Automated Classification of Causes of Mortality

Francisco Duarte
francisco.ribeiro.duarte@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

September 2017

Abstract

This work addresses the automatic assignment of ICD-10 codes for causes of death by analyzing free-text descriptions in death certificates, together with the associated autopsy reports and clinical bulletins, from the Portuguese Ministry of Health. The proposed method leverages a deep neural network that combines word embeddings, recurrent units, and neural attention as mechanisms for the generation of intermediate representations of the textual contents. The neural network explores the hierarchical nature of the input data, by building representations from the sequences of words within individual fields, which are then combined according to the sequences of fields that compose the input. Moreover, innovative mechanisms for initializing the weights of the final nodes of the network are explored, leveraging co-occurrences between classes together with the hierarchical structure of ICD-10. Experimental results attest to the contribution of the different neural network components. The best model achieves accuracy scores over 89%, 81%, and 76%, respectively for ICD-10 chapters, blocks, and full-codes. Through examples, it is also shown that the proposed method can produce interpretable results, useful for public health surveillance.

Keywords: Automated ICD Coding, Clinical Text Mining, Deep Learning, Natural Language Processing, Artificial Intelligence in Medicine

1. Introduction

The systematic collection of high-quality mortality data is essential for the surveillance of a population's health, and for conducting mortality and other epidemiologic studies. For these and other legal purposes, doctors have to write death certificates, i.e. reports containing personal data of the deceased and textual descriptions for the causes of death, as well as any contributing conditions or injuries. In Portugal, doctors are submitting death certificates in electronic format to the Death Certificate Information System (SICO), for data collection and registry purposes [1]. The analysis of causes of death includes classifying the death certificates according to revision 10 of the International Statistical Classification of Diseases and Related Health Problems (ICD[1]), which is distributed by the World Health Organization. ICD defines diseases, and other health conditions, in a comprehensive hierarchical fashion. Despite having all the data centrally in digital form, the assignment of ICD-10 codes to the free-text descriptions provided by doctors is still made manually by mortality coders with specific expertise, after submission to SICO.

Figure 1 presents a screen-shot of the online form presented by SICO to collect a death certificate. The form has two parts, delimited by the solid lines in the figure. Part I comprises up to four fields of text (i.e., boxes marked from *a)* to *d)*) for reporting a chain of events leading directly to death, where the underlying cause of death should be given in the lowest line and the immediate cause in the first one. Part II is optional, and it is used for reporting other significant diseases, conditions, or injuries that contributed to death, but are not part of the main causal sequence leading to death. In complement to the death certificate, a clinical information bulletin is also filled by the doctor be-

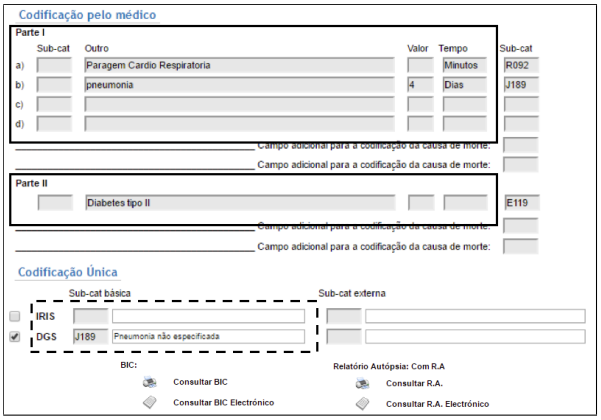

Figure 1: The form used in Portugal for death certificates registration and entering ICD-10 codes.

[1] http://www.who.int/classifications/icd/

fore the death certificate itself, describing relevant clinical information of the patient. The clinical bulletin is mandatory, but doctors often do not associate the clinical bulletin to the death certificate. In case of violent and unknown causes of death, an autopsy report can be requested by the Public Ministry. These auxiliary reports can be accessed from the death certificate form within SICO, as shown at the bottom of Figure 1. After a manual review of the data, the mortality coder should assign the ICD-10 code corresponding to the underlying cause of death in the box shown under the dashed line.

The manual coding of the free-text contents in death certificates and/or autopsy reports is a challenging, expensive, and time consuming task [2], which slows down the process of disseminating mortality statistics and prevents death surveillance by disease in real time. However, given the past efforts in manually coding death certificates, these classified certificates can be used to inform supervised machine leaning methods capable of assigning codes automatically. Such automated approaches can be used to speed-up the process of publishing mortality statistics, by quickly producing results that can latter be revised through manual coding. When integrated into existing platforms, automated approaches can also facilitate the task of manual coding, by providing coding hints. If sufficiently accurate, automatic coding also has the potential to reduce the cost of physician involvement, and to increase coding consistency.

In this article, we propose a deep neural network that processes the full-text contents of death certificates, clinical bulletins, and autopsy reports. The network is trained end-to-end from a set of manually coded instances of death certificates, and it combines different mechanisms for generating intermediate representations, including two levels of Gated Recurrent Units (GRUs) for modeling sequential data within and between the textual fields that compose the inputs [3, 4], averages of word embeddings similarly to the proposal by Joulin et al. [5], and neural attention mechanisms for highlighting relevant parts of the inputs [6, 4].

