

Deep Learning for Multi-Label ICD-9 Classification of Hospital Discharge Summaries

Rafael Miranda, Bruno Martins, Mário Silva, Nuno Silva, Francisca Leite

Abstract—Hospital discharge summaries are composed of free-text and drop-down textual fields, describing relevant information about the patient’s hospital stay and treatments. The variability of medical vocabulary makes the processing of these records an important natural language processing challenge. We leverage a deep neural network that combines word, word-type and character embeddings, recurrent and convolutional units and neural attention, for the generation of intermediate representations of the textual contents. The neural network also explores the hierarchical nature of the input data, by building representations from the sequences of words within individual fields. An auxiliary memory mechanism is also leveraged for providing a *a priori* prediction of classes, being provided as input. The model is evaluated on the automatic assignment of ICD-9 codes to diagnosis on discharge summaries from both Hospital Beatriz Ângelo (Portugal) and the MIMIC III dataset, showing to be flexible and adaptable.

Index Terms—Patient discharge summaries; Coding clinical text; Deep learning; Natural Language Processing; Multi-label classification.

1 INTRODUCTION

ELECTRONIC Health Records (EHR) are a common tool used in modern medicine, where the information about patient journey is recorded. In EHR are included the hospital discharge summaries, reports prepared by an health professional, outlining the patient’s complaint, diagnostic findings, therapy administered, etc. [1] Those aggregate the clinical information, containing structured and/or unstructured data. For quality and billing purposes, discharge summaries need to be coded according to International Classification of Diseases (ICD) nomenclature from the World Health Organization (WHO). This is a time consuming process which is performed by a team of coders i.e., physicians with specific training to code the events, analyze the data in the EHR and attribute an ICD code. Artificial intelligence algorithms can contribute to the optimization of this process as a computer aided decision support system for classification by combining free-text and structured fields.

Methodologies of this kind, based on supervised training of classification models involving the use of deep neural networks, were successfully applied to death certificates at the Portuguese Director General of Health (Direcção Geral da Saúde) [2]. The present project intends to extend the previously developed methodologies, achieving (1) the development of neural architectures capable of combining free-text information with structured information, (2) the development of mechanisms that allow multi-label ICD classification and (3) the development of effective mechanisms for addressing “few-shot learning”.

The rest of this article is organized as follows. Section 2 surveys important concepts on artificial neural networks and previous related work on natural language processing and, specifically, on discharge summaries. Section 3 details the proposed approach, presenting deep neural network architecture developed. Section 4 report experimental evaluation of the proposed method, presenting data pre-processing and statistical analysis, results obtained in ten different ablation experiments. Performance by chapter, a real-life

application and alternative datasets assessment are also presented. At last, Section 5 outlines the main conclusions and possible developments for future work.

2 CONCEPTS AND RELATED WORK

This section describes fundamental concepts on neural models applied to NLP problems and a related work revision.

2.1 Neural Models for Natural Language Processing

A neural network model [3] presents nodes arranged in layers that are connected to each other, allowing the information flow. Between the input layer and the output layer, there are “hidden” layers that produce a response accordingly to an activation function, deciding if input information is relevant or not.

In the simplest case, a single-node neural network computes a single output from multiple real-valued inputs by forming a linear combination according to input weights, and then processing the output through an activation function. Equation 1 shows how this can be written mathematically, where y refers to the returned prediction, $\mathbf{x} = \langle x_1, \dots, x_n \rangle$ is the vector of inputs, \mathbf{w} denotes the vector of weights, b is a bias term, and $\varphi(\cdot)$ is an activation function.

$$y = \varphi \left(\sum_{i=1}^n w_i \cdot x_i + b \right) = \varphi \left(\mathbf{w}^T \cdot \mathbf{x} + b \right) \quad (1)$$

More complex architectures are possible to be explored using this simple idea as fundamental unit. A Multi-Layer Perceptron (MLP) consists of a set of nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer. The input signal propagates through the network layer-by-layer, until it reaches the output node(s). In a feed-forward network with a single hidden

layer, the corresponding computations can be written as shown in Equation 2.

$$y = \varphi(\mathbf{B} \cdot \varphi'(\mathbf{A} \cdot \mathbf{x} + \mathbf{a}) + \mathbf{b}) \quad (2)$$

In the previous equation, \mathbf{x} is a vector of inputs and \mathbf{y} a vector of outputs. The matrix \mathbf{A} represents the weights of the first layer and \mathbf{a} is the bias vector of the first layer, while \mathbf{B} and \mathbf{b} are, respectively, the weight matrix and the bias vector of the second layer. The functions φ' and φ both denote an element-wise non-linearity.

Training the neural network corresponds to adapting all weights and bias parameters to their optimal values, given an input training set \mathbf{x} and the corresponding outputs \mathbf{y} . This problem can be solved applying the back-propagation algorithm. Back-propagation in neural networks moves backward from the final error through the outputs, weights and inputs of each layer, assigning those weights responsibility for a portion of the error. Adaptive Moment Estimation (Adam) algorithm [4] is extensively used with the purpose of adjusting network weights and biases by increasing or decreasing them whichever gradient direction decreases error, computing parameter updates leveraging an exponentially decaying average of past gradients, together with adaptive learning rates for each parameter.

As an example of more complex architectures, we firstly present convolutional neural networks (CNNs). Consider a sequence of words $\mathbf{x} = x_1, \dots, x_n$, each with their corresponding d_{emb} dimensional word embedding $\epsilon(x_i)$. A 1D convolution layer of width k operates by moving a sliding window of size k over the sentence and applying the same filter to each window in the sequence $\epsilon(x_i), \epsilon(x_{i+1}), \dots, \epsilon(x_{i+k-1})$. Let the concatenated vector of the i th window be $\mathbf{w}_i = \epsilon(x_i), \epsilon(x_{i+1}), \dots, \epsilon(x_{i+k-1})$, $\mathbf{w}_i \in \mathbb{R}^{k d_{emb}}$. The result of the convolution layer is m vectors $\mathbf{p}_1, \dots, \mathbf{p}_m$, $\mathbf{p}_i \in \mathbb{R}^{d_{conv}}$ where:

$$\mathbf{p}_i = \sigma(\mathbf{w}_i \cdot \mathbf{W} + \mathbf{b}) \quad (3)$$

σ is a non-linear activation function that is applied element-wise, $\mathbf{W} \in \mathbb{R}^{k \cdot d_{emb} \times d_{conv}}$ and $\mathbf{b} \in \mathbb{R}^{d_{conv}}$ are parameters of the network. Each \mathbf{p}_i is a d_{conv} dimensional vector, encoding the information in \mathbf{w}_i . The m vectors are then combined using a max pooling layer, resulting in a single d_{conv} dimensional vector \mathbf{c} , as described by Equation 4.

$$c_j = \max_{1 < i \leq m} \mathbf{p}_i[j] \quad (4)$$

$\mathbf{p}_i[j]$ denotes the j th component of \mathbf{p}_i . The effect of the max-pooling operation is to get the most salient information across window positions.

