IXA-AAA at CLEF eHealth 2020 CodiEsp

Automatic classification of medical records with Multi-label Classifiers and Similarity Match Coders

Alberto Blanco*[1], Alicia Pérez[1], and Arantza Casillas[1]

HiTZ Center - Ixa, University of the Basque Country UPV/EHU,
Manuel Lardizabal 1, 20080 Donostia, Spain
{alberto.blanco, alicia.perez, arantza.casillas}@ehu.eus

Abstract. These working notes present the participation of the IXA-AAA team on the CodiEsp Track, as part of the CLEF 2020. The track is about automatic coding of clinical records according to the International Classification of Diseases 10th revision (ICD-10). There are three sub-tasks: CodiEsp-D, CodiEsp-P and CodiEsp-X. The two main tasks, CodiEsp-D and CodiEsp-P, aim to develop systems able to automatically classify clinical texts according to the ICD-10, respectively for diagnostics and procedures. CodiEsp-X, by contrast, is an exploratory sub-task within the framework of Explainable AI in which the goal is to detect the text fragment that motivates the presence of the ICD code. For the IXA-AAA team participation, we have developed several systems to cope with the three sub-tasks, including tree-based multi-label classifiers, similarity match strategies, and ensemble models. For the similarity match, we have explored several approaches and algorithms from string edit distances as Levenshtein to dense representation with Transformers grounded BERT models. Our best results overall are achieved by the combination of models, with a MAP of 69.8% for CodiEsp-D and 48.1% for CodiEsp-P. Regarding the exploratory task, CodiEsp-X, our best coder achieve a micro F1-Score of 30.6%.

Keywords: CLEF · CodiEsp · Clinical records · Similarity Match · Multi-label classifier

1 Introduction

Here we gather the contribution of IXA-AAA team in the CodiEsp Track from the eHealth CLEF 2020 – Multilingual Information Extraction [10,15]. The task consists in the automatic classification of clinical notes according to the ICD-10 codes, considering both procedures and diagnosis. The track contains three independent sub-tasks, two of them considered as the main tasks and the other

* Corresponding author.

regarded as exploratory. The main tasks require systems able to perform ICD assignments (diagnosis and procedures) to a given clinical note. In the exploratory task, the systems must also submit the text that motivated each code assigned. Therefore, the three sub-tasks are a) Diagnosis Coding main (CodiEsp-D): automatic ICD-10-CM (i.e. diagnosis) code assignment. b) Procedure Coding main (CodiEsp-P): automatic ICD-10-PCS (i.e. procedure) code assignment. c) Explainable AI exploratory (CodiEsp-X): automatic ICD-10-CM and ICD-10-PCS code assignment and text position for reference designation.

These tasks present several challenges regarding the text, multi-label and ICD classification domain. The documents, written in Spanish, come from a set of clinical case studies showing properties of both, the biomedical and medical literature, as well as clinical records. Moreover, they cover a variety of medical topics, including oncology, urology, cardiology, pneumology or infectious diseases, which increases both the quantity and the diversity of the ICD codes present in the dataset. Each clinical note can have several diagnoses or procedures and, therefore, we face a multi-label classification task. The text multi-label classification alone is an open challenge in the machine learning field but, conjugating this with the large label-set yielded by the ICD-10 codes, with the low frequency and with imbalance of labels, then, the task involves overcoming multiple and varied barriers. Moreover, we are confronted with a zero-shot learning paradigm, where the clinical cases from the different data partitions (train, dev, test) have non-overlapping label-sets. Regarding the exploratory task, the identification of the text position reference for a given code is not trivial since the non-standard medical language in the text can differ heavily from the standard terms in the ICD. Besides, apart from the continuous references, there are also discontinuous references (i.e. references with several parts distributed along the clinical note). In practical terms, the evaluation of the discontinuous references is carried out taking the beginning of the first fragment and the end of the last.

2 Related Work

The automatic classification of medical records according to the ICD is an active field of research with a presence on shared tasks competitions [19] and Natural Language Processing literature [21]. Through the years, numerous techniques and systems have been developed to solve these tasks, such as Dictionary lookups [4], statistical models like Topic Modeling [18], machine learning models and, lately, Deep Learning models [1,2].