We report on experiments with a dataset referring to $121,536$ deceased individuals, covering all cases between 2013 and 2015 in Portuguese territory, except for neonatal and perinatal mortality. Using 25% of the dataset as a testing set, we evaluated the predictive capabilities of the proposed approach by measuring results in terms of classification accuracy, as well as macro-averaged precision, recall, and F1-scores. Given the hierarchical organization of ICD-10 (i.e., the codes are organized hierarchically into chapters, blocks and full-codes), we also measured results according to different levels of code specification.

Our complete model achieved an accuracy of 89.2%, 81.2%, and 75.9%, respectively when considering ICD-10 chapters (i.e., a total of 19 distinct classes appearing in our dataset), blocks (611 distinct classes) and full-codes ($1,418$ distinct classes). We argue that the obtained results indicate that automatic approaches leveraging supervised machine learning can indeed contribute to a faster processing of death certificates. Our experiments also show that the neural attention mechanisms led to an increased performance, offering at the same time much needed model interpretability, by allowing us to see which parts of the input are attended to, when making predictions.

In complement to the main set of experiments, we also report on tests with a second dataset, referring to the year of 2016 and still undergoing the process of manual coding at the time of preparing this article. Leveraging the full-model from the first round of tests, trained with 75% of the data from 2013-2015, we again measured the predictive accuracy of the proposed method, in an attempt to see if it could generalize across time periods, obtaining similar results.

The rest of this document is organized as follows: Section 2 surveys previous related work. Section 3 details the proposed approach, presenting the architecture of the deep neural network that was considered the task. Section 4 presents the experimental evaluation of the proposed method, detailing the datasets, the evaluation methodology, and the obtained results. Finally, Section 5 summarizes our main conclusions and presents possible directions for future work.

2. Related Work

Various studies have addressed the automatic assignment of ICD codes to clinical text. Different methods were for instance presented at the 2007 Computational Medicine Challenge (CMC), which involved about 50 participants [7]. The goal was to automate the assignment of ICD-9 codes to free-text radiology reports, with basis on a training set of 978 documents and a test set of 976 documents. The top-performing system used an ensemble of multiple models that achieved a micro-averaged F1-score of 0.89, while the mean F1-score among all participants was of 0.77. The inter-annotator agreement, measured as the F1-score of individual annotators against an aggregated score obtained through majority voting, was also found to be comparable to those of the best systems.

Perotte et al. stressed how the current volume of health care data can be used to support the automated assignment of ICD codes to clinical text [8]. The authors used the publicly available Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC II) repository of records for patients in

Intensive Care Units (ICUs), to assess the performance of standard text classification methods for automatically coding patient discharge summaries. The MIMIC II dataset comprises 22,815 records collected between 2001 and 2008 from a variety of ICUs consisting of multiple fields (e.g., clinical notes and reports for cardiac catheterization, ECGs, radiology and echo tests) with a total of 5,030 distinct ICD-9 codes. Two different classification methods were tested, namely a flat classifier based on Support Vector Machines (SVMs), with one binary SVM per ICD-9 class, and a method based on a tree with 8 levels of SVM models, leveraging the hierarchical structure of ICD-9 (i.e., a method where the classifier associated with a given code in the hierarchy is applied only if its parent code has been classified as positive). Perotte et al. showed that the proposed hierarchical method outperformed the simpler approach that treated each ICD-9 code independently.

Yan et al. [9] and Wang et al. [10] have both proposed methods for automated ICD coding of data within electronic health records, combining linear classifiers (i.e., logistic regression or SVMs) with model regularization procedures that explore inter-code relationships (e.g., label co-occurrences over the training data, or other available prior knowledge) for improving multi-label classification. For instance Wang et al. compared different multi-label classification methods for ICD-9 coding, also using the MIMIC II dataset. The most innovative aspect in the work from Wang et al. relates to the proposal of a novel classification method based on logistic regression (i.e., the authors used a logistic loss combined with a $\ell_{2,1}$ -norm for inducing sparsity in the model parameters), which incorporates a graph structure that reflects the correlations between diseases (i.e., the regularization term of the model combines the feature weights with a class affinity matrix where each cell corresponds to the cosine similarity between a pair of classes, with basis on the class associations to individual training instances). The novel method was compared against previous approaches specifically designed for multi-label classification, using metrics that are also specific for multi-label problems (i.e., the Hamming loss and the ranking loss). The method leveraging disease correlations outperformed 6 alternative classification approaches and, in most cases, the note features had better results than the chart features.