Another example of neural networks are Recurrent neural networks (RNNs). In RNN architectures is taken not just the current input instance but also what was perceived one step back in time. More formally, given a sequence $\mathbf{x} = (x_1, x_2, \dots, x_t)$, a RNN updates its recurrent hidden state \mathbf{h}_t by sequentially processing the input sequence and computing:

$$\mathbf{h}_t = \varphi(\mathbf{W} \cdot \mathbf{x}_t + \mathbf{U} \cdot \mathbf{h}_{t-1}) \quad (5)$$

In brief, we have that the hidden state \mathbf{h}_t at time step t is a function of the input at the same time step \mathbf{x}_t , modified by a weight matrix \mathbf{W} . This result is added to

the hidden state of the previous time step \mathbf{h}_{t-1} , multiplied by its own hidden-state-to-hidden-state matrix \mathbf{U} . Previous research has noted that standard RNNs have difficulties in modeling long sequences, and extensions have been proposed to handle this problem. A well-known example are GRUs, originally proposed by Cho et al. [5] and detailed further ahead in this paper.

2.2 Natural Language Processing and Text Classification

NLP is defined as the automatic manipulation of natural language e.g., speech and text, by software. Large amounts of data, lack of structured information or alternate orthography are some of the major difficulties that arise on NLP. Since NLP relates also to classification problems, multi-label classification and lack of balanced data are also questions to be considered. This section is intended to present some of the state-of-the-art approaches presented in recent years.

A textual structure can be thought of as an hierarchical structure i.e., a text is composed of sentences, a sentence is composed of words, a word is composed of characters. Horn et al. [6] proposes an extension of the word2vec model called context encoders (ConEc) which allows effortless creation of out-of-vocabulary embeddings as well as better representation of words with multiple meanings by multiplying word2vec embeddings by word context vectors. In clinical text context, Patel et al. [7] propose a method to add task specific information to pre-trained word embeddings by adapting CBOW algorithm, adding information from medical coding data, in order to deal with clinical synonyms and abbreviations. Going deeper on word structure, Bojanowski et al. [8] propose an approach based on the Mikolov's skipgram model, learning representations for character n -grams, representing words as the sum of n -gram vectors.

Deep neural network models represent the state-of-the-art on models for NLP and Text Classification, thus being presented thereafter. Johnson and Zhang [9] proposed a 15 weight layers-deep CNN architecture that can efficiently represent long-range associations in text. The architecture combines a first layer for text region embedding with a convolution/pooling block stacking. Adding attention mechanisms, Shen et al. [10] propose bi-directional block self-attention network (Bi-BloSAN), a model that splits the entire sequence into length-equal blocks and applies an intra-block self-attention network (SAN) for modeling local context, then applying inter-block SAN to the outputs for all blocks to capture global dependency.

Other state-of-the-art approach is the introduction of auxiliary memory mechanisms in deep neural network models. Memory-based Parameter Adaptation (MbPA), proposed by Sprechmann et al. [11], is a method for augmenting neural networks with an episodic memory that stores examples and then uses a context-based look-up to locally adapt neural network parameters. Wang et al. [12] propose to enhance neural network models by allowing training set instance-level information from k -nearest neighbor (kNN) of the input text. kNN is also used as an external memory, being the final prediction made based on the text embedding of the input text, the attentive kNN label distribution and the attentive kNN text embedding.

Since multi-class and multi-label scenarios are likely to be associated with NLP and text classification problems, approaches to address high label space dimension and unbalanced training sets are also useful to look upon. On the extreme multi-label text classification (XMTC) topic, Liu et al. [13] present an approach based on a CNN model that takes into account multi-label co-occurrence patterns (XML-CNN). On the one-shot learning topic, Kaiser et al. [14] presented a large-scale life-long memory module, consisting of key-value pairs and exploiting fast nearest-neighbor algorithms, with ability to return not only the single nearest neighbor but also return a number of them to be processed by other layers of the network.

2.3 Clinical Text Coding

On what regards to discharge summaries, the most common EHR, a large variety of studies have been developed. Most of this studies have been experimented on global databases e.g., MIMIC database, written in English.

Baumel et al. [15] presented a bidirectional GRU with a Hierarchical Attention mechanism (HA-GRU) approach for multiple ICD code assignment, which consist on a hierarchical model with two levels of bi-GRU encoding, the first operating over tokens and the second encoding the document, an attention mechanism over the second GRU and a fully-connected layer with softmax activation for output prediction. Also in the topic of recurrent neural networks, Yani et al. [16] present the Grounded RNN (GRNN), a RNN architecture for Intensive Care Unit (ICU) discharge summaries multi-label prediction which explicitly ties labels to specific dimensions of the recurrent hidden state. Prakash et al. [17] present a condensed memory neural network (C-MemNNs) model for multi-label classification from free-text, a model with iterative condensation of memory representations that preserves the hierarchy of features in memory.

Mullenbach et al. [18] reported Convolutional attention for multi-label classification (CAML). An attention mechanism selects the parts of the document that are most relevant for each possible code and a regularizer is employed so that, if a code is rarely observed, it will encourage its parameters to be similar to those of other codes with similar WHO descriptions. Rios and Kavuluru [19] model comprehends an approach that applies two distinct neural networks over (1) a support subset containing instances that support the classification and over (2) an input instance, projected accordingly to the application of distinct convolutional filters. Each input will be associated with k distinct labels, being k predicted using an additional output unit (*MetaLabeller*).

Referring to a non-English dataset, Boytcheva work [20] presents max-wins voting-multi-class SVM as an approach to automatic mapping of ICD-10 codes to diagnoses extracted from discharge letters in Bulgarian language. Particular attention was given to the using of abbreviation sets in Latin and Bulgarian to substitute abbreviation words by its expanded terms meanings in Latin and Bulgarian language.

3 PROPOSED APPROACH

This work proposes a deep neural network model for automatic assignment of ICD-9 codes to discharge summaries, leveraging as input pre-selected free-text (i.e., patient's complaint at admission, physician diagnosis and clinic resume of the event) and complementary structured information (i.e., age group and department of stay), taking inspiration on previous work by Duarte et al. [2].