[21] indicates that it is troublesome to evaluate the advances in the field since neither the models nor the evaluation results are generally comparables across related works. Hence, it is a significant milestone to establish standard datasets along with evaluation systems like in this and in past CLEF eHealth editions since 2012 [8]. In 2018, the sixth annual edition of the CLEF eHealth evaluation lab [22], the organizers bestowed a multilingual information extraction lab, with ICD-10 coding of death certificates as the main task. The dataset contained free-text descriptions in 5 languages of causes of death as reported by practitioners in

the standardized causes of death forms, and the teams must extract ICD-10 codes from the raw lines of death certificate text. The best system was provided by the IxaMed team [3], which cast the problem following a sequence-to-sequence prediction paradigm. The authors leveraged only the organizers-provided datasets, namely, ICD-10 dictionaries and the different sets of death reports texts with their corresponding ICD codes, which fed to an encoder-decoder model were able to deliver high-quality, while language-independent, results. On the last year edition of the CLEF eHealth evaluation lab [12] the main task consisted of the classification of non-technical summaries of German animal experiments according to the ICD-10 codes. Although the dataset consisted of veterinary texts, it still comprised a biomedical lexicon, which combined with the use of ICD codes, brought a narrowly related task. The WBI team [20] approached the task as a multi-label classification problem and leveraged the BERT Multilingual model, extended by an output layer which produced the individual probabilities for each possible ICD-10 code. With this setup, the authors succeeded to get the best results on both Precision and F-Measure metrics. However, the MLT-DFKI team [2] managed to improve their recall. While the authors also employed a BERT-based model, in this case, they applied its biomedical variant BioBERT [13], in conjunction with an automatic translation system from German to English, (as the BioBERT model is trained based on the English BERT model, instead of the multilingual). It is worth noting that the WBI team also made use of extra training data from the German Clinical Trials Register and tried ensemble techniques to improve the overall performance. On this year edition, the clinical notes yield longer texts while preserving the challenges related to the clinical language, the non-standard terms and the ICD-10 large label-set. Besides, the Explainable-AI-related assignment brings a new challenge regarding to the interpretability of models.

3 Materials

For the resolution of the three sub-tasks, the organization has provided both main and additional data, and we have employed extra additional resources. The main data consists of 1,000 clinical studies which are coded manually according to the ICD-10 by practising physicians and clinical documentalists. Table 1 shows a brief quantitative description of the main datasets regarding the texts.

Partition	**docs**	**sent/doc**	**words**	**words/doc**	**vocab**	**OOV**
Train	500	17.62	172,533	345±162	26,298	N/A
Dev	250	18.45	86,913	347±165	16,768	8,016
Test	250 + 2,751	19.25	1,110,601	370±304	92,900	74,753
All	3,751	18.44	1,370,047	354±210	105,038	N/A

Table 1. Quantitative description of the main dataset by partition. Number of documents, sentences per doc, total words, average length in number of words, vocabulary size (unique words) and Out-of-Vocabulary (OOV) words for dev and test sets are given.

Note that the row with the test data is in fact 'test + background' and that there is a gap between the number of clinical studies which are coded manually (1,000) and the full number of available documents (3,751). The reason is that the test set is intentionally inflated with a so-called 'background' test with ∼2.700 documents, added to the real test documents (250) to prevent manual predictions. The systems will only be evaluated on the 250 test set documents, but since we cannot discern, the statistics shown are for the test and background sets in conjunction. Regarding the 1,000 coded documents, the partitions split proportion is 50/25/25. Including the background set, there is a total of 71,190 sentences, 1,370,047 words from 105,038 unique words leading to 354±210 words mean ± standard deviation length documents. It is relevant to mention that the texts comprise biomedical, medical literature and clinical records, involving a variety of medical topics such as oncology, urology, cardiology, pneumology or infectious diseases. Hence the variety of technical lexicon is increased, which increases the challenge. It is relevant to note that relative to the OOVs, the percentage of OOV words in the dev set is 47.81% while it is 80.47% in the test + background set, meaning that both sets do not follow a similar pattern concerning the lexical distribution (note that since it also includes the background set we cannot claim that this divergence prevails considering only the test set).

