applied sciences

Article

A Comparison of Deep Learning Methods for ICD Coding of Clinical Records

Elias Moons [1,*], Aditya Khanna [1,2], Abbas Akkasi [1] and Marie-Francine Moens [1]

1 Department of Computer Science, KU Leuven, 3000 Leuven, Belgium; adityakhanna@iitb.ac.in (A.K.); abbas.akkasi@kuleuven.be (A.A.); sien.moens@cs.kuleuven.be (M.-F.M.)
2 Department of Electrical Engineering, IIT Bombay, Maharashtra 400076, India
* Correspondence: elias.moons@cs.kuleuven.be

Received: 29 June 2020; Accepted: 28 July 2020; Published: 30 July 2020

Abstract: In this survey, we discuss the task of automatically classifying medical documents into the taxonomy of the International Classification of Diseases (ICD), by the use of deep neural networks. The literature in this domain covers different techniques. We will assess and compare the performance of those techniques in various settings and investigate which combination leverages the best results. Furthermore, we introduce an hierarchical component that exploits the knowledge of the ICD taxonomy. All methods and their combinations are evaluated on two publicly available datasets that represent ICD-9 and ICD-10 coding, respectively. The evaluation leads to a discussion of the advantages and disadvantages of the models.

Keywords: ICD coding; hierarchical classification; electronic healthcare

1. Introduction

The International Classification of Diseases (ICD), which is endorsed by the World Health Organization, is the diagnostic classification standard for clinical and research purposes in the medical field. ICD defines the universe of diseases, disorders, injuries, and other related health conditions, listed in a comprehensive, hierarchical fashion. ICD coding allows for easy storage, retrieval, and analysis of health information for evidenced-based decision-making; sharing and comparing health information between hospitals, regions, settings, and countries; and data comparisons in the same location across different time periods (https://www.who.int/classifications/icd/en/) . ICD has been revised periodically to incorporate changes in the medical field. Today, there have been 11 revisions of the ICD taxonomy, where ICD-9 and ICD-10 are the most studied when it comes to their automated assignment to medical documents. In this paper, we compare state-of-the-art neural network approaches to classification of medical reports written in natural language (in this case English) according to ICD categories.

ICD coding of medical reports has been a research topic for many years [1]. Hospitals need to label their patient visits with ICD codes to be in accordance with the law and to gain subsidies from the government or refunds from insurance companies. When the documents are in free text format, this process is still done manually. Automating (a part of) this process would greatly reduce the administrative work.

In this paper, we compare the performance of several deep learning based approaches for ICD-9 and ICD-10 coding. The codes of ICD-9 consist of, at most, five numbers. The first three numbers represent a high level disease category, a fourth number narrows this down to specific diseases, and a fifth number differentiates between specific disease variants. This leads to a hierarchical taxonomy with four layers underneath a root node. The first layer ($L1$) consists of groups of 3-numbered categories, the next three layers ($L2$ through $L4$) correspond to the first 3, 4, or 5 numbers of the ICD code as is displayed in the upper part of Figure 1. In the lower part of this figure, a concrete example of the coding is shown.

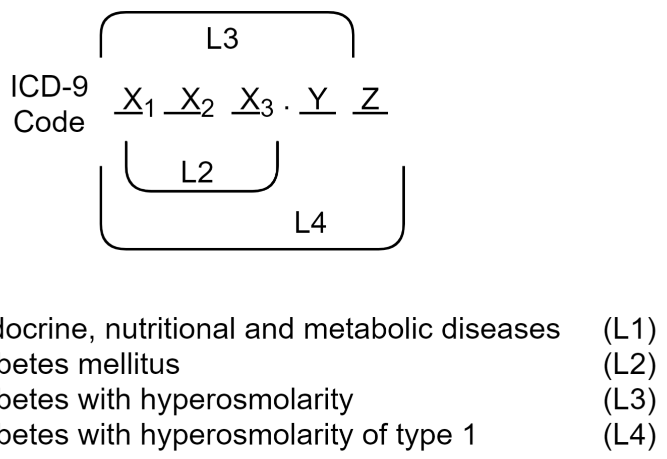
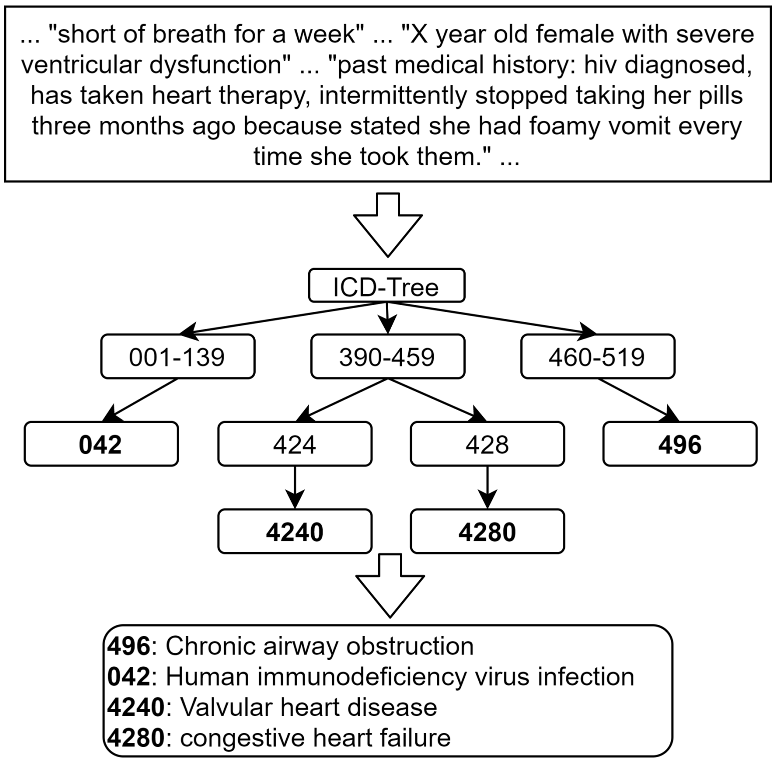