Figure 1 presents the proposed neural network, which is detailed in the next subsection. The entire model is trained end-to-end from a set of coded discharge summaries, leveraging the back-propagation algorithm [21] in conjunction with the Adam optimization method [4]. The implementation of the model relied mostly on the Keras¹ deep learning

1. <https://keras.io/>

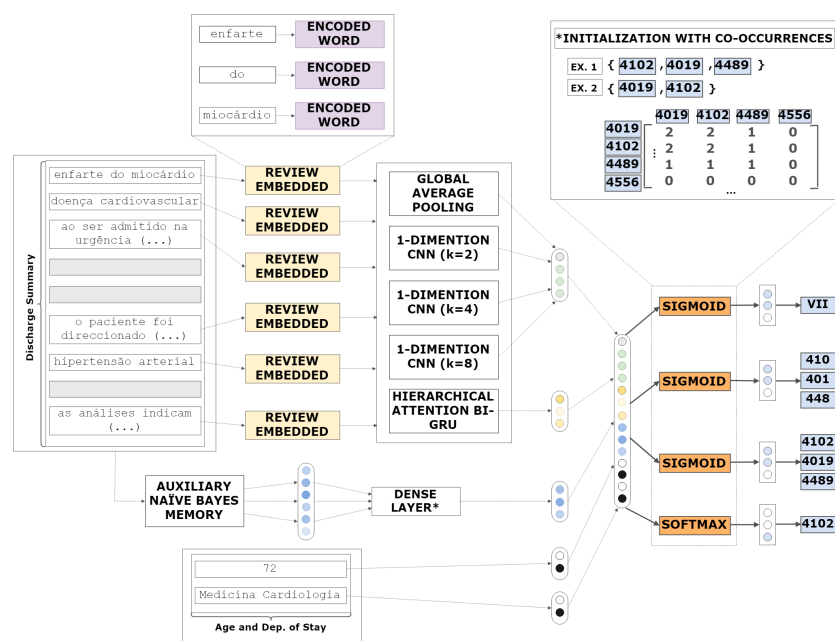


Fig. 1: The proposed neural network architecture.

library, using as computational backend TensorFlow². Resources from other machine learning and NLP libraries such as scikit-learn³, scikit-multilearn⁴ or NLTK⁵ were also used for specific operations.

3.1 A Hierarchical Attention Model Combined with a Convolutional Neural Network and a Memory Mechanism

The proposed model takes three hierarchical text inputs i.e., character-level input, word-level and syntax-level (i.e., a level that associates a semantic case to each token accordingly to the type of characters that compose it), age group and department of hospital stay, fed to the model as one-hot vectors, and an auxiliary memory mechanism fed as a 2D array of class probabilities.

For each of the three hierarchical inputs, the model first builds representations of individual fields, then aggregating those into an encompassing representation. The embedded representation, which mechanism is displayed on Figure 2, of these inputs will be leveraged by three 1D convolutional layers, with kernel sizes of, respectively, 2, 4 and 8. Each convolutional layer leverage the encompassing representation of the initial hierarchical inputs, being the output a result from max-pooling operation. This mechanism is represented at Figure 3. The outputs from each of the three convolutional layers are concatenated with the embedding average of the encompassing representation, a simpler mechanism that takes inspiration from Joulin et al. [22].

Subsequently, a bi-directional GRU is used at both word and field levels to build the representations. Mechanism is shown at Figure 4. Notice that GRUs in the first level of the model leverage the encompassed representation of the three hierarchical inputs as input, whereas the second level uses as input the field representations generated at the first level.

A GRU computes the next hidden state \mathbf{h}_t given a previous hidden state \mathbf{h}_{t-1} and the current input \mathbf{x}_t using two gates (i.e., a reset gate \mathbf{r}_t and an update gate \mathbf{z}_t), that control how the information is updated, as shown in Equation 6. The update gate (Equation 7) determines how much past information is kept and how much new information is added, while the reset gate (Equation 9) is responsible for how much the past state contributes to the candidate state. In Equations 6 to 9, $\tilde{\mathbf{h}}_t$ stands for the current new state, \mathbf{W} is

2. <http://www.tensorflow.org>
3. <http://scikit-learn.org>
4. <http://scikit.ml/>
5. <https://www.nltk.org/>

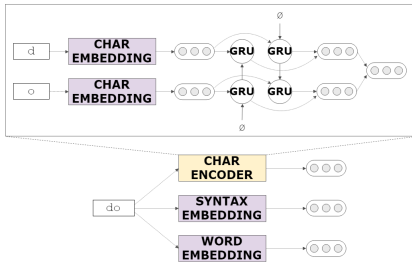


Fig. 2: Word, syntax and character embedding mechanism. Review embedded will result from concatenating this results, obtained for each field of each discharge summary.

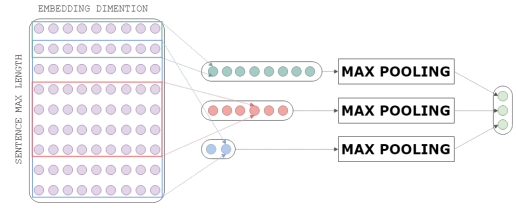


Fig. 3: 1D convolutional layers. Figure presents three convolutional layers for kernel sizes of 2, 4 and 8.

the parameter matrix for the actual state, \mathbf{U} is the parameter matrix for the previous state, and \mathbf{b} a bias vector.

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \quad (6)$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \cdot \mathbf{x}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1} + \mathbf{b}_z) \quad (7)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{r}_t \odot (\mathbf{U}_h \cdot \mathbf{h}_{t-1} + \mathbf{b}_h)) \quad (8)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot \mathbf{x}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1} + \mathbf{b}_r) \quad (9)$$

Bi-directional GRUs perceive the context of each input in a sequence by outlining the information from both directions. Concatenating the output of processing a sequence forward $\overrightarrow{\mathbf{h}}_{it}$ and backwards $\overleftarrow{\mathbf{h}}_{it}$ grants a summary of the information around each position, $\mathbf{h}_{it} = [\overrightarrow{\mathbf{h}}_{it}, \overleftarrow{\mathbf{h}}_{it}]$.

Since different words and fields can be differently informative in specific contexts, the model also includes two levels of attention mechanisms (i.e., one at the word-level and one at the field-level). For instance, in the case of the word-level part of the network, the outputs \mathbf{h}_{it} of the bi-directional GRU encoder are fed to a feed-forward node (Equation 10), resulting in vectors \mathbf{u}_{it} representing words in the input. The attention weights α_{it} are calculated as shown in Equation 11, using a context vector \mathbf{u}_w that is randomly initialized. The importance weights in α_{it} are then summed over the whole sequence, as shown in Equation 12.

$$\mathbf{u}_{it} = \tanh(\mathbf{W}_w \cdot \mathbf{h}_{it} + \mathbf{b}_w) \quad (10)$$

$$\alpha_{it} = \frac{\exp(\mathbf{u}_{it}^T \cdot \mathbf{u}_w)}{\sum_t \exp(\mathbf{u}_{it}^T \cdot \mathbf{u}_w)} \quad (11)$$

$$\mathbf{s}_i = \sum_t \alpha_{it} \cdot \mathbf{h}_{it} \quad (12)$$

The vector \mathbf{s}_i from Equation 12, which corresponds to a weighted sum of the bi-GRU outputs, is finally taken as

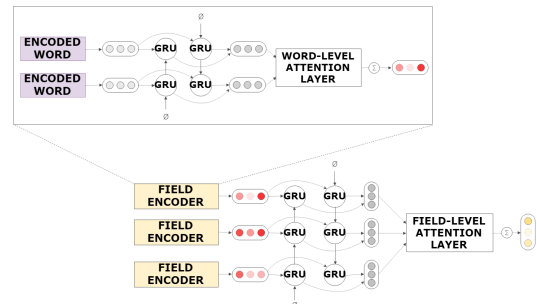


Fig. 4: Hierarchical attention mechanism and bi-directional Gated Recurrent Unit mechanism.

the representation of the input. The part of the network that processes the sequence of fields similarly makes use of bi-GRUs with an attention mechanism, taking as input the representations produced for each field.

