IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 4, 2020, accepted May 12, 2020, date of publication May 25, 2020, date of current version June 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997241

ICD-10 Coding of Spanish Electronic Discharge Summaries: An Extreme Classification Problem

MARIO ALMAGRO[1], RAQUEL MARTÍNEZ UNANUE[1], VÍCTOR FRESNO[1], AND SOTO MONTALVO[2]

[1]Department of Computer Languages and Systems, National University of Distance Education (UNED), 28040 Madrid, Spain
[2]Department of Computer Science, King Juan Carlos University (URJC), 28933 Madrid, Spain

Corresponding author: Mario Almagro (malmagro@lsi.uned.es)

This work was supported by the Spanish Ministry of Science and Innovation through the MAMTRA-MED Project, AEI/FEDER, UE, under Grant TIN2016-77820-C3-2-R.

ABSTRACT Objective: Medical coding is used to identify and standardize clinical concepts in the records collected from healthcare services. The tenth revision of the International Classification of Diseases (ICD-10) is the most widely-used coding with more than 11,000 different diagnoses, affecting research, reporting, and funding. Unfortunately, ICD-10 code sets tend to follow biased, unbalanced, and scattered distributions. These distribution attributes, along with high lexical variability, severely restrict performance when coded clinical records are used to infer code sets in uncoded records. To improve that inference, we explore a combination of example-based methods optimized to capture codes with different appearance frequencies in data sets. Materials and Methods: The proposed exploration has been carried out on Spanish hospital discharge reports coded by experts, excluding all sentences without any biomedical concept. Representations based on semantic and lexical features are explored, using both global and label-specific attributes. In turn, algorithms based on binary outputs, groups of subsets and extreme classification are compared. Lists of codes together with their confidence values (certainty probabilities) are suggested by each method. Results: Diverse spectral behaviors are shown for each method. Binary classifiers seem to maximize the capture of more popular codes, while extreme classifiers promote infrequent ones. In order to exploit such differences, ensemble approaches are proposed by weighting every output code according to the method, confidence value and appearance frequency. The rule-based combination reaches a 46% Precision at 10 ($P@10$), which means a 15% improvement over the best individual proposal. Conclusion: Assembling methods based on weighting each code according to training frequency and performance can achieve better overall Precision scores on extreme distributions, such as ICD-10 coding.

INDEX TERMS Extreme classification, XMTC, ICD-10 coding, text mining.

I. INTRODUCTION

Most information coming from healthcare services remains unstructured, preventing direct, and easy interpretation of clinical data. The standardization of medical concepts in Electronic Health Records (EHRs) is a necessary preliminary step for deeper analysis.

ICD is a clinical cataloging system that enables statistical analyses of morbidity and mortality by defining more than 11,000 diseases, abnormal findings, complaints, social circumstances, external causes of injury, signs, and symptoms. The tenth revision (ICD-10) is one of the main blocks in the clinical information analysis workflow as it is increasingly used for reporting causes of death and clinical research, audits and funding. ICD-10 is structured in chapters grouping codes of 3 and 4 characters in length. The Spanish version (CIE-10-ES[1]) extends the specificity of the hierarchical structure with 7-character codes, increasing the amount to approximately 69,000 diagnoses and 72,000 procedures (notice that ICD-10 does not contain procedures). In particular, CIE-10-ES codes are organized in three-character categories and can, in turn, belong to different nested subcategories. Final CIE-10-ES codes can consist of 3 to 7 characters, depending on the specificity of the diagnosis or procedure. More general and shorter codes are assigned when there is a lack of information and longer ones are given in association

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

[1] https://eciemaps.mscbs.gob.es/ecieMaps/

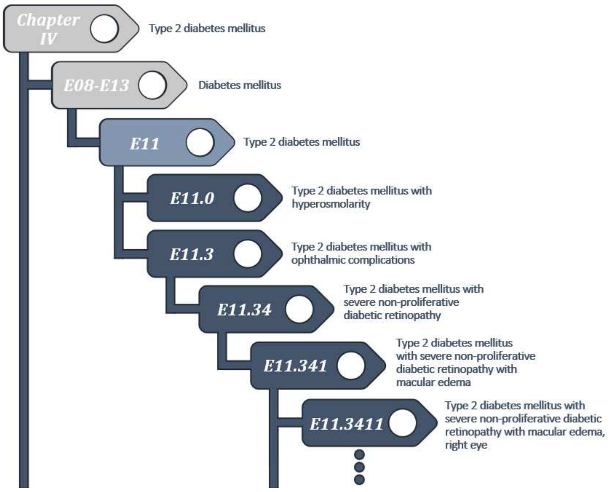
Chapter IV
Type 2 diabetes mellitus
E08-E13
Diabetes mellitus
E11
Type 2 diabetes mellitus
E11.0
Type 2 diabetes mellitus with hyperosmolarity
E11.3
Type 2 diabetes mellitus with ophthalmic complications
E11.34
Type 2 diabetes mellitus with severe non-proliferative diabetic retinopathy
E11.341
Type 2 diabetes mellitus with severe non-proliferative diabetic retinopathy with macular edema
E11.3411
Type 2 diabetes mellitus with severe non-proliferative diabetic retinopathy with macular edema, right eye

FIGURE 1. **Example of hierarchy in the structure of CIE-10-ES codes.**

with more detail. For example, Figure 1 shows the connection among several codes of the same family, *Type 2 diabetes mellitus*.

CIE-10-ES coding entails great difficulty. Certain diseases are much more frequent than others resulting in collections of hundreds of very popular codes and thousands of infrequent codes. Therefore, the prevalence of diagnoses and procedures leads to extremely unbalanced data sets. The tens of thousands of rare CIE-10-ES codes from known data entails a large sparsity in the final distribution. Significant biases are also common in data sets as a consequence of the strong dependency on local factors, such as environmental conditions, lifestyles or clinical services offered. Given bias, imbalance and sparsity as the main attributes, code sets tend to follow an exponential rather than a uniform distribution. Besides, the task is carried out at the document level, and although each record contains lexical expressions that could locally be associated with some code, disseminated information is required to propose the final codes. Thus, it can be considered a multi-label classification of one-to-many, with more than 140,000 possibilities. Finally, code descriptions are designed to aggregate multiple clinical concepts, thus employing more abstract language and general terminology. This means that more different lexical forms can be associated with the same code.

The combination of the rich diversity of lexical forms and the existence of an enormous quantity of codes with only a few examples severely complicates the attainment of high-quality automatic outputs. For this reason, even though the automation is a key priority in most health institutions, coding is performed with human intervention, involving considerable financial resources.

None of the state-of-the-art ICD-10 coding approaches deals effectively with the constrains imposed by extreme distributions. For this reason, the task is addressed as an eXtreme Multi-label Text Classification (XMTC) problem in this paper, focusing on the frequency of the codes to be inferred.

The purpose of automatic coding is to support coders by generating a list of possible candidate codes. Data distribution favors the prediction of the most common codes, which provide less information to coders. So, one of the main challenges is to exploit that distribution avoiding inferring only frequent codes while promoting the assignment of infrequent ones. To this end, we explore multiple methods and their different behaviors according to the number of code instances. As far as we know, XMTC algorithms have never been used in CIE-10-ES classification problems and we think they could significantly contribute to the inference of rare codes.

