# Fraunhofer AICOS at CLEF eHealth 2020 Task 1: Clinical Code Extraction From Textual Data Using Fine-Tuned BERT Models

João Costa[1], Inês Lopes[1], André Carreiro[1], David Ribeiro[1], and Carlos Soares[1,2]

[1] Fraunhofer Portugal AICOS
Rua Alfredo Allen, 455/461, 4200-135 Porto, Portugal
https://www.aicos.fraunhofer.pt/
{joao.antonio,ines.lopes,andre.carreiro,david.ribeiro,
carlos.soares}@fraunhofer.pt
[2] Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
https://www.fe.up.pt

**Abstract.** Nosology is an important branch of Medical Science that concerns the classification and coding of diseases, conditions, procedures, and other medical information. This is a vital task for all stakeholders of the health sector, from hospitals and health regulators, to insurance companies and governments. The ICD10 system is the current revision of a Nosology system managed by the World Health Organization, being widely used internationally. Since medical coding is based on manual analysis of clinical textual data, it is ripe for automation, with Natural Language Processing (NLP) techniques used to address this challenge. This paper describes our contribution to the CLEF eHealth 2020 Task 1 Challenge, regarding Information Extraction of ICD10 codes on unstructured Spanish clinical text. We present two approaches for ICD10 code extraction based on Conditional Random Fields (CRFs) and the *BERT* Deep Learning Language Model. The *BERT*-based methodology achieved a mean average precision of 0.517 and 0.445 for ICD10-CM and ICD10-PCS codes, respectively, and a F1 score of 0.505 for the Explainable AI subtask. The results obtained show the flexibility and robustness of pre-trained Deep Learning models for NLP, only requiring fine-tuning for a particular task, leading to reduced requirements both for labelled data and computational effort.

**Keywords:** Medical Information Extraction · ICD10 Codes · *BERT* Language Model

## 1 Introduction

Medical coding, also known as Nosology, is an important area for the health sector, with dedicated specialists manually annotating a large number of relevant clinical documents, such as in- and outpatient clinical reports. This operation is essential for several stakeholders, including health information management systems, insurance companies, governments, researchers, among others [5].

The most commonly used medical coding system is the International Classification of Diseases (ICD), developed and maintained by the World Health Organization (WHO), which aims to provide the nations of the world with a Nosology standard for disorders, diseases, and other conditions, structured in a hierarchical fashion. Currently it is in its tenth revision (ICD10) [35], with a new major release (ICD11) planed for early 2022 [5].

The ICD10 code system has been further augmented by WHO member states, including the USA, where the Centers for Medicare and Medicaid Services have the ICD10-CM (Clinical Modification) [1] and ICD10-PCS (Procedure Coding System) [2] systems. The ICD10-CM focuses on morbidity data (diseases, conditions, etc.), whereas the ICD10-PCS is used to code medical procedures (surgeries, implants, among others). Other countries have translated and adapted these Nosology systems, such as Spain with the CIE-10-ES *Diagnósticos* [3] and CIE-10-ES *Procedimientos* [4] systems, respectively.

The annotation process is based on the manual analysis of clinical reports, leading to significant costs and time spent by specialists. Therefore, the introduction of (semi-)automatic processes of annotation is an important challenge. This is where Natural Language Processing (NLP) comes into play. Recently, the paradigm of NLP has shifted to the application of large, deep models, pre-trained on extensive corpora, and fine-tuned on a particular task at hand [37]. This allows highly accurate models for a plethora of applications with low fine-tuning effort, when compared to tailor-made systems that require high amounts of annotated data, with high computational costs and training time.

In this paper we describe our contributions to CLEF eHealth 2020 Task 1, which have their basis on the application of Conditional Random Fields (CRF), and the *BERT* language model [12], pre-trained on Spanish corpora [10], and fine-tuned on NER of ICD10 codes.