Also, an auxiliary memory mechanism arise from applying a baseline Naive Bayes model to the training set and then feeding the predicted probabilities of each label to the model as an input. Multi-label problem transformation strategies were applied to the dataset e.g., Label Powerset (LP) and Dataset Extension (DE), as described at Tsoumakas et al. [23] as *PT3*, *PT4* and *PT5*, respectively, in order to allow baseline model application. LP considers each different set of labels that exist in the multi-label dataset as a single label. It so learns one single-label classifier $H : X \rightarrow P(L)$, where $P(L)$ is the power set of L . In contrast, DE decomposes each example (x, Y) into $|Y|$ examples (x, l) for all $l \in Y$ and learns one single-label *coverage-based* classifier from the transformed dataset afterward.

The representation that is produced as the output of the field-level attention mechanism is finally concatenated with the alternative representation built through the CNN, the output from the Naïve Bayes auxiliary memory mechanism and the age and department inputs.

The final layer is composed by four distinct output nodes, namely (i) a softmax node that outputs the ICD-9 full-code of the main diagnosis, (ii) a sigmoid node that outputs multiple ICD-9 codes, (iii) a sigmoid node for ICD-9 blocks and (iv) a sigmoid node for ICD-9 chapters. Nodes (i), (ii) and (iii) can be initialized through a non-negative matrix factorization [24] [25] over a label co-occurrence matrix, considering a number of components for the decomposition that is equal to the dimensionality of the combined input representation.

4 EXPERIMENTAL METHODOLOGY

This section presents experimental methods and consequent results. Statistical dataset analysis, ablation tests results and performance on alternative datasets are some of the discussed themes.

TABLE 1: Statistical characterization of the main dataset used in the experiments.

Number of distinct ICD-9 main codes	3,511
Number of distinct ICD-9 codes	5,793
Number of distinct ICD-9 blocks	873
Number of distinct ICD-9 chapters	18
Number of ICD-9 codes per instance	1.868
Number of ICD-9 blocks per instance	1.818
Number of ICD-9 chapters per instance	1.531
Number of entries in the dataset	112,044
Number of entries with multiple ICD-9 codes	36,456
Average number of sentences per instance	14.770
Average number of words per instance	80.772
Average number of words per sentence	4.788
Training set vocabulary size	145,376
Number of out-of-vocabulary word types in the test set	37,689

4.1 Pre-Processing and Dataset Analysis

Prior to using the data collection provided (after local Institutional Review Board ethical approval), a pre-processing phase had to take place in order to obtain a suitable dataset.

A discharge summary instance is composed by 116 fields, which can be Boolean, drop-down list one-liners or short/long free-text. Through qualitative analysis, options were narrowed down to five fields, due to medical and ICD coding significance, filling percentage, structure and low-rate of numerical quantities. Clinician diagnosis (9 strings), patient’s complaint at admission (1 string) and clinic resume of the event (25 strings), among other non-textual inputs such as age and department of stay (one-hot vectors), were the selected input fields. The provided data collection presented initially 151,536 instances, of which 112,044 remained after removing the ones where the number of fields was different from 121 or the field relative to clinician diagnosis was unfilled. Each of the 35 strings is padded at the beginning and at the end with a special unique token that encodes the beginning/termination of a sentence. Numerical token replacement took place in cases in which the ratio between number of digits and total number of characters of the token was larger than 0.5, replacing the numerical token with a common token <NUMBER>.

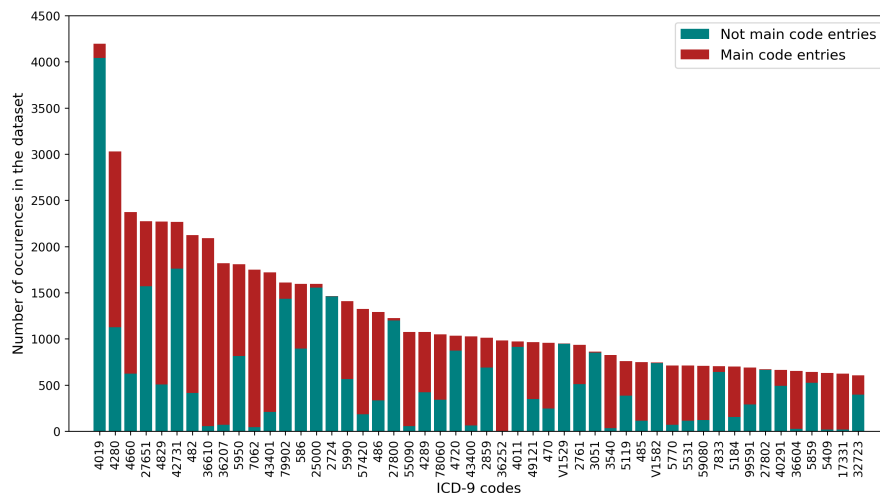


Fig. 5: Number of occurrences of the 50 most common ICD-9 codes in the dataset.

ICD-9 codes were provided together with its textual description and multiple codes in the same string, separated by "Enter" e.g. 182.3.5- OCLUSÃO DA RETINA; 169.7-PARAGEM CARDIO-RESPIRATÓRIA;. Textual descriptions were removed and each discharge summary was matched with a list of the corresponding ICD-9 codes. Codes occurring less than two times in the dataset as main code (i.e., first code of the list) were removed.

The dataset used in the experiments presents the statistical profile described in Table 1. Input information is stored together with the ICD-9 main full-code and the ICD-9 full-codes corresponding to every diagnosis identified in one discharge note.

As already stated, one of the main challenges related to the aforementioned dataset is class unbalancing. Figure 5 shows the distribution for the number of occurrences of the 50 most common ICD-9 full-codes. In fact, 5,270 codes out of the 5,793 distinct ICD-9 codes found (i.e., around 90% of the code set) present a number of counts below 75. 75% of the code set present a count number below 20 and almost half of the code set present a count number below 5 occurrences.

Another challenge related to this dataset is the multi-label scenario. Figure 6 presents the frequency of discharge notes associated to each number of ICD-9 codes, blocks and chapters per note. 68% of the instances of the dataset are associated to only one ICD-9 code, therefore being useful to evaluate single-label scenario. Also, the number of instances associated to only one ICD-9 block is larger than the one related to full-code, which demonstrates that a discharge summary can be associated to distinct codes from the same block. ICD-9 chapters case is analogous, proving the hierarchical structure of the ICD nomenclature.

