Extreme Multi-Label Classification of Disease Codes From Medical Text

Modestas Filipavicius, Orhun Ozbek, Akmaral Yessenalina, Brynja Sigurdottir

D-INFK, ETH Zürich

Abstract

Automatic diagnosis code assignment from clinical notes is an important problem since performing it manually is error-prone and requires considerable time and human resources. To solve this multi-label classification problem, we use label representations obtained from the label co-occurence graph to generate embeddings, which are fed into an attentional convolutional network. Although the method does not perform well, potential reasons for sub-optimal results and possible improvements are discussed. Code: `github.com/mfilipav/auto-diagnosis`

1 Introduction

The International Classification of Diseases (ICD) provides a hierarchy of diagnostic codes for classifying diseases and procedures. The assignment of ICD codes to medical records is a tedious and error-prone task which requires staff with extensive training. The goal of this work is to build a model that accurately assigns ICD codes to the free-text discharge summaries.

MIMIC-III [10] consists of hospital records collected between 2001-2012, comprising over 58,000 hospital admissions and discharge notes for 38,645 adults and 7,875 neonates. It is a combination of structured data (vital signs, laboratory tests, medications) and unstructured data (clinical notes). Here we focus on the discharge summaries, a type of clinical note, written when the patient leaves the hospital. Each admission is linked to a discharge summary, and tagged by multiple ICD codes, which are the labels for the multi-label classification task.

Assignment of a set of ICD codes to the discharge summaries obtained from MIMIC-III clinical database is an extreme multi-label classification (XML) task due to the large number of possible diagnosis codes. Current approaches in the literature can be separated into three parts:

- Framing the problem as multiple binary classification tasks, as done by Baumel et al. [3] and Mullenbach et al. [12].
- Ranking all available labels and choosing the top k highest scoring labels, where k is determined by another network which decides on the number of tags to be assigned to a document, as seen in Rios and Kavuluru [18] and Du et al. [6].
- Training different classifiers for each decision point in ICD9 code hierarchy (see Perotte et al. [14]).

Despite the popularity in XML literature of the graph based embedding methods, which embed the label co-occurrence graph to a latent space, we have not came across a work which applies this technique to the specific problem of medical code multi-label classification. Since diseases are not independent events and tend to co-occur, we assume that an embedding-based approach might increase the accuracy of current classifiers.

In this work, we utilized graph embedding methods to represent nodes in a continuous vector space. These label embeddings are used to create representations for each discharge record. Then, we used convolutional neural networks (CNN) in order to regress the document representation for each document. As a final step, a modified version of k-nearest neighbor algorithm is used for the final classification task.

2 Previous Work

2.1 Text Representations

Word embeddings represent words as vectors. Well-known approaches for deriving embeddings are: Glove [13], word2vec [11], fastText [10], and most recently, BERT [5]. Recently, Biosentvec, a sentence encoder trained on PubMed biomedical journal articles and MIMIC-III was released [21].

2.2 Extreme Multi-Label Classification

The objective in XML is to learn a classifier that can automatically tag a data point with the most relevant subset of labels from a large label set. Extreme classification is an active research area, because it allows the reformulation of commercial learning problems such as consumer recommendation and ranking. Extreme classification is challenging as it necessitates dealing with large number of labels, feature dimensions and training points [16].

Sparse Local Embeddings for Extreme Multi-label Classification (SLEEC) [4] is one example of XML-method which relies on learning an embedding transformation to map label vectors into low-dimensional representation. SLEEC accurately predicts infrequently occurring labels by learning an ensemble of local distance-preserving embeddings, thus exploiting the similarity among labels to improve classification, and learning the representations for clusters of labels.

Other popular methods learn a hierarchy from the training data: a root node is initialized to contain the entire label set, then the node partitioning formulation is optimized to determine which labels should be assigned to the left child and which to the right. Nodes are recursively partitioned till each leaf contains only a small number of labels. During prediction, a novel data point is passed down the tree until it reaches a leaf node. A base multi-label classifier of choice can then be applied to the data point focusing exclusively on the leaf node label distribution. This leads to prediction costs that are sub-linear or even logarithmic if the tree is balanced. Tree based methods often outperform 1-vs-All classifiers in terms of prediction accuracy, yet are expensive to train. FastXML [16] is faster to train then other state-of-the-art tree-based methods, such as Multi-label Random Forest [2] and Label Partitioning by Sublinear Ranking [19].