## 2 Task and Data

The CLEF eHealth 2020 challenges researchers with real-world datasets, fostering the application of state of the art NLP methodologies on the medical and clinical domains [14]. In particular, the proposed Task 1 in this series concerns Information Extraction (IE), focusing on the extraction of ICD10 codes from clinical textual data in Spanish [19].

The dataset is composed of Spanish text data from clinical reports, partitioned into *train* (500 reports), *dev* (250), and *test* (250) subsets. In addition, 2751 reports are provided with the *test* set as a *background* set, to discourage

manual corrections and promote scalable solutions. In total, 2172 ICD10-CM and 696 ICD10-PCS unique codes are referenced in the *train* and *dev* set.

Task 1 is subdivided into 3 subtasks:

**1** — **ICD10-CM codes assignment** predict ICD10-CM codes present in a given clinical report, ranked by confidence;
**2** — **ICD10-PCS codes assignment** predict ICD10-PCS codes present in a given clinical report, ranked by confidence;
**3** — **Explainable AI** predict ICD10-CM and ICD10-PCS codes present in a given report and provide a textual reference (character span) that justifies said codes.

Subtasks 1 and 2 can be regarded as a multi-label classification problem on a report level, whereas subtask 3 is related to multiclass classification on a word-by-word level, i.e., a Named Entity Recognition (NER) problem, where each word present in a given text has 1 label associated with it (ICD10 codes in this case).

## 3 Related Work

Medical coding has been a task mainly reserved to specialized personnel, although there are some recent efforts to automate this process.

Early attempts for automatic medical coding were mainly rule-based systems for ICD9 [34] code assignment [11,13]. These systems automatically find relations between ICD9 codes, their descriptions, and medical text, creating a list of associations that allows medical text labelling. Another approach is described in [16], where the authors created an ICD10 coding system by applying Support Vector Machine classifiers in a cascaded architecture to automatically assign cancer related medical codes to death certificates. Although all these systems achieve good results, their reach is reduced, with each only encompassing a small subset of the full ICD code list. Nevertheless, these systems have high interpretability, which makes them valuable and interesting to use in specific, smaller scopes.

New methodologies developed with Deep Learning (DL) models and architectures have had a great impact in recent NLP research. Most of the tasks and challenges of this area have benefited from deep word embedding strategies, from global and context-free embeddings such as *word2vec* [18], *GloVe* [22], and *fasttext* [8], to context-aware embeddings, such as *ELMo* [23], *OpenAI GPT* [25,26,9], and *BERT* [12]. Contextual embedding models such as the aforementioned have led to an evolution of the NLP paradigm, allowing the use of Transfer Learning techniques, with models pre-trained on huge corpora and fine-tuned to achieve state-of-the-art results in specific tasks for which much smaller data is available [37].

More and more of these DL methods are being applied in the field of clinical NLP, including in the extraction of ICD codes from text data [36]. For this task, the annual CLEF eHealth IE challenges have had significant contributions with

the application of state of the art models on multilingual clinical text corpora [20,21].

The contributions of Amin et al. [6] and Sanger et al. [27] for the CLEF eHealth 2019 IE challenge [21] illustrate the capabilities of deep NLP language models such as *BERT* to extract ICD10 codes from clinical text. For this particular challenge, the goal was to extract ICD10 codes from non-technical summaries of animal experiments, written in German.

Amin et al. [6] applied an English version of *BioBERT* (*BERT* model trained on biomedical text data) [17] on machine-translated versions of the German summaries for multi-label classification. The best results were achieved by performing an ensemble of the predictions using the *BioBERT* model and a Code Attentive LSTM network [6] with pre-trained PubMed *word2vec* embeddings,[1] reaching an $F1$ score of 0.78 on the test set of the challenge.

Sanger et al. [27] adapted the multilingual version of *BERT*, adding a linear output layer to the sequence embedding generated by the model, behaving as a one-vs-rest classification task for each of the ICD10 codes present in the training set. The ensemble of different instances of the trained model (using different random seeds) was also studied, but the single *BERT* multi-label model achieved the best results of the challenge ($F1$ metric of 0.80 on the test set).