Available data was split into two subsets, with 75% (84,033 instances) for model training and 25% (28,011 instances) for testing.

Fields related to patient's age group and department of stay were considered useful inputs since correlation between this fields and ICD nomenclature was found.

Figure 7 presents an heatmap stating co-occurrences between ICD-9 chapters and 6 distinct age groups. For

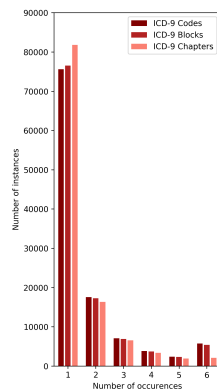


Fig. 6: Number of ICD-9 full-codes, blocks and chapters per instance of the dataset. Columns associated with number of occurrences equal to 6 refer to the sum of occurrences above 6.

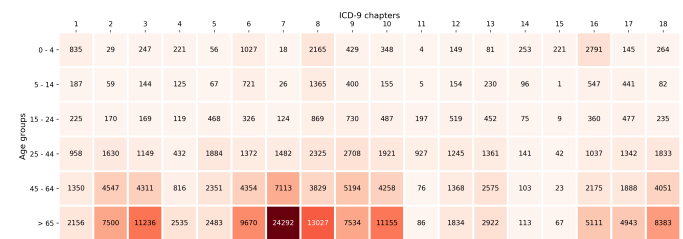


Fig. 7: Heatmap with number of ICD-9 code occurrences per age group.

instance, code occurrences related to ICD-9 chapter 11 i.e., *Complications of Pregnancy, Childbirth, and the Puerperium*, is associated 72% of the times with ages between 25 and 44 (87% in ages between 15 and 44), being unlikely for the model to match instances associated to ages outside this range to codes from chapter 11.

In addition, Figure 8 presents the percentage of hospital discharges associated to the top 10 ICD-9 main codes by age group. It is possible to infer that diseases such as senile cataract, diabetic macular edema and inguinal hernia only appear in the three latter age groups. Also, for the 45 to 64 and the over-65 age group it is possible to infer that the percentage is more evenly distributed than for prior age groups.

Similarly to age groups, department of stay have also shown correlation to ICD-9 nomenclature. For instance, in the case of ophthalmology surgery, 96% of code occurrences associated to this department are also associated to ICD-9 chapter 6 i.e., *Diseases of the Nervous System and Sense Organs*.

Besides numerical token replacement, which allowed a decrease from a vocabulary size of 411,498 to 145,376, tokens with a single occurrence were removed from the vocabulary, narrowing down vocabulary size to 70,598. Out-of-vocabulary words were substituted by the most similar word on the vocabulary, according to the Jaro-Winkler string distance metric [26].

Embedding layers in the first level of the model and GRU output considered a dimensionality of 150. Model training was made in batches of 15 instances, using the Adam optimization algorithm [4] with default parameters. Model training also considered a stopping criteria based on the combined training loss, finishing when it stopped

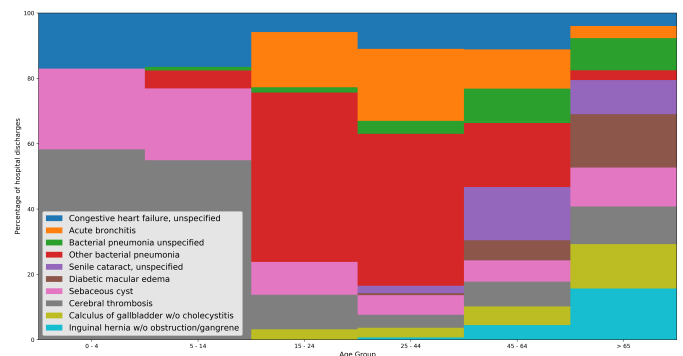


Fig. 8: Percentage of hospital discharges associated to the 10 most common ICD-9 codes appearing as main code.

TABLE 2: Performance metrics for neural network baselines and ablation tests. Assessed metrics were Accuracy (Acc.), Precision (Prec.), Recall (Rec.), F1-Score (F1), Hamming Loss(HL), Exact Match (Match), One Error (OneE) and MRR.

	ICD Level	Single-label					Multi-label							
		Acc.	Prec.	Rec.	F1	MRR	Acc.	Prec.	Rec.	F1	HL	Match	OneE	MRR
(1) Average of Word Embeddings	Full-code	34.12	3.22	4.10	3.19	41.37	26.34	31.95	27.43	28.06	2.8E-5	22.17	67.43	39.12
	Block	45.98	10.43	10.31	9.34	-	39.60	47.94	42.02	42.40	2.3E-5	32.73	50.14	53.22
	Chapter	66.27	50.64	49.73	49.56	-	52.94	61.89	61.28	58.21	1.5E-5	39.19	34.39	73.09
(2) Model 1 + CNN	Full-code	41.38	14.16	13.43	13.93	46.64	34.10	40.47	36.15	36.39	2.5E-5	28.27	58.64	52.41
	Block	53.64	24.70	22.31	22.56	-	49.56	59.32	51.43	52.60	1.9E-5	42.00	39.56	69.19
	Chapter	70.27	59.75	58.93	59.23	-	65.35	77.16	67.53	69.25	1.2E-5	55.39	21.86	84.42
(3) Duarte et al. [2]	Full-code	44.65	16.65	15.62	15.00	50.04	35.46	42.34	37.59	37.93	2.4E-5	29.31	56.15	55.02
	Block	57.35	27.34	25.17	25.19	-	51.41	61.44	53.31	54.57	1.9E-5	43.53	36.99	70.87
	Chapter	73.68	64.45	62.66	63.38	-	67.87	79.35	70.63	71.96	1.1E-5	57.26	19.02	86.54
(4) Model 3 + CNN	Full-code	44.58	16.49	15.42	14.61	50.05	34.95	42.24	37.02	37.41	2.5E-5	28.81	56.46	53.05
	Block	57.17	28.58	25.19	25.26	-	50.80	61.58	52.75	54.07	1.9E-5	42.75	36.96	69.39
	Chapter	72.97	63.13	62.26	62.54	-	66.22	77.94	69.51	70.53	1.1E-5	55.06	20.30	85.06
(5) Model 4 + Structured Inputs	Full-code	44.07	17.31	16.51	15.48	49.98	35.79	43.03	36.70	37.89	2.4E-5	30.57	56.48	53.55
	Block	56.18	28.52	26.23	25.79	-	50.91	61.36	52.01	53.86	1.9E-5	43.69	37.83	70.02
	Chapter	72.76	63.00	62.57	62.64	-	67.74	79.81	69.80	71.69	1.1E-5	57.56	19.11	86.41
(6) Model 5 + DE AuxMem	Full-code	44.81	16.89	15.54	14.90	50.92	35.03	41.54	40.31	38.29	2.4E-5	27.03	55.29	54.95
	Block	56.75	28.46	25.41	25.47	-	50.18	59.41	55.28	54.22	1.9E-5	40.10	36.90	71.27
	Chapter	73.39	63.80	63.34	63.39	-	67.04	77.90	71.35	71.55	1.1E-5	55.12	19.64	86.13
(7) Model 5 + LP AuxMem	Full-code	45.72	17.53	16.63	15.73	51.22	36.65	43.89	38.60	39.11	2.4E-5	30.56	55.06	55.08
	Block	57.68	30.14	26.66	26.78	-	51.89	62.23	53.87	55.15	1.9E-5	43.81	36.40	71.60
	Chapter	74.33	64.29	62.88	63.40	-	68.14	80.00	70.92	72.33	1.1E-5	57.32	18.54	86.68
(8) Model 6 + Char/Word-type Embeddings	Full-code	38.36	8.49	8.13	6.98	45.20	31.16	38.36	31.83	33.13	2.6E-5	26.29	61.42	49.31
	Block	51.23	18.50	16.59	15.74	-	46.40	56.52	47.11	49.11	2.1E-5	39.73	42.95	65.51
	Chapter	70.07	60.18	57.10	58.26	-	64.45	76.60	65.65	68.08	1.2E-5	55.22	22.88	83.61
(9) Model 8 leveraging Sparse Attention	Full-code	40.36	10.69	9.97	8.79	46.85	31.64	38.75	31.83	33.44	2.6E-5	27.25	61.13	48.59
	Block	54.85	22.63	19.96	18.96	-	49.16	60.00	49.65	51.97	2.0E-5	42.35	39.66	66.57
	Chapter	73.37	64.67	60.09	61.24	-	66.65	78.25	69.06	70.59	1.1E-5	56.51	20.50	85.07
(10) Proposed Approach	Full-code	42.71	14.13	13.42	12.76	48.81	33.22	40.56	33.91	35.14	2.5E-5	29.15	59.05	51.13
	Block	56.59	25.77	23.18	22.01	-	50.56	61.12	51.11	53.53	2.0E-5	42.91	38.05	68.13
	Chapter	73.24	64.61	61.12	61.20	-	67.69	79.23	69.77	71.06	1.1E-5	57.22	18.93	86.41