Specifically on what regards death certificates, Koopman et al. described the use of SVM classifiers for identifying cancer related causes of death in natural language descriptions [11]. The textual contents were encoded as sparse binary feature vectors (i.e., term n-grams, vectors encoding the presence of terms, and SNOMED CT concepts recognized by a clinical natural language processing system named Medtex), and these representations were used as features to train a two-level hierarchy of SVM models: the first level was a binary classifier for identifying the presence of cancer, and the second level consisted of a set of classifiers (i.e., one for each cancer type) for identifying the type of cancer using the ICD-10 classification system (i.e., according to 85 different ICD-10 blocks, of which 20 instances corresponded to 85% of all cases). The system was highly effective at identifying cancer as the underlying cause of death, and at determining the type of common cancer having obtained a macro-averaged F1-score of 0.94 and 0.7 for each task, respectively.

Lavergne et al. described a large-scale dataset prepared from French death certificates, suitable to the application of machine learning methods for ICD-10 coding [12]. The dataset comprised a total of 93,694 death certificates referring to 3,457 unique ICD-10 codes, and it was made available for international shared tasks organized in the context of CLEF. The 2016 CLEF eHealth shared task was defined at the level of each statement (i.e., lines varying from 1 to 30 words, with outliers at 120 words and with the most frequent length at 2 tokens) in a death certificate, and statements could be associated with zero, one or more ICD-10 codes. The best-performing system achieved a micro-averaged F1-score (i.e., harmonic mean of precision and recall weighted by the class size) of 0.848, leveraging dictionaries built from the shared task data. At the time of preparing this article, the 2017 edition of the CLEF eHealth shared task was still underway.

Leveraging the dataset from the 2016 CLEF eHealth competition, Zweigenbaum et al. presented hybrid methods for ICD-10 coding of death certificates [13], combining dictionary linking with supervised machine learning (i.e., an SVM classifier leveraging tokens, character trigrams, and the year of the certificate as features). The best hybrid model corresponded to the union of the results produced by the dictionary-based and learning-based methods, outperforming the best system at the 2016 edition of the CLEF eHealth shared task with a micro-averaged F1-score of 0.8586.

Although different approaches for ICD coding of clinical text have been proposed in the literature, some of which specifically focusing on death certificates and/or autopsy reports, the current state-of-the-art is still relying on methods that are much simpler than those that constitute the current best practice on other text classification problems. Our work builds on ideas from the work surveyed in this section, in particular exploring class co-occurrences and the hierarchical nature of ICD-10, but we in-

troduce recent machine learning approaches based on the supervised training of deep neural networks that involve mechanisms such as recurrent nodes and neural attention.

3. The Proposed Approach

This work presents a deep neural network for assigning ICD-10 codes to underlying causes of death, by analysis of the free-text contents from death certificates, each associated with the respective clinical bulletin and autopsy report, taking inspiration on previous work by Yang et al. [4]. Considering the SICO platform from the Portuguese Ministry of Health's Directorate-General of Health (DGS), illustrated on Figure 1, the coding task was modeled as follows: given different strings encoding events leading to death, our model outputs the ICD-10 code of the underlying cause of death. For an in-depth introduction to deep neural networks for natural language processing, the reader can refer to the tutorial by Yoav Goldberg [14].

The network presented in Figure 2 explores a combination of different mechanisms to generate intermediate representations for the textual contents, such as word embeddings, a hierarchical arrangement of recurrent units, and neural attention. It also considers multiple outputs in an attempt to leverage existing relations between ICD-10 classes to further improve classification results, assuming that, given the hierarchical class structure of ICD-10 and since most of the full-codes are only sparsely used in the training data, using ICD-10 blocks can further assist the model training procedure. Moreover, this work also explores innovative mechanisms for initializing the weights of the final nodes of the network, leveraging co-occurrences between classes in the training data, together with the hierarchical structure of ICD-10.

The entire model is trained end-to-end from a set of coded death certificates, leveraging the back-propagation algorithm [15] in conjunction with the Adam optimization method [16]. At the output nodes of the network, the model training procedure combines loss functions computed from the ICD-10 full-code and the ICD-10 block for the main cause of death (i.e., categorical cross-entropy in the two softmax nodes shown in Figure 2), and from the ICD-10 codes encoding auxiliary and contributing conditions (i.e. a binary cross-entropy in the sigmoid node from the bottom of Figure 2, taking inspiration on a suggestion from Nam et al. [17]), respectively with weights 0.8, 0.85 and 0.75. The implementation of the model relied mostly on the keras[2] deep learning library, although the scikit-learn[3] machine learning package was also used for specific operations such as for computing the considered evaluation metrics.

In Section 3.1 a detailed overview of the Hierarchical attention model combined with the average of the embeddings is made. After that, in Section 3.2 there is a description of the method used to leverage the model using label co-occurrence to initialize parameters in the network.

3.1. A Hierarchical Attention Model Combined with the Average of the Embeddings

The inputs to the proposed model can be seen as having an hierarchical structure in which words form different fields, and the fields from the death certificate, clinical bulletin, and autopsy report form an input entry. The model first builds representations of individual fields, and then aggregates those into an encompassing representation. This two-level hierarchical approach is illustrated in Figure 2, with the word-level part of the model (i.e., the part that generates a representation from a given field) shown in the box at the top. A recurrent neural network node known as a Gated Recurrent Unit (GRU) is used at both levels to build the representations. Notice that the GRUs in the first level of the model leverage word embeddings as input, whereas the second level uses as input the field representations generated at the first level.