Both of these contributions show the versatility of deep language models for ICD10 code extraction, and in particular of the *BERT* architecture, achieving the top results for this challenge.

## 4 Methodologies

Two runs were submitted for subtasks 1 and 2, comprising two distinct methodologies to tackle this challenge: Conditional Random Fields (CRF) and the *BERT* Deep Learning Model. For subtask 3, we submitted a single run using the *BERT*-based model.

Both methodologies tackle all subtasks at once, by considering the challenge as a NER problem, similar to what is described in Subtask 3. In this case, each token present in a given clinical report is classified using the available ICD10 codes (including an *O* tag for tokens that do not have a code associated with them). The predictions for Subtask 1 and 2 are derived from this NER schema by identifying all predicted ICD10-CM and ICD10-PCS codes (respectively) in a given clinical report, and associating them to the clinical report identifier, in a descending order of confidence. Thus, the only changing factor across methodologies is how token classification is performed: using CRF, or using BERT.

Each of the tested methodologies was implemented in Python and is described below in more detail.

[1] https://archive.org/details/pubmed2018_w2v_400D.tar

### 4.1 Conditional Random Fields

CRFs are a commonly used technique for NER, since they take into account context around neighbouring words to create a statistical model that can infer their type [29].

For text pre-processing, namely tokenization, lemmatisation, Part-of-Speech tagging, among other morphological token characteristics extraction, the StanfordNLP toolbox `stanza`[2] [24] is used, taking advantage of the available Universal Dependencies and NER Spanish models.

The methodologies applied for ICD10 code extraction are similar to those described by Tawara et al. [32], namely the calculation of features and score. In addition to these, other features are considered relating to string search and matching with a dictionary of ICD10 code descriptions. These features are then considered alongside the extracted n-grams and are used as input for the CRF model [15,30].

This approach is used to produce two distinct models:

**CRF CM** CRF model trained exclusively on ICD10-CM codes;
**CRF PCS** CRF model trained exclusively on ICD10-PCS codes.

Each of these models used to create predictions for subtasks 1 and 2, respectively, identifying all codes present in a given report and ranking them by confidence.

### 4.2 *BERT*

A schema of the steps followed for the *BERT* methodology is shown in Figure 1.

The *BERT* model was employed as a means of performing NER on the clinical reports of the dataset. Since the reports are written in Spanish, it is adequate to use models pre-trained on Spanish corpora; thus we use the $BETO_{base,uncased}$ pre-trained model [10], available[3] on the `transformers`[4] Python package [33]. This model was pre-trained on text that was first pre-processed to lowercase. $BETO_{base,cased}$,[5] where words can remain capitalised, was also considered, but since they provided similar results, the uncased version was chosen for simplicity.

Since $BETO_{base,uncased}$ (henceforth *BETO*) is a pre-trained model, most of the computational effort has already been performed, with the model only requiring fine-tuning for the NER task. This is performed by addition of a linear classification layer for the token embedding outputs of the model, and training all model parameters with this new task [12].

However, since the type of text present in clinical reports can vastly differ from commonly used corpora for model pre-training, a first Language Modelling

[2] https://stanfordnlp.github.io/stanza/
[3] https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased
[4] https://github.com/huggingface/transformers
[5] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased

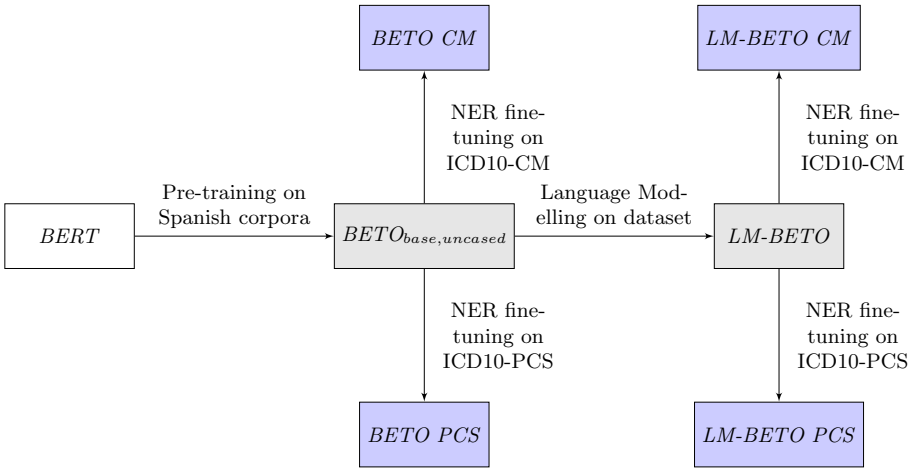


**Fig. 1.** *BERT*-based methodology steps, illustrating the pre-training, language modelling, and NER fine-tuning steps and their relations to the evaluated models (in blue).

(LM) step was considered, where the *BETO* model is first tuned on the *train* set of the challenge's dataset, by performing Masked Language Modelling and Next Sentence Prediction (both unsupervised tasks) [12]. This is expected to further improve results by modelling the particular architecture, choice of words, and medical jargon used in clinical reports.

Furthermore, since ICD10-CM and ICD10-PCS codes differ in application, it is reasonable to separate the classification of each into two distinct models. Therefore, each pre-trained *BETO* model is fine-tuned on two distinct NER tasks for each type of ICD code, generating two classifiers: *BETO CM* and *BETO PCS*. Each of these is used to predict exclusively ICD10-CM or ICD10-PCS codes, respectively, with their predictions combined to identify all relevant codes in a given text.

Consequently, four different *BERT*-based models are developed and tested for NER:

**BETO CM** Pre-trained on Spanish corpora, fine-tuned on ICD10-CM NER;
**LM-BETO CM** Pre-trained on Spanish corpora, further LM tuning on the challenge's dataset, fine-tuned on ICD10-CM NER;
**BETO PCS** Pre-trained on Spanish corpora, fine-tuned on ICD10-PCS NER;
**LM-BETO PCS** Pre-trained on Spanish corpora, further LM tuning on the challenge's dataset, fine-tuned on ICD10-PCS NER.

**Model Training** The dataset was first pre-processed and converted to a NER dataset, with segmented sentences. Tokenization is performed by the *BERT* WordPiece tokenizer [12].

The *BETO* model was trained for LM for 50 epochs, using a linearly decreasing learning rate, starting at $5 \times 10^{-5}$. Training was done with a batch size

of 16 and a block size of 256 (maximum number of tokens per input sequence). This block size was found to be sufficient, since most sentences have a much smaller number of tokens present. According to the results of this training step, a suitable number of epochs was chosen as a basis for the *LM-BETO* models, to avoid overfitting.

All models are trained for NER for 15 epochs, with a batch size of 8, and block size of 256. The learning rate is determined by a cosine scheduler with warmup (2 epochs) and hard restarts (2 cycles), with a maximum of $5 \times 10^{-5}$.

## 5 Results

Two main sets of results are here reported, based on the evaluation on the *dev* (Subsection 5.1) and the *test* (Subsection 5.2) set.

Analysing the results on the *dev* set, two runs were submitted for evaluation: one using the *CRF* methodology, and another based on the *LM-BETO CM* and *LM-BETO PCS* models. The *test* set results were provided by the task evaluators[6] after run submission.

### 5.1 *Dev* Set Results

**Language Modelling Results** The evolution of LM training of *BETO* on the *train* set can be seen on Figure 2, with both training loss and *dev* set perplexity shown for each training epoch. Perplexity is a commonly used metric for LM, which measures how good a language model is at predicting an unknown sample, and is described in Equation 1, with $H(m)$ being the cross-entropy loss of a given model $m$. Lower perplexity values indicate a better predicting language model.

$$PP(m) = 2^{H(m)} \tag{1}$$

As seen of Figure 2, the perplexity decreases with more training time, although it somewhat stabilizes after approximately 10 epochs. To avoid overfitted *LM-BETO* models, their 10 epoch weights are used henceforth. This model is then used as a basis for fine-tuning *LM-BETO CM* and *LM-BETO PCS* on NER of ICD codes.

**NER Results** The results obtained for NER on the *dev* set for all considered trained models are shown in Table 1. The considered metrics are the micro-averaged precision ($P$), recall ($R$), and F1-score ($F1$).

Note that this evaluation was performed considering the NER task, i.e., all presented metrics are on a token by token basis. Furthermore, although models were only trained on the codes present in the *train* set, metrics are shown considering all unique *train* and *dev* ICD10 codes.

[6] https://github.com/TeMU-BSC/CodiEsp-Evaluation-Script

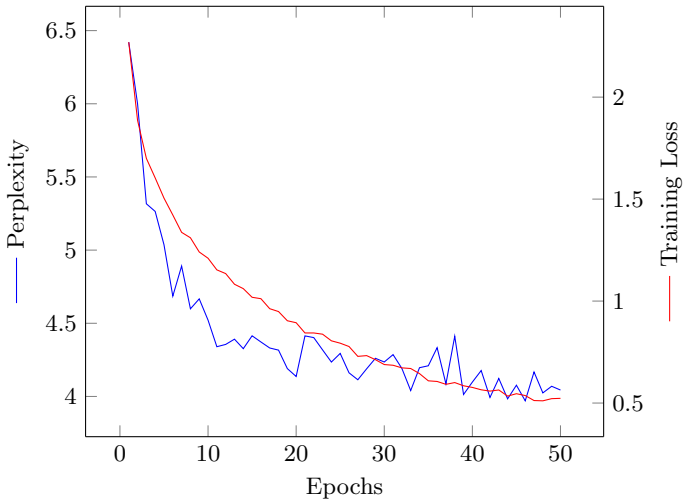


**Fig. 2.** Evolution of perplexity and training loss during language model training of *BETO*.

**Table 1.** NER results on the *dev* set.

| | P | R | F1 |
|---|---|---|---|
| CRF CM | **0.693** | 0.304 | **0.513** |
| BETO CM | 0.498 | 0.477 | 0.487 |
| LM-BETO CM | 0.498 | **0.480** | 0.489 |
| CRF PCS | 0.335 | **0.582** | **0.439** |
| BETO PCS | 0.440 | 0.341 | 0.384 |
| LM-BETO PCS | **0.443** | 0.344 | 0.388 |

### 5.2 *Test* Set Results

The submitted runs were evaluated by the task organizers, with reported results shown in Tables 2, 3, and 4, for subtasks 1, 2, and 3, respectively. Bold metrics indicate the official metric used for model evaluation. Bold values indicate the model that achieved the best value for a particular metric.

For the subtasks, three distinct evaluation modes were considered:

**All codes** Considers all unique ICD10 codes present in the *train*, *dev*, and *test* sets;

**Train + Dev codes** Considers only codes present in the *train* and *dev* sets, codes that are unique to the *test* set are ignored;

**Category** Only consider as labels the first 3 digits of ICD10-CM codes, and the first 4 digits of ICD10-PCS codes.

For Tables 2 and 3, the metrics used are the Mean Average Precision (MAP) and the Mean Average Precision at $K$ (MAP@K) ($K = 30$ for ICD10-CM codes, and $K = 10$ for ICD10-PCS codes) [28]. Micro averaged precision (P), recall (R) and F1-score (F1) are also reported.