decreasing or the number of training epochs was equal to 50, and a model checkpoint, saving the model that presented the minimum value of combined training loss. Maximum sentence length was of 30 words and maximum word length was of 20 characters.

4.2 Experimental Results on Ablation Tests

Ablation tests were conducted in order to assess the improvement enabled by each feature added to the model proposed in Section 3. We tested the influence of (i) aggregating a convolutional neural network to this model, (ii) providing an auxiliary memory mechanism, (iii) substituting the regular attention mechanism by a sparse attention mechanism and (iv) leveraging (or not) the label co-occurrence non-negative matrix factorization for initializing the weights of the output nodes.

Thereby, results will be presented for 10 distinct neural network architectures. These will be evaluated both on single-label and multi-label scenarios. Therefore, the considered models are the following:

- 1) A model that only uses the word embedding average mechanism;
- 2) A model that combines a three-layer convolutional neural network with the word embedding average mechanism;
- 3) Duarte et al. [2] proposed architecture, without considering mechanisms for initializing the weights of the output nodes;
- 4) A model that combines the previous architecture with a three-layer convolutional neural network;
- 5) The previous model leveraging age group and department of hospital stay inputs, besides the already provided textual input;
- 6) A model that combines the previous architecture with a baseline Naïve Bayes auxiliary memory mechanism;
- 7) A model similar to the previous one, leveraging the auxiliary memory mechanism with a Label Power-set multi-label problem transformation;
- 8) The full model, as described in Section 3, without considering mechanisms for initializing the weights of the output nodes;
- 9) The previously described model, considering a sparse attention mechanism instead of the regular attention mechanism;
- 10) The full model, as described in Section 3.

Table 2 presents the obtained results for each of the 10 models on both single and multi-label scenarios. Besides Mean Reciprocal Rank (MRR), single-label scenario performance metrics are calculated using scikit-learn package definitions. All multi-label metrics were custom made, based on Pereira et al. [27] work, since scikit-learn metrics presented some limitations on multi-label scenario.

In terms of single-label scenario, model 7 presents the best accuracy values for full-code, block and chapter prediction i.e., 45.72%, 57.68% and 74.33%, respectively. Model 9 presents the best macro-average precision value for chapter

TABLE 3: Performance metrics results for each of the ICD-9 chapters. The column *Percentage* gives the fraction of codes in the dataset, corresponding to each ICD-9 chapter. Assessed metrics were Precision (Prec.), Recall (Rec.) and F1-Score (F1).

Chapter	Percentage	Prec.	Rec.	F1
1	2.745	51.40	48.83	50.09
2	6.654	72.56	73.80	73.17
3	8.233	57.99	39.78	47.19
4	2.049	51.20	51.06	51.13
5	3.504	81.80	69.98	75.43
6	8.334	89.55	89.17	89.36
7	15.743	75.29	78.85	77.03
8	11.239	75.79	79.37	77.54
9	8.109	81.82	83.47	82.64
10	8.740	72.25	78.57	75.28
11	0.616	67.69	68.89	68.28
12	2.535	79.46	80.10	79.78
13	3.653	82.17	81.40	81.79
14	0.391	48.51	47.11	47.80
15	0.203	7.14	11.54	8.82
16	5.744	37.28	32.31	34.62
17	4.420	75.35	71.37	73.31
18	7.088	49.90	46.24	48.00
Macro AVG:		64.29	62.88	63.40

prediction (64.67%) and Model 6 presents the best macro-average recall value for chapter prediction (63.34%).

In multi-label scenario, in which every full-code, block or chapter presenting a probability above the defined threshold is selected, Model 7 presents the best results for 6 out of 8 multi-label metrics, for full-code, block and chapter predictions. Model 6 presents the best recall performance values for full-code, block and chapter predictions i.e., 40.31%, 55.28% and 71.35%, respectively. Full-code and chapter exact match prediction values are the best for Model 5 i.e., 30.57% and 57.56%, respectively.

The introduction of age group and department of stay inputs produce an increase on single-label macro-averaged metrics, although producing a decrease on single-label accuracy. Also, on multi-label scenario, accuracy, precision and exact match present an increase for full-code prediction, confirming the relevance of this inputs for model training.

On other topic, adding a memory mechanism produces an increase on performance, since an *a priori* prediction is provided as input of the neural network model. Although the LP-leveraged memory mechanism produces a better performance than dataset extension approach, it also produces large amounts of memory usage, obligating to several model adaptations in order to decrease complexity e.g., lowering dense layer dimensions, batch size and string size or producing smaller and less complex vocabulary.

Although Models 8, 9 and 10 present a lower performance than less complex models, this is a case of fine tuning of the model parameters, not achieved due to processing memory limitations and shortness of time to test alternatives.