L3
ICD-9 Code
X_1 X_2 X_3 . Y Z
L2
L4
240-279: Endocrine, nutritional and metabolic diseases (L1)
250: Diabetes mellitus (L2)
250.2: Diabetes with hyperosmolarity (L3)
250.21: Diabetes with hyperosmolarity of type 1 (L4)

Figure 1. ICD-9 code structure with second through fourth layer representations and diabetes as an example.

In this paper, we survey state-of-the-art deep learning approaches for ICD-9 coding. We especially focus on the representation learning that the methods accomplish.

Experiments with ICD-9 are carried out on the MIMIC-III dataset [2]. This dataset consists of over 50,000 discharge summaries of patient visits in US hospitals. These summaries are in free textual format and labeled with corresponding ICD-9 codes, an example snippet is visible in Figure 2. Most discharge summaries are labeled with multiple categories, leading to a multiclass and multilabel setting for category prediction.


... "short of breath for a week" ... "X year old female with severe ventricular dysfunction" ... "past medical history: hiv diagnosed, has taken heart therapy, intermittently stopped taking her pills three months ago because stated she had foamy vomit every time she took them." ...

496: Chronic airway obstruction
042: Human immunodeficiency virus infection
4240: Valvular heart disease
4280: congestive heart failure

Figure 2. Example snippet of discharge summary from the MIMIC-III dataset with corresponding target International Classification of Diseases (ICD) codes.

Codes from the ICD-10 version are very similar to those of ICD-9. The main difference is that they consist of up to seven characters of which at least the first three are always present, the latter four are optional. The first character is an uppercase alphabetic letter, all other characters are numeric. The first three characters indicate the category of the diagnoses, and the following three characters indicate the etiology, anatomic site, severity, or other clinical details. A seventh character indicates an extension. An example of the ICD-10 structure is visible in Figure 3, it visualizes the same diagnosis as in Figure 1 but for ICD-10 instead of ICD-9.

Experiments with ICD-10 are conducted on the CodiEsp dataset, which is publicly available. This dataset consists of 1000 discharge summaries of patient visits in Spain. The documents are in free text format, which is automatically translated to English from Spanish, and they are manually labeled with ICD-10 codes by healthcare professionals.

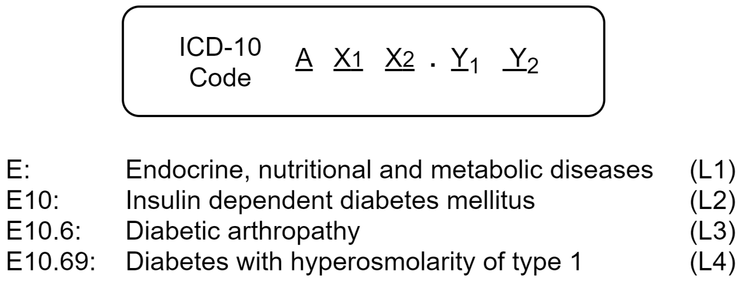

ICD-10 Code
A X1 X2 . Y1 Y2
E: Endocrine, nutritional and metabolic diseases (L1)
E10: Insulin dependent diabetes mellitus (L2)
E10.6: Diabetic arthropathy (L3)
E10.69: Diabetes with hyperosmolarity of type 1 (L4)

Figure 3. ICD-10 code structure with second through fourth layer representations and diabetes as an example.

The deep learning methods that we discuss in this paper encompass different neural network architectures including convolutional and recurrent neural networks. It is studied how they can be extended with suitable attention mechanisms and loss functions and how the hierarchical structure of the ICD taxonomy can be exploited. ICD-10 coding is especially challenging, as in the benchmark dataset that we use for our experiments the ICD coding model has to deal with very few manually labeled training data.

In our work we want to answer the following research questions. What are the current state-of-the-art neural network approaches for classifying discharge summaries? How do they compare to each other in terms of performance? What combination of techniques gives the best results on a public dataset? We hypothesize the following claims. (1) A combination of self-attention and convolutional layers yields the best classification results. (2) In a setting with less training samples per category, attention on description vectors of the target categories improves the results. (3) Using the hierarchical taxonomy explicitly in the model improves classification on a small dataset. The most important contribution of our work is an extensive evaluation and comparison of state-of-the-art deep learning models for ICD-9 and ICD-10 coding which currently does not exist in the literature.

The remainder of this paper is organized as follows. In Section 2, related work relevant for the conducted research will be discussed. Section 3 will elaborate on the datasets used in the experