SehMIC: Semi-hierarchical Multi-label ICD code Classification

Sedigheh Eslami[1,2], Peter Adorjan[1], and Christoph Meinel[2]

[1] Data4Life, Germany
{sedigheh.eslami, peter.adorjan}@data4life.care
[2] Hasso Plattner Institute, Germany
{sedigheh.eslami, christoph.meinel}@hpi.de

Abstract. Automatic ICD code assignment to clinical notes is a beneficial, but challenging task due to the large number of possible ICD codes and a small number of available data. It becomes even more challenging in multilingual settings with resource-poor languages, in which the number of available annotated *textual* data is generally very small. In this work, we present *SehMIC*, a semi-hierarchical multi-label classification approach which leverages the knowledge about the structure of ICD codes to assign them to Spanish discharge letters. This approach classifies different sections of the ICD code separately for a given letter. It achieves the final ICD code by concatenation of the predicted code sections and pruning the unlikely combinations by using an empirical a priori distribution. Moreover, we utilize a transfer learning approach using pre-trained multilingual BERT to achieve contextual document representations for Spanish discharge letters. Data augmentation is also performed in order to exploit more data in the learning process. SehMIC achieves 0.1 and 0.004 MAP scores on the dev and test datasets, respectively. This work is done by our nlp4life team at CLEF eHealth 2020 Task 1 challenge on Multilingual Information Extraction.

Keywords: Automated ICD code assignment · Multi-label classification · Transfer learning · multilingual BERT.

1 Introduction

Motivation. Electronic health records (EHR) include a collection of patients' health related longitudinal data [10]. They contain patients' demographic data, medical histories, symptoms, diagnoses, etc both in structured and unstructured text format. International Classification of Diseases (ICD) codes are diagnostic codes used in EHRs in order to uniquely describe the patient's diagnosis for

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

CEUR-WS.org/Vol-2696/paper_179.pdf

billing and reimbursement purposes [18]. Healthcare systems train several human coders who specifically learn the medical terminologies, the ICD coding system and its rules so that they can undertake assigning the ICD codes to patients' records manually. This process is not only time consuming and expensive, but also intrinsically introduces human errors in detecting the correct codes. Therefore, it is beneficial to develop an automated computational solution to detect the associated ICD codes for given clinical notes. In this work, we investigate how to assign ICD codes directly from discharge letters since it is assumed that discharge letters contain the diagnosis ground truths along with the symptoms, procedures, examinations information of the patient [11,21].

Related work. Accurate automated ICD code assignment to clinical texts is a challenging problem. To name a few reasons: 1. clinical texts contain many typos, specific medical terminologies and keywords, 2. the number of possible ICD codes is huge and respectively, there is not enough samples per ICD code to learn from, 3. real-world data suffers from the imbalanced data problem. This task has been previously investigated via rule-based [9], machine learning [5, 19, 26] and deep learning based [2, 11, 12] approaches. Rule-based systems require human experts to find the patterns in text and design the rules. These manual human efforts make the rule-based approaches difficult to scale. In contrast, learning-based approaches mostly depend on the underlying data distributions to find common patterns and decision procedures. With the recent success of deep learning in language modeling and contextual word embedding solutions, end-to-end deep neural networks have been studied for automated ICD code assignment and achieved competing results [2,16]. Recently, this task has also been carried out in multilingual settings [7,8,17] . In [1], authors utilize a transfer learning approach using Bidirectional Encoder Representation from Transformers (BERT) [6] for the bilingual German-English automated ICD code assignment.

Our contributions. In this paper, we describe our work on the *ICD10-CM code assignment* subtask from CLEF eHealth 2020 challenge Task 1 [15]. We developed a *semi-hierarchical* multi-label classifier by leveraging the knowledge about the structure of the labels in order to assign ICD10-CM codes to Spanish discharge letters. We fine-tuned multilingual BERT in each hierarchy of the classification. Additionally, we applied a data augmentation mechanism in order to exploit more diverse samples per label in the learning phase.

2 Problem and concepts definition

The following set of notations is used throughout this paper:

- Vocabulary of words $V = \{w_1, w_2, \ldots, w_v\}$ of size v,
- Set of word-embeddings $E = \{e_1, e_2, \ldots, e_v\}$ of size v, in which $e_i \in \mathbb{R}^d$ is the word embedding vector for the word w_i,
- Set of discharge summaries $S = \{s_1, s_2, \ldots, s_n\}$, in which s_j is a sequence of words from the vocabulary V,

- Set of features $X = \{X_1, X_2, \ldots, X_n\}$, in which $X_j \in \mathbb{R}^m$ is the contextual feature vector representing discharge letter s_j,
- Set of all labels $L = \{l_1, l_2, \ldots, l_\ell\}$ corresponding to ICD codes,
- For a given discharge summary s_j, we represent the set of associated labels as $L_j = \{0,1\}^\ell$.

Given $\{(X_j, y_j)\}_{j=1}^n$ where $X_j \in \mathbb{R}^m$ and $y_j \in L_j$, our objective is to train a multi-label classifier $C : \mathbb{R}^m \longrightarrow \{0,1\}^\ell$ such that $C(X_j) = y_j$ for any $j \in \{1, \ldots, n\}$.

3 Approach

In this section, we describe our proposed approach for multi-label ICD code assignment to discharge letters. Our approach includes two main steps: 1. data augmentation 2. semi-hierarchical multi-label classification (SehMIC).

3.1 Data augmentation

In learning-based approaches, the more and diverse data we have, the better our model learns the underlying distributions and patterns in the data. Data augmentation is used in several fields, e.g., computer vision and natural language processing, in order to increase the diversity of the training data without actually collecting new sets of data. In the CLEF 2020 eHealth challenge, we perform data augmentation primarily because there exists very few discharge letters for a lot of the ICD codes in the training data. Inspired by the work in [24], we use a lexical substitution approach using word-embeddings. We create the *Synonyms Dictionary* (SD) based on the similarity of the words in the embedding space using the word-embeddings set E. We define *synonyms* of each word to be the set of all the words whose similarity in the embedding space is greater than a given similarity threshold θ,

$$\begin{gathered}\mathrm{SD}(w_i) : w_i \longrightarrow \mathrm{synonyms}(w_i), \\ \mathrm{synonyms}(w_i) = \{w_j\} \\ \text{s.t.} \\ \mathrm{sim}(e_i, e_j) \geq \theta, \text{ for all } j \in \{1, \ldots v\} \text{ where } i \neq j.\end{gathered}$$

Notice that depending on the threshold θ, a word can end up with an empty set of synonyms. Afterwards, given the SD and a discharge letter s_j, we iterate over the words in the letter, randomly select a synonym from its set of synonyms stored in SD, and finally substitute the word with the selected synonym. We repeat this process k_j times per letter s_j in which:

$$k_j = \frac{\max_{l \in L}(\text{number of samples for } l)}{\min_{l' \in L_j}(\text{number of samples for } l')}.$$

The reason for repeating the text generation k_j times per letter s_j is two-fold: first, in order to balance the label distribution in terms of the number of available samples for each label, as a result, the data augmentation generates fewer sample for the majority labels and more samples for the minority ones; secondly, since we have multiple synonyms per word, repetition of text generation utilizes the synonyms as many as possible per word in the augmentation step. Algorithm 1 provides a summary pseudo-code of our data augmentation approach.

Algorithm 1 Data augmentation

Input: data $\{s_j, y_j\}_{j=1}^{n}$, SD
Output: augmented labeled data $\{a_j, y_j\}_{j=1}^{k}$

procedure AUGMENT(data, SD)
    $aug\_data \leftarrow$ `init_empty_data()`
    $max\_cnt \leftarrow$ `max`(number of samples over all labels)
    $L \leftarrow$ `unique_labels`(data)
    $min\_cnt\_dict \leftarrow$ `dict()`
    for l `in` L **do**
        $min\_cnt\_dict(l) \leftarrow$ `min`(number of samples for l)
    for *text*, *label* `in data` **do**
        $K \leftarrow \dfrac{max\_cnt}{min\_cnt\_dict(label)}$
        for k `in range(`K`)` **do**
            $aug\_text \leftarrow$ `""`
            for w `in` *text* **do**
                $syn \leftarrow$ `random(SD(`w`), 1)`
                $aug\_text \leftarrow aug\_text + syn$
            aug_data`.append(`aug_text`,` *label*`)`
    `return` aug_data

3.2 SehMIC

Often in the task of automated ICD code assignment, the number of available samples per label is not sufficient. In this case, a flat classifier, i.e., a classifier that does not consider an inherent hierarchy between the labels, cannot receive enough samples per label to learn from. Thus, minimizing the training error will lead to overfitting. On the other hand, training a full-hierarchical classification system requires training thousands of local-classifiers considering the intermediate hierarchies [22] which is time consuming. In order to overcome these problems, we propose *SehMIC*, a heuristic semi-hierarchical multi-label classification solution in which we leverage the knowledge we have about the hierarchical structure of ICD codes. In this work, we explain our method with regards to ICD10-CM codes, but the same concepts can be applied for other types of ICD codes as well.

