Convolutional Attention Models with Hierarchical Post-Processing Heuristics at CLEF eHealth 2020

Elias Moons and Marie-Francine Moens

KU Leuven, Belgium
elias.moons@cs.kuleuven.be

Abstract. In this paper, we compare state-of-the-art neural network approaches to the 2020 CLEF eHealth task 1. The presented models use the neural principles of convolution and attention to obtain their results. Furthermore, a hierarchical component is introduced as well as hierarchical post-processing heuristics. These additions successfully leverage the information that is inherently present in the ICD taxonomy.

1 Introduction

In this paper, we compare different neural network approaches in the context of the CLEF eHealth 2020 task 1[2][5]. More specifically, we have submitted predictions for subtasks 1 and 2 which evaluate systems that predict diagnostic and procedural ICD-codes, respectively. Diagnostic codes represent all the different diagnoses and their variants themselves. Procedural codes identify what was done to or given to a patient (medication, surgeries, etc.). Our strategy combines the principles of convolutional neural networks and attention mechanisms. Furthermore, these models are extended with a hierarchical objective, corresponding to the underlying ICD taxonomy. Lastly, hierarchical heuristics are used for post-processing the results.

The dataset consists of 1,000 clinical cases, tagged with various ICD-10 codes by health specialists. The original text fragments are in Spanish but an automatically translated version in English is also provided by the organisers. This version was used in this research as the described models are optimized for English texts. Assessing the influence of using this translated version instead of the original Spanish texts would be an interesting addition in future works. The dataset contains a split of 500 training samples, 250 development samples and 250 test samples. In total the 1,000 documents comprise of 16,504 sentences and 396,988 words, with an average of 396.2 words per clinical case. For the first subtask, these documents are trained with corresponding diagnostic ICD-code tags. For the second subtask, these same documents were trained with their procedural ICD-codes instead. The biggest hurdle while training with this dataset is the size and consequently the small number of training samples for each category present. For the diagnostic ICD-codes for example, there are in total 1,767 different categories spread out over only 500 training documents. Every document is labeled with on average 11.3 different categories and each category is on average represented by 3.2 training examples. Only seven categories have more than 50 training examples. For the case of procedural ICD-codes, these numbers are slightly lower with 563 different categories, 3.1 categories per example and only 2.7 training examples for each category, leading to a very similar distribution. Figure 1 gives a sorted view of all categories present in the diagnostic training dataset (left) as well as the procedural training dataset (right) and the amount of examples tagged with that specific category.

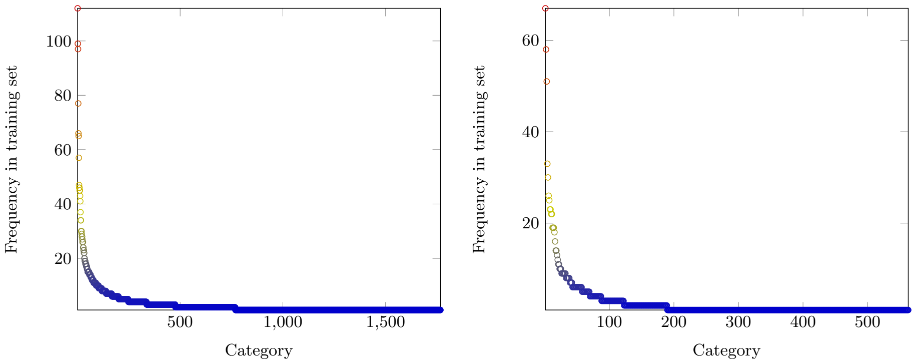

Fig. 1. Category frequencies of CodiEsp training dataset (diagnostic on the left, procedural on the right).

In this paper we hypothesize that exploiting the knowledge of the hierarchical label taxonomy of ICD-10 helps the performance of automated coding when limited training examples that are manually coded are available.

The remainder of this paper is organized as follows. In section 2 related work relevant for the conducted research will be discussed. The evaluated deep learning methods are described in section 3. These methods are evaluated on the benchmark CodiEsp ICD-10 dataset and all findings are reported in section 4. The most important findings will be recapped in section 5.

2 Related Work

The most prominent and more recent advancements in categorizing medical reports with standard codes will shortly be described in this section.

In [7] an hierarchical support vector machine (SVM) is shown to outperform that of a flat SVM. Results were reported based of F-measure scores on the Mimic-II dataset. [3] show that datasets of different sizes and different numbers of distinct codes demand different training mechanisms. For small datasets, feature and data selection methods serve better. The authors have evaluated ICD coding performance on a dataset consisting of more than 70,000 textual EMRs (Electronic Medical Records) from the University of Kentucky (UKY) Medical Center tagged with ICD-9 codes.

A deep learning model that encompasses an attention mechanism is tested by [9] on the Mimic-III dataset. LSTMs are used for both character and word level representations. A soft attention layer here helps in making predictions for the top 50 most frequent ICD-9 codes in the dataset.

More recently, [1] have introduced the Hierarchical Attention bidirectional Gated Recurrent Unit model (**HA-GRU**). By identifying relevant sentences for each label, documents are tagged with corresponding ICD-9 codes. Results are reported both on the Mimic-II and Mimic-III datasets. [6] presents the Convolutional Attention for Multi-Label classification (**CAML**) model that combines the strengths of convolutional networks and attention mechanisms. They propose adding regularization on the long descriptions of the target ICD codes, especially to improve classification results on less represented categories in the dataset. This approach is further extended with the idea of multiple convolutional channels by [8] with max pooling across all channels. The authors also shift the attention from the last prediction layer, as in [6], to the attention layer. [6] and [8] achieve state-of-the art results for ICD-9 coding on the MIMIC-III dataset. As an addition to these models, in this paper a hierarchical variant of each of them is constructed and evaluated. Furthermore, if the target output space of categories follows a hierarchy of labels - as is also the case in ICD coding - the trained models can efficiently use this hierarchy for category assignment [7][10][4]. During categorization the models apply a top-down or a bottom-up approach at the classification stage. In a top-down approach parent categories are assigned first and only children of assigned parents are considered as category candidates. In a bottom-up approach only leaf nodes in the hierarchy are assigned which entail that parent nodes are assigned. The hierarchical structure of a tree leads to various parent-child relations between its categories. For the models discussed in this paper, an hierarchical variant will also be tested which exploits the information of the tree structure and shows that it can enhance the classification performance. Recent research shows the value of these hierarchical dependencies using hierarchical attention mechanisms [1] and hierarchical penalties [11] which are also integrated in this paper.

3 Methods

In this section, we explain the used models for ICD code prediction. First, the preprocessing step is shortly discussed. Then, two recent state-of-the-art models in the field of ICD coding are explained in detail. These models are implemented by the authors following the original papers and are called **DR-CAML** [6] and **MVC-(R)LDA** [8], respectively. We discuss in detail the attention mechanisms and loss functions of these models. Afterwards, as a way of handling the hierarchical dependencies of the ICD-codes, we propose various ways of their integration in all models. This is based on advancements in hierarchical classification as inspired by [11]. Lastly, heuristics are described for post-processing of the predictions given by the models. This leads in section 4 to a clear comparison between all tested models among themselves as well as with their hierarchical novel variants and the introduced post-processing.

3.1 Preprocessing

The preprocessing follows as standard procedure described in [6], i.e., tokens that contain no alphabetic characters are removed and all tokens are put to lowercase. Furthermore tokens that appear in fewer than three training documents are replaced with the 'UNK' token. All documents are then truncated to a maximum length of 2500 tokens.

All discussed models have for each document i as input, a sequence of word vectors x^i as their representation and as output, a set of ICD-codes y^i.

3.2 Convolutional models

This subsection describes the details of recent state-of-the-art models presented in [6] and [8] in the way they are used for the experiments in section 4.

DR-CAML **DR-CAML** is a CNN based model adopted for ICD coding [6]. When an ICD code is defined by the WHO, it is accompanied by a label definition expressed in natural language to guide the model towards learning the appropriate parameter values of the model. For this purpose the model employs a per-label attention mechanism enabling it to learn distinct document representations for each label. It has been shown that for labels for which there are very few training instances available, this approach is advantageous. The idea is that the description of a target code is itself a very good training example for the corresponding code. Similarity between the representation of a given test sample and the representation of the description of a target code gives extra confidence in assigning this label.

In general, after the convolutional layer, DR-CAML employs a per-label attention mechanism to attend to the relevant parts of text for each predicted label. An additional advantage is that the per-label attention mechanism provides the model with the ability of explaining why it decided to assign each code by showing the spans of text relevant for the ICD code.

MVC-(R)LDA Both **MVC-LDA** and **MVC-RLDA**, can be seen as extensions of DR-CAML. Similar to that model, they are based on a CNN architecture with a label attention mechanism that considers ICD coding as a multi-task binary classification problem. The added functionality lies in the use of parallel CNNs with different kernel sizes to capture information of different granularity.

In general, these multi-view CNNs are constructed with four CNNs that have the same number of filters but with different kernel sizes. This convolutional layer is followed by a max-pooling function across all channels to select the most relevant span of text for each filter.

Loss function The loss functions used to train DR-CAML and the multi-view models MVD-(R)LDA are calculated in the same way. The general loss function is the binary cross entropy loss $loss_{BCE}$. This loss is extended by regularization on the long description vectors of the target categories

Given N different training examples x_i. The values of $\hat{y}_l$ and max-pooled vector z_l can be calculated by getting the description of code l out of all L target codes. In this figure and the following formulas β_l is a vector of prediction weights and v_l the vector representation for code l. Assuming n_y is the number of true labels in the training data, the final loss is computed by adding regularization to the base loss function as:

$$\hat{y}_l = \sigma(\beta_l^t v_l + b_l) \tag{1}$$

$$loss_{BCE}(X) = -\sum_{i=1}^{N}\sum_{l=1}^{L} y_l \log{(\hat{y}_l)} + (1 - y_l)\log{(1 - \hat{y}_l)} \tag{2}$$

$$loss_{Model}(X) = loss_{BCE} + \lambda \frac{1}{n_y} \sum_{i=1}^{N}\sum_{l=1}^{L} \|z_l - \beta_l\|_2 \tag{3}$$

3.3 Modelling hierarchical dependencies

In this section we investigate the modelling of hierarchical dependencies as extensions of the models described above. A first part integrates the hierarchical dependencies directly into the structure of the model. This leads to **Hierarchical models**, which are layered variants of the already discussed approaches. The second way hierarchical dependencies are explicitly introduced into the model is via the use of a hierarchical loss function to penalize hierarchical inconsistencies across the model's prediction layer.

Hierarchical models Hierarchical relationships can be shaped directly into the architecture of any of the described models above. The ICD-10 taxonomy can be modeled as a tree with a general ICD root and 4 levels of depth. On the highest level, codes have 1 character, the next 2 levels represent categories with respectively 3 and 4 characters. The rest of the codes are combined in the last layer. This leads to a hierarchical variant of any of the models. In this variant, not 1 but 4 identical models will be trained, one for each of the different layers in the ICD hierarchy (corresponding to the length of the codes).

An overview of the approach is given in figure 2. The input for each layer is partially dependent on an intermediary representation from the previous layer as well as the original input through concatenation of both. Layers are stacked

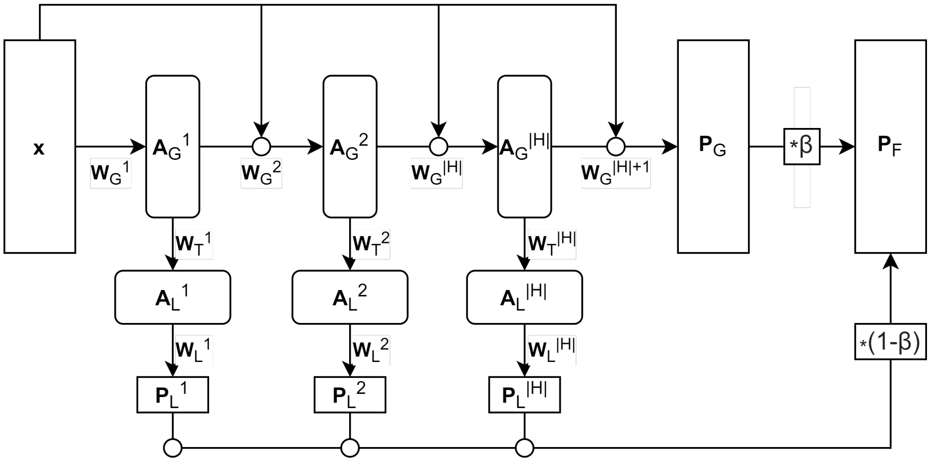

Fig. 2. Overview of hierarchical variant of a model, inspired by [11].

from most to least specific or from leaf to root node in the taxonomy. Models corresponding to different layers will then rely on different features, or characteristics, to classify the input vectors. This way the deepest, most advanced representations, can be used for classifying the most abstract and broad categories. On the other hand, for the most specific categories, word level features can directly be used to make detailed decisions between classes that are very similar.