**Table 2.** Subtask 1 (ICD10-CM codes assignment) *test* set results.

| | All codes | | | | | Train + Dev codes | | | | | Category | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **MAP** | MAP@30 | P | R | F1 | MAP | MAP@30 | P | R | F1 | P | R | F1 |
| CRF CM | 0.239 | 0.239 | **0.759** | 0.198 | 0.314 | 0.286 | 0.286 | **0.759** | 0.230 | 0.354 | **0.835** | 0.238 | 0.370 |
| LM-BETO CM | **0.517** | **0.517** | 0.551 | **0.638** | **0.591** | **0.604** | **0.603** | 0.551 | **0.743** | **0.633** | 0.624 | **0.736** | **0.676** |

**Table 3.** Subtask 2 (ICD10-PCS codes assignment) *test* set results.

| | All codes | | | | | Train + Dev codes | | | | | Category | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **MAP** | MAP@10 | P | R | F1 | MAP | MAP@10 | P | R | F1 | P | R | F1 |
| CRF PCS | 0.407 | 0.407 | **0.537** | 0.432 | 0.479 | 0.468 | 0.468 | **0.537** | 0.524 | 0.530 | **0.591** | 0.476 | 0.527 |
| LM-BETO PCS | **0.445** | **0.444** | 0.454 | **0.527** | **0.488** | **0.509** | **0.508** | 0.454 | **0.639** | **0.531** | 0.509 | **0.579** | **0.541** |

For Table 4, the used metrics are the micro-averaged precision, recall, and F1-score. For this particular subtask, correct predictions are only considered when the correct code is predicted and its reference position is also correct, with an error tolerance of 10 characters.

**Table 4.** Subtask 3 (Explainable AI) *test* set results.

| | All codes | | | Train + Dev codes | | |
|---|---|---|---|---|---|---|
| | P | R | **F1** | P | R | F1 |
| LM-BETO CM+PCS | 0.534 | 0.478 | 0.505 | 0.534 | 0.562 | 0.548 |

## 6 Discussion

Tables 2 and 3 show the positive results obtained using the *BERT* methodology, with an achieved MAP of 0.517 and 0.445 for ICD10-CM and ICD10-PCS codes, respectively. Furthermore, a F1 score of 0.505 was obtained for subtask 3, which considering the large amount of possible codes (2172 ICD10-CM and 696 ICD10-PCS), shows DL models can effectively perform NER on unconstrained clinical texts and extract clinically relevant information from them.

Interestingly, CRF methodologies achieved the best NER results on the *dev* set evaluations, as seen on Table 1. Nevertheless, precision and recall are considerably unbalanced, when compared with the *BERT*-based classifiers, which may indicate the tendency of CRF models to over- or under-estimate ICD10 codes. This possibility is further corroborated by the results obtained in the *test* set, with high precision scores, but low recall, resulting in lower F1 scores. This methodology appears to be more conservative on its labelling, returning a low number of codes that are mostly correct, but many instances of ICD10 codes are ignored. This can be an indication of an increased difficulty in labelling rarer codes, with the model only confidently labelling a token as a code if it has a particular high frequency in the training data. A more in-depth analysis of the models' behaviours is required, namely threshold analysis, and precision-recall trade-off.

Although the models have similar results for the identification of ICD10-PCS codes (only differing in 0.04 on MAP), they are worse than the best results for the first subtask, even when there is a considerable less amount of ICD10-PCS codes to consider. A possible reason for this may be how medical procedures are referenced in clinical text, which differs from how medical conditions are mentioned: many procedures are often encompassed in one or two words of text, with much of its underlying information (such as location of the procedure, method, etc) implicit or scattered along the report. This makes human coding trivial, since humans can detect this implicit information across long texts, but severely hampers performance of both employed methodologies.

## 7 Conclusions

In this paper we present two methods for ICD10 code extraction from non-structured clinical text in Spanish, achieving a MAP of 0.517 and 0.445 for ICD10-CM and ICD10-PCS codes, respectively, and a F1 score of 0.505 for

NER. This is achieved by employing two *BERT*-based models, both LM tuned to the dataset, and fine-tuned on NER of ICD10-CM and ICD10-PCS codes.

It is important to note that this *BERT*-based methodology was applied for Spanish clinical texts, but could have easily been applied to different languages, simply by using a model that is pre-trained on that specific language, or using a multilingual model, and performing fine-tuning as described here.

The achieved results show the flexibility of novel DL based NLP models for the execution of a number of tasks on several different fields of application, taking advantage of pre-trained models to fine-tune classifiers with little computational effort and small amount of data, bringing Transfer Learning to NLP.