As a result, combinations of methods are proposed to improve the assignment of codes in different frequency ranges. The goal is to maximize each contribution in terms of Precision improvements.

II. RELATED WORK

There are multiple proposals for addressing the automatic ICD-10 coding to assist coders. Most of them have focused on alleviating lexical variability, while a few have tried to reduce the imbalance effect. Next, extreme classification algorithms are introduced following this latest trend.

A. ICD-10 CODING

The large amount of ICD codes and the different hospital record instances associated with each one is the main issue as it requires an unavailable volume of coded data. On this basis, different ways to try to capture more instances have been proposed in the state-of-the-art.

The most widespread way is to handle the high lexical variability through external knowledge bases. For example, some authors have explored lexical similarities by enriching the representation through dictionaries [1]–[3]. In a similar way, other proposals have used documents as queries, applying the expansion with ontologies [4]–[6]. Following this tendency, repositories of medical terminology have been explored to improve the representation of documents before applying machine learning [7].

As an alternative to biomedical dictionaries, other authors choose to reduce bias and extend collections by transforming other data sets. Subotin *et al.* use the General Equivalence Mappings (GEMS[2]) between ICD-10 and ICD-9 to supplement the small size of the training corpus through reports annotated with ICD-9 [8]. In turn, Almagro *et al.* explore the application of Machine Translation techniques to expand the data set with foreign resources [9].

Another way to deal with lexical variability is to work directly with meanings. In this line, Chen *et al.* have explored the Longest Common Subsequence (LCS) of concepts as a feature for the classification [10], and Ning *et al.* have exploited the hierarchical structure using a distributional

[2] https://www.asco.org/practice-guidelines/billing-coding-reporting/icd-10/general-equivalence-mappings-gems

semantic [11]. Other approaches have applied neural networks fed with word embeddings trained on external corpora [12], [13]. Following this line, Amin *et al.* use BERT pre-trained on PubMed and exploit the information provided by a language model in the clinical domain to represent medical concepts [14].

In addition to reducing variability, some authors have focused on harnessing the ICD-10 hierarchy to reduce imbalance, grouping features of similar codes [15], [16] or avoiding the assignment of multiple similar ones [17]. Furthermore, generative rather than discriminative models, such as Latent Dirichlet Allocation (LDA), have been used in the extraction of topics as features for binary classifiers [18].

As for the Spanish version of the ICD-10, there are few publications. Almagro *et al.* conduct a preliminary study on the application of supervised and unsupervised methods in CIE-10-ES coding [19]. In turn, Blanco *et al.* explore how considering different numbers of codes during training affects deep learning algorithms [20]. Recently, Pérez *et al.* presented an approach based on the extraction of topic models using Latent Dirichlet Allocation (LDA). Subsequently, it uses topics as features for applying binary classifiers. The authors obtain positive results, but considering only the 124 most frequent CIE-10-ES codes.

B. EXTREME CLASSIFICATION

So far, ICD-10 coding has not been addressed as an extreme classification problem. However, the high data sparsity associated with very biased and unbalanced data sets fits perfectly into that research area. XMTC deals with extreme distributions by using sublinear algorithms to assign each document the most relevant subset of labels from a large space of categories. Most approaches fall into three main families: decision tree-based, embedding-based, and deep learning-based methods.

Decision tree-based methods [21]–[23] start with the whole label space and learn a hierarchy from training data by determining which labels should be assigned to the left or right child node. Then, nodes are recursively partitioned until each leaf contains a small number of labels. Each leaf node supplies a binary base classifier for only dealing with two subsets of labels. The most representative method in this family is FastXML [24]. It learns the hierarchical structure of label subsets from training instances and optimizes an NDCG-based objective at each node of the hierarchy. The goal is to have all the documents in each subset sharing similar label distribution.

The embedding-based methods try to make the training and prediction tractable by assuming low-rank training label matrix. For this purpose, those methods linearly transform the high-dimensional label vectors into low-dimensional ones reducing the effective number of labels [25]–[28]. Among these type of methods SLEEC [29] is the most representative as it achieves significant improved accuracy on some benchmark data sets, being computationally efficient. Its architecture works in two steps: learning embeddings and using k-nearest neighbor (kNN) classifiers. It learns $\hat{L}$-dimensional embeddings from the original L-dimensional label vectors that non-linearly capture label correlations. At prediction time, the approach performs a kNN search for projecting a novel document in the $\hat{L}$-dimensional embedding space.

As regards extreme deep learning methods, the main idea is to design new approaches by focusing on the multi-label task. Zhang *et al.* recently proposed a deep embedding method, DXML [11], non-linearity modeling the feature space and label graph structure in a XMTC context. On the other hand, Liu *et al.* present a new Convolutional Neural Network (CNN) model tailored for XMTC problems [30]. This approach, XML-CNN, uses a dynamic max pooling scheme that captures richer information from different regions of the documents as well as for reducing model size. It obtains encouraging results in well-known XMTC benchmark data sets, improving in many cases to FastXML, in most cases to SLEEC, and in all cases to other CNN models.

Other approaches which focused on multi-label classification of very unbalanced data sets have not been cataloged as XMTC. In particular, Rubin *et al.* design a text classifier that takes advantage of LDA principles to model dependencies between labels [31]. No XMTC model always achieves the best result when evaluating with multiple corpora but some models seem to consistently obtain good results in every different benchmark data sets, such as Amazon [32], Wiki10 [33], and EURLex [34].

III. MATERIALS AND METHODS

A. DATA SET

Our entire collection consists of 7,254 Spanish hospital discharged reports collected in Hospital Universitario Fundación Alcorcón for the years 2016-2018. This data set has restrictions of use due to the European General Data Protection Regulation (EU GDPR), so it cannot be made public for the research community, even if anonymized. In total, 76,525 CIE-10-ES codes were identified by coders, approximately 7,000 different ones. That code set follows the above-mentioned distribution, as can be seen in Figure 2. The graph shows how the number of different ICD codes (on the y-axis) varies depending on the number of documents in which it appears (on the x-axis) on a logarithmic scale. The higher the frequency in documents, the fewer the number of different CIE-10-ES codes.

Regarding the textual content, documents are written in natural language, which includes different types of information such as clinical judgment, diagnosis, history, results of clinical trials, and treatments. In general, these reports contain a great deal of data, with an average length of 4,000 words, so it is necessary to select the relevant information. The evaluation will be carried out on the 10 most reliable codes as 10.55 is the average number of CIE-10-ES codes per document. Tables 1 and 2 summarize statistical values for describing the collection.