Hierarchical loss function To capture the hierarchical relationships in a given model, the loss function of the above models can be extended with an additional term. This leads to the definition of a **Hierarchical loss function** ($loss_H$). This loss function penalizes classifications that contradict the inherent ICD hierarchy. More specifically, when a parent category is not predicted to be true, none of its child categories should be predicted to be true. The hierarchical loss between a child and its parent in the tree is then defined as the difference between their computed probability scores, with 0 as a lower bound. More formally, for the entire loss function $loss_{H\_Model}$ for a category of layer X, combining the regular training loss $loss_{Model}$ described above and the hierarchical loss $loss_H$, is calculated as follows:

$$P(X) = Probability(X == True) \tag{4}$$

$$Par(X) = Probability(Parent(X) == True) \tag{5}$$

$$L(X) = True\,label\,of\,X(0\,or\,1) \tag{6}$$

$$loss_H(X) = Clip(P(X) - Par(X), 0, 1) \tag{7}$$

$$loss_{H\_Model}(X) = (1-\lambda)loss_{Model}(X) + \lambda loss_H(X) \tag{8}$$

which leaves a parameter λ to optimize the loss function.[1]

3.4 Hierarchical post-processing

As a final step in the classification process, a heuristical post-processing will be applied to some of the submitted models. All considered heuristics are explained below. They are all reliant on the distance of any pair of target categories in the ICD-10 taxonomy and reweigh the prediction values accordingly. The heuristics are numbered from $H1$ until $H7$ for efficient referencing in the result section.

Node distance (H1) Given all L predictions y_i made for document i by any given model, the new prediction values $y_i^{post_1}$ can be calculated as follows:

$$y_i^{post_1} = \sum_{j=1}^{L} (\frac{y_j}{(1 + dist(i,j))}. \tag{9}$$

The newly calculated prediction values are the result of a weighted sum of all previously calculated prediction values, taking into account the relative distances of all target categories in the ICD taxonomy. In general, $dist(i,j)$ gives the distance between categories i and j in the ICD tree, e.g., the distance between a parent and its child is 1, the distance between two siblings is 2 and the distance of an element to itself is 0.

Node distance from child to ancestor (H2) This heuristic functions the same way as the heuristic described above but differs in behavior if the lowest common ancestor (LCA) of categories i and j which is not j itself. y_j will only be added to the total new score of category i if j is an ancestor of i. This can be formally described as follows:

$$y_i^{post_2} = \sum_{j=1}^{L} dist_{a,c}(i,j) * y_j; \tag{10}$$

$$dist_{a,c}(i,j) = \begin{cases} \frac{1}{(1 + dist(i,j)}, & \text{if ancestor}(i,\ j) == \text{True} \\ 0, & \text{if ancestor}(i,\ j) == \text{False.} \end{cases} \tag{11}$$

Node distance from ancestor to child (H3) This heuristic functions analogous to heuristic $H2$ but in the opposite direction. y_j will only be added to the total new score of category i if i is an ancestor of j. This gives:

$$y_i^{post_3} = \sum_{j=1}^{L} dist_{c,a}(i,j) * y_j; \tag{12}$$

[1] Parameter λ is optimized over the training set.

$$dist_{c,a}(i,j) = \begin{cases} \dfrac{1}{(1 + dist(i,j)}, & \text{if ancestor}(j,\, i) == \text{True} \\ 0, & \text{if ancestor}(j,\, i) == \text{False.} \end{cases} \tag{13}$$

Node distance between ancestors and children (H4) Heuristic $H4$ combines the ideas presented in the previous two heuristics, only adding y_i when either i is an ancestor of j or j is an ancestor of i. Using equations 11 and 13, this evaluates to:

$$y_i^{post_1} = \sum_{j=1}^{L} (dist_{a,c}(i,j) + dist_{c,a}(i,j)) * y_j. \tag{14}$$

