UMLS mapping and Word embeddings for ICD code assignment using the MIMIC-III intensive care database

Henning Schäfer[1], and Christoph M. Friedrich[1,2*], *Member*, *IEEE*

Abstract— **Diagnosis codes are used as a billing mechanism in the Electronic Health Record and have the capability to benefit decision support systems, which aim to assist coders by suggesting a relevant subset of potential codes to choose from. Due to the large set of possible labels and length of patient records, automatic ICD code assignment is considered to be a challenging task within the field of multi-label classification. This paper introduces a baseline for automatic ICD code assignment using Support Vector Machines (SVM) and FastText with Unified Medical Language System (UMLS) metathesaurus mappings into word embedding models. Training data is obtained from the Medical Information Mart for Intensive Care (MIMIC-III) database and extended with 'is-a' relationships from ICD-9 hierarchy. FastText is evaluated with different label count estimations, of which an approach based on label cardinality yields a F1-Score of 62.2%. FastText achieves high recall results and mentionable performance improvements over previous models. Reported values are obtained through 10-fold cross-validation.**

I. INTRODUCTION

Discharge summaries are free text narratives created during patient stays. They are associated with a set of codes from a medical classification system called International Statistical Classification of Diseases and Related Health Problems (ICD), which is maintained by the world health organization (WHO). ICD describes diagnosis and procedures of hospital stays in a standardized way. The main challenges of this task are (1) handling the large number of labels to assign with over 14,000 ICD-9 codes and around 70,000 ICD-10 codes, (2) building a model, which covers the relations inbetween ICD codes, (3) addressing the long document instances containing approximately 1,900 words on average and (4) deciding the correct subset from predicted codes.

Because manual coding of documents can be time-consuming, automatic ICD code assignment has been studied in the past and several models to tackle this task have already been proposed. Recent approaches to mention are ML-NET [1] with a label count prediction network and CAML [2] with an attention mechanism.

The presented model reduces the MIMIC-III [3] discharge summary document length by mapping a maximum window of 5 tokens sequentially over the UMLS metathesaurus [4]. Thus each document corpus is represented only by their standardized biomedical concept embeddings (order remains). A simple bag-of-words (BoW) sentence representation trained by a binary SVM through one-vs-rest classifier is used as the baseline while FastText is used as the main model.

II. PREVIOUS WORK

Perotte et al. [5] proposed a hierarchy based approach using SVM, where the 'is-a' relation between ICD-9 codes was used to model label dependencies. The hierarchy based classifier outperformed the flat SVM, which did not consider code dependencies. Apart from label dependency, other approaches have identified label density and label noise as useful characteristics [6], while others empirically evaluated co-occurrences between labels [7].

ML-NET [1] followed the hierarchy based approach and extended the coding of documents. Their deep neural network consists of an additional network to estimate label count. Instead of splitting relevant vs. irrelevant labels by a threshold, a label count prediction network was built by taking the document vector as input.

Baumel et al. [8] evaluated 4 different models for ICD code assignment using data from both, MIMIC-II and MIMIC-III datasets. They introduced a continuous bag-of-words model [9] (CBOW), a convolutional neural network, a SVM one-vs-all model and a bidirectional Gated Recurrent Unit model with hierarchical attention, which they refer to as (HA-GRU).

Mullenbach et al. [2] proposed attention mechanisms to extract n-grams from the text that are influential in the prediction of each label (CAML) and (DR-CAML).

III. DATASET

A. Records

Training data is obtained from the MIMIC-III [3] dataset, which comprises de-identified records from Beth Israel Deaconess Medical Center intensive care unit (ICU) stays collected between 2001 and 2012. The training set consist of the corpus of discharge summaries and associated ICD-9 codes. Descriptive data is shown in Table I.

TABLE I

MIMIC-III DATASET DESCRIPTIVE STATISTICS

	MIMIC-III
Number of records with ICD code	59,652
Number of unique tokens	119,171
Avg. number of tokens / record	1,947
Avg. number of sentences / record	112
Avg. number of labels / record	11.48
Label Density	0.0018
Number of labels in collection	6,918
Number of labels in extended collection	8,790

[1] Department of Computer Science, University of Applied Sciences and Arts Dortmund (FHDO), Dortmund, NRW Germany

[2] Institute for Medical Informatics, Biometry and Epidemiology (IMIBE), University Hospital Essen, Essen, NRW Germany

* Corresponding author
christoph.friedrich@fh-dortmund.de

978-1-5386-1311-5/19/$31.00 ©2019 IEEE

B. *Codes*

ICD-9 codes follow a hierarchy and each node (maximum depth is 8) can have a varying number of children. The same tree structure Perotte et al. [5] used is applied to MIMIC-III training data. It is available for download at the National Center for Biomedical Ontology (NCBO) BioPortal. [10]

IV. Methods

A. *Embedding UMLS terms*

Biomedical terminologies often contain more information than mere terms and their inter-relations, which lead to the idea of trading in the loss of some contextual information for more substantial and standardized terms. In both models the Unified Medical Language System (UMLS) is used to reduce record length by removing any non-matching terms.