ICD10-CM codes are three to seven character codes separated by a dot. The first three characters describe the *category* of the medical condition. *Details*

about the condition in the category section are represented by the characters appearing after the dot. The first character in the category code is called the *chapter* code which describes the main type of the medical condition, e.g., injury. The next two characters provide more information about the problem in the chapter code, e.g., location or the severity of the problem [18]. Figure 1 depicts this structure with an example ICD10-CM code.

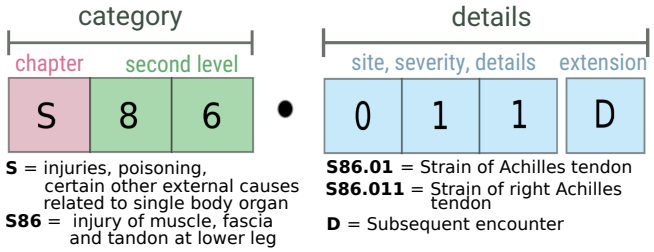

Fig. 1. Example of ICD10-CM code structure

Considering this structure, we translate the ICD code classification as follows:

1. Solve the multi-label classification of *chapter* given the discharge letter.
2. Solve the multi-label classification of the *second level* given the discharge letter.
3. Achieve the preliminary candidate *category* codes by concatenating the results from 1 and 2.
4. Prune the preliminary category codes with respect to unlikely code combination by multiplying an empirically estimated conditional a priori distribution and reach the final category codes:

$$P(\text{second level}|\text{discharge letter}) \times P(\text{second level}|\text{chapter})$$

5. Solve the multi-label classification of *details* given the discharge letter.
6. Concatenate the results from 4 and 5 to reach the preliminary *ICD10-CM* codes.
7. Prune the codes with respect to unlikely details and category combinations by multiplying an empirically estimated conditional a priori distribution and reach the final ICD10-CM codes:

$$P(\text{details}|\text{discharge letter}) \times P(\text{details}|\text{category}).$$

Figure 2 illustrates this approach with an example. For a given discharge summary, SehMIC predicts *S* and *T*, *89* and*99*, *02* and *9* for chapter, second level and details codes, respectively. Concatenating the predicted chapter and second level codes results in *S89, S99, T89, T99* from which *T89, T99* will be pruned by the conditional a priori distribution as they are invalid ICD codes and their corresponding probabilities are zero. Similarly, combining and pruning the category codes and the predicted detail codes results in the final *S89.02, S89.9,*

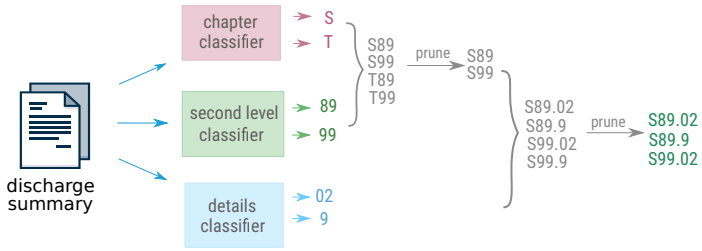

Fig. 2. The process of predicting ICD10-CM code via SehMIC

S99.02 ICD codes to be assigned to the discharge summary.

Multi-label classification. All of the classifications in steps 1, 2 and 5 are multi-label, i.e., multiple labels are predicted per sample discharge letter. Two main approaches exist for performing multi-label classification: first, *problem transformation* methods, i.e., methods that transform the multi-label problem into many single-label classification problems; second, *algorithm adaptation* methods, i.e., methods that directly adapt algorithms to handle the multi-label classification [23]. In this work, we adapt and fine-tune multilingual BERT for sequence classification to directly support multi-label classification. BERT provides a sequence-level contextual embedding represented for the $[CLS]$ special token [6]. Fine-tuning BERT for single-label classification is done by adding a feed forward fully connected layer with softmax activation function in the output layer on top of the sequence level BERT embeddings [6]. In contrast, for the multi-label classification setting we use the sigmoid activation in the output layer. This is because the probabilities computed by sigmoid are independent and do not need to sum up to one. As a result, our network can allow more than one correct label for a given sample. Given a decision probability threshold, we select all the labels whose probability is more that the threshold to be the predicted labels.

