinomial NB	0.56	0.22	0.14	0.16	39.07
SVM	0.78	0.69	0.58	0.62	18.89
SVM (rbf)	0.78	0.69	0.58	0.62	18.89
KNN	0.75	0.62	0.52	0.56	21.68
Decision Tree	0.80	0.63	0.60	0.64	17.68
Random Forest	0.74	0.64	0.51	0.54	22.32
AdaBoost	0.75	0.60	0.45	0.51	22.47
RoBERTa fine-tuned (t=0.5)	**0.89**	0.79	**0.74**	**0.80**	**10.32**
RoBERTa fine-tuned (t=0.6)	0.88	**0.80**	**0.74**	**0.80**	10.66
RoBERTa fine-tuned (t=0.65)	0.88	0.79	**0.74**	0.79	10.95
RoBERTa fine-tuned (t=0.7)	0.88	0.79	**0.74**	0.79	10.94
MKaan (sigmoid, t=0.5)	0.81	0.13	0.0	0.07	17.19
MKaan (sigmoid, t=0.7)	0.86	0.19	0.0	0.11	13.01
MKaan (sigmoid, t=0.8)	0.87	0.24	0.0	0.15	11.91
MKaan (sigmoid, t=0.9)	0.87	0.33	0.01	0.23	11.32
MKaan (sigmoid, t=0.95)	0.87	0.42	0.06	0.34	11.50
MKaan (softmax, t=0.01)	0.88	0.38	0.24	0.44	10.74
MKaan (softmax, t=0.05)	0.86	0.55	0.43	0.59	12.25
MKaan (softmax, t=0.1)	0.85	0.61	0.50	0.63	13.54
MKaan (softmax, t=0.3)	0.81	0.66	0.61	0.66	16.46
MKaan (softmax, t=0.5)	0.79	0.66	0.60	0.63	18.62

Table 3: Results of the different approaches trained and tested on the whole dataset (72429 training samples, 31042 test samples).

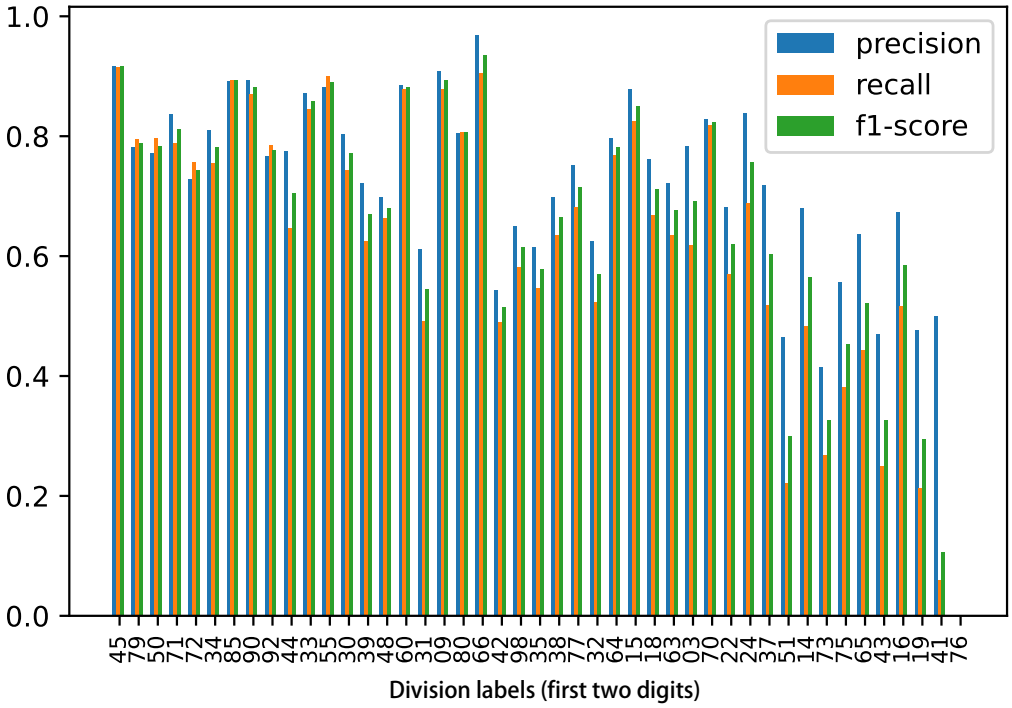

Figure 3: Results of the RoBERTa fine-tuned model (t=0.5) per label. We preserve the order presented in Figure 2, from more represented labels ('45') to less represented labels ('76').