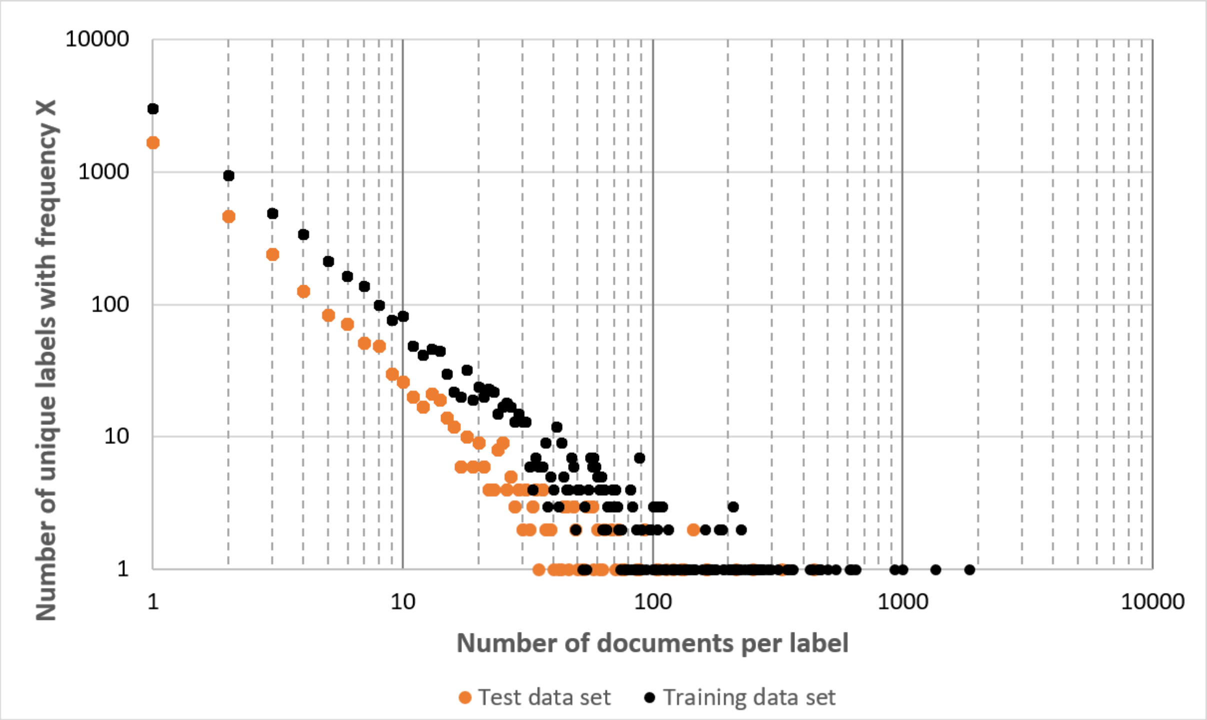

FIGURE 2. Distribution of the Spanish hospital discharged reports data set. The training data set is in black and the test data set is in orange. How the number of different ICD codes (on the y-axis) varies depending on the number of documents in which it appears (on the x-axis) is plotted on logarithmic scale.

Label dimensionality refers to the number of different codes available in the data set.

B. TEXT REPRESENTATIONS

As mentioned above, discharge reports are long text documents with a considerable amount of clinical information. Hence, it seems necessary to apply a preprocessing step in order to discard the information not relevant to the ICD-10 coding task. The IxaMedTagger[3] tool [35] has been used for the selection of sentences. This is a Spanish clinical part-of-speech tagging software that uses the SNOMED CT terminology [4] to identify body structures, qualifiers, medicines, allergies, and diseases. It has been assumed that all sentences without any of those entities are not relevant for coding, therefore they have been discarded. The content of each discharged report used in the classification is made by grouping together all sentences in which clinical entities were detected. Finally, the replacement of capital letters and accented characters, the removal of punctuation marks, and a stemming process have been carried out. No specific negation detection has been used.

Different features have been extracted from those summaries to feed the various methods. Table 3 shows the dimensions of the features. Both global and label-specific lexical features have been explored, using word N-grams. Bags of Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF) have been applied to represent attributes for all codes. In turn, Term Frequency - Bi-Normal Separation (TF-BNS) is used to characterize each code with particular features as Forman proposes [36]. At the same time, semantic features have been explored with word embeddings. Spanish clinical word embeddings have been generated using the fastText approach proposed by Bojanowski *et al.* [37]. The Spanish Billion Word Corpus,[5] more than 150,000 uncoded hospital records and thousands of medical PhD dissertations have been used for the transfer learning process.

C. METHODS

As mentioned, ICD-10 coding is a extreme multi-label task, where a document is associated with a subset of codes. To deal with multiple outputs, some simplifications have typically been made, such as assuming independence between labels and considering whole subsets as the only possibilities. Alternatively, other algorithms directly adapted to multi-label outputs have been used, such as k-nearest neighbors, decision trees, and neural networks. In particular, XMTC algorithms appear to extend those by reducing the imbalance effects. In this paper we explore and compare approaches based on each of these foundations to exploit the overlapping and dissimilarity between methods within the context of ICD coding.

[3] http://ixa2.si.ehu.eus/prosamed/resources

[4] SNOMED CT (http://www.snomed.org/) is a clinical terminology included in the Unified Medical Language System (UMLS)

[5] https://crscardellino.github.io/SBWCE/

TABLE 1. Data set features.

Feature	Value
CIE-10-ES Codes	76,525
Label Dimensionality	7,078
Number of Training Documents	5,803
Number of Test Documents	1,451

TABLE 2. Data set statistics.

Attribute	Average	Median
Documents per Label	9.56	2
Labels per Document	10.55	9
Words per Document	4,060	2,939

Codes can be processed separately by ignoring the dependencies between them. In this way, one classifier for each code can be defined following a One-vs-Rest (OvR) strategy, which produces a binary output representing presence or absence. Then, the final CIE-10-ES code subset associated with a document would consist of those codes whose output indicates its presence. For example, Support Vector Machines (SVMs) and Multi-Layer Perceptrons (MLPs) have been trained for experimentation in binary classification. Each classifier has been fed with label-specific features, by using TF-BNS, as it only perceives the differences between documents with the corresponding code and those that do not have it. Boosting methods to collect predictions of multiple weak models have also been explored. Adaptive Boosting (AdaBoost) iteratively modifies the sample distribution by fitting the weights of each instance, while Gradient Boosting (GBoost) uses a gradient descent function to optimize the remaining errors. In addition, an approach based on TF-IDF similarity has been proposed using the Kullback-Leibler Divergence (KLD) as the term selection method. Estimating KLD provides the best terms characterizing each code in such a way that the terms representing codes and those in documents can be compared.

Fixing code subsets as default labels promotes the assignment of infrequent codes. The inference of codes from a new document would be estimated using the subset belonging to the most similar training document. The transformation of documents into TF-IDF vectors and the estimation of their similarity has been explored (Document-Similarity). Instead of assigning the code subset of the most closely resembling document, a statistical average has been computed to improve the robustness, avoiding the inefficiency of a simple label aggregation. The final labels have been collected by applying voting to the CIE-10-ES codes from the 30 most similar documents.

Regarding adapted algorithms, no assumption is necessary. These methods can infer more than one output from data. In this line, a Long Short-Term Memory (LSTM) fed with word embeddings has been applied to the data set. This approach and the other general multi-label methods do not take the main feature of ICD distributions into account: the number of relevant codes for each document is orders of magnitude smaller than the number of irrelevant ones. For that, XMTC methods focuses on dealing with imbalance, optimizing the retrieval of relevant labels. In particular, a Convolutional Neural Network is explored (XML-CNN), which minimizes a binary cross-entropy loss and exploits dynamic max pooling mechanisms. In turn, the most widespread XMTC approaches split feature spaces or compress label dimension in order to determine the differences between codes. FastXML uses decision trees as bases and binary classifiers to establish criteria in nodes. This is used with TF-IDF, starting with the entire code set in the main node and recursively dividing it into different subsets. Alternatively, SLEEC is based on reduced code vectors and uses KNN and TF-IDF vectors to search similar code projections. Finally, an adaptation of the Latent Dirichlet Allocation for capturing word probabilities for groups of labels is explored (Dependency-LDA). BoW representations are used to estimate those probabilities.