2.3 Medical Code Classification

Automated ICD coding problem was actively approached over the past few years. Perotte et al. [14] utilized the hierarchical structure of ICD9 codes to build SVM classifiers, while Huang et al. [9] demonstrated that deep-learning-based methods such as CNN and RNN outperform conventional machine learning approaches like one-vs-all SVM or Logistic Regression. Most recently, a CNN is used to extract features from discharge notes, while augmenting the CNN with an external memory over a support set consisting of similar documents Rios and Kavuluru [18]. Prakash et al. [17] presented condensed memory neural networks (C-MemNNs), that efficiently store condensed representations in memory, thereby maximizing the utility of limited memory slots.

Baumel et al. [3] proposed HA-GRU (Hierarchical Attention-bidirectional GRU), which outperformed SVM, CBOW (continuous-bag-of-words) and CNN approaches. They exploited two layers of bidirectional GRU: encoding sentences applied over tokens, and encoding the document applied over the encoded sentences. Moreover, they used an attention mechanism over the second GRU that allowed the model to focus on the relevant sentences for each label. Similarly, Du et al. [6] introduced a hierarchical approach with LSTMs instead of GRUs.

2.4 Graph embeddings

Given a graph $G = (V, E)$ a graph embedding is a mapping: $f : v_i \to y_i \in R^d \quad \forall i \in [n]$ such that $d << |V|$ and the function f preserves some proximity measure defined on graph G.

An embedding maps each node to a low dimensional feature vector while trying to preserve the connection strengths $S_{i,j}$ between vertices v_i and v_j. An embedding preserving first order proximity is obtained by minimizing $\sum_{i,j} S_{i,j}||y_i - y_j||_2^2$.

Recently, the focus of embedding research has shifted towards the more scalable algorithms that can be applied to graphs with million of nodes and edges. Goyal and Ferrara [7] proposed a categorization of embedding methods: Factorization, Random Walk and Deep Learning based.

Factorization based algorithms represent the connection between nodes in the form of a matrix (adjacency, Laplacian, transition probability matrix, etc.) and factorize it to obtain the embedding. The factorization methods differ based on the matrix properties: eigenvalue decomposition is used for positive semidefinite matrices, and gradient descent for unstructured matrices [7].

Random walks are used to approximate graph properties such as node centrality and similarity. They are especially useful when one can either only partially observe the graph or if the graph is too large to measure in its entirety. Three examples of such embedding techniques to obtain label representations are DeepWalk [15], Node2vec [8] and AttentionWalk [1].

Deep Neural network based graph methods such as Deep autoencoders are used for dimensionality reduction due to their ability to model non-linear structure in the data. More recent algorithms were introduced to generate an embedding which captures the non-linearity in graphs [7].

2.4.1 DeepWalk

DeepWalk preserves higher order proximity between nodes by maximizing the probability of observing the last k nodes and the next k nodes in the random walk centered at v_i, i.e. maximizing the $\log P(v_{i-k}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+k}|Y_i)$ where $2k+1$ is the length of the random walk. The model generates multiple random walks each of length $2k+1$ and optimizes each random walk. A dot-product based decoder is used to reconstruct the edges from the node embeddings [15].

2.4.2 AttentionWalk

AttentionWalk differs from methods above, in that the parameters such as the length of a random walk can be learned via training and backpropagation, instead of manual tuning. It is a novel attention model on the power series of the transition matrix, which guides the random walk to optimize an upstream objective. Attention parameters are utilized exclusively on the data itself (e.g. on random walk) but are not used for inference[1].

2.5 Multi-label k-NN

A multi-label lazy learning approach named ML-KNN [20], is derived from the traditional K-nearest neighbour (KNN) algorithm. Given a training set of instances with known label sets, the task is to build a multi-label classifier to predict the label sets of unseen instances.