Regarding the labels, namely the ICD-10 codes, there are 10,711 annotated codes with 2,925 unique ones from both the ICD-10-CM (diagnostics) and ICD-10-PCS (procedures). Table 2 presents an overview of the statistics of the train and dev partitions (which were the available annotated partitions of the corpus before the submission, and consequently the data used for training the models).

partition	label-set	label count	unique labels	cardinality	max imb. ratio
Train	CM	5,661	1,767	11.3	0.009
	PCS	1,550	563	3.6	0.015
Dev	CM	2,683	1,158	10.7	0.02
	PCS	817	375	3.7	0.025
All	CM	7,211	2,196	11.0	0.014
	PCS	3,500	729	3.6	0.02

Table 2. Statistical description of the labels of the main dataset by partition

One can see that all the codes from train + dev set only represent a small percentage of the full ICD-10 codes (98,287 for ICD-10-CM and 87,169 for ICD-10-PCS), but still portray a large label-set, especially taking into account the low representativeness of some codes (i.e. only 200 CM labels appears on 1% or more of the clinical cases from the train set) and the extreme imbalance. But more important is the question of the disjoint codes among sets, and especially, unseen codes in the test set. In fact, there may be unseen codes in the test set, and in general, there are codes which only appear on one partition, since the partitions were obtained via a random split of training, dev and test (i.e, there are 1,036 CM and 352 PCS labels on dev set not seen on train set). This question leads to a zero-shot learning environment where a standard classifier will make predictions solely among the seen codes on the training phase, and therefore, fail to predict the unseen codes.

The CodiEsp-X sub-task requires to detect the text reference position, so the available corpus also brings the start and end position noted. Also, keep in mind that there are continuous and discontinuous codes, the formers implies that all the words related to the code appear sequentially in the text, while for the latter there are several fragments of texts related to the code. Nevertheless, in both cases, the way to evaluate the detection as correct is to give the start position of the first (or unique) fragment and end position of the last (or unique) fragment, regardless of the number of fragments. The organization also provides additional resources, and from those, we have used the Spanish abstracts from Lilacs and Ibecs with ICD-10 codes, to expand the dictionary of ICD and non-standard descriptions. The in-house resources employed by our team consist of additional non-standard term descriptions for some ICDs. Moreover, we have applied a Medical Named Entity Recognition (NER) system to extract medical terms such as diagnostic and procedure terms, and to reduce noisy words. This alternative representation of texts has helped us with the augmentation of the train and dev sets.

4 Methods

The systems developed to deal with each subtask are of two different kinds. First, we have applied a tree-based multi-label classifier based on gradient boosting machines [9], to cope with CodiEsp-D and CodiEsp-P, presented in section 4.1.

Furthermore, we have developed a coder based on string similarities, which can cope with CodiEsp-D and CodiEsp-P sub-tasks, but also CodiEsp-X, introduced in section 4.2. Besides, we combined the outputs from the classifiers and the coders to improve the overall results.

Regarding the text representation, we have applied a Medical Named Entity Recognition (NER) tool to extract medical entities from the raw texts. Particularly, it classifies each word as 'Disease', 'Procedure', 'Drug', 'Part of the body' or 'Others'. Taking that classification, we have extracted three alternate representations of the raw clinical notes following two strategies; i) Medical terms (NER Med): Aims for noise removal, preserving only those words not classified as 'Others' and ii) Diagnostics (NER D) or Procedures (NER P): Preserve only the words marked as 'Disease' or 'Procedure', accordingly. These alternate representations can also be concatenated to the raw texts, as a data augmentation technique.

4.1 Tree-based multi-label classifier: Gradient Boosting Machines

The Gradient Boosting Machine or GBM is an ensemble classifier. Ensemble classifiers rely on the combination of several base-classifiers to make a final prediction. Specifically, the boosting technique consists in training several classifiers sequentially, in a manner that each classifier learns from the errors made by the previous ones. The objective of each individual classifier is to reduce the loss function, in this case, binary cross-entropy (CE), given by expression (1), where log is the natural logarithm, y is the binary label and p is the prediction or membership probability to the given class.

$$CE = -\left[y \log p + (1-y) \log (1-p)\right] \tag{1}$$

The optimization of the function uses a gradient descent algorithm to minimize the loss when adding new classifiers [5]. To cope with the multi-label paradigm, we applied the one-versus-rest approach, in which as many binary classifiers as present labels are trained. Thus, for the i-th binary classifier, label i is treated as the positive class and all the remaining labels as negative. The training procedure is then followed by a post-processing stage where the optimal threshold must be found. However, note that the evaluation metric to be applied in this task is the Mean Average Precision (MAP) well suited for candidate-ranking. For consistency with this metric, the output from our system is a ranking of all the possible labels ordered by probability. That is, the system should provide all the labels ($1,767$ for CM and 563 for PCS) even though from the data analysis (in Table 2) one could expect the system to provide just around 11.3 labels in CM and 3.6 in PCS. More about this question is discussed in section 6. We have applied the XGBoost implementation [6] with the Scikit-learn [17] wrapper.

4.2 Similarity Match

Our Similarity Match algorithms set their foundation on the similarity between two strings. For this work, we have implemented two variations that, although they follow this same approach, differ severely on the core of the algorithm, i.e. the computation of the similarity itself. We have named them, respectively PartialMatch and BERTMatch coders.

First, let us describe the shared logic behind the two coders using our specific use-case as an example. On the one hand, we get a clinical record with several sentences. Naturally, in each sentence one or more terms associated with a given ICD code can appear. As an example, here we are a sentence from the corpus:

Sentence:

```
'Realizamos frotis sanguíneo para justificar la causa de la
anemia y trombocitopenia'
```

On the other hand, there is an ICD code dictionary, which relates each code to one or more arbitrary-length description strings (either standard diagnostic terms from the ICD or gold mentions from the corpus). An entry from the ICD dictionary, as shown below, conveys the ICD code (D69.6) and one or more standard ways to refer to that code (e.g. plaquetopenia, tombocitopenia, trombocitopenia, trombopenia).

Dictionary entry:

```
D69.6:
       plaquetopenia
       tombocitopenia
       trombocitopenia
       trombopenia
```

The dictionary can include a variety of terms including standard and non-standard, single-word descriptions and even phrases frequently associated with the code like 'enfermedad de graves basedow' for the 'E05.00' code, which differs harshly from the standard ICD description ('tirotoxicosis con bocio difuso sin crisis tirotoxica ni tormenta tiroidea').

Next, a Similarity Match algorithm will cycle through all the associated strings of each ICD code, and through all the texts, computing the similarity between pairs of standard and non-standard terms and text fragments. The texts fragments are extracted with a sliding window in which the length is set to the number of words of the current ICD description. Following the example, the process to find the likelihood of the D96.6 code on the sample text is as follows: compute the similarity between the 'plaquetopenia' term and each word of the target text, and store the maximum value. Then, repeat for the rest of the associated terms (tombocitopenia, trombocitopenia...), and finally get the overall maximum value. As the similarity metric is normalized in the [0, 1] range, it can be interpreted as a membership probability for each code on each clinical record. In the case of CodiEsp-X, which requires identifying the range, it is only necessary to search for the range of the text fragment that leads to the maximum similarity.

The **similarity computation** is then what differentiates the two developed coders. Let us describe a similarity function as one that for a given pair of strings as input generates a similarity coefficient, as described in (2), where s_1 and s_2 is a pair of strings, sim is the similarity coefficient normalized in a range $[0, 1]$ and Σ is the vocabulary. Regarding the interpretation, $sim(s_1, s_2) = 1$ means that s_1 and s_2 are the same string while $sim(s_1, s_2) = 0$ means that are completely different.

$$\begin{array}{rcl} sim : \Sigma^* \times \Sigma^* & \longrightarrow & [0, 1] \\ (s_1, s_2) & & sim(s_1, s_2) \end{array} \tag{2}$$

On this basis, the Partial Match coder applies a regular string similarity algorithm, such as the Jaro Winkler [24] or Levenshtein Distance [14] (and we also enable an ‘Auto’ configuration that dynamically choose one or the other based on the length of the given term).

On the other hand, the BERTMatch coder leverages the BERT Multilingual model [7] to come out with a similarity between strings, the process is as follows. First, for each string (s_i) a dense representation ($v(s_i) \in \mathbb{R}^n$ with $n = 768$) is extracted from the representation of the texts generated internally by the BERT model. Then, the similarity between the vectors $v(s_1)$ and $v(s_2)$ is computed via the Cosine Similarity [11]: $sim_{BERTMatch}(s_1, s_2) = cos(v(s_1), v(s_2))$. Note that the BERTMatch algorithm is far more computationally demanding than PartialMatch, hence, it was not applied to the test set predictions (indeed, the test set is, curiously enough, the largest set, as shown in Table 2).

Following the example, the matching score between the word ‘t**r**ombocitopenia’ from the model sentence and the word ‘tombocitopenia’ from the dictionary entries gives a similarity value (in the range $[0.0, 1.0]$, being 0.0 completely different words and 1.0 exactly the same word) of 0.98 with the JaroWinkler (as it is almost the same word but with a slight spelling mistake) but only 0.46 with the BERT embeddings. However, the score between ‘trombocitopenia’ and ‘plaquetopenia’, is as low as 0.57 with JaroWinkler (although they are synonyms) and 0.78 with the BERT embeddings, a much more appropriate score since both words mean the same thing.

Finally, it should be noted that we have developed all the classifiers and coders in a way so that their outputs can be combined. Combining is done using simple aggregation functions such as the mean, minimum, or maximum over the similarity scores or probabilities, which is a straightforward but practical way to improve results by ensembling strategies.

5 Results

The results from the submissions, on the test set, as reported by the CLEF organizers for the CodiEsp-D/P and X sub-tasks, are presented on this section. Table 3, 4 and 5 show our team submission results, including the predictions from the test set (250 docs) (and excluding the 2,751 background set docs). For the official results, only predictions for test files and labels from train and dev sets were considered.

Note that during the development phase of the challenge, it was reported that there were codes present in the test set that were not present in the train and validation sets, but only after the submission phase was reported that these codes would not be taken into account for the evaluation of the results. Therefore, the systems were developed considering that all metrics would be computed taking into account also the predictions for the codes present only in the test set, which could have had significant harmful effects on the results.

The official metrics for the subtasks are MAP for the CodiEsp-D/P and F-Score for the CodiEsp-X, but other metrics were also computed and reported. Specifically, MAP@30, Precision and Recall for CodiEsp-D/P and Precision and Recall for CodiEsp-X. Finally, in CodiEsp-D, Precision, Recall and F-score were also computed for categories, considering a category the first three digits of an ICD-10-CM code. (I.e. codes P96.5 and P96.89 are mapped to P96). Therefore, systems that predict the code P96.89 for a document whose correct code is P96 would be correct. In CodiEsp-P, Precision, Recall and F-score are also computed for categories. In this case, considering categories the first four digits of the code.

In relation with the name of the columns, **M** stands for **MAP**; **M30** stands for **MAP@30**; **P** stands for **Precision**; **R** stands for **Recall**; and **F1** stands for **F1-Score**. The **T** suffix stands for **Test** and the **C** suffix stands for **Category**. Those columns with the **Test** suffix show the results evaluated only on the labels from the train and dev sets (without the only test set labels), while those with the **Category** suffix show the results considering the category labels.

For all sub-tasks, each submitted run is the result of applying different techniques. Table 3 presents the results from the CodiEsp-D sub-task, and each run corresponds to the following setup: 1) XGBoost classifier, trained with documents and diagnostics labels from train and dev sets, augmenting the clinical texts with the outputs of the NER Med and the NER D. 2) Partial Match coder with the Jaro Winkler similarity algorithm and predicting only the diagnostics labels present on the train and dev sets. 3) The combination of the outputs from 1) and 2).

Regarding the official metric for this task, namely the MAP, or more precisely, the MAP evaluated only on the test set (**M-T** column), the XGBoost classifier prevails over the Partial Match strategy with 63.8 and 57.1 points respectively. However, the best result comes from the run3, with the combination of both methods, leading to 69.8 MAP points.

Table 4 presents the results from the CodiEsp-P sub-task, and each run corresponds to the following setup: 1) XGBoost classifier, trained with documents and procedure labels from train and dev sets, augmenting the clinical texts with

Run	M	M-T	M30	M30-T	P	R	F1	P-T	R-T	F1-T	P-C	R-C	F1-C
run1	0.543	0.638	0.529	0.622	0.004	0.858	0.009	0.004	1.0	0.009	0.01	0.968	0.021
run2	0.485	0.571	0.469	0.553	0.004	0.858	0.009	0.004	1.0	0.009	0.01	0.968	0.021
run3	0.593	0.698	0.578	0.681	0.004	0.858	0.009	0.004	1.0	0.009	0.01	0.968	0.021

Table 3. Submission results for the CodiEsp-D sub-task as reported by the CLEF organization.

the outputs of the NER Med and the NER P. 2) Partial Match coder with the Jaro Winkler similarity algorithm and predicting only the procedure labels present on the train and dev sets. 3) The combination of the outputs from 1) and 2).

Run	M	M-T	M30	M30-T	P	R	F1	P-T	R-T	F1-T	P-C	R-C	F1-C
run1	0.412	0.46	0.395	0.441	0.004	0.825	0.008	0.004	1.0	0.008	0.005	0.857	0.01
run2	0.362	0.414	0.339	0.389	0.004	0.825	0.008	0.004	1.0	0.008	0.005	0.857	0.01
run3	0.425	0.481	0.401	0.455	0.004	0.825	0.008	0.004	1.0	0.008	0.005	0.857	0.01

Table 4. Submission results for the CodiEsp-P sub-task as reported by the CLEF organization.

Similarly to the D sub-task, the best M-T result from single models is achieved by the XGBoost classifier, with 46.0 points, while the Partial Match strategy stays about 5 points below, with 41.4 points. Once again, the combination of both methods manages to improve individual performance, with a solid 48.1 MAP points.

Table 5 presents the results from the CodiEsp-X sub-task, and each run corresponds to the following setup: 1) Partial Match coder with the Jaro Winkler similarity algorithm. 2) Partial Match coder with the Auto configuration for the similarity algorithm. 3) Partial Match coder with the Levenshtein similarity algorithm. For each setup, predicting only the diagnostics and procedure labels present on the train and dev sets.

Run	P	R	F1	P-T	R-T	F1-T
run1	0.043	0.318	0.075	0.043	0.374	0.076
run2	0.144	0.301	0.195	0.144	0.354	0.205
run3	0.288	0.278	0.283	0.288	0.327	0.306

Table 5. Submission results for the CodiEsp-X sub-task as reported by the CLEF organization.

For the CodiEsp-X task, the official metric is the F1-Score, particularly, the F1-Score evaluated only on the test set (**F1-T** column). We can see that the Jaro

Winkler algorithm, which dominated on the D/P tasks, here is, curiously, the worst-performing one with 7.6 points. The 'Auto' configuration, that mixes the Jaro Winkler and Levenshtein algorithms, gets an increased 20.5 points. Finally, the Levenshtein algorithm improves that mark by approximately 10 points, with a solid F1 score of 30.6, which is our best result overall for the CodiEsp-X task. Although we have not been able to apply the Similarity Match algorithm based on BERT embeddings on the test + background set for computational reasons, our experiments in the dev set suggest that the BERTMatch algorithm is able to overcome the Jaro Winkler.

6 Discussion

For sub-tasks CodiEsp-D and CodiEsp-P, the code predictions must be ranked, this is, generating a list of possible codes ordered by confidence. The main metric for evaluating these outputs is the Mean Average Precision or MAP. It is computed iteratively; First, precision is computed considering only the first ranked code, then, it is computed considering the first two codes, etc. Finally, precision values are averaged over the number of gold codes. The organizers claim that the MAP is the most standard ranking metric among the TREC community, and it has shown good discrimination and stability [16]. However, the way to exploit the MAP metric is to output all the considered codes, without discrimination, and ranked by confidence. In other words, ranking all the considered ICD codes and not establishing a threshold for a discrete "Yes/No" decision. Our scripts yield this output because it is the way to maximize the MAP metric and face the competition but we believe that this way of evaluating might not be the most desirable, since the notion of an "automatic classifier" that "decides" whether or not a code belongs in a given document is shaded. We feel that instead of ranking all the labels available within the ICD, the system should just limit the output to a subset of labels that correspond to the document. Nevertheless, MAP metric favours a rank over all the labels above a rank of a sub-set of labels. In brief, the ability to state whether a code is present or not in the in the given medical record is not regarded by the MAP metric. Accordingly, a weakness of this task is the need of a threshold for accepting and discarding codes given the ranked list. By contrast, the CodiEsp-X sub-task does not present this drawback, since the main evaluation metric is the micro F-Score, and therefore each predicted code that does not belong to the ground truth carry a penalty.