### 7.1 Future Work

The aforementioned methodologies can be significantly improved in the future to create an even more robust system that can tackle a larger number of ICD10 codes.

A clear gap in the DL methodology is the lack of input provided by the ICD10 code descriptions, which often contain very relevant information regarding certain aspects of a given condition, disease or procedure. Moreover, there exists vast data online regarding these same codes and the underlying concepts they represent, which can be taken advantage to build a more robust system, employing techniques similar to those by Bai et al. [7].

*BERT* models can also be improved for NER with several different techniques. For instance, Souza et al. [31] add a CRF layer to *BERT* to improve Portuguese NER, allying the transfer capabilities of *BERT* with the structure predictions of CRF. This can also be considered for this challenge, since the designations of conditions or procedures follow a given structure, which can be captured more effectively by CRFs.

Finally, the scope of the solution here presented was limited to the ICD10 codes present in the training set, which are a very small percentage of the total number of codes in this Nosology system (98288 ICD10-CM and 87170 ICD10-PCS codes). A truly robust ICD IE system would have the possibility to predict any code, as well as have an inherent representation of their hierarchical structure, being able to predict the most accurate code for a given sequence when possible, or a more general but suitable code when not.

## References


1. 2020 ICD-10-CM — CMS, https://www.cms.gov/Medicare/Coding/ICD10/2020-ICD-10-CM, accessed on 2020-07-06
2. 2020 ICD-10-PCS — CMS, https://www.cms.gov/Medicare/Coding/ICD10/2020-ICD-10-PCS, accessed on 2020-07-06
3. eCIE-Maps - CIE-10-ES Diagnósticos, https://eciemaps.mscbs.gob.es/ecieMaps/browser/index_10_mc.html, accessed on 2020-07-06
4. eCIE-Maps - CIE-10-ES Procedimientos, https://eciemaps.mscbs.gob.es/ecieMaps/browser/index_10_pcs.html, accessed on 2020-07-06

5. WHO — International Classification of Diseases (ICD) Information Sheet, http://www.who.int/classifications/icd/factsheet/en/, accessed on 2020-07-03
6. 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. Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum p. 15 (2019)
7. Bai, T., Vucetic, S.: Improving Medical Code Prediction from Clinical Text via Incorporating Online Knowledge Sources. In: The World Wide Web Conference. pp. 72–82. WWW '19, Association for Computing Machinery (2019), DOI: 10.1145/3308558.3313485, https://doi.org/10.1145/3308558.3313485
8. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association of Computational Linguistics **5**, 135–146 (2017), DOI: 10.1162/tacl_a_00051, https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00051
9. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners (2020), http://arxiv.org/abs/2005.14165
10. Cañete, J., Chaperon, G., Fuentes, R., Pérez, J.: Spanish pre-trained BERT model and evaluation data. In: To Appear in PML4DC at ICLR 2020 (2020)
11. Crammer, K., Dredze, M., Ganchev, K., Pratim Talukdar, P., Carroll, S.: Automatic Code Assignment to Medical Text. In: Biological, Translational, and Clinical Language Processing. pp. 129–136. Association for Computational Linguistics (2007), https://www.aclweb.org/anthology/W07-1017
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019), http://arxiv.org/abs/1810.04805
13. Farkas, R., Szarvas, G.: Automatic construction of rule-based ICD-9-CM coding systems. BMC Bioinformatics **9**(3), S10 (2008), DOI: 10.1186/1471-2105-9-S3-S10, https://doi.org/10.1186/1471-2105-9-S3-S10
14. 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., Cappellato, L., Ferro, N. (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)
15. Greene, E.: Extracting Structured Data From Recipes Using Conditional Random Fields, https://open.blogs.nytimes.com/2015/04/09/extracting-structured-data-from-recipes-using-conditional-random-fields/, accessed on 2020-07-15
16. Koopman, B., Zuccon, G., Nguyen, A., Bergheim, A., Grayson, N.: Automatic ICD-10 classification of cancers from free-text death certificates. International Journal of Medical Informatics **84**(11), 956–965 (2015), DOI: 10.1016/j.ijmedinf.2015.08.004, http://www.sciencedirect.com/science/article/pii/S1386505615300289
17. 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 p. btz682 (2019), DOI: 10.1093/bioinformatics/btz682, http://arxiv.org/abs/1901.08746