Squared node distance (H5) This heuristic functions as heuristic $H1$ but squares the value of its distance function. As a result, it gives relatively more weight to predictions made for categories that are closer the observed category in comparison to $H1$. This leads to the following relationship:

$$y_i^{post_5} = \sum_{j=1}^{L} (\frac{y_j}{(1 + dist(i,j)^2)}. \tag{15}$$

Squared node prediction values (H6) Heuristic $H6$ differs from the first heuristic in that it rescales the starting prediction values y_i. Instead of using the calculated values it will use the squares of these values, making discrepancies in prediction values relatively more prominent. The resulting values can be calculated via:

$$y_i^{post_6} = \sum_{j=1}^{L} (\frac{y_j^2}{(1 + dist(i,j))}. \tag{16}$$

Squared node distances and prediction values (H7) This heuristic combines the ideas that comprise heuristics $H5$ and $H6$, leading to the following relationship:

$$y_i^{post_7} = \sum_{j=1}^{L} (\frac{y_j^2}{(1 + dist(i,j)^2)}. \tag{17}$$

4 Results

For both the subtasks of predicting diagnostic and procedural codes, 5 different models were trained, this was the maximum amount allowed in the competition. Since the size of the dataset was a problem during training, the authors chose to only train models for the top-50 most represented categories in the training dataset. During training of the hierarchical models, ancestors of the top-50 categories were added as well, but only the performance on the original 50 categories was taken into account for calculating the result metrics. A selection of models was chosen aiming for much variety to be able to assess the influence of both proposed models (CAML and MVC-RLDA), the hierarchical objective and post-processing using a heuristic. The chosen models are summarized below and are the same for both subtasks:

1. CAML
2. CAML + hierarchical objective
3. MVC-RLDA + hierarchical objective
4. CAML + hierarchical post-processing H1
5. MVC-RLDA + hierarchical objective + hierarchical post-processing H1

First, one baseline without use of the hierarchy and heuristics was chosen. Since CAML got slightly better results than MVC-RLDA on the development set, this model was selected. Second, to assess the influence the hierarchy can have on the classification results, both CAML and MVC-RLDA models were trained with a hierarchical objective. The last 2 models were chosen with the post-processing heuristic in mind. Only heuristic H1 was chosen for this (based on higher performance on the development set), once in a setting without hierarchical objective (with CAMl) and once with the hierarchical objective (and MVC-RLDA). Since the models used in this paper had a lot of difficulties with the small number of training examples, the prediction probabilities of all categories were rather close together (often in the range of 0.3 to 0.5 instead of from 0.0 until 1.0). For this reason, the prediction files were generated using the top-5 highest predicted categories instead of using a fixed cut-off point. This is not optimal for obtaining a high MAP, where it is better to submit more categories leading to lower performance values. The results obtained by these prediction files are visible in tables 1 and 2 for diagnostic and procedural subtasks respectively.

Table 1. Results on the diagnostic codes subtask.

		MAP	Precision	Recall	F1
Diag.	**CAML**	0.011	0.066	0.029	0.041
	CAML + Hier.	0.015	0.073	0.032	0.044
	MVC-RLDA + Hier.	0.006	0.040	0.018	0.024
	CAML + H1	**0.044**	0.124	0.055	0.076
	MVC-RLDA + Hier. + H1	0.002	0.013	0.006	0.008

For the case of diagnostic codes, visible in table 1, the best performance is achieved by the CAML model in combination with heuristical post-processing $H1$. Adding the heuristic to CAML leads to a clear improvement in classification quality. Comparing CAML with CAML+Hier. leads to the conclusion that the hierarchy can as well lead to an improvement, but it is less prominent than using the post-processing heuristic. Furthermore, it is clear that the MVC-RLDA model gets outperformed by CAML. This is most likely due to the fact that the

Table 2. Results on the procedural codes subtask.

		MAP	Precision	Recall	F1
Diag.	**CAML**	0.007	0.015	0.010	0.012
	CAML + Hier.	**0.020**	0.046	0.020	0.028
	MVC-RLDA + Hier.	nan	nan	0.0	nan
	CAML + Heuristic	**0.017**	0.051	0.034	0.041
	MVC-RLDA + Hier. + Heuristic	nan	nan	0.0	nan

former model contains more trainable parameters than CAML but having only a small amount of training examples.