TABLE 3. Feature dimensionality.

Feature	Value
TF-IDF	56,449
TF-BNS	1,000
Bags of Words	900,305
Word embedding sequence	1,500,000

D. EVALUATION

Rank-based assessment metrics are commonly used to compare methods in the XMTC domain. In this line, the evaluation has focused on Precision and normalized Discounted Cumulative Gain at top K, $P@K$ and $nDCG@K$ respectively. $P@K$ would be the number of relevant codes in the K first predicted codes (Equation 1). r is a binary array, where i element indicates the presence or absence of the i suggested code in the gold standard.

$$P@K = \sum_{i=1}^{K} \frac{r(i)}{K} \tag{1}$$

Although Precision estimation is usually complemented by Recall and F-measure values to quantify the correlation between relevant and retrieved codes, this is not necessary when fixing the number of retrieved codes. Instead, $nDCG@K$ would measure the distribution of those relevant codes by giving more importance to the top positions. $nDCG@K$ is described in Equations 2, 3, and 4, where r is the same binary array and $|REL|$ is the number of best ratings up to position K.

$$DCG_K = \sum_{i=1}^{K} \frac{r(i) - 1}{log_2(i + 1)} \tag{2}$$

$$IDCG_K = \sum_{i=1}^{|REL|} \frac{r(i) - 1}{log_2(i+1)} \quad (3)$$

$$nDCG@K = \frac{DCG_K}{IDCG_K} \quad (4)$$

Alternatively, another metric based on the distance between the suggested code set and the gold standard is explored in Equation 6: $S@K$. Similarity values between pairs of codes are calculated exploiting the hierarchical structure as proposed in [38]. Equation 5 deals with the Information Content (IC) of code 1 ($IC(i)$), code 2 ($IC(j)$), and the least common subsumer ($IC(LCS(i,j))$). The IC has been established as the number of characters. Considering that the size of the final CIE-10-ES codes can range from 3 to 7 characters, then $IC \in [3, 7]$.

$$C(i,j) = \frac{2 \cdot IC(LCS(i,j))}{IC(i) + IC(j)} \quad (5)$$

The code set similarity (S) is finally proposed as the maximum weight matching in a bipartite graph $G = (V, E)$, where the vertices are the union of two subsets $V = V_1 \cup V_2$, with V_1 being the suggested codes and V_2 being the gold standard codes, and the edges between both subsets (E) have a cost based on the code similarity $C_{i,j}$ in Equation 5. Such maximization is defined in Equation 6, where N_g is the number of codes in the gold standard and $X_{i,j}$ is a binary value indicating the assignment of code i to code j. As a constraint, there must be only one positive value of X for each i. The Hungarian method has been used for the optimization [39].

$$S@K = \frac{max \sum_{i=1}^{K} \sum_{j=1}^{N_g} C_{i,j} X_{i,j}}{K} \quad (6)$$

One could impose $P@K = S@K$ by restricting $S@K$ to be only the sum of the cost functions of the code pairs that match exactly. Therefore, it is interesting to note that $P@K$ is the same as $S@K$ when there are no partial similarities. So the difference $S@K - P@K$ indicates the percentage of partial code overlap, excluding exact code matches.

Regarding the generation of results, several K values have been computed to evaluate different ranges, but all decisions made are based on the top 10 retrieved codes as this is roughly the average number of codes per document. In this way, $P@10$, $S@10$, and $nDCG@10$ aim to quantify the performance of a system capable of predicting 10 CIE-10-ES codes per document. All approaches described in Section III-C have been applied on the data set using a 5-fold cross-validation with an 80-20 split. Evaluation metrics have been computed based on micro average.

IV. RESULTS

Global scores are shown in Table 4. All $S@K$ values are higher than $P@K$ values, indicating that some of the incorrect suggested codes belong to the same hierarchical branch as some of the unpredicted codes in the report. Moreover, P as a function of K is shown in Figure 3, which gives an idea of the trend of the metrics by varying the number of codes assigned to each document.

In addition to the previous described methods, a baseline consisting of always assigning the most frequent codes is explored. Despite the existence of thousands of codes, the baseline reaches 30%, 19%, and 14% Precision when only predicting the 1, 5, and 10 most frequent codes respectively. It also yields similarity values from 40% to 20% for the suggested code sets. In particular, 14% $P@10$ and 23% $S@10$ means that one of the 10 codes recommended by the baseline usually matches completely (sometimes 2) and several of the other 9 usually match partially without exceeding together more than 100% in the percentage of coincidence. The nDCG values close to 45% suggest that these codes tend to be slightly lower in the output rankings.

The performances of LSTM and KLD barely exceed the baseline as they require large quantities of annotated examples not available in these collections. While LSTM is more effective in predicting few codes, its effectiveness decreases rapidly as the number of codes increases. In contrast, the variation in KLD Precision at different K-values is less pronounced, with higher S values. For example, 28% $P@1$ value is almost double 51% $S@1$ value, which means that almost 3 out of 10 predicted codes (one per document) usually match, while the other 7 codes often overlap categories or subcategories with a total superposition of 30%, e.g. 4 out of 7 suggested codes could match half of the characters with 4 codes from the gold standard.

As for XMTC classifiers, $P@10$ and $S@10$ increase to values greater than 25% and 35% respectively, while nDCG is around 70%. The one based on neural networks (XML-CNN) achieves lower values despite the dynamic max pooling mechanisms counteracting the scarcity of examples, closely followed by SLEEC. Conversely, FastXML and Dependency-LDA obtain more promising results, for both small and large values of K. Both differ in behavior for different K values: Dependency-LDA has a less pronounced Precision slope. Document-Similarity also achieves similar scores by only comparing examples, with 3 out of 10 codes retrieved per document being correct and 15% of the remaining 7 codes matching the categories or subcategories, i.e. the additional 10% to $P@10$ spread over 7 codes.

Regarding binary classifiers, those algorithms achieve the best overall values. Gradient Boosting produces the best performance, with 69% success rate retrieving only one code and 40% for predicting the top 10. It also reaches the maximum similarity, 80% when suggesting a single code and 50% with 10. SVMs and MLPs behave in a similar way, reaching S scores from 80% to 45%. Adaptive Boosting is also around these values, but it shows a smaller variance depending on K.

At first glance, it would seem reasonable to suppose that binary methods more efficiently capture frequent codes, for which there is enough information to conduct a quality characterization, while the others that exploit the dependencies are able to capture rarer codes. A breakdown has been made

TABLE 4. Results of CIE-10-ES predictions for each method. The scores are shown as a percentage.