GRUs model sequential data by having a recurrent hidden state whose activation at each time step is dependent on that of the previous time step. A GRU computes the next hidden state h_t given a previous hidden state h_{t-1} and the current input x_t using two gates (i.e., a reset gate r_t and an update gate z_t), that control how the information is updated, as shown in Equation 1. The update gate (Equation 2) determines how much past information is kept and how much new information is added, while the reset gate (Equation 4) is responsible for how much the past state contributes to the candidate state. In Equations 1 to 4, $\tilde{h}_t$ stands for the current new state, W is the parameter matrix for the actual state, U is the parameter matrix for the previous state, and b a bias vector.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (1)$$

$$z_t = \sigma\big(W_z x_t + U_z h_{t-1} + b_z\big) \quad (2)$$

$$\tilde{h}_t = \tanh\big(W_h x_t + r_t \odot (U_h h_{t-1} + b_h)\big) \quad (3)$$

$$r_t = \sigma\big(W_r x_t + U_r h_{t-1} + b_r\big) \quad (4)$$

We specifically considered bi-directional GRUs [3], as they perceive the context of each input in a sequence by outlining the information from both directions. Concatenating the output of processing a sequence forward $\overrightarrow{h}_{it}$ and backwards $\overleftarrow{h}_{it}$ grants a summary of the information around each position, $h_{it} = [\overrightarrow{h}_{it}, \overleftarrow{h}_{it}]$.

[2] http://keras.io

[3] http://scikit-learn.org

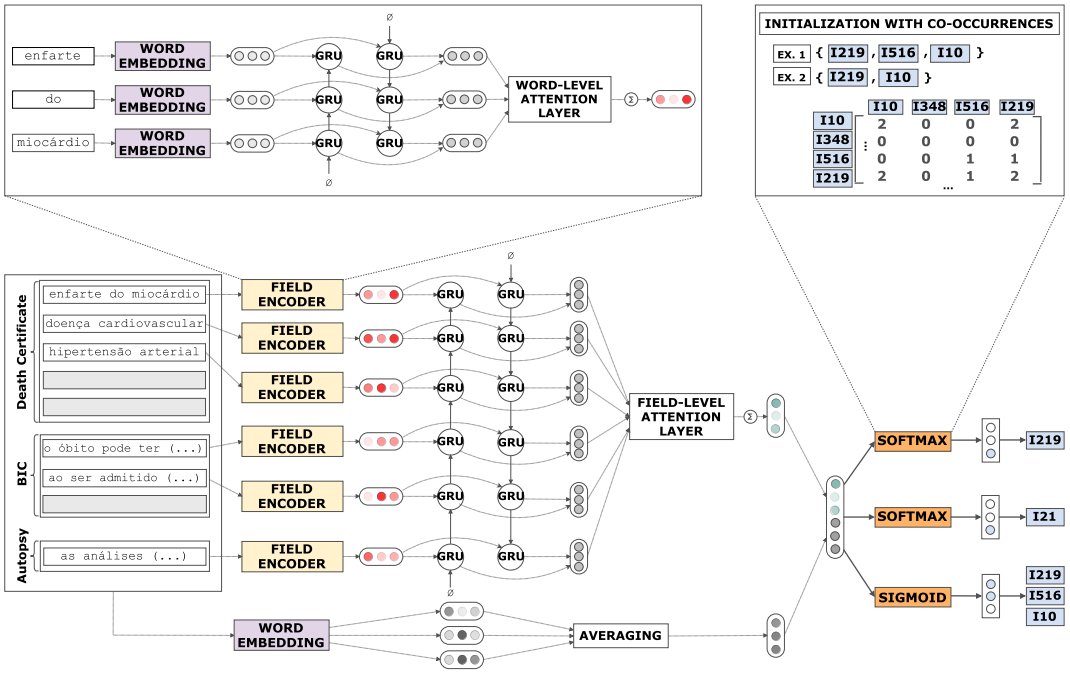

Figure 2: The proposed neural network architecture.

Since the 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), that let the model to pay more or less attention to individual words/fields when constructing representations (i.e., different weights will be used for the elements in the sequence of GRU ouptuts).

For instance, in the case of the word-level part of the network, the outputs h_{it} of the bi-directional GRU encoder are fed to a feed-forward node (Equation 5), resulting in vectors u_{it} representing words in the input. A normalized importance α_{it} (i.e., the attention weights) is calculated as shown in Equation 6, using a context vector u_w that is randomly initialized. Then, it is summed over the whole sequence, as shown in Equation 7.

$$u_{it} = \tanh\left(W_w h_{it} + b_w\right) \tag{5}$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{6}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{7}$$

The vector s_i from Equation 6 is finally taken as the representation of the input. The part of the network that processes the sequence of fields similarly makes use of bi-directional GRUs with an attention mechanism, taking as input the representations produced for each field, as shown in Figure 2.