The goal is to create the category vector $\mathbf{y_t}$ for a given test instance t where its lth component takes the value 1 if t belongs to class l and 0 otherwise. Firstly, for the test sample t ML-KNN identifies its KNNs *N(t)* in the training set. After that, we count the number of neighbours of the test instance t which belong to the lth class to get the *membership counting vector*:

$$C_t(l) = \sum_{a \in N(t)} y_a(l)$$

Consequently, maximum a posteriori (MAP) principle is used:

$$y_t(l) = \operatorname*{argmax}_{b \in \{0,1\}} P(H_b^l | E_{C_t^l}^l)$$

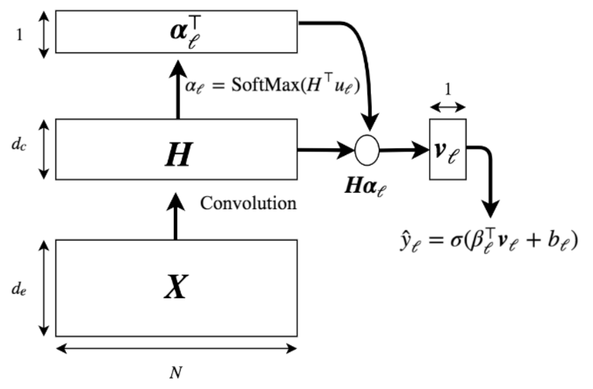
where H_1^l is the event that t has label l, whereas H_0^l be the event that t has not label l; $E_j^l (j \in 0, 1, ..., K)$ is the event that t has exactly j neighbours with label l.

It can be shown that by using the Bayesian rule the above formula can be rewritten through prior probabilities $P(H_b^l)$ and posterior probabilities $P(E_{C_t^l}^l | H_b^l)$. In fact, these probabilities can be estimated from the training set based on frequency counting.

Zhang and Zhou [20] showed that ML-KNN outperfoms BoosTexter, AdaBoost and Rank-SVM algorithms on three different tasks: Yeast gene functional analysis, natural scene classification and automatic web page categorization. It is worth to note that the choice of k does not significantly affect the performance of the algorithm. However, the experiments that the authors conducted consisted of 5-40 categories with around 2-5 labels in average per instance, which is significantly less than in our XML problem.

2.6 Convolutional Attention for Multi-Label classification (CAML)

We build our study based on a state-of-the-art model by Mullenbach et al. [12] which predicts a set of codes for both ICD-9 diagnoses and procedures, and is composed of shared embedding and CNN layers between all codes and an individual attention layer for each code. This allows the model to learn appropriate document representations for each label. Furthermore, the model incorporates the text descriptions for each code, by training an embedding for them, thus encouraging the parameters of rare codes to be similar to the parameters from codes with similar descriptions. Their best model, which reached a micro F1-score of 53.9% on MIMIC-III, is interestingly also the base model without the secondary embedding with code descriptions.



$\alpha_\ell^\top$
1
$\alpha_\ell = \text{SoftMax}(H^\top u_\ell)$
1
d_c
H
v_ℓ
$H\alpha_\ell$
Convolution
$\hat{y}_\ell = \sigma(\beta_\ell^\top v_\ell + b_\ell)$
d_e
X
N

Figure 1: CAML architecture with per-label attention shown for one label. In a max-pooling architecture, H is mapped directly to the vector v_l by maximizing over each dimension.

2.6.1 Convolutional architecture

Overall architecture of CAML is shown in Figure 1 taken from Mullenbach et al. [12]. We store our embeddings in matrix $X = [x_1, ..., x_N] \in R^{d_e \times N}$, where N is the total number of words, each represented by a d_e-long embedding, are vertically concatenated. The length of each document is truncated to $N = 2,500$ words. For each document in the training set, embeddings pass through a convolutional filter $W_c \in R^{k \times d_e \times d_c}$ to form a zero-padded matrix $H \in R^{d_c \times N}$, where d_e, d_c, k are the size of the input embedding, the size of the filter output, and the filter width, respectively. Thus, at each token n the following non-linear activation is computed: $h_n = g(W_c * x_{n:n+k-1} + b_c)$, with a convolutional operator $*$, activation function g and bias $b_c \in R^{d_c}$.