Type	Method	$P@1$	$P@5$	$P@10$	$S@1$	$S@5$	$S@10$	$nDCG@5$	$nDCG@10$
-	Baseline	30.39	19.31	14.59	41.50	28.77	23.07	46.00	47.48
Binary	SVMs	67.26	49.19	37.06	76.41	58.69	45.64	79.39	76.72
Binary	MLPs	66.99	48.48	35.28	75.35	58.08	43.95	79.00	76.26
Binary	AdaBoost	58.44	47.62	36.36	69.42	57.91	45.06	76.77	74.46
Binary	GBoost	**69.47**	**53.30**	**40.88**	**80.73**	**64.00**	**49.98**	**80.71**	**78.44**
Binary	KLD	28.33	20.96	16.52	51.02	35.09	26.46	56.24	55.20
Grouping	Document-Similarity	56.44	39.59	29.37	66.22	51.22	39.86	72.50	70.02
Adapted	LSTM	37.28	22.14	15.08	48.73	33.06	23.95	51.19	51.19
XMTC	XML-CNN	52.31	34.31	24.99	62.54	46.38	35.26	68.53	65.76
XMTC	FastXML	60.65	41.81	29.87	68.57	52.43	39.32	75.04	72.55
XMTC	SLEEC	51.00	37.08	27.00	63.72	49.45	37.87	68.76	67.18
XMTC	Dependency-LDA	59.61	41.28	31.96	71.52	53.48	41.93	75.34	72.19

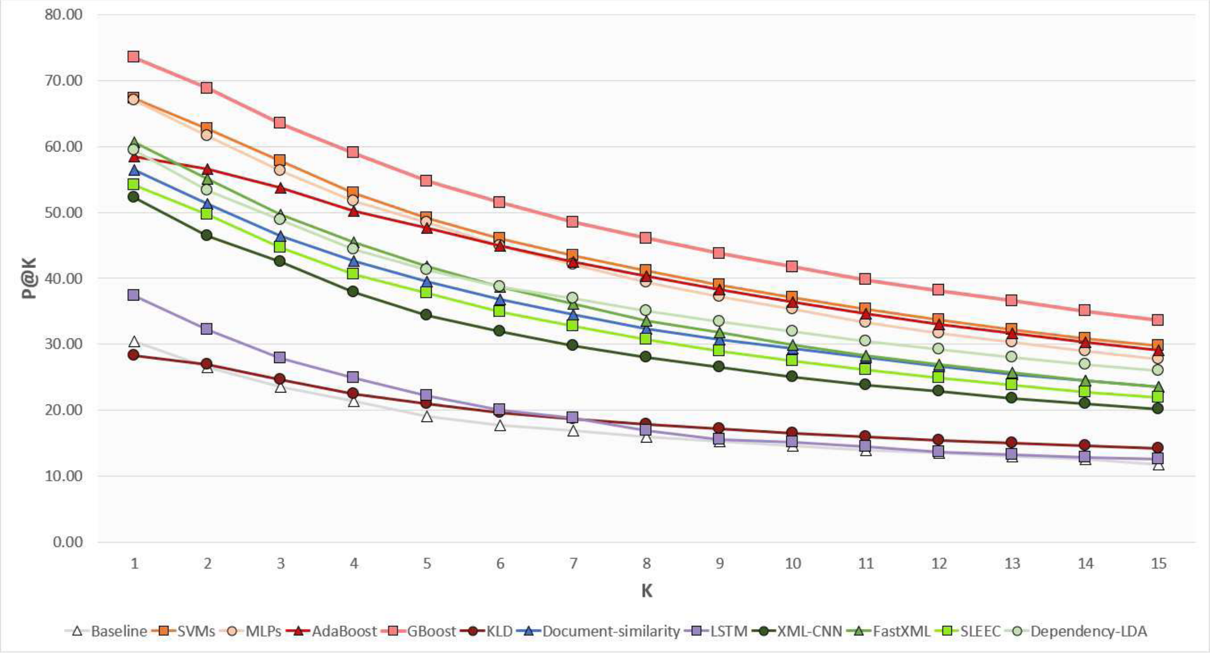

FIGURE 3. Precision at different *K*-values for all methods proposed for experimentation. The scores are shown as a percentage.

below to provide further details on which codes each method is predicting.

A. ANALYSIS

In Figure 4 a detailed analysis of $P@10$ using the frequency of training instances has been carried out to discern differences in retrieved codes. $P@10$ is plotted on the y-axis, breaking down the results by codes grouped into 8 clusters according to the number of instances in the training data set.

The used frequency ranges follow a logarithmic scale to balance the percentage of instances in each one. In addition, the number of different CIE-10-ES codes and the impact on the test data set for each group are shown in parentheses and brackets respectively on the x-axis.

Figure 4 shows three separate sections in which different methods work best: up to 5, from 6 to 278, and from 279 instances in the training data set. As one can see, the LSTM focuses on the very common codes without getting the best scores, like the baseline. XMTC and Document-Similarity approaches tend to balance Precision for all frequency ranges by exploiting code co-occurrences. Dependency-LDA outperform all methods for the least frequent codes in the first section, which contains 5,676 different codes and only represents 15% of the test collection.

Conversely, binary classifiers surpass the other methods for those codes appearing more than 5 times. In particular, SVMs together with boosting methods get the highest Precision in the second section, which collects about 63% of codes in the test data. On the contrary, KLD and MLPs seems to be far better than the others with those codes appearing more than 278 times in the training data set. Despite the poor overall KLD scores, it seems to perform efficiently for the higher ranges.

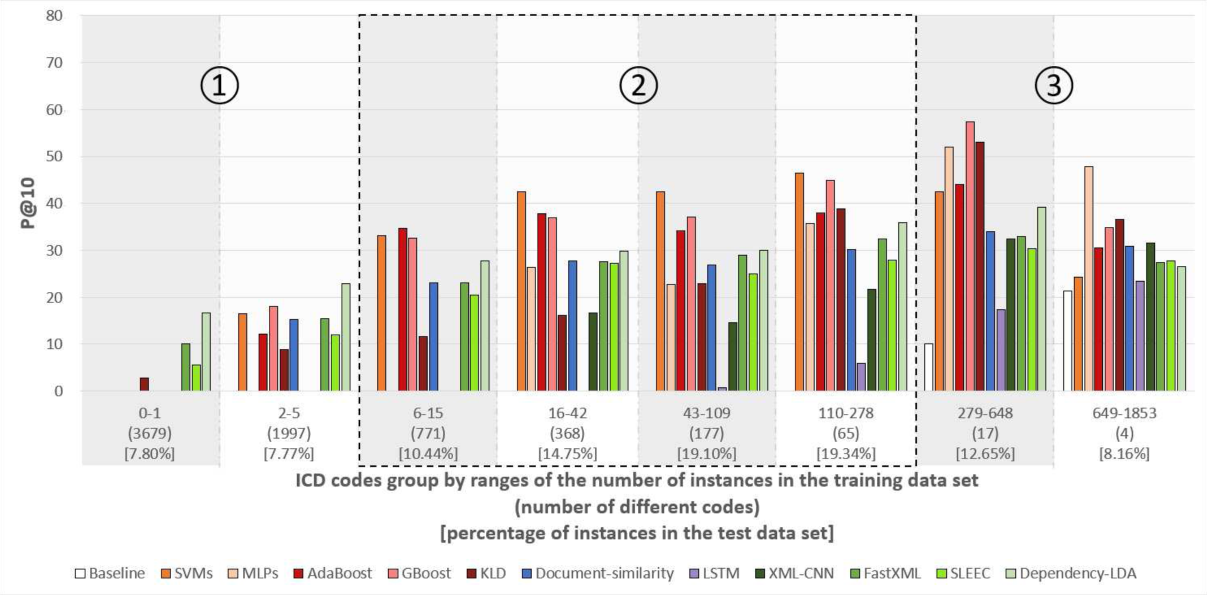

FIGURE 4. Frequency analysis of the CIE-10-ES predictions for each method. The scores are shown as a percentage.

