Caring is Sharing – Exploiting the Value in Data for Health and Innovation
M. Hägglund et al. (Eds.)

doi:10.3233/SHTI230264

Learning to Classify Medical Discharge Summaries According to ICD-9

Leonardo MOROS[a,1], Jérôme AZÉ[a], Sandra BRINGAY[a,b], Pascal PONCELET[a],
Maximilien SERVAJEAN[a,b] and Caroline DUNOYER[c,d]
[a] *LIRMM UMR 5506, University of Montpellier, CNRS, Montpellier, France*
[b] *AMIS, Paul-Valéry University, Montpellier, France*
[c] *Medical Information Department, CHU Montpellier, Montpellier, France*
[d] *IDESP, UMR UA11, INSERM - University of Montpellier, Montpellier, France*

Abstract. Context: We present a post-hoc approach to improve the recall of ICD classification. **Method:** The proposed method can use any classifier as a backbone and aims to calibrate the number of codes returned per document. We test our approach on a new stratified split of the MIMIC-III dataset. **Results:** When returning 18 codes on average per document we obtain a recall that is 20% better than a classic classification approach.

Keywords. Supervised learning, constrained optimization, NLP

1. Introduction

Healthcare professionals meticulously record each patient's hospital visit via structured and semi-structured documents, which contain information about treatments, procedures and diagnoses. For financial reasons, healthcare institutions must associate billing codes from the International Classification of Diseases (ICD) with the aforementioned hospital visits. ICD coding is currently done manually by specialists. It is a very complex, tedious, subjective, costly, time-consuming and error-prone task. For example, in the United States, the cost of coding a single discharge summary is estimated to be about $172 [1]. The desire to limit costs has made the field of automatic coding a very active area of research. In recent years, the best results have been achieved by approaches using deep learning models. Mullenbach et al. [2] propose an approach combining convolutional neural networks and attention. Shi et al. [3] combine character level and sentence level LSTMs to code specific sections of the discharge summary. Teng et al. [4] combine convolutional neural networks, graph embeddings for code hierarchy, and attention. More recently, transformer based pre-trained language models have also been used. However, they do not outperform recurrent neural networks [5] in clinical coding. In this work, we propose a classifier that incorporates a budget approach. It is important to note that an error is the result of either an omitted label or of a wrong prediction. In the former case, the clinical coder must read the entire document to identify the omitted label. In the latter case, to invalidate the label, the coder only focuses on the parts of the document used by the model to make its prediction. In this context, our goal is to maximize recall.

[1] Corresponding Author: Leonardo Moros, E-mail: leonardo.moros@lirmm.fr.

The majority of papers evaluate their approaches on the existing MIMIC-III splits [2]. These splits have important stratification issues, which complicate the evaluation and comparison of methods. We evaluate the architecture of our classifier by comparing it to LAAT [6] on a new split of the MIMIC-III dataset that guarantees the stratification of labels in all the subsets.

2. Method

Let $\mathcal{X}$ be the input space (the discharge summaries associated with each patient) and $\mathcal{Y}$ the nodes of the ICD-9 hierarchy. The cartesian product $\mathcal{X} \times P(\mathcal{Y})$ is a probability space with a joint probability measure $P_{X,Y}$ where $Y \in \{0,1\}^L \sim P(\mathcal{Y})$ is a binary vector (representing the flattened ICD-9 hierarchy) that indicates whether a label is present or not. We want to minimize the following risk which is the inverse of recall:

$$\mathcal{R}(S) = \mathbb{E}_{X,Y}\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}\left[Y_j = 1, Y_j \notin S(X)\right]\right]$$

We add **two budget constraints** ($1a$, $1b$) to prevent minimizing the risk by returning all labels and a **hierarchical constraint** (2):
1a) Between K' and K codes are returned per document:

$$\forall_x \in \mathcal{X}, K' \leq |S(x)| \leq K$$

1b) On average K'' codes are returned over all documents: $\mathbb{E}_X[|S(X)|] \leq K''$
2) Hierarchical constraint: If a leaf node is returned then all of its parent nodes must also be returned:

$$\forall_x \in \mathcal{X}, \forall_y \in \mathcal{Y}, \forall_{\tilde{y}} \in ancestors(y), y \in S(x) \Rightarrow \tilde{y} \in S(x)$$

Our goal is to construct a function $S : \mathcal{X} \to P(\mathcal{Y})$ that satisfies some combinations of the aforementioned constraints. In this paper, we test three different combinations of these constraints: 1) Top-*K* (1a and 2), 2) Average-*K* (1b and 2), and 3) Hybrid (1a, 1b and 2), which is an Average-*K* method with both an upper and a lower bound to prevent returning too many or too few codes.

3. Experiments and Results

Dataset: MIMIC-III [7] is a freely available clinical database. Most studies use the two splits created by Mullenbach et al. [2]. The first split contains the 50 most frequent ICD-9 codes (11,368 records) and the second contains all 8,929 ICD-9 codes (52,722 records) (see Table \ref{tab:donnees}). The code distribution is considerably unbalanced. For example, code *567.2* occurs 211 times, while code *276.5* occurs 1,294 times, or six times as often. Furthermore, in the current splits, there is no guarantee that all the codes will be found in the learning, validation, and test sets. For example, the code *276.5* appears 1,293 times in the learning set, once in the testing set and does not appear at all in the validation set. We therefore decided to make our own split using the 1,000 most frequent codes in the dataset. We applied a stratification algorithm [8] to ensure that each code was represented in the same proportions in the training, test and validation sets. We also ensured that patients appearing in the training set do not appear in the test/validation sets. The final distribution of the data is presented in Table 1 (MIMIC-III-1000).

Table 1. Statistics on the MIMIC-III splits

Split	Train	Val	Test	Total
MIMIC-III-50	8,066	1,573	1,729	11,368
MIMIC-III-1000[2]	44,592	2,716	5,327	52,635
MIMIC-III-Full	47,719	1,631	3,372	52,722

Implementation: We built an estimator of probabilities $\hat{\eta}$ that uses a neural network on the nodes at the lowest level of the hierarchy and we estimate the parent probabilities on the basis of children scores. We chose LAAT [6] with the optimal parameters mentioned in their paper. We trained the model with a learning rate of 0.001 and a batch size of 8 for 50 epochs. We used early stopping by monitoring micro F1; if there was no improvement after 5 consecutive epochs, we stopped the training. We used the word2vec[3] embeddings trained on all the discharge summaries and a dropout of 0.3 between the embedding and LSTM layers. For text preprocessing, we removed all tokens not containing alphabetic characters and we lowercased all the text. Once the estimator was built, we used it with the Top-*K*, Average-*K* and Hybrid rules.

Evaluation: We evaluated our methods on all the splits but we only present here the results on the MIMIC-III-1000 stratified split. We also evaluated LAAT [6] on our split as our baseline and used the hierarchical recall as our metric. We drew curves that show the trade-off between budget size and micro recall and tested the 3 budget methods on two configurations: 1) taking only leaves from the hierarchy and 2) enriching the labels with their parents.

Analysis and interpretation: Figure 1 shows the results for the different budget approaches. On the ordinate, we have the recall, on the abscissa we have the size of the budget. The red line represents the recall obtained by the baseline, LAAT [6]. The green line shows the average number of labels in the dataset. **The graph on the left** shows the results on the leaves of the hierarchy. Both methods obtain better results than the baseline. For Top-*K*, starting with a budget of 14 (the average number of labels per document), the method improves on the baseline. For Average-*K*, starting with a budget of 12, the method improves on the baseline. On average, Average-*K* achieves a recall that is 5.6% greater than Top-*K*. **The graph on the right** shows the results when using the hierarchy (enriching the labels with parent information). We have removed the leaves that have no siblings and their parents. To be able to compare to a method that does not take the hierarchy into account, we drew curves for when the entire budget is used on the leaves, the parents associated with the selected leaves being added manually. Starting with a budget of 24 (the average number of labels per document), Average-*K* improves on the baseline. For Top-*K*, we need a budget of at least 27 to improve on the baseline. On

[2] Splits created for this publication. https://github.com/leo90v/MIMIC-1000

[3] https://github.com/aehrc/LAAT/tree/master/data/embeddings

average, Average-*K* achieves a recall 5.61% greater than Top-*K*. Therefore, allocating the budget to the hierarchy provides a very small improvement in results.

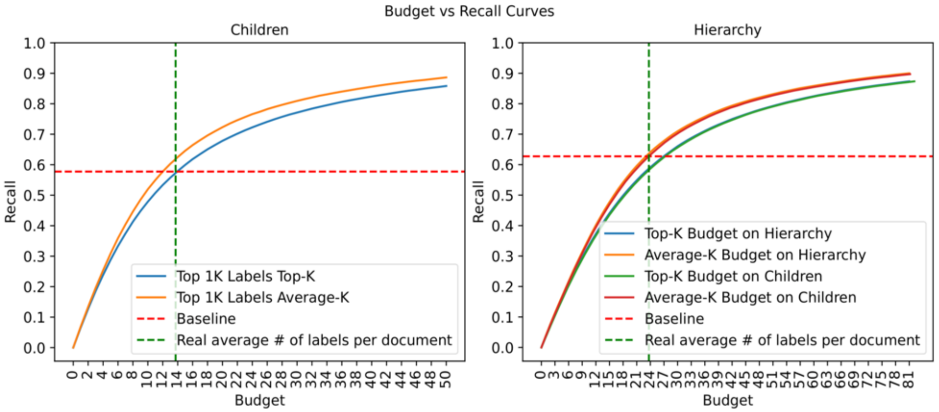

Figure 1. Top-*K* vs Average-*K*. In the left graph, we only consider leaf nodes. In the right graph, we consider the hierarchy.