In the CodiEsp-X sub-task, there are some codification errors on which the assigned ICD code and the text which has motivated the assignation of code mismatches involving those errors. We have found slight differences with the main ICD block (the first three digits of the ICD) remain while the modifiers (other digits) vary. However, the evaluation entails the F-Score of the full-code, without considering the relationship between codes according the hierarchy. This type of errors (confounding two closely related diseases) penalize as any other error (i.e. confounding un-related diseases). For example, for the record with ID 'S0211-69952011000500011-3', the label 'K85.10 - **Billiary acute** pancreatitis

WITHOUT NECROSIS OR INFECTION' is assigned, motivated by the following text fragment: '*acute non-lithiasic pancreatitis*'. The mistake is that the record elucidates that it is '**non**-lithiasic pancreatitis', but the code corresponds to that of 'lithiasic' or 'biliary' pancreatitis. The label assigned by our system is 'K85.90 - ACUTE PANCREATITIS WITHOUT NECROSIS OR INFECTION, UNSPECIFIED', and although we cannot claim that the K85.90 is the correct label, it seems that is, at least, more accurate than K85.10, but it is counted as an error.

On document 'S2254-28842013000300009-1' we have the following text fragment: '*Mujer de 73 años de edad con antecedentes personales de [...],* ***histerectomía por prolapso uterino*** *y [...]*'. Our system gives a confidence of 98.5% to the 'Z90.710 - ACQUIRED ABSENCE OF BOTH CERVIX AND UTERUS' code, which describes a hysterectomy (the surgical removal of the uterus, which may also include the cervix and other surrounding structures [23]). The Z90.710 code is considered as incorrect, and there is no other code that matches the '*histerectomía*' word (though it is coded with 'N81.2 - INCOMPLETE UTEROVAGINAL PROLAPSE' due to '*prolapso uterino*', which is the cause of the hysterectomy, and seems correctly coded). There are abundant examples of this type of missing codes in the ground truth that, unfairly lead to False Positives. Accordingly, we believe that the evaluation results of these tasks should be regarded with prudence.

7 Concluding remarks and future work

The CodiEsp Track proposes two different sub-tasks based on the classification of medical texts according to the ICD-10 CM and PCS codes. The CodiEsp-D/P sub-tasks aim to the automatic classification of diagnostic and procedures codes, while the CodiEsp-X sub-task strives to bring explainability to the challenge.

We have developed several systems to cope with these tasks, two strategies with five different algorithms for the D and P sub-tasks, and one strategy with four algorithms capable of producing explainable results, in conjunction with the ability of ensembling the distinct models, enhanced by techniques that yield alternate representations of the medical texts with tools as Medical NER, while experimenting also with different label-sets.

Regarding the D and P sub-tasks, the similarity match based algorithms perform better, on average, than the multi-label classifiers. However, we conclude that the NER techniques for enriching the medical text inputs for the classifiers accomplish to improve the performance of the classifiers, resulting in the best overall results being achieved with the combination of both methods.

It seems that the best similarity algorithm for the diagnostics and procedures individually is the Jaro Winkler, while it is the Levenshtein for the CodiEsp-X sub-task as a whole. We have not delved in this topic, but might be related to divergences among the average length of diagnostic and procedure terms.

The similarity match algorithm based on the BERT dense representations appears to be weaker than the traditional approaches but shows promising results when applying it to the extraction of diagnostic and procedure terms boundaries.

The consideration of the full ICD-10 codes instead of those from the train set degrades the performance. It can be observed in every sub-task, and we believe that this is due to the large number of extra codes considered with respect to the actual number of codes that only appear in the dev set. Improving NER and looking for combined match approaches might lead to further improvements.