The distribution of the results confirms that an independent code characterization rewards the prediction of common codes, which are assigned to a substantial number of instances with which to establish certain coding criteria, while limiting the suggestion of less frequent codes due to their lack of information. Given the diverse behaviors, an ensemble approach could exploit the dissimilarity and overlapping between methods.

B. ENSEMBLE

The idea of combining such complementary methods is to leverage this better modeling of usual codes and the aggregation of scarce ones. Each representation and method can contribute with different information. As for the way to combine the methods, emphasizing contributions according to the code frequency seems the obvious option. For this purpose, two combination methods have been explored: voting and regression.

The first one is implemented using an extension of the Borda count to try to adjust the relevance of each predicted code i to the results shown in Figure 4. The output per document of each method consists of a list of candidate codes sorted by confidence. In this scenario, all the codes suggested by the methods for a document are grouped together and sorted according to the new value $Score_v(i)$ in Equation 7. Different partial scores are assigned to each output candidate i according to the identifier of the method m that suggests the code, the positions of the code i in the rankings provided by the methods ($p_{i,m}$), and the appearance frequency of the code i in the training data set (f_i).

$$Score_v(i) = \sum_{m=1}^{M} W(p_{i,m}) \cdot \alpha(m, f_i) \tag{7}$$

The sum of partial scores for the same code indicates the final code score (Equation 7), which will determine the position in the ordination. M is the number of methods, W is an exponential decay function, and α is a matrix with coefficients associated with the different methods and frequency ranges in Figure 4. The coefficients are proportional to the performance in each section and system. For example, the coefficient of the codes suggested by Dependency-LDA and which appear between 6 and 15 times in the training data set is the same as the coefficient assigned to the codes predicted by Adaboost and with a frequency greater than 648 occurrences in the training data set.

The intended effect is to penalize those codes that are less reliable in each method and to promote those that tend to be the most successful. The individual treatment per frequency range avoids gathering all the predictions for the most frequent codes by improving the distribution of the results.

Regarding regression, the methods described in Section III-C are applied to the training data set. Each code assigned to a document by some classifier is a training instance. Similar output code attributes have been used: code position per method ($p_{i,m}$), appearance frequency (f_i), and length (L_i). Equation 8 describes the final code score $Score_r(i)$, where M is the number of methods again, β are the intercept constants and slope coefficients, and ϵ is the residual value. $Score_r(i)$ is set to a positive constant value during the learning process if the code i is in the gold standard; otherwise it is zero.

$$Score_r(i) = \beta_0 + \beta_1 \cdot f_i + \beta_2 \cdot L_i + \sum_{m=1}^{M} \beta_{m+2} \cdot p_{i,m} + \epsilon \tag{8}$$

The regressor must estimate the probability that the code i has been assigned to the document. Again, the codes for

TABLE 5. Results of the combination methods, including GBoost with the best scores. The scores are shown as a percentage.

Method	$P@1$	$P@5$	$P@10$	$S@1$	$S@5$	$S@10$	$nDCG@5$	$nDCG@10$
GBoost	69.47	53.30	40.88	80.73	64.00	49.98	80.71	78.44
Voting	62.51	49.22	**46.75**	70.90	57.42	**53.17**	72.93	70.49
Regression	**73.53**	**54.73**	41.73	**82.75**	**66.31**	51.25	**83.56**	**80.80**

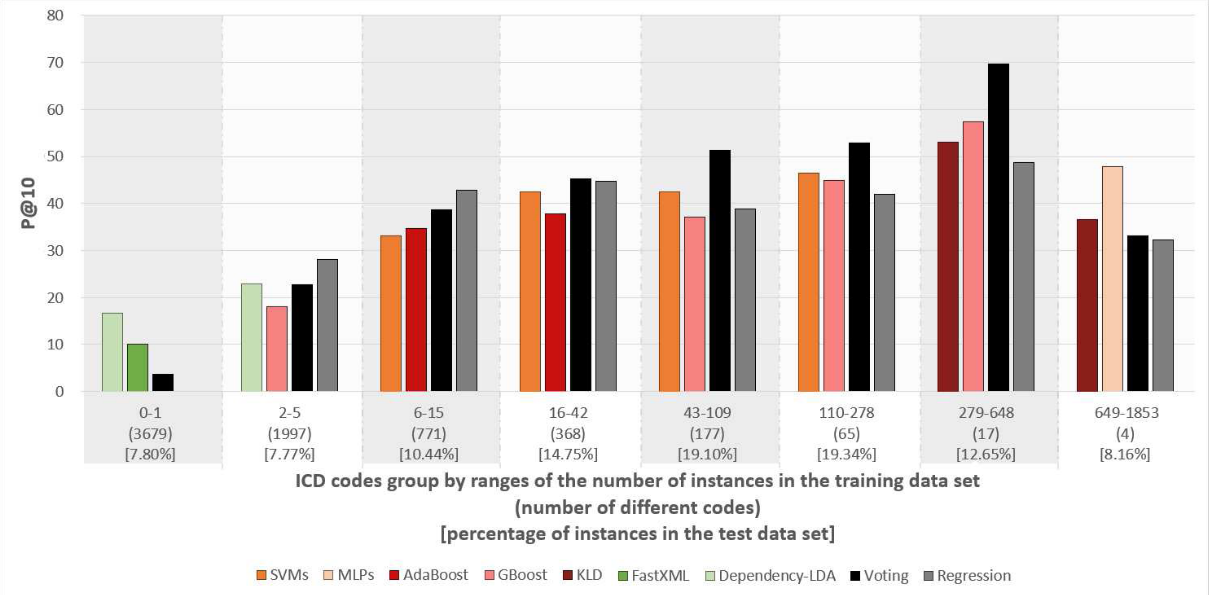

FIGURE 5. Frequent analysis for the predictions of the ensemble approaches and the two best methods per section. The scores are shown as a percentage.

the same document will be sorted according to that value. Although different estimators have been explored to compute final scores, Bayesian linear regression has achieved the best scores on the output rankings. The results are shown in Table 5.

Both combinations have been designed to optimize $P@10$, considering 20 candidates per method during the fusion. The Voting method reaches 46% $P@10$, which is a 15% improvement over the Gradient Boosting. However, the relative increase in S is lower (only 6%), which indicates that almost all codes are successful in full and hardly any in part. In turn, $nDCG@5$ and $nDCG@10$ are smaller, indicating the movement of valid codes to lower positions in the output ranking. On the contrary, Regression method does not reach such high scores at both $P@10$ and $S@10$, but surpasses all other metrics. The increase to 83% in $nDCG@10$ means an approach of the valid codes to the highest positions. Although the distance in $P@10$ between Voting and Regression is 5%, the partial match of the Regression compensates for this difference by reducing the deviation in $S@10$ to less than 2%.