The representation that is produced as the output of the field-level attention mechanism, which encompasses the entire output, is also concatenated with an alternative representation built through a simpler mechanism which, taking inspiration on the good results reported by Joulin et al. [5], computes the average of the embeddings for all words in the input fields. The word embeddings are randomly initialized and adjusted during model training. They are also shared by the hierarchical attention and the averaging mechanisms, and thus while one part of the model uses multiple parameters to compute representations for the inputs, the other part of the model can more directly propagate errors back into the embeddings, to be updated.

3.2. Initializing Model Parameters through Label Co-Occurrence

In the neural architecture illustrated on Figure 2, the representations resulting from the different fields are finally passed to feed-forward output nodes. Three separate outputs are considered in the model, namely (i) a softmax node that outputs the ICD-10 full-code of the underlying cause of death, (ii) another softmax node that outputs the ICD-10 block of the underlying cause of death, and (iii) a sigmoid node that outputs multiple ICD-10 codes, corresponding to all contributing and auxiliary conditions, together with the the cause of death.

Following the suggestion of Nam et al. [17], the proposed model relies on the sigmoid activation function and the binary cross-entropy loss function in the case of the node with the model outputs corresponding to multiple ICD-10 codes, given its superior performance in handling multi-label classification problems.

All three output nodes of the model can be initialized with weights that, given the list of auxil-

iary codes associated to each instance in the training set, try to capture the co-occurrences between ICD-10 codes. Two different approaches to compute the weight matrices of the output nodes were tested. One of these approaches is based on the previous work of Kurata et al. [18] and leverages the the Apriori algorithm [19] to find the most significant and frequent label co-occurrence patterns. The second approach uses a non-negative matrix factorization [20, 21] 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 (i.e., the node before the output nodes).

In the first strategy, the first part of the Apriori algorithm proposed by Agrawal and Srikant [19] is used for finding the sets ICD-10 codes that frequently appear together in the training data (i.e., finding frequent itemsets). These sets of auxiliary codes are used to initialize the weight matrices for the output nodes, following the method proposed by Kurata et al. [18].

The second technique considered for initializing the weights of the output nodes leverages the components of the decomposition that result from a Non-negative Matrix Factorization (NMF), applied to a matrix that encodes label co-occurrences in the training dataset. A square matrix $X_{m,m}$, where m stands for the dimensionality of the output node, is first built from the training data with basis on label co-occurrence information (i.e., each matrix cell corresponds to the number of co-occurrences of a pair of ICD-10 labels, and the values at the diagonal simply reflect the frequency of the label in the training data). To reduce the impact of the most common labels and their prevalence in co-occurrence information, the $X_{m,m}$ is scaled with a binary logarithm (i.e., $\log_2(1 + x_{i,j})$ for each matrix entry $x_{i,j}$). The NMF is then used to decompose the $X_{m,m}$ matrix into a product of two matrices, namely $X_{m,m} \approx W_{m,n} \times H_{n,m}$, where n stands for the dimensionality of the hidden node that captures the representation of the input. The matrix $H_{n,m}$ is finally used for initializing the node weights.

The problem of finding two non-negative matrices W and H whose product is approximately equal to the original non-negative matrix X relies on minimizing the following objective function with an alternating minimization of W and H:

$$\arg\min_{W,H} \frac{1}{2}\|X - WH\|^2_{Frobenius} = \frac{1}{2}\sum_{i,j}(X_{ij}WH_{ij})^2 \quad (8)$$

4. Experimental Evaluation

We first present a statistical characterization of the datasets that supported our tests, together with the considered experimental methodology. Then, Section 4.2 presents and discusses the obtained results.

Table 1: Statistical characterization of the dataset used in the experiments.

Number of distinct ICD-10 codes	1,418
Number of distinct ICD-10 blocks	611
Number of distinct ICD-10 chapters	19
Number of distinct ICD-10 codes	2,446
Number of entries in the dataset	121,536
Number of entries with filled death certificates	114,228
Number of entries with autopsy reports	5,653
Number of entries with clinical bulletins	3,003
Number of textual fields	274,501
Average number of words per textual field	6,68
Training set vocabulary size	29,284
Number of out-of-vocabulary words in the test set	5,260

4.1. Datasets and Experimental Methodology

The main dataset used in the experiments consists of the death certificates in SICO for the years 2013 to 2015, excluding neonatal and perinatal mortality. All supplemental clinical bulletin and autopsy reports were included, although these cases mostly corresponded to deaths associated to accidents, suicides, or homicides (see a simple statistical profile of the dataset in Table 1).

For each death certificate, the textual contents of fields labeled from *a)* to *d)* in Part I of the SICO form, as well as the contents from Part II were used as inputs to the model, in each case concatenating the strings labeled as *Outro*, *Valor* and *Tempo* (i.e., the fields named *Valor* and *Tempo* can be used to encode the approximated interval between the onset of the respective condition and the date of death, which can be relevant in cases like a stroke that occurred much before the time of death).