4.3 Single-label Performance Assessed by ICD Chapter

As already described at Subsection 4.1, this dataset presents a very uneven distribution of ICD-9 codes. 75% of the code set presents less than 20 occurrences and, in contrast, codes 4019, 4280 and 4660, the top-3 codes, present frequencies higher than 2,000 occurrences. In fact, codes 4019 and 4280

belong to chapter 7 and code 4660 to chapter 8, which correspond to the most populated chapters of the dataset. Table 3 presents the fraction of codes in the dataset and the single-label performance metrics values for each of the ICD-9 chapters. Chapter 7, *Diseases of the Circulatory System*, and chapter 8, *Diseases of the Respiratory System*, represent 15.743% and 11.239% of chapter frequency, respectively. In contrast, chapters 14, *Congenital Anomalies*, and 15, *Certain Conditions Originating in the Perinatal Period*, represent only 0.391% and 0.203% of the dataset chapter frequency.

Chapter 15 is the chapter presenting the worse model performance in single-label prediction, with performance metrics of 7.14%, 11.54% and 8.82% (precision, recall and F1-score, respectively) for chapter prediction. Codes between blocks 764 and 779 have their fifth digit defined by birthweight of the newborn. Since numerical tokens were substituted by a common token, it is not possible to achieve differentiation at this level. This argument, associated to low code frequency, explain low performance of the model in this chapter.

Chapter 18, *Supplementary Classification Of Factors Influencing Health Status And Contact With Health Services*, presents performance below average although presenting an above average fraction of codes in the dataset i.e., 7.088%. In fact, codes belonging to this chapter are usually associated to preexisting conditions, therefore not being main codes, typically. Since codes occurring less than two times in the dataset as main code were removed, some codes belonging to this chapter became main codes, producing prediction mistakes in single-label scenario.

In contrast, chapters 6, *Diseases Of The Nervous System And Sense Organs*, 9, *Diseases Of The Digestive System* and 13, *Diseases Of The Musculoskeletal System And Connective Tissue* present values for performance metrics above 80%, presenting a high effectiveness chapter prediction.

TABLE 4: Probability of finding at least one of the discharge summaries' labels in the first n predicted labels, $n \in \{1, 5, 10, 25, 50, 100\}$.

HR@1	HR@5	HR@10	HR@25	HR@50	HR@100
38.867	59.755	67.124	75.206	80.290	84.920

4.4 Real-life Application Performance Assessment

An approach of decision support model i.e., the model output being a list of the top- n ranked labels, ordered by prediction probability, was also assessed. Observing MRR values for best model's performance i.e., 51.22% for single-label full-code and 55.08%, 71.60% and 86.68% for multi-label full-code, block and chapter predictions, respectively, it is possible to infer that a less strict usage of the model can be also useful.

In order to assess model performance in this scenario, a Hit Rate (HR) [28] was measured. HR states that if at least one of the true labels was included in the top- n predicted labels, the result would be favorable (1). Otherwise, the result is unfavorable (0). Table 4 presents the probability of finding at least one of the discharge summaries' labels in the first n predicted labels.

TABLE 5: Statistical characterization of MIMIC III and Death Certificates alternative datasets.

	MIMIC III	Death Certificates
Number of distinct ICD main codes	1,768	1,418
Number of distinct ICD codes	6,496	2,446
Number of distinct ICD blocks	929	611
Number of distinct ICD chapters	19	19
Number of ICD codes per instance	12.264	2.239
Number of ICD blocks per instance	11.267	2.200
Number of ICD chapters per instance	6.534	1.851
Number of entries in the dataset	46,539	121,536
Number of entries with multiple ICD codes	46,368	85,553
Average number of sentences per instance	17.131	2.255
Average number of words per instance	98,890	11,893
Average number of words per sentence	10,378	4,812
Training set vocabulary size	24,795	14,230

Considering a list of 25 relevant predictions, a performance of approximately 75% is achieved. This corresponds to narrowing down the number of possible ICD-9 labels from over 17,500 to 25, which represents decreasing the set of possible labels in 99.9%. Nonetheless, the model would need to be tuned to allow a threshold-like approach in which the model would provide the top-25 list only if a quality threshold could be surpassed.

4.5 Full Model Performance on Alternative Datasets

As a method of best-performance model validation, performance was assessed in two additional datasets: (i) MIMIC III and (ii) death certificates from Duarte et al. [2] work. Table 5 presents a similar statistical analysis to the one presented at Table 1 for HBA dataset.

MIMIC III dataset is statistically more complex than HBA dataset, comprehending a larger label space on a lower dataset entry number, with a ratio of 7.138 dataset entries per distinct code whereas HBA dataset presents a ratio of 19.341. Also, the number of codes per instance is 6.565 times bigger than for HBA dataset. In terms of the textual fields, MIMIC III dataset exhibits larger average number of sentences and words per instance and of words per sentence than HBA dataset. Nonetheless, since this dataset is English written, misspellings due to accented characters are less probable, being produced a smaller amount of new tokens due to this situation.

In contrast, Duarte et al. [2] dataset of death certificates, from here mentioned as *DDC dataset*, presents a dataset entry per distinct code of 49.688, for a similar dataset entry count to HBA dataset. Textual inputs for this dataset are short descriptions of the underlying cause of death, instead of reports of the patient stay at the hospital, resulting in a shorter vocabulary size.

Table 6 presents the obtained results for model 7 performance on each of the alternative datasets for single-label and multi-label scenario.

Model 7 had to suffer some alterations due to dataset constraints in order to allow model training. For MIMIC III dataset, age group input was discharged since this information is not available on this dataset. In the case of DDC dataset, besides age group, department of stay was also discharged since this EHR is not compatible with this information. Also, the number of textual input sentences was narrowed from 35 to 9 sentences, corresponding to the original input form of Duarte et al. [2] architecture. Another important issue of concern is that MIMIC III equivalent to department of stay is a free-text field. 26 distinct departments were considered valid and object of textual homogenization. The remaining textual options were mapped to a generic department.

From Table 6 analysis for Model 7 performance on DDC dataset it is possible to verify that this model can not achieve results reported by Duarte et al. [2]. As already explained, model parameters had to be adapted in order to allow HBA dataset training without causing memory exhaustion. Nonetheless, it is possible to infer that the model performs well on a multi-label scenario, with a MRR of 88.82% for ICD-10 full-code prediction, as well as a precision of 77.64%.

On MIMIC III dataset, our best model performs at the same level of some state-of-the-art models mentioned as related work. Multi-label precision results on MIMIC III are in line with the performance of the same model on HBA dataset, whereas accuracy, recall and F1-score are much lower due to the fact that the average number of predicted labels by the model is much lower than the true number of labels for each discharge summary. Nonetheless, MRR values (e.g., 69.73% for full-code multi-label prediction) are

TABLE 6: Performance metrics for neural network baselines and ablation tests. Assessed metrics were Accuracy (Acc.), Precision (Prec.), Recall (Rec.), F1-Score (F1), Hamming Loss(HL), Exact Match (Match), One Error (OneE) and MRR.