2.6.2 Per-Label Attention

Since each discharge summary is labeled with multiple codes, we want to dissect the relevant combinations of words that activate the network for each label $l \in L$. To this end, each label is parameterized with $u_l \in R^{d_c}$, and we generate an attention distribution vector, α_l, of the most likely locations in the document:

$$\alpha_l = \mathrm{SoftMax}(H^T u_l)$$

Eventually, we compute the vector representations for each label,

$$v_l = \sum_{n=1}^{N} \alpha_{l,n} h_n$$

2.6.3 Classification and Training

In the output layer we classify the document representation for a given vector l with:

$$\hat{y_l} = \sigma(\beta_l^T v_l + b_l),$$

where $\sigma(x)$ is sigmoidal nonlinearity, while $\beta_l \in R^{d_c}$ are learned model parameters with biases b_l. During testing, the output values greater than 0.5 are assigned as present (1) and the rest are assigned as absent (0).

As a training objective we use the binary cross entropy loss on the last layer's outputs:

$$L_{BCE}(X, y) = -\sum_{l=1}^{L} y_l \log(\hat{y_l}) + (1 - y_l) \log(1 - \hat{y_l})$$

3 Models and Methods

The classification scheme used (Figure 2) in our approach is highly similar to Zhang et al. [21]. Instead of using label vectors as the target variables directly, we obtained label embeddings by using AttentionWalk algorithm, and aggregated label embeddings corresponding to each document to obtain a new target vector which is lower dimensional. Then, we used CAML architecture [12] for multi-target regression of these document embeddings (see section 2.6). Resulting document embeddings are mapped to the label power set using ML-KNN algorithm.

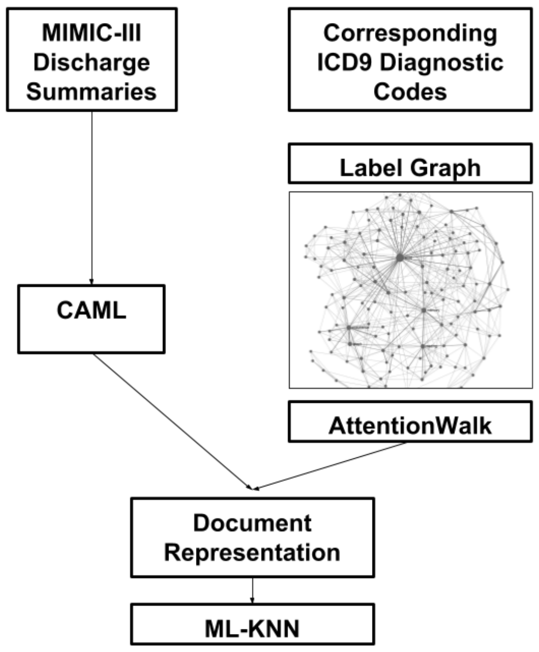

MIMIC-III Discharge Summaries
Corresponding ICD9 Diagnostic Codes
Label Graph
CAML
AttentionWalk
Document Representation
ML-KNN

Figure 2: Multi-label Classification scheme for medical discharge records

3.1 Pre-processing

Each document is pre-processed by following steps:

- Remove all de-identified information marked by [** deidentified-info**]
- Remove all numerical tokens and replace them with "num". This is an important step since target ICD9 codes can be encountered in the discharge records.
- Lowercase the document.
- Remove English stop words.

After the pre-processing, documents are tokenized using regular expression ((?u)\b \w \w+ \b) to tokenize each document and used a vocabulary size of 40,000 to create a dictionary. Any token which is not included in the vocabulary is mapped to "UNK" token. As a final step, document size is limited to 2,500

	AUC		F1		P@n
Model	macro	micro	macro	micro	8
SVM (tf-idf)	-	-	0.034	0.379	-
Logistic Regression (tf-idf)	0.870	0.974	0.016	0.354	0.434
CNN (Mullenbach et al. [12])	0.806	0.969	0.042	0.402	0.581
CAML (Mullenbach et al. [12])	0.895	0.986	0.088	0.539	0.709
HAN (Baumel et al. [3])	-	-	-	0.405	-
Our approach	0.536	0.948	0.014	0.250	0.242

Table 1: Results for MIMIC-III discharge record multi-label classificat

tokens. Shorter documents are padded using ”PAD” token. For word embedddings, word2vec embeddings that are trained on the training set is used. Each document is transformed to a (300 x 2,500) matrix.