to multiclass problems).

A limitation of our approach is the lack of measures for balancing input data. Typically, this would risk having our CPV classifier performing well only for the classes with more representation. However, as shown in Figure 3, our CPV classifier shows an excellent performance for most categories, and has an acceptable performance for classes with less data available (except fpr extremely rare categories '41' and '76'). We suspect that in addition to the number of training instances, the generality of the divisions and the overlap between them also play a role in the differences in performance. For example, divisions '42' and '43' represent "Industrial machinery" and "Machinery for mining, quarrying, construction equipment", respectively. Words similar to "machinery" will therefore appear frequently in descriptions of both divisions, leading to false positives/negatives. In Figure 3, we can in fact confirm that both divisions have worse performance than the immediate surrounding divisions having a similar amount of instances.

7 Conclusions and Future Work

This paper presents an approach to classify CPV code divisions for Spanish public procurement descriptions. Our work evaluated classical machine learning algorithms, showing that SVM had an excellent performance, surpassing the previous existing transformed-based approach for the task. Additionally, we fine-tuned the RoBERTa transformed-based model trained on a corpus of the BNE (Spanish National Library), that outperformed all the previous approaches. All data, data processing scripts and training notebooks have been made available through a public code repository, Zenodo (Navas-Loro, Garijo, and Corcho, 2022)[11] and a Research Object[12] for the sake of reproducibility. This material is also planned to be used in the AI4Gov international master.[13]

Our approach covers only CPV division classification, and therefore it does not yet address the CPV over-generalization problem when assigning CPVs to text (i.e., some codes

[11] https://zenodo.org/record/6554843
[12] https://w3id.org/dgarijo/ro/sepln2022
[13] https://ai4gov-master.eu/

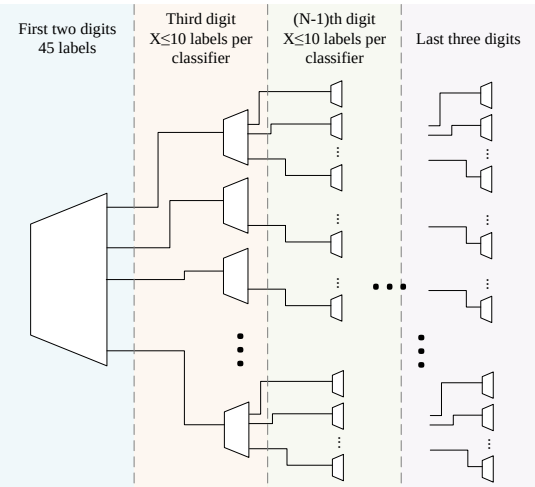

Figure 4: Hierarchical approach to the CPV classification problem. The first classifier would be responsible for categorizing the first two digits of the code, i.e., its division. The next level would attempt to predict the next digit based on the previous digits. For example, if the first classifier determined that a description corresponds to the labels '45' and '48', that description would be passed to the classifiers that determine the next digit trained with examples of those two codes.

are systematically not used in preference to more generic codes, even though the specific codes in disuse are much better suited to the topic of the description). Our future work includes designing a sequence of models that successively classify the digits of CPVs, as depicted in Figure 4, to be able to predict more specific CPVs. Alternatively, we plan on assessing techniques based on sentence embeddings against CPV descriptions, in order to suggest more specific CPVs despite the lack of training instances. Designing more specific classifiers will also require dealing with noise in data, e.g., when annotators assign different CPVs to the same contract description or incorrect CPVs. We also plan to increase the dataset, including contracting information from several years and also retrieving and making use of additional information from contracting processes. These include features such as the cost, that could help in the disambiguation of general words such as "service" or "work", that can be used in very different situations. Additionally, we will also enhance the preprocessing of the data in order to improve the quality in the dataset, a well-known problem in this kind of classification problem.