Of the six fields of the clinical bulletin, only three were considered: circumstances of admission, clinical situation, and diagnosis. These fields were used in the experiments, since the remaining fields are significantly less informative. An autopsy report consists of a single small textual description of the autopsy results.

Each instance in the dataset thus consists of 9 different strings, some of them possibly empty: 5 strings for each field in the death certificate, 3 for the clinical bulletin, and 1 for the autopsy report. Each of the 9 strings is padded with special symbols to encode the beginning/termination of the textual contents. The input information is stored together with the ICD-10 full-code corresponding to the underlying cause of death, the ICD-10 block for the underlying cause of death, and ICD-10 codes corresponding to conditions or injuries present in the deceased, other than those from the underlying cause of death.

The available data was split into two subsets, with 75% (91,152 instances) for model training and 25% (30,384 instances) for testing. We noticed that the dataset was very unbalanced in the number of certificates per ICD code, and that some ICD-10 chapters had no instances, given that the cor-

responding health problems are seldom related to death (i.e., Chapter VII, corresponding to diseases of the eye and adnexa).

The word vocabulary considered by the model was generated using the instances of the training subset. When pre-processing the testing set, out-of-vocabulary words (i.e., words from the testing set that were not present in the training set) were substituted by the most similar word on the vocabulary, according to the Jaro-Winkler string distance metric [22]. This set of words, $5,260$ in total, corresponds to approximately 18% of the vocabulary built from the training set. A manual analysis of the results showed that the certificates often include misspellings or alternative spellings for words (e.g., without diacritics), and hence the use of string similarity for matching related words.

To further test the performance of the proposed method, and to assess its generalization capabilities and its effectiveness in a near-real-time surveillance scenario, a second dataset was used, consisting of $86,071$ instances corresponding to deaths occurring in 2016, also manually assigned to ICD-10 codes.

For assessing the quality of the model predictions, the classification accuracy over the test split was measured, as well the macro-averaged precision, recall and F1-scores (i.e., macro-averages assign an equal importance to each class, thus providing useful information in the case of datasets with a highly unbalanced class distribution and when the system is required to perform consistently across all classes, regardless of how densely populated these are). Given the hierarchical organization of ICD-10, results according to the level of specialization of ICD-10 terms were also measured, considering chapters, blocks, and full-codes. Similar measurements were also taken with the dataset of instances from 2016.

All experiments relied on the keras[4] deep learning library, and the tests involving non-negative matrix factorization relied on an implementation from the scikit-learn library[5]. Model training used the Adam optimization algorithm [16] with default parameters. Model training also considered a stopping criteria based on the combined training loss, finishing when the difference between epochs was less than 0.3.

4.2. Experimental Results

The first set of experiments compared six different neural network architectures, in an attempt to assess the contribution of the different mechanisms: (i) A model that only uses the average word embedding mechanism; (ii) A hierarchical model with two levels of GRUs but without the attention mechanisms, thus using the hidden states at the edges of the sequences in order to build the intermediate representations; (iii) A hierarchical model with two levels of GRUs and with the attention mechanisms at each level, inspired on the proposal from Yang et al. [4]; (iv) A model that combines the previous hierarchical attention approach with the average word embedding mechanism; (v) The full model combining hierarchical attention and average word embeddings, as described in Section 3, with 3 output nodes and initializing the weights of the output nodes by exploring frequent co-occurrence patterns; (vi) The full model, as described in Section 3, leveraging non-negative matrix factorization for initializing the weights of the output nodes.

Table 2 presents the results obtained by each model. The best value in terms of accuracy for full-code prediction was obtained by the full model leveraging initialization with non-negative matrix factorization, corresponding to a value of 75.948%.

To further assess the overall performance of the proposed method, the Mean Reciprocal Rank (MRR) of the correct class was also computed, when sorting classes according to the probability assigned prior to performing the softmax operation associated to full ICD-10 codes. Model 6 has a MRR of 0.804 when assigning full-codes, 0.845 for blocks, and 0.915 for ICD-10 chapters, again attesting to the good predictive accuracy of the proposed neural network architecture.

The most common causes of death in the dataset correspond to ICD-10 Chapters II (i.e, neoplasms) and IX (i.e., diseases of the circulatory system). Together, these ICD-10 codes represent approximately 56.6% of the instances. Table 3 further details the results obtained by Model 6 in these two important chapters. It is also noticeable that deaths with underlying cause in Chapter XVIII (i.e., symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified) were predicted with high effectiveness (i.e., an F1-score of 89.542%, the third largest in terms of individual chapters).