QuickUMLS [11] was used to extract biomedical concepts from discharge summaries. The library implements simstring [12] to perform fast approximate string matching by defining a threshold. Available similarity measures are cosine, Jaccard and Dice, of which cosine similarity is used with a similarity matching threshold of $t = 0.8$. A window of $n = 5$ tokens is considered for matching. By allowing a threshold, variations caused by tense, abbreviations or typos are less likely to be missed out and more standardized matches end up in the training set. To further increase recall, all terms are folded to lowercase. Embeddings of UMLS terms decreased the overall length of the discharge summary corpus by 34.5%.

B. *SVM*

Based on Perotte et al.'s flat and hierarchical SVM used on MIMIC-II dataset, a similar model is applied on the updated MIMIC-III dataset to demonstrate basic performance improvements by adding mapping and hierarchical dataset augmentation. Thus, a multi-label binary SVM classifier with Scikit Learn [13], [14] implementation (C-Support Vector Classification) is used to build two models *SVM-UMLS* and *eSVM-UMLS* (hierarchically extended) as the baseline.

Due to the multi-label classification problem, the model has to fit a binary SVM classifier for each code against all other codes (one-vs-rest). Features are determined through bag-of-words and $tf \cdot idf$ weights for each code. A dimension of 6000 is available for the top vocabulary of UMLS embeddings.

The non-stratified *RepeatedKFold* function has been used to perform 10-fold cross-validation. Within repeated random_state, the first 90% of the dataset has been used for training and the remaining 10% for testing. Table II shows the hyperparameters that were used.

TABLE II

Hyperparameters used for sklearn libsvm

Hyperparameter Name	Hyperparameter Value
C (cost parameter)	1.0
gamma	'auto'
kernel	'linear'
shrinking	True
decision function shape	'ovr' (one-vs-rest)

C. *FastText*

FastText [15], [16] has been proposed to solve text classification problems for large corpora and large output space. The model explains the input-output relation by a large matrix $C\epsilon\mathbb{R}^{p\times q}$. Figure 1 shows the FastText model. The classifier computes a document vector, which can be described as a linear bag of words for that document. The document vector is an average representation of all document word vectors and forms the hidden variable. To improve training efficiency, hierarchical softmax [18], [9] is used to minimize softmax output over training data $\{(x_n, y_n)\}_{n=1}^{N}$. By default, class probabilities are computed with the regular softmax, but when dealing with large datasets such as the MIMIC-III training set, hierarchical softmax should be preferred. It decomposes the output layer to a binary tree, which reduces complexity to obtain probability from $O(N)$ to $O(log(N))$. Results slightly improve when using the regular softmax.

For label prediction with FastText, the hyperparameters shown in Table III have been used to train *FastText-UMLS* and *eFastText-UMLS* models.

TABLE III

Hyperparameters used for FastText

Hyperparameter Name	Hyperparameter Value
dim (dimension for word)	200
epoch	90
lr (learning rate)	1.3
wordNGrams	2
loss function	hierarchical softmax (hs)

Non-stratified, 10-fold cross-validation is performed by repeating the training procedure with the above mentioned hyper parameters in a random_state for each fold. Similar to the SVM baseline, the first 90% of shuffled data is used for training and the remaining 10% for testing. For the extended model *eFastText-UMLS*, training data is augmented after each fold by adding the ancestors for each code over each document. Labels within the testing data remain untouched, because ancestors and descendants are resolved and processed during evaluation.

Model implementation is online available on GitHub [17].

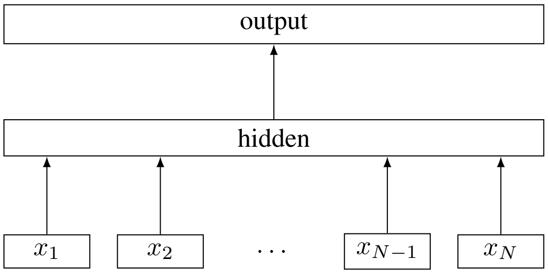

Fig. 1. Model architecture of `FastText` for a sentence with N ngram features $x_1, \dots, x_N$. The features are embedded and averaged to form the hidden variable. [16]

TABLE IV

Results on MIMIC-III full label set. (*) Denotes models applied to the previous version (MIMIC-II). Bold indicates mentionable performance improvements. Scores are given in percent and have been rounded to one decimal place followed by their standard deviation from 10-fold cross-validation data. For eFastText-UMLS ’P@5’ and ’P@8’ are reported seperately. It refers to the same model for each label count estimation.