	ICD Level	Single-label					Multi-label							
		Acc.	Prec.	Rec.	F1	MRR	Acc.	Prec.	Rec.	F1	HL	Match	OneE	MRR
Death Certificates	Full-code	72.42	18.11	16.46	15.71	78.62	60.30	77.64	65.68	68.42	1.2E-5	36.13	14.83	88.82
	Block	77.31	30.09	25.53	25.47	-	63.79	80.78	69.01	71.83	1.0E-5	39.00	11.94	91.05
	Chapter	87.33	64.71	58.40	60.32	-	77.96	87.62	84.09	83.97	0.6E-5	58.19	5.40	96.54
MIMIC III	Full-code	35.82	10.56	9.21	8.91	43.03	9.30	47.15	10.26	15.28	7.4E-5	0.01	46.35	69.73
	Block	44.41	16.86	14.40	14.40	-	16.06	56.70	18.34	25.16	6.1E-5	0.02	32.77	78.95
	Chapter	65.42	41.36	41.91	41.08	-	48.43	71.15	62.03	63.00	2.1E-5	0.33	9.63	94.59

higher on this dataset than for HBA dataset.

TABLE 7: Probability of finding at least one of the death certificates' labels in the first n predicted labels, $n \in \{1, 5, 10, 25, 50, 100\}$.

HR@1	HR@5	HR@10	HR@25	HR@50	HR@100
84.768	93.599	95.300	96.952	97.940	98.730

Table 7 presents Hit Rate metrics for Death Certificates classification. Results reveal that is possible to find at least one correct label between the top-5 highest ranked predicted labels with a probability of 93.599%, proving the proposed architecture potential as recommendation system.

5 CONCLUSIONS AND FUTURE WORK

In the present work it was developed a deep neural network model to automatically classify hospital discharge summaries which combines different mechanisms for generating intermediate representations including word, syntax and character representations, two levels of GRU, a convolutional network similar to the proposal by Liu et al. [13] and neural attention mechanisms. An auxiliary memory mechanism based on an *a priori* prediction is provided as an additional input to the neural network. The model was tested in three distinct datasets e.g., HBA, MIMIC III and death certificates, performing distinctly according to the assessed dataset. Experimental results are in line with the actual state-of-the-art methods. Consequently, it also highlights the challenges of applying machine learning with real-world data from difference institutions and database provenance. Stop-word filtering, medical abbreviate and acronym unroll, word and character n -gram embeddings, feature selection or sparsemax activation function usage are some future work considerations to be held.

ACKNOWLEDGMENTS

I would like to express my gratitude to INESC-ID and Grupo Luz Saúde for allowing and creating the necessary structure to develop this project. A special thanks to professors Bruno Martins and Mário Silva and Eng. Nuno Silva and Eng. Francisca Leite for mentoring this work.

REFERENCES

- [1] I. Mosby, *Mosby's medical dictionary*. Mosby, 2006.
- [2] F. Duarte, B. Martins, C. S. Pinto, and M. J. Silva, "Deep neural models for icd-10 coding of death certificates and autopsy reports in free-text," *Journal of biomedical informatics*, vol. 80, pp. 64–77, 2018.
- [3] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, 2016.
- [4] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations*, 2015.
- [5] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of the Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.
- [6] F. Horn, "Context encoders as a simple but powerful extension of word2vec," in *Proceedings of the ACL Workshop on Representation Learning for NLP*, 2017.
- [7] K. Patel, D. Patel, M. Golakiya, P. Bhattacharyya, and N. Birari, "Adapting pre-trained word embeddings for use in medical coding," in *Proceedings of the ACL-SIGBioMed Workshop on Biomedical Natural Language Processing*, 2017.
- [8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, 2017.
- [9] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2017, pp. 562–570.
- [10] T. Shen, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Bi-Directional Block Self-Attention for Fast and Memory-Efficient Sequence Modeling," *ArXiv e-prints*, Apr. 2018.
- [11] P. Sprechmann, S. M. Jayakumar, J. W. Rae, A. Pritzel, A. P. Badia, B. Urias, O. Vinyals, D. Hassabis, R. Pascanu, and C. Blundell, "Memory-based parameter adaptation," *arXiv preprint arXiv:1802.10542*, 2018.
- [12] Z. Wang, W. Hamza, and L. Song, "k-nearest neighbor augmented neural networks for text classification," *CoRR*, vol. abs/1708.07863, 2017. [Online]. Available: <http://arxiv.org/abs/1708.07863>
- [13] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17. New York, NY, USA: ACM, 2017, pp. 115–124. [Online]. Available: <http://doi.acm.org/10.1145/3077136.3080834>
- [14] L. Kaiser, O. Nachum, A. Roy, and S. Bengio, "Learning to remember rare events," in *Proceedings of the International Conference on Learning Representations*, 2017.
- [15] T. Baumel, J. Nassour-Kassis, M. Elhadad, and N. Elhadad, "Multi-label classification of patient notes a case study on icd code assignment," *arXiv preprint arXiv:1709.09587*, 2017.
- [16] A. Vani, Y. Jernite, and D. Sontag, "Grounded recurrent neural networks," *arXiv preprint arXiv:1705.08557*, 2017.
- [17] A. Prakash, S. Zhao, S. A. Hasan, V. V. Datla, K. Lee, A. Qadir, J. Liu, and O. Farri, "Condensed memory networks for clinical diagnostic inferring," in *AAAI*, 2017, pp. 3274–3280.
- [18] J. Mullenbach, S. Wiegrefe, J. Duke, J. Sun, and J. Eisenstein, "Explainable prediction of medical codes from clinical text," *arXiv preprint arXiv:1802.05695*, 2018.
- [19] A. Rios and R. Kavuluru, "Emr coding with semi-parametric multi-head matching networks," 2018.
- [20] S. Boytcheva, "Automatic matching of ICD-10 codes to diagnoses in discharge letters," in *Proceedings of the ACL-SIGBioMed Workshop on Biomedical Natural Language Processing*, 2011.
- [21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive Modeling*, vol. 5, no. 3, 1988.
- [22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2017.
- [23] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.
- [24] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, 1999.
- [25] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, 2007.
- [26] W. E. Winkler, "The state of record linkage and current research problems," Statistical Research Division of the US Census Bureau, Tech. Rep. 2006-2, 2006.
- [27] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. Merschmann, "Correlation analysis of performance measures for multi-label classification," *Information Processing Management*, vol. 54, no. 3, pp. 359 – 369, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306457318300165>
- [28] Q. Han, M. Ji, I. M. d. R. de Troya, M. Gaur, and L. Zejnilovic, "A hybrid recommender system for patient-doctor matchmaking in primary care," *arXiv preprint arXiv:1808.03265*, 2018.