8 Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Technology (PAT-MED PID2019-106942RB-C31) and by the Basque Government (Elkartek KK-2019/00045, IXA IT-1343-19, Predoctoral Grant PRE-2019-1-0158).

References

1. Almagro, M., Unanue, R.M., Fresno, V., Montalvo, S.: Icd-10 coding of spanish electronic discharge summaries: An extreme classification problem. IEEE Access **8**, 100073–100083 (2020)
2. Amin, S., Neumann, G., Dunfield, K., Vechkaeva, A., Chapman, K.A., Wixted, M.K.: Mlt-dfki at clef ehealth 2019: Multi-label classification of icd-10 codes with bert. CLEF (Working Notes) (2019)
3. Atutxa, A., Casillas, A., Ezeiza, N., Fresno, V., Goenaga, I., Gojenola, K., Martínez, R., Anchordoqui, M.O., Perez-de Viñaspre, O.: Ixamed at clef ehealth 2018 task 1: Icd10 coding with a sequence-to-sequence approach. In: CLEF (Working Notes). p. 1 (2018)
4. Bounaama, R., Abderrahim, M.E.A.: Tlemcen university at celf ehealth 2018 team techno: Multilingual information extraction-icd10 coding. In: CLEF (Working Notes) (2018)
5. Cauchy, A.: Méthode générale pour la résolution des systemes d'équations simultanées. Comp. Rend. Sci. Paris **25**(1847), 536–538 (1847)
6. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Ferro, N.: What happened in clef... for a while. Crestani et al.[94] (2019)
9. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
10. Goeuriot, L., Suominen, H., Kelly, L., Miranda-Escalada, A., Krallinger, M., Liu, Z., Pasi, G., Saez Gonzales, G., Viviani, M., Xu, C.: Overview of the CLEF eHealth evaluation lab 2020. In: Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., andNicola Ferro, L.C. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020). LNCS Volume number: 12260 (2020)

11. Han, J., Kamber, M., Pei, J.: 2 - getting to know your data. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining (Third Edition), pp. 39 – 82. The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Boston, third edition edn. (2012). https://doi.org/https://doi.org/10.1016/B978-0-12-381479-1.00002-2, http://www.sciencedirect.com/science/article/pii/B9780123814791000022
12. Kelly, L., Suominen, H., Goeuriot, L., Neves, M., Kanoulas, E., Li, D., Azzopardi, L., Spijker, R., Zuccon, G., Scells, H., et al.: Overview of the clef ehealth evaluation lab 2019. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 322–339. Springer (2019)
13. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
14. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
15. Miranda-Escalada, A., Gonzalez-Agirre, A., Armengol-Estapé, J., Krallinger, M.: Overview of automatic clinical coding: annotations, guidelines, and solutions for non-english clinical cases at codiesp track of CLEF eHealth 2020. In: Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings (2020)
16. Mogotsi, I.: Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval (2010)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
18. Pérez, J., Pérez, A., Casillas, A., Gojenola, K.: Cardiology record multi-label classification using latent dirichlet allocation. Computer methods and programs in biomedicine **164**, 111–119 (2018)
19. Pestian, J., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Biological, translational, and clinical language processing. pp. 97–104 (2007)
20. Sänger, M., Weber, L., Kittner, M., Leser, U.: Classifying german animal experiment summaries with multi-lingual bert at clef ehealth 2019 task. CLEF (Working Notes) (2019)
21. Stanfill, M.H., Williams, M., Fenton, S.H., Jenders, R.A., Hersh, W.R.: A systematic literature review of automated clinical coding and classification systems. Journal of the American Medical Informatics Association **17**(6), 646–651 (2010)
22. Suominen, H., Kelly, L., Goeuriot, L., Névéol, A., Ramadier, L., Robert, A., Kanoulas, E., Spijker, R., Azzopardi, L., Li, D., et al.: Overview of the clef ehealth evaluation lab 2018. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 286–301. Springer (2018)
23. Thomson, A.P.: Handbook of Consult and Inpatient Gynecology 1st ed. Springer (2016)
24. Winkler, W.E.: The state of record linkage and current research problems. In: Statistical Research Division, US Census Bureau. Citeseer (1999)