Overall, our positive results are a step forward towards the creation of a decision support system to help in CPV classification, allowing a more transparent and efficient public procurement in Spain and Europe.

Aknowledgments

This work has been supported by NextProcurement European Action (grant agreement INEA/CEF/ICT/A2020/2373713-Action 2020-ES-IA-0255) and the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation). We also acknowledge the participation of Jennifer Tabita for the preparation of the initial set of notebooks, and the AI4Gov master students from the first cohort for their validation of the approach. Source of the data: Ministerio de Hacienda.

References

Aggarwal, C. C. and C. Zhai. 2012. A survey of text classification algorithms. In *Mining text data*. Springer, pages 163–222.

Ahmia, O. 2020. *Assisted strategic monitoring on call for tender databases using natural language processing, text mining and deep learning*. Ph.D. thesis, Université de Bretagne Sud, 03.

Bhatia, K., H. Jain, P. Kar, M. Varma, and P. Jain. 2015. Sparse local embeddings for extreme multi-label classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Boser, B. E., I. M. Guyon, and V. N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Breiman, L. 2001. Random forests. *Machine learning*, 45(1):5–32.

Chang, W.-C., H.-F. Yu, K. Zhong, Y. Yang, and I. S. Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3163–3171.

Deloitte. 2020. Study on up-take of emerging technologies in public procurement. Technical report, Deloitte.

European Commission. 2020. *eForms : policy implementation handbook*. Publications Office.

Freund, Y. and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Gargiulo, F., S. Silvestri, M. Ciampi, and G. De Pietro. 2019. Deep neural network for hierarchical extreme multi-label text classification. *Applied Soft Computing*, 79:125–138.

Gutiérrez-Fandiño, A., J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, A. Gonzalez-Agirre, C. Armentano-Oller, C. R. Penagos, and M. Villegas. 2021. Spanish language models. *CoRR*, abs/2107.07253.

Hand, D. J. 2007. Principles of data mining. *Drug safety*, 30(7):621–622.

Hastie, T., S. Rosset, J. Zhu, and H. Zou. 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360.

İşgüder-Şahin, G. G., H. R. Zafer, and E. Adah. 2014. Polarity detection of turkish comments on technology companies. In *2014 International Conference on Asian Language Processing (IALP)*, pages 136–139. IEEE.

Kayte, S. and P. Schneider-Kamp. 2019. A mixed neural network and support vector machine model for tender creation in the european union ted database. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 139–145. INSTICC, SciTePress.

Liu, J., W.-C. Chang, Y. Wu, and Y. Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 115–124.

Minaee, S., N. Kalchbrenner, E. Cambria, et al. 2021. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), apr.

Minsky, M. 1961. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

Navas-Loro, M., D. Garijo, and O. Corcho. 2022. Code repository for multi-label text classification for public procurement in spanish, May.

Prabhu, Y. and M. Varma. 2014. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 263–272.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning*, 1(1):81–106.

Siblini, W., P. Kuntz, and F. Meyer. 2018. CRAFTML, an efficient clustering-based random forest for extreme multi-label learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4664–4673. PMLR.

Soylu, A., Corcho, B. Elvesæter, C. Badenes-Olmedo, F. Yedro-Martínez, et al. 2022. Data quality barriers for transparency in public procurement. *Information*, 13(2).

Suta, A. 2019. Multilabel text classification of public procurements using deep learning intent detection. Master's thesis, KTH, Mathematical Statistics.

Zhang, M.-L. and Z.-H. Zhou. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048.