In 2016 there were a total of $111,279$ deaths in Portugal and, by July of 2017, a fraction of 77.3% of these cases, corresponding to $86,071$ death certificates, had already been manually reviewed and coded according to ICD-10. A second round of experiments attempted to classify these $86,071$ instances from 2016, leveraging Model 6 from the previous experiments, trained with data from 2013-2015. The number of instances for each of the ICD-10 chapters in the 2016 dataset is similar to the one in the dataset from the years 2013 to 2015, and the performance metrics for ICD-10 chapters, blocks and full-codes can be seen in Table 4. The accuracy values are very similar to those obtained from the test subset (i.e., an accuracy of 75.901%

[4] http://keras.io

[5] http://scikit-learn.org

Table 2: Performance metrics for different variants of the neural model.

	ICD Level	Accuracy	Macro-averages		
			Precision	Recall	F1-Score
Average of Word Embeddings	Chapter	74.362	38.733	39.679	38.219
	Block	54.930	9.512	9.163	8.616
	Full-code	49.760	4.487	4.679	4.120
Hierarchical GRUs	Chapter	83.570	52.227	51.115	51.582
	Block	72.420	27.712	24.210	24.675
	Full-code	67.647	18.032	16.139	15.983
Hierarchical GRUs with Attention	Chapter	88.938	65.228	62.406	63.265
	Block	80.588	36.569	34.667	34.033
	Full-code	75.043	24.386	23.913	22.584
Combined Model	Chapter	89.267	**68.522**	63.780	**65.478**
	Block	81.132	37.022	35.125	34.398
	Full-code	75.632	23.222	23.174	21.619
Combined Model with Frequent Itemset Initialization	Chapter	**89.320**	67.656	**64.297**	65.372
	Block	**81.349**	38.792	36.011	35.782
	Full-code	**76.112**	25.136	24.228	23.084
Combined Model with NMF Initialization	Chapter	89.159	64.092	62.202	62.907
	Block	81.207	**44.649**	**39.900**	**40.505**
	Full-code	75.947	**29.513**	**27.773**	**27.042**

Table 3: Results for blocks and full-codes within ICD Chapters II and IX.

	ICD Level	Accuracy	Macro-averages		
			Precision	Recall	F1-Score
Chapter II	Block	90.518	34.762	31.317	32.546
	Full-code	86.743	31.846	29.914	29.756
Chapter IX	Block	82.313	18.199	14.487	15.492
	Full-code	78.389	17.812	14.201	15.027

for full-codes, 80.615% for blocks, and 89.129% for chapters), confirming that the proposed approach can generalize across different time periods.

Besides applications in near real-time death cause surveillance, the proposed approach can also be useful for assisting human coders. The results from Table 2, particularly when comparing the cells corresponding to Models 2 and 3, have already shown that the neural attention mechanisms can led to an increased performance. More interestingly, neural attention can also offer model interpretability, by allowing users to see which parts of the input (i.e., which fields and which words) are attended to, when making predictions for underlying causes of death.

Figure 3 illustrates the attention weights calculated as shown in Equation 6, for the contents of two death certificates in the testing set. These instances were not associated to a clinical bulletin or an autopsy report, and thus the figure is only showing the first four textual fields.

The certificate shown in Figure 3a was correctly assigned to code C719 (i.e., malignant neoplasm of brain, unspecified) with a confidence of 95.21%, and the figure shows the words *glioblastoma multiforme* having a significant impact. In turn, the certificate in Figure 3b was correctly assigned to code J40 (i.e., bronchitis, not specified as acute or chronic) with a confidence of 92.39%. In this example, the words *insuficiência cardíaca descompensada* in the first field have much less impact than the word *traqueobronquite* on the second field.

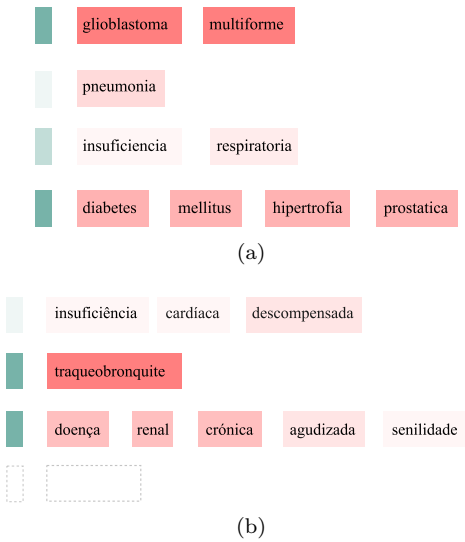

Figure 3: Distribution of attention weights given to different fields and tokens in two instances.

5. Conclusions and Future Work

We presented a deep learning method for assigning ICD-10 codes to the free-text descriptions for the underlying cause of death, included in death certificates, clinical bulletins, and autopsy reports obtained from the Portuguese Ministry of Health's Directorate-General of Health, according to ICD-10. Experimental results show that although ICD-10 coding is a difficult task, due to the large number of classes that are sparsely used, we can still obtain a high classification accuracy. We argue that

Table 4: Performance metrics over the 2016 dataset.