Figure 5 shows the breakdown by frequency of both combinations. It also shows those methods that reach the first or second position per range. In general, combinations outperform any other method in range 43-648, where those seem to exploit the dissimilarities and different criteria of each method. Although the best scores in lower ranks are not exceeded, the combinations succeed in adopting different behaviors approaching the best methods in each case. As for the 4 most frequent codes, performance has been decreased to favor predictions of other codes.

V. DISCUSSION

Binary discriminative methods work properly by suggesting individual codes but tend to focus on the most common codes and ignore the rest. Noise is introduced and Precision is reduced when there are a large number of labels with a limited number of instances. In this case, there are 6,447 categories in the used training data set (91% of the total) with less than 16 instances. Although this is only 26% of the volume of the test data set, the prediction of these codes seems to be more interesting for coders as their criteria and evidences are more difficult to learn. That huge imbalance makes the use of XMTC approaches convenient as they focus on subsets instead, balancing results in different frequency ranges. For example, Dependency-LDA always yields a $P@10$ score between 20% and 40% for all frequency ranges.

A more desirable behavior would be a system with the best of both families of algorithms: a high predisposition to guess frequent codes as they involve most instances and imply more automatic activity for the coder, while keeping the ability to suggest rare codes to handle greater complexity. Assembling methods is one way of trying to combine both attributes.

The overlap of results in a joint system has been explored through Regression and Voting. It should be noted that the remarkable skewness of documents per label distribution produces a tendency for labels with a larger number of documents to be predicted more often and therefore to appear higher at the intersections between results, pushing down the rest of the codes and extending the unbalance. Regression is a discriminative learning method that requires a minimum number of instances to identify patterns, and Voting method selects codes based on their occurrence frequency. Neither method deals with imbalance, so some mechanisms must be implemented to counteract the promotion of common codes.

Including methods that propose more diverse codes such as XMTC classifiers, introducing frequency as a feature to identify rare codes and increasing their relavance compensate for the imbalance. The proposed combinations of methods have demonstrated how to harness different representations and selection criteria to suggest lists of candidate codes more relevant to coder task, predicting common and not-so-common codes. As Figure 5 shows, there is an overall improvement in code prediction, both in high-frequency and low-frequency sections. Hardly any of the codes that appear only once are matched, which seems an acceptable weakness for a data-driven system. The counterbalancing mechanisms used to avoid the constant suggestion of the most frequent codes have penalized the score for those 4 codes that appear more than a thousand times and constitute 8% of the data set volume.

VI. CONCLUSION

This paper has addressed the prediction of the Spanish modification of the ICD-10 (CIE-10-ES) coding as a classification problem with more than 7,000 classes. The main ICD challenge is to deal with extreme distributions, containing few very frequent codes and many infrequent ones. As far as we know, this is the first data-driven proposal to deal with CIE-10-ES coding considering so many codes.

The proposal is conceived to be applied in a real system, suggesting a list of the 10 most probable codes to experts. The idea is to provide the coders with additional information that helps them to focus the search for diagnoses reducing manual annotation time. For this purpose, it is important to consider that coders can more easily recognize very frequent codes in reports than less frequent codes, precisely because they are more used to the former. So, a system capable of suggesting also less frequent codes with precision might be useful to them. For that reason, the proposed approach has focused on avoiding the tendency to always predict the same codes and to promote other less common ones.

Different methods have been explored, with special attention paid to $P@10$ as it indicates the degree of accuracy on the 10 codes, being 10 the average per document. The best $P@10$ score is achieved by Gradient Boosting (40%), followed by SVMs, Dependency-LDA, and FastXML. Conversely, the worst values are reached by LSTM and KLD approaches. None of these methods achieves the best results in all frequency ranges. The idea of combining these methods is based on exploiting their different strengths to improve the results. A rule-based method by voting reaches 46% $P@10$ while a learning-based regressor get 5% less Precision but locating the right codes at the top of the rankings.

In this domain, identifying negation as well as enriching lexical diversity are important factors. Therefore, it is planned for the future to include an effective detection of denied expressions in combination with techniques based on medical knowledge bases in order to improve the representation of reports. The intention is also to explore more effective fusion methods that focus on promoting more less frequent codes and on the rough estimation of the number of codes in each document through the diversity of terms.

REFERENCES

[1] E. M. Van Mulligen, Z. Afzal, S. Akhondi, D. Vo, and J. Kors, "Erasmus MC at CLEF ehealth 2016: Concept recognition and coding in French texts," in *Proc. CLEF*, 2016, pp. 1–4.

[2] L. M. Ho-Dac, "Litl at clef ehealth2017: Automatic classication of death reports," in *Proc. CLEF*, 2017, pp. 1–16.

[3] P. Zweigenbaum and T. Lavergne, "Hybrid methods for ICD-10 coding of death certificates," in *Proc. 7th Int. Workshop Health Text Mining Inf. Anal.*, 2016, pp. 96–105.

[4] S. G. Rizzo, D. Montesi, A. Fabbri, and G. Marchesini, "ICD code retrieval: Novel approach for assisted disease classification," in *Proc. Int. Conf. Data Integr. Life Sci.* Cham, Switzerland: Springer, 2015, pp. 147–161.

[5] D. Zhang, D. He, S. Zhao, and L. Li, "Enhancing automatic ICD-9-cm code assignment for medical texts with pubmed," in *Proc. BioNLP*, 2017, pp. 263–271.

[6] M. T. Chiaravalloti, R. Guarasci, V. Lagani, E. Pasceri, and R. Trunfio, "A coding support system for the ICD-9-cm standard," in *Proc. IEEE Int. Conf. Healthcare Informat.*, Dec. 2014, pp. 71–78.

[7] S. Boytcheva, "Automatic matching of ICD-10 codes to diagnoses in discharge letters," in *Proc. 2nd Workshop Biomed. Natural Lang. Process.*, 2011, pp. 11–18.

[8] M. Subotin and A. Davis, "A system for predicting ICD-10-PCS codes from electronic health records," in *Proc. BioNLP*, 2014, pp. 59–67.

[9] M. Almagro, R. Martínez, S. Montalvo, and V. Fresno, "A cross-lingual approach to automatic icd-10 coding of death certificates by exploring machine translation," *J. Biomed. Informat.*, vol. 94, Jan. 2019, Art. no. 103207.

[10] Y. Chen, H. Lu, and L. Li, "Automatic icd-10 coding algorithm using an improved longest common subsequence based on semantic similarity," *PLoS ONE*, vol. 12, no. 3, 2017, Art. no. e0173410.

[11] W. Ning, M. Yu, and R. Zhang, "A hierarchical method to automatically encode chinese diagnoses through semantic similarity estimation," *BMC Med. Informat. Decis. making*, vol. 16, no. 1, p. 30, 2016.

[12] Z. Miftahutdinov and E. Tutubalina, "KFU at clef ehealth 2017 task 1: ICD-10 coding of english death certicates with recurrent neural networks," in *Proc. CEUR Workshop*, 2017.