18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013), http://arxiv.org/abs/1301.3781
19. 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)
20. Neveol, A., Robert, A., Grippo, F., Morgand, C., Orsi, C., Pelikan, L., Ramadier, L., Rey, G., Zweigenbaum, P.: CLEF eHealth 2018 Multilingual Information Extraction task overview: ICD10 coding of death certificates in French, Hungarian and Italian. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum p. 18 (2018)
21. Neves, M., Butzke, D., Dorendahl, A., Leich, N., Hummel, B., Schonfelder, G., Grune, B.: Overview of the CLEF eHealth 2019 Multilingual Information Extraction. Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum p. 9 (2019)
22. Pennington, J., Socher, R., Manning, C.: GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics (2014), DOI: 10.3115/v1/D14-1162, https://www.aclweb.org/anthology/D14-1162
23. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations (2018), http://arxiv.org/abs/1802.05365
24. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python Natural Language Processing Toolkit for Many Human Languages (2020), http://arxiv.org/abs/2003.07082
25. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving Language Understanding by Generative Pre-Training p. 12 (2018)
26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners p. 24 (2019)
27. Sanger, M., Weber, L., Kittner, M., Leser, U.: Classifying German Animal Experiment Summaries with Multi-lingual BERT at CLEF eHealth 2019 Task. Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum p. 12 (2019)
28. Schütze, H., Manning, C.D., Raghavan, P.: Introduction to Information Retrieval, vol. 39. Cambridge University Press Cambridge (2008)
29. Settles, B.: Biomedical Named Entity Recognition using Conditional Random Fields and Rich Feature Sets. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (NLPBA/BioNLP). pp. 107–110. COLING (2004), https://www.aclweb.org/anthology/W04-1221
30. Silva, N., Ribeiro, D., Ferreira, L.: Information Extraction from Unstructured Recipe Data. In: Proceedings of the 2019 5th International Conference on Computer and Technology Applications - ICCTA 2019. pp. 165–168. ACM Press (2019), DOI: 10.1145/3323933.3324084, http://dl.acm.org/citation.cfm?doid=3323933.3324084
31. Souza, F., Nogueira, R., Lotufo, R.: Portuguese Named Entity Recognition using BERT-CRF (2020), http://arxiv.org/abs/1909.10649
32. Tawara, Y., Omura, M., Miura, M.: Incorporating Unsupervised Features into CRF based Named Entity Recognition. In: Proceedings of the 11th NTCIR Conference. p. 4 (2014)

33. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: HuggingFace's transformers: State-of-the-art natural language processing **abs/1910.03771** (2019)
34. World Health Organization: International Classification of Diseases : [9th] Ninth Revision, Basic Tabulation List with Alphabetic Index. World Health Organization (1978), https://apps.who.int/iris/handle/10665/39473, accepted: 2012-06-16T14:05:20Z Journal Abbreviation: ICD-9 : basic tabulation list with alphabetic index
35. World Health Organization: International Statistical Classification of Diseases and Related Health Problems. World Health Organization (2016), oCLC: 910334285
36. Wu, S., Roberts, K., Datta, S., Du, J., Ji, Z., Si, Y., Soni, S., Wang, Q., Wei, Q., Xiang, Y., Zhao, B., Xu, H.: Deep learning in clinical natural language processing: A methodical review. Journal of the American Medical Informatics Association **27**(3), 457–470 (2020), DOI: 10.1093/jamia/ocz200, https://academic.oup.com/jamia/article/27/3/457/5651084
37. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent Trends in Deep Learning Based Natural Language Processing (2018), http://arxiv.org/abs/1708.02709