3.2 Graph Embeddings

We used AttentionWalk model to train 256 dimensional label embeddings. For each document, corresponding label embeddings are aggregated by taking by averaging. Resulting representation of the document is used as the target vector for multi-target regression.

3.3 Regression

For multi-target regression, CAML architecture as demonstrated in Section 2.6 is used. We used a filter size of 15 and trained 256 filters. A dropout with probability of 0.2 is applied after the embedding layer. Mean squared error is used as the loss function.

$$L_{MSE}(Y,\hat{Y}) = -\frac{1}{n}\sum_{n=1}^{N}(Y-\hat{Y}_i)^2$$

For training, used Adam optimization (learning rate 0.0001) and trained our model for 100 epochs with a batch size of 32.

3.4 Classification

To map the label embedding space back to a binary vector which represents the label power set, we used ML-KNN algorithm. We set $k = 3$ and smoothing parameter $s = 0.5$.

4 Results and Discussion

For evaluation, we used a train-test split of 90%-10%. Our results are disappointing, the classifier performed significantly worse than our soft baseline (Table 1).

To understand whether this under-performance is due to the embeddings learned by AttentionWalk model, representations for the test label sets are created by averaging embeddings corresponding to each label, then classified using the ML-KNN algorithm. Using this method, we obtained 0.475 f1-micro score on the test set, which implies there is a problem about the label embeddings or the aggregation method. We still believe that a label embedding method would improve the discriminative ability of classifiers. Possible reasons for our performance are listed below:

- Similar to this work, SLEEC uses k-nearest neighbors algorithm in order to make its classifications; since the algorithm is designed to learn embeddings which preserve pairwise distances between close label vectors, the method is more compatible with KNN. However, this is not the case for the embeddings learned by AttentionWalk model. Hence, we trained various classifiers including multi-layer perceptron and support vector machines for the final classification which did not yield any improvements on the scores.
- Averaging is used for the aggregation of the label embeddings to get document representations. A more sophisticated model could be used to get more better document representations.

5 Summary

We tried to exploit label co-occurence information by learning label embeddings using AttentionWalk algorithm to tackle the extreme multi-label classification of digital health records. We obtained document representations by aggregating label embeddings generated by AttentionWalk, then used CAML by Mullenbach et al. [12] to regress these representations. Finally, we used ML-KNN for multi-label classification to obtain a set of labels.

Results show that our approach could not surpass simple machine learning algorithms. We suggest possible flaws in our approach that might lead to unsatisfactory results and possible areas of development.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. Watch your step: Learning graph embeddings through attention. *arXiv preprint arXiv:1710.09599*, 2017.

[2] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, pages 13–24. ACM, 2013.

[3] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noemie Elhadad. Multi-label classification of patient notes a case study on icd code assignment. *arXiv preprint arXiv:1709.09587*, 2017.

[4] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738, 2015.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. Ml-net: multi-label classification of biomedical texts with deep neural networks. *arXiv preprint arXiv:1811.05475*, 2018.

[7] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[8] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[9] Jinmiao Huang, Cesar Osorio, and Luke Wicent Sy. An empirical evaluation of deep learning for icd-9 code assignment using mimic-iii clinical notes. *arXiv preprint arXiv:1802.02311*, 2018.

[10] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[11] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[12] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[14] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21 (2):231–237, 2013.

[15] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations, 2014.

[16] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 263–272. ACM, 2014.

[17] Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek V Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Condensed memory networks for clinical diagnostic inferencing. In *AAAI*, pages 3274–3280, 2017.

[18] Anthony Rios and Ramakanth Kavuluru. Emr coding with semi-parametric multi-head matching networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2081–2091, 2018.

[19] Jason Weston, Ameesh Makadia, and Hector Yee. Label partitioning for sublinear ranking. In *International Conference on Machine Learning*, pages 181–189, 2013.

[20] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[21] Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 100–107. ACM, 2018.