Method	Precision	Recall	F1-Score	P@5	P@8
Binary relevance SVM [5]*	57.7%	30.0%	39.5%	-	-
MT-CNN-net [1]	31.1%	44.2%	36.5%	-	-
MT-CNN-net-threshold [1]	50.1%	37.3%	42.8%	-	-
HA-GRU [8]	-	-	40.5%	-	-
CAML [2]	-	-	53.9%	-	**70.9%**
DR-CAML [2]	-	-	52.9%	-	69.0%
SVM-UMLS	70.1% (0.357)	20.2% (0.399)	31.4% (0.378)	-	-
eSVM-UMLS	54.9% (0.548)	33.7% (0.612)	41.8% (0.578)	-	-
FastText-UMLS	-	-	-	56.4% (0.121)	44.3% (0.236)
eFastText-UMLS	-	-	-	**68.7%** (0.196)	63.0% (0.230)
eFastText-UMLS$_{\text{cardinality}}$	58.0% (0.479)	**66.8%** (0.629)	**62.2%** (0.544)	-	-
eFastText-UMLS$_{\text{threshold}}$	**71.2%** (0.534)	42.8% (0.642)	53.5% (0.583)	-	-
eFastText-UMLS$_{\text{read-off}}$	60.1% (0.236)	61.2% (0.337)	60.6% (0.278)	-	-

V. Evaluation Pipeline and Metrics

F1-Score is the most relevant and widely used evaluation metric for ICD code assignment. Using F1-Score allows model comparison with prior and future work [1], [2], [5], [8]. In addition to that, precision at n (’P@n’) is being reported. This is motivated by the potential use case as a decision support system, where a fixed number of predicted codes is presented to the coder. For this use case, high precision is preferred over high recall. Again, to allow for comparison with prior work [2] $n = 5$ and $n = 8$ are being reported. Mean precision, mean recall and mean F1-Score are obtained from the repeated folds and are reported along with the standard deviation.

A. SVM

Decision outcome of the extended SVM is processed with the ICD-9 hierarchy tree downloaded from the National Center for Biomedical Ontology (NCBO) BioPortal [10]. The hierarchy was not changed. Each code is binary tested against the SVM to be reported as either ’positive’ or ’negative’ (one-vs-rest). For each code all positive ancestors are then considered to be also positive and negative descendants to be also negative.

For example having ICD-9 code *398.91* (Rheumatic heart failure) reported as ’positive’ while goldstandard set only contains the next ancestor *398.9* (Other and unspecified rheumatic heart diseases) still counts as a hit. Evaluation for the non-extended SVM is reported directly through binary testing against each code.

B. Precision at n

To report precision at n (’P@n’) for the *FastText-UMLS* model, FastText’s *test* function is fed with the label count parameters $n = 5$ and $n = 8$ for each cross-validation fold.

For the extended *eFastText-UMLS* model, FastText’s *predict* function is used for each document within the testing data for that fold. FastText’s report outputs are piped and also processed with the NCBO BioPortal hierarchy tree [10] to resolve ancestors and descendants. Precision and recall is reported consequently by assuming ancestors of an assigned ICD-9 code must also be positive while descendants of a negative ICD-9 code must also be negative (example available in Section V-A).

C. Label count for multi-label classification

FastText lacks a native way to estimate label count. For this reason, FastText F1-Scores are obtained through 3 different ways in the corpus: (1) By using the average truncated number of assigned codes (label *cardinality*) from all bootstrap samples $n = 11$, (2) by *thresholding* the prediction subset with FastText’s *predict-prob* function with a threshold parameter of $t = 0.09$ also obtained from bootstrap training and for comparing purpose (3) in an oracle approach by using the original label count given by each test set, that is the amount to predict is obtained directly from the test set (*read off*).

VI. Discussion

To map free text narratives to concepts from knowledge sources like UMLS and then train classifiers on document representations that include UMLS Concept Unique Identifiers (CUIs) as features is a common approach [19]. Using basic UMLS concept mappings for word embeddings evaluated with ICD code assignment has not been tested in prior work and performs well. However the UMLS embeddings only consider predefined matched UMLS concept terms. Therefore relevant context information around biomedical concepts is not being covered by the model. A performance improvement can be expected by including additional embeddings to augment the text representation.

Free text narratives -including discharge summaries- often contain information about past patient illness, which is not always compliant to the ICD coding for the corresponding

ICU stay. Previous work showed that attention or text segmentation can be useful to detect and weigh relevant parts of discharge summaries [2].