	ICD Level	Accuracy	Macro-averages		
			Precision	Recall	F1-Score
All Chapters	Chapter	89.129	59.994	52.748	54.510
	Block	80.615	34.938	29.525	30.363
	Full-code	75.901	21.349	19.343	18.832
Chapter II	Block	89.991	27.142	24.210	25.203
	Full-code	86.367	24.495	22.085	22.197
Chapter IX	Block	80.811	14.874	10.432	11.687
	Full-code	77.107	13.939	10.761	11.353

our approach can indeed contribute to a faster processing of death certificates, supporting near real-time surveillance of relevant ICD-10 blocks. Our approach can also help in the task of manual coding the certificates, providing ICD-10 code suggestions that are interpretable based on visualizations built from the neural attention weights highlighting the elements of the input data that contribute to particular predictions.

Despite the already interesting results, there are also many open possibilities for future work. Although other previous studies have advanced methods for ICD coding of death certificates, their results are not directly comparable ours, given the focus on different languages and different formulations of the task. Some of these studies considered a single textual field as input, and the prediction tasks also differed in the number of classes and/or in accepting multiple codes as output. To comparatively assess our approach, a possible experiment would involve testing an adapted version of our neural architecture over the French and English datasets from the CLEF eHealth shared task [12].

Our model leverages GRUs to encode sequences, but other types of recurrent nodes have also been recently been proposed. For instance, the Minimal Gated Unit approach [23, 24] relies on a simplified model with just a single gate. Having less parameters to train can contribute to improving the model effectiveness. In contrast, Multi-Function Recurrent Units (Mu-FuRUs) adopt an elaborate gating mechanism that allows for additional differentiable functions as composition operations, leading to models that can better capture the nuances involved in encoding sequences [25]. Other alternatives include Long Short-Term Memory (LSTM) networks with coupled gates [26], Structurally Constrained Recurrent Networks [27], IRNNs [28], and many other LSTM or GRU variants [26, 29].

References

[1] Cátia Sousa Pinto, Robert N. Anderson, Cristiano Marques, Cristiana Maia, Henrique Martins, and Maria do Carmo Borralho. Improving the mortality information system in Portugal. *Eurohealth*, 22(2), 2016.

[2] Hercules Dalianis. Clinical text retrieval-an overview of basic building blocks and applications. *Professional Search in the Modern World*, 8830:147–165, 2014.

[3] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of the Workshop on Synthax, Semantics and Structure in Statistical Translation*, 2014.

[4] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.

[5] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2015.

[7] John P Pestian, Christopher Brew, Paweł Matykiewicz, Dj J Hovermale, Neil Johnson, K Bretonnel Cohen, and Włodzisław Duch. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on Biological, Translational, and Clinical Language Processing*, 2007.

[8] 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), 2013.

[9] Yan Yan, Glenn Fung, Jennifer G. Dy, and Romer Rosales. Medical coding classification by leveraging inter-code relationships. In *Proceedings of the ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, 2010.

[10] Sen Wang, Xiaojun Chang, Xue Li, Guodong Long, Lina Yao, and Quan Z Sheng. Diagnosis code assignment using sparsity-based disease correlation embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2016.

[11] Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. Automatic ICD-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics*, 84(11), 2015.

[12] Thomas Lavergne, Aurélie Névéol, Aude Robert, Cyril Grouin, Grégoire Rey, and Pierre Zweigenbaum. A dataset for ICD-10 coding of death certificates: Creation and usage. In *Proceedings of the Workshop on Building and Evaluating Resources for Biomedical Text Mining*, 2016.

[13] Pierre Zweigenbaum and Thomas Lavergne. Hybrid methods for ICD-10 coding of death certificates. In *Proceedings of International Workshop on Health Text Mining and Information Analysis*, 2016.

[14] Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 2016.

[15] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Neurocomputing: foundations of research*, 1988.

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*, 2015.

[17] Jinseok Nam, Jungi Kim, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification - revisiting neural networks. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2017.

[18] Gakuto Kurata, Bing Xiang, and Bowen Zhou. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.

[19] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of the International Conference on Very Large Data Bases*, 1994.

[20] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 1999.

[21] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10), 2007.

[22] William E Winkler. The state of record linkage and current research problems. Technical report, RR99/04. 1999.

[23] Guo-Bing Zhou, Jianxin Wu, Chen-Lin Zhang, and Zhi-Hua Zhou. Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing*, 13(3), 2016.

[24] Joel Heck and Fathi M. Salem. Simplified minimal gated unit variations for recurrent neural networks. *CoRR*, abs/1701.03452, 2017.

[25] Dirk Weissenborn and Tim Rocktäschel. MuFuRU: The multi-function recurrent unit. In *Proceedings of the Association for Computational Linguistics Workshop on Representation Learning for Natural Language Processing*, 2016.

[26] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 99:1–11, 2016.

[27] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michaël Mathieu, and Marc'Aurelio Ranzato. Learning longer memory in recurrent neural networks. *CoRR*, abs/1412.7753, 2014.

[28] Quoc V. Le, Navdeep Jaitly, and Geoffrey E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *CoRR*, abs/1504.00941, 2015.

[29] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015.