Finally, the results of the hybrid method are shown in Figure 2. Since the hierarchy did not improve the results, we tested this method only on the leaves. We drew the Top-*K* and Average-*K* curves as well since they represent the minimum and maximum performance we can obtain with the hybrid method. We set the upper bound (K) as a function of the budget (K''). **In the graph on the left**, we show the micro recall. With $1.2K''$, the performance is on average 3.2% better than with Top-*K* and 2.25% worse than with Average-*K*. We also tested with $1.8K''$, but did not draw it since the hybrid and Average-*K* curves overlap at that point. These results show that with relatively small bounds, the performance is similar to Average-*K*, while also preventing having an unmanageable number of predictions for some documents. **In the graph on the right**, we show the macro recall. We notice similar behavior but with considerably lower performance. For example, with $1.5K''$ the micro recall is 22% greater on average than the macro recall. This is mainly due to the imbalance of the data set. Even with stratified data, the model has difficulties with some labels.

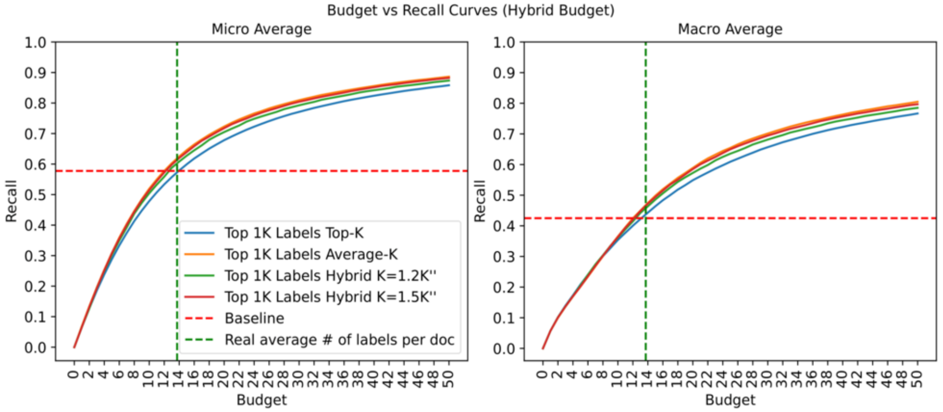

Figure 2. Hybrid method (micro recall on the left and macro recall on the right).

4. Conclusion

We have presented a new approach for semi-automatic clinical coding. We have tested our approach on the splits of the literature and on our stratified split. Via an adaptive predictor, we predict more or less codes per document and at different levels of the hierarchy. Our solution is applicable to any classifier. We plan to use other models in the future, such as LAAT trained with the LDAM loss function [9] designed for unbalanced datasets. We could also use transformers (e.g. Longformer), adapted to long documents [10], but they have not yet surpassed the state of the art for this task. We plan to develop approaches with a weighted budget to give more importance to certain codes. Finally, we plan to experiment our approach using ICD10 and more contemporary data from CHU Montpellier.

References

[1] Richman BD, Kaplan R, Kohli J, Purcell D, Shah M, Bonfrer I, et al. Billing And Insurance-Related Administrative Costs: A Cross-National Analysis. Health affairs. 2022;41 8:1098-106.

[2] Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable Prediction of Medical Codes from Clinical Text. In: 2018 Chapter of the ACL: Human Language Technologies, Volume 1; 2018. p. 1101-11.

[3] Shi H, Xie P, Hu Z, Zhang M, Xing EP. Towards Automated ICD Coding Using Deep Learning. CoRR. 2017;abs/1711.04075.

[4] Teng F, Yang W, Chen L, Huang L, Xu Q. Explainable Prediction of Medical Codes With Knowledge Graphs. Frontiers in Bioengineering and Biotechnology. 2020;8.

[5] Pascual D, Luck S, Wattenhofer R. Towards BERT-based Automatic ICD Coding: Limitations and Opportunities. In: Proceedings of the 20th Workshop on Biomedical Language Processing. Online; 2021. p. 54-63.

[6] Vu T, Nguyen DQ, Nguyen A. A Label Attention Model for ICD Coding from Clinical Text. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20; 2020. p. 3335-41. Main track.

[7] Johnson A, Pollard T, Shen L, Lehman Lw, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Scientific Data. 2016 05;3:160035.

[8] Sechidis K, Tsoumakas G, Vlahavas I. On the Stratification of Multi-label Data. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M, editors. Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg; 2011. p. 145-58.

[9] Cao K, Wei C, Gaidon A, Arechiga N, Ma T. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems; 2019. p. 1567–1578.

[10] Beltagy I, Peters ME, Cohan A. Longformer: The Long-Document Transformer. ArXiv. 2020;abs/2004.05150.