For the case of procedural codes, visible in table 2, the best results are now obtained by a combination of CAML with a hierarchical objective. This is closely followed by CAML with a post-processing heuristic. Both techniques improve the classification scores significantly but the overall scores are lower than for the task of classifying diagnostic codes. Lastly, both MVC-RLDA models predicted invalid codes for all documents in the test set, not being able to learn significant relations present in the data.

As an extra experiment to assess the performance of the described heuristics, a CAML model got post-processed with 7 different heuristics. In this case, not only the top-5 categories were retained but the top-50 categories were all sorted by confidence. These resulting files were then evaluated by the evaluation file provided by the competition and results are reported in table 3.

Table 3. Comparison of all post-processing heuristics.

	Diagnostic(MAP)	Procedural(MAP)
CAML	0.042	0.052
CAML + H1	**0.075**	**0.060**
CAML + H2	0.042	0.052
CAML + H3	0.050	0.052
CAML + H4	0.050	0.052
CAML + H5	0.053	0.058
CAML + H6	0.054	0.052
CAML + H7	0.047	0.052

For both the subtasks of classifying diagnostic and procedural codes, the use of heuristic $H1$ is the clear winner. It is worth noting that in no case, the results of the baseline got worse because of the use of a post-processing heuristic. Furthermore, in most cases this has led to an improvement of the results strengthening the claim that post-processing heuristics based on the ICD-10 taxonomy can be a valuable tool. Next to $H1$, the best performing heuristic is $H5$ which squares the distances between nodes in the classification tree. Since all heuristics that try to give more weight to nodes closer to the observed node underperform with respect to $H1$, it might be interesting to see whether the opposite can further improve the classification process.

5 Conclusion

In this paper we trained 5 models for participation in 2 subtasks of the 2020 CLEF eHealth task 1. For both subtasks, experiments were conducted, yielding interesting results. The hierarchical component as well as the use of post-processing heuristics proved their value in this setting. The use of a multi-view neural network led to an abundance of trainable parameters which ultimately made the model unable to efficiently generalize over the training samples. An extra experiment was conducted to asses the influence of the presented post-processing heuristics. This led to the conclusion that these heuristics can be a powerful tool for the classification of ICD codes.

References

1. Baumel, T., Nassour-Kassis, J., Elhadad, M., Elhadad, N.: Multi-label classification of patient notes a case study on icd code assignment (2018)
2. 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)
3. Kavuluru, R., Rios, A., Lu, Y.: An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. Artificial intelligence in medicine **65**(2), 155–166 (2015)
4. Kowsari, K., Brown, D.E., Heidarysafa, M., Meimandi, K.J., Gerber, M.S., Barnes, L.E.: Hdltex: Hierarchical deep learning for text classification. In: 2017 16th IEEE International Conference on Machine Learning and Applications (Dec 2017)
5. 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)
6. Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., Eisenstein, J.: Explainable prediction of medical codes from clinical text. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1101–1111. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1100, https://www.aclweb.org/anthology/N18-1100
7. Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., Elhadad, N.: Diagnosis code assignment: models and evaluation metrics. J Am Med Inform Assoc **21**(2), 231–237 (Mar 2014), 24296907[pmid]

8. Sadoughi, N., Finley, G.P., Fone, J., Murali, V., Korenevski, M., Baryshnikov, S., Axtmann, N., Miller, M., Suendermann-Oeft, D.: Medical code prediction with multi-view convolution and description-regularized label-dependent attention. arXiv preprint arXiv:1811.01468 (2018)
9. Shi, H., Xie, P., Hu, Z., Zhang, M., Xing, E.P.: Towards automated icd coding using deep learning. arXiv preprint arXiv:1711.04075 (2017)
10. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22**(1), 31–72 (Jan 2011)
11. Wehrmann, J., Cerri, R., Barros, R.: Hierarchical multi-label classification networks. In: Dy, J., Krause, A. (eds.) ICML. Proceedings of Machine Learning Research, vol. 80, pp. 5075–5084. PMLR, Stockholmsmässan, Stockholm Sweden (10–15 Jul 2018)