Another recent approach is to learn rich representations of medical language by using CNNs. Gehrmann et al. [20] showed that CNN representation outperfoms both rule-based entity extraction and n-gram-based methods by evaluating the model on ten different phenotyping tasks. CNNs are able to enrich representation by identifying the relevant phrases that lead to a positive classification. To train a knowledge-guided CNN model with word embeddings and UMLS CUIs entity embeddings for clinical text representation is promising and considerable for future work [21].

VII. Conclusion

This paper introduces SVM and FastText with basic UMLS concept mappings into word embedding models evaluated with ICD code assignment. Because prior work within this field did not use knowledge sources as text representation, comparability is limited and performance improvements can be expected in all models. FastText achieves high recall results and mentionable performance improvements when evaluating with different label count estimation approaches on a hierarchy extended model. Despite the large initial dataset, training time is considerably small. That is because of the reduced corpus, the hierarchical softmax as output function and the general efficiency of FastText, which also allows for k-fold cross-validation. eFastText-UMLS$_{\text{cardinality}}$ yielded a F1-Score of 62.2% and outperformed previous models. While label count estimation through thresholding can be useful for high precision, it lacks in high recall. DR-CAML model outperforms FastText in precision at n.

References

[1] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, and Z. Lu, ”ML-Net: Multi-label classification of biomedical texts with deep neural networks,” *arXiv preprint arXiv:1811.05475*, 2018.

[2] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, ”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), New Orleans, Louisiana, USA, June, 2018, pp. 1101–1111.

[3] A. E. W. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, ”MIMIC-III, a freely accessible critical care database,” *Scientific data*, vol. 3, p. 160035, 2016.

[4] C. Lindberg, ”The Unified Medical Language System (UMLS) of the National Library of Medicine,” *Journal (American Medical Record Association)*, vol. 61, no. 5, pp. 40–42, 1990.

[5] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad, ”Diagnosis code assignment: Models and evaluation metrics,” *Journal of the American Medical Informatics Association*, vol. 21, no. 2, pp. 231–237, 2013.

[6] N. SpolaôR, E. A. Cherman, M. C. Monard, and H. D. Lee, ”A comparison of multi-label feature selection methods using the problem transformation approach,” *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135–151, 2013.

[7] R. Kavuluru, A. Rios, and Y. Lu, ”An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records,” *Artificial intelligence in medicine*, vol. 65, no. 2, pp. 155–166, 2015.

[8] T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad, and N. Elhadad, ”Multi-label classification of patient notes: Case study on ICD code assignment,” *in Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, February 2-7, 2018, pp. 409–416.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ”Efficient estimation of word representations in vector space,” *in 1st International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, May 2-4, 2013, arXiv:1301.3781.

[10] P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, and M. A. Musen, ”BioPortal: Enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications,” *Nucleic acids research*, 39(suppl 2), pp. W541–W545, 2011.

[11] L. Soldaini and N. Goharian, ”QuickUMLS: a fast, unsupervised approach for medical concept extraction,” *in Proceedings of the 2nd SIGIR workshop on Medical Information Retrieval (MedIR)*, Pisa, Italy, July 17-21, 2016.

[12] N. Okazaki and J.'i. Tsujii, ”Simple and Efficient Algorithm for Approximate Dictionary Matching,” in Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), Beijing, China, August 23-27, 2010, pp. 851–859.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., ”Scikit-learn: Machine learning in Python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[14] C.-C. Chang and C.-J. Lin, ”LIBSVM: A library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, ”Enriching Word Vectors with Subword Information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[16] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, ”Bag of Tricks for Efficient Text Classification,” *in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Volume 2 (Short Papers) Valencia, Spain, April 3-7, 2017, pp. 427–431.

[17] GitHub Repository for model implementation is online available. https://github.com/0xhesch/UMLS-Mimiciii-Word-Embeddings.

[18] F. Morin and Y. Bengio, ”Hierarchical probabilistic neural network language model,” *in Proceedings of the international workshop on artificial intelligence and statistics (AISTATS)*, Barbados, January 6-8, 2005, pp. 246–252.

[19] V.N. Garla and C. Brandt, ”Knowledge-based biomedical word sense disambiguation: an evaluation and application to clinical document classification,” *Journal of the American Medical Informatics Association*, 20(5), pp. 882–886, 2012.

[20] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote Jr, E. T. Moseley, D.W. Grant, P.D. Tyler and L.A. Celi, ”Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives,” *PloS one 13*, no. 2, p. e0192360, 2018.

[21] L. Yao, C. Mao, Y. Luo, ”Clinical text classification with rule-based features and knowledge-guided convolutional neural networks,” *BMC medical informatics and decision making*, vol. 19, no. 3, p. 71, 2019.