[13] A. Atutxa, A. Casillas, N. Ezeiza, V. Fresno, I. Goenaga, K. Gojenola, R. Martínez, M. O. Anchordoqui, and O. Perez-de Vi naspre, "Ixamed at clef ehealth 2018 task 1: ICD 10 coding with a sequence-to-sequence approach," in *Proc. CEUR Workshop*, 2018.

[14] S. Amin, G. Neumann, K. Dunfield, A. Vechkaeva, K. Chapman, and M. Wixted, "MLT-DFKI at CLEF eHealth 2019: Multi-label classification of ICD-10 codes with BERT," in *Proc. CLEF*, 2019, pp. 1–15.

[15] S. W. Chen, P. T. Lai, Y. L. Tsai, J. K. C. Chung, S. S. H. Hsiao, and R. T. H. Tsai, "NCU IISR system for NTCIR-11 MedNLP-2 task," in *Proc. NTCIR*, 2014, pp. 1–4.

[16] D. Arifoğlu, O. Deniz, K. Aleçakır, and M. Yöndem, "Codemagic: Semi-atomatic assignment of ICD-10-AM codes to patient records," in *Information Sciences and Systems*. Cham, Switzerland: Springer, 2014, pp. 259–268.

[17] P. Xie and E. Xing, "A neural architecture for automated ICD coding," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1066–1076.

[18] J. Pérez, A. Pérez, A. Casillas, and K. Gojenola, "Cardiology record multi-label classification using latent Dirichlet allocation," *Comput. Methods Programs Biomed.*, vol. 164, pp. 111–119, Oct. 2018.

[19] M. Almagro, R. Martinez, V. Fresno, and S. Montalvo, "Preliminary study of the automatic annotation of hospital discharge report with ICD-10 codes," *Procesamiento Lenguaje Natural*, no. 60, pp. 45–52, May 2018.

[20] A. Blanco, A. Casillas, A. Pérez, and A. D. de Ilarraza, "Multi-label clinical document classification: Impact of label-density," *Expert Syst. Appl.*, vol. 138, Mar. 2019, Art. no. 112835.

[21] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma, "Multi-label learning with millions of labels: Recommending advertiser bid phrases for Web pages," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 13–24.

[22] J. Weston, A. Makadia, and H. Yee, "Label partitioning for sublinear ranking," in *Proc. 30th Int. Conf. Mach. Learn. (PMLR)*, 2013, pp. 181–189.

[23] S. Si, H. Zhang, S. S. Keerthi, D. Mahajan, I. S. Dhillon, and C. J. Hsieh, "Gradient boosted decision trees for high dimensional sparse output," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3182–3190.

[24] Y. Prabhu and M. Varma, "Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 263–272.

[25] K. Balasubramanian and G. Lebanon, "The landmark selection method for multiple output prediction," 2012, *arXiv:1206.6479*. [Online]. Available: http://arxiv.org/abs/1206.6479

[26] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1529–1537.

[27] W. Bi and J. Kwok, "Efficient multi-label classification with many labels," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 405–413.

[28] M. M. Cisse, N. Usunier, T. Artieres, and P. Gallinari, "Robust Bloom filters for large multilabel classification tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1851–1859.

[29] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 730–738.

[30] J. Liu, W. C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 115–124.

[31] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Mach. Learn.*, vol. 88, nos. 1–2, pp. 157–208, 2012.

[32] J. Leskovec and R. Sosic, "Snap: A general-purpose network analysis and graph-mining library," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, p. 1, 2016.

[33] A. Zubiaga, "Enhancing navigation on wikipedia with social tags," 2012, *arXiv:1202.5469*. [Online]. Available: http://arxiv.org/abs/1202.5469

[34] E. L. Mencia and J. Fürnkranz, "Efficient pairwise multilabel classification for large-scale problems in the legal domain," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2008, pp. 50–65.

[35] K. Gojenola, M. Oronoz, A. Pérez, A. Casillas, and I. Taldea, "Ixamed: Applying freeling and a perceptron sequential tagger at the shared task on analyzing clinical texts," in *Proc. SemEval COLING*, 2014, pp. 361–365.

[36] G. Forman, "Bns feature scaling: An improved representation over TF-IDF for SVM text classification," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 263–270.

[37] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[38] Z. Jia, X. Lu, H. Duan, and H. Li, "Using the distance between sets of hierarchical taxonomic clinical concepts to measure patient similarity," *BMC Med. Informat. Decis. making*, vol. 19, no. 1, p. 91, 2019.

[39] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

MARIO ALMAGRO was born in Madrid, Spain. He received the B.S. degree in industrial engineering from the Carlos III University of Madrid, in 2012, and the M.S. degree in artificial intelligence from the National University of Distance Education (UNED), Spain, in 2017, where he is currently pursuing the Ph.D. degree in intelligent systems engineering. From 2015 to 2017, he was a Research Assistant with the Natural Language Processing (NLP) Department, UNED. He has authored over ten articles in different conferences and journals. His research interest includes development of text mining techniques in the biomedical domain, focused on electronic health record processing.

RAQUEL MARTÍNEZ UNANUE was born in Portugalete, Spain. She received the B.S. and Ph.D. degrees in computer science from University of Deusto, Bilbao, in 1985 and 2000, respectively. She has extensive experience in teaching and researching with several Spanish universities, such as Cádiz University, Cádiz, Complutense University of Madrid, and King Juan Carlos University, Madrid. She is currently a Full Professor with the Computer Systems and Languages Department, National University of Distance Education (UNED), Madrid, Spain. She has been a Project Manager with several competitive research projects. She has authored over 90 articles in different conferences and journals. Her current research interests include text mining, including monolingual and multilingual document, focus on the representation of documents using natural language processing techniques, and classification and clustering in different specialty domains.

VÍCTOR FRESNO was born in Madrid, Spain. He received the B.S. degree in theoretical physics from the Autonomous University of Madrid (UAM), in 1999, the M.S. degree in telecommunication engineering from the Polytechnic University of Madrid (UPM), in 2004, and the Ph.D. degree in computer science from King Juan Carlos University (URJC), in 2006. From 2000 to 2001, he was a Research Assistant with the Spanish National Research Council (CSIC). He was a Teaching Assistant and a Lecturer with URJC. Since 2007, he has been an Associate Professor and an Assistant Professor with the National University of Distance Education (UNED), Spain. He was also a Visiting Faculty with the Queen College, The City University of New York (CUNY), in 2012. He has been a principal investigator with several competitive research projects. He is the author of more than 80 articles. His research interests include document representation models for classification/clustering, information retrieval, and NLP tools and techniques for text mining.

SOTO MONTALVO was born in Segovia, Spain. She received the B.Sc. degree in computer science from the Polytechnic University of Madrid, in 2000, and the M.Sc. and Ph.D. degrees in computer science from King Juan Carlos University (URJC), Madrid, Spain, in 2003 and 2013, respectively. She is currently an Associate Professor with the Department of Computer Science, URJC. She has extensive experience in teaching and researching, participating in different research projects. She has authored over 40 articles in different conferences and journals. Her research interests include text mining and application of natural language processing techniques to biomedical documents.