4 Experiments

4.1 Dataset

As participants of the CLEF eHealth 2020 challenge [15], we conduct our experiments using the Spanish corpus released in this challenge. The average length of the letters in all three of the training, development and test sets is 350. Table 1 represents a simple statistic over these sets. During the challenge, around 3000 letters were released for the testing phase from which only 250 were the actual test corpus used in the evaluations. The rest of the letters were considered as background texts. The fraction of labels with only one sample in Table 1 illustrates that if we simply ignore the labels with very few samples, we will lose more than half of the labels. Moreover, about 37% of the unique ICD10-CM

codes of the development set are not present in the training set. Similarly, only about 68% of the ICD10-CM codes in test set overlap with the union of the codes in training and development set and the rest are missing. Thus, our data explorations show that this challenge also includes tackling the missing labels problem.

Table 1. CLEF2020 eHealth dataset statistic

	#samples	**#labels**	**Fraction of labels with only one sample**
Train	500	1767	56%
Dev(elopment)	250	1158	64%
Train + Dev	750	2194	53%
Test	250	1143	60%

4.2 Experimental setup

Data augmentation. In the augmentation step, we use pre-trained fastText embeddings [3] from the Spanish Billion Word Corpus and Embeddings project [4] in order to configure the synonyms and create the synonyms dictionary. We set the similarity threshold to 0.7[3] and use cosine similarly to calculate the word embedding similarities. We concatenate train and development discharge letters and perform the data augmentation on the concatenated set in order to unravel the missing labels problem in the training set. Stopwords and the ICD10-CM codes mentioned in the letters are skipped in our setting. The average number of synonyms per word in the resulting synonyms dictionary is 3 and 40% of the words end up with no synonyms. The maximum number of synonyms is 20 in the dictionary. In the training phase, the augmented dataset and the original training set are used together for training the classification models. The final set used for training includes 41750 discharge letters and 2196 unique ICD codes.

Classification setup. We fine-tuned the pre-trained *bert-base-multilingual-cased* model[4] using the Hugging Face Transformers library [25] which is based on Pytorch [20]. Since our problem is a multi-label classification task, we adapted the *BertForSequenceClassification* class from Hugging Face to use the sigmoid activation on the output layer along with binary cross entropy loss [5]. We set the maximum sequence length to 512 and train each of the three classifiers for 3 epochs with learning rate of 0.00003 and AdamW optimizer [14]. For chapter, category and ICD codes we set the decision threshold to 0.5, 0.001 and 0.001, respectively.

[3] Similarities are normalized values in the range of $[0, 1]$.
[4] More details at huggingface.co/transformers/pretrained˙models.html.
[5] Link to source code: github.com/sarahESL/CLEFeHealth2020-multilabel-bert.

Conditional a priori distributions. In order to calculate the empirical a priori distributions, the following is used. We use $S86.011$ code as example for illustration.

$$p(\text{second level} = \text{“86”}|\text{chapter} = \text{“}S\text{”}) = \frac{\#\text{ samples with category “}S86\text{”}}{\#\text{ samples with chapter code “}S\text{”}}$$

$$p(\text{details} = \text{“011”}|\text{category} = \text{“}S86\text{”}) = \frac{\#\text{ samples with code “}S86.011\text{”}}{\#\text{ samples with category code “}S86\text{”}}$$

4.3 Results and insights

The experimental result of our proposed method is depicted in Table 2. We use the Mean Average Precision (MAP) [13] metric in our evaluations as it was the evaluation metric in the CLEF2020 eHealth challenge. [15]. On the chapter level, our classifier achieves the MAP score of 0.97 and 0.43 on the development and test sets, respectively. Although the MAP score for the category code prediction in the development setting is 0.69, we see a tremendous degradation in the test result. Furthermore, the final ICD10-CM codes are predicted with the MAP score of 0.1 and 0.004 for the development and test sets. This is due to the fact that the data used for training is highly imbalanced and the default BERT does not handle imbalanced classes. Additionally, predicting the second level code directly by the last two characters in category results in misclassifications. This is due to the fact that the same site, severity, etc are represented with different second level codes. For instance, both $Z94.0$ and $S37.0$ codes describe a condition about *kidney*. However, the kidney is represented by 94 and 37 for transplant (Z) and injury (S) conditions. We think modeling a latent semantic variable for the second level code will improve the category prediction performance. The same explanation applies to details code as well.

Table 2. MAP scores on development and test sets

	chapter	**category**	**ICD10-CM**
Dev	0.97	0.69	0.1
Test	0.43	0.008	0.004

We suspect that using the development letters in our data augmentation step causes letters that are very similar to the development set to appear in the augmented data. As a result, our classifiers already have seen some development-like data in their training phases. Therefore, we interpret the dev results in Table 2 as training results.

5 Conclusion

In this work, we presented our (nlp4life team) submission to the CLEF eHealth 2020 Task 1 challenge. This challenge required overcoming imbalanced data distributions and missing labels problems. Additionally, the number of available samples per unique

labels was small, which made it particularly challenging to train a fully flat and supervised classification model. In this work, we proposed a lexical substitution data augmentation and a semi-hierarchical classification approach for assigning ICD10-CM codes to discharge letters. Our approach results in misclassifying a noticeable number of category and ICD codes. In future work, we would like improve these results by modeling the latent semantic variables to improve second level and details code predictions. Moreover, we plan to investigate context-aware approaches using ICD code embeddings in order to improve the classification performance and overcome the missing labels problem.

Acknowledgement

We would like to thank Matthias Steinbrecher for the helpful discussions and comments.

References

1. 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. In: CLEF (Working Notes) (2019)
2. Baumel, T., Nassour-Kassis, J., Cohen, R., Elhadad, M., Elhadad, N.: Multi-label classification of patient notes: case study on icd code assignment. In: Workshops at the thirty-second AAAI conference on artificial intelligence (2018)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
4. Cardellino, C.: Spanish Billion Words Corpus and Embeddings (August 2019)
5. Dermouche, M., Velcin, J., Flicoteaux, R., Chevret, S., Taright, N.: Supervised topic models for diagnosis code assignment to discharge summaries. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 485–497. Springer (2016)
6. 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)
7. Dörendahl, A., Leich, N., Hummel, B., Schönfelder, G., Grune, B.: Overview of the clef ehealth 2019 multilingual information extraction (2019)
8. 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)
9. Goldstein, I., Arzumtsyan, A., Uzuner, Ö.: Three approaches to automatic assignment of icd-9-cm codes to radiology reports. In: AMIA Annual Symposium Proceedings. vol. 2007, p. 279. American Medical Informatics Association (2007)
10. Gunter, T.D., Terry, N.P.: The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. Journal of medical Internet research **7**(1), e3 (2005)

11. Huang, J., Osorio, C., Sy, L.W.: An empirical evaluation of deep learning for icd-9 code assignment using mimic-iii clinical notes. Computer methods and programs in biomedicine **177**, 141–153 (2019)
12. Li, M., Fei, Z., Zeng, M., Wu, F.X., Li, Y., Pan, Y., Wang, J.: Automated icd-9 coding via a deep learning approach. IEEE/ACM transactions on computational biology and bioinformatics **16**(4), 1193–1202 (2018)
13. Liu, L., Özsu, M.T.: Encyclopedia of database systems, vol. 6. Springer New York, NY, USA: (2009)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
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. Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., Eisenstein, J.: Explainable prediction of medical codes from clinical text. arXiv preprint arXiv:1802.05695 (2018)
17. Névéol, 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. In: CLEF (Working Notes) (2018)
18. Organization, W.H.: International statistical classification of diseases and related health problems, vol. 1. World Health Organization (2004)
19. Pakhomov, S.V., Buntrock, J.D., Chute, C.G.: Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. Journal of the American Medical Informatics Association **13**(5), 516–525 (2006)
20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)
21. Prakash, A., Zhao, S., Hasan, S.A., Datla, V., Lee, K., Qadir, A., Liu, J., Farri, O.: Condensed memory networks for clinical diagnostic inferencing. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
22. Sun, A., Lim, E.P.: Hierarchical text classification and evaluation. In: Proceedings 2001 IEEE International Conference on Data Mining. pp. 521–528. IEEE (2001)
23. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM) **3**(3), 1–13 (2007)
24. Wang, W.Y., Yang, D.: That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 2557–2563 (2015)
25. 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. ArXiv **abs/1910.03771** (2019)
26. Yan, Y., Fung, G., Dy, J.G., Rosales, R.: Medical coding classification by leveraging inter-code relationships. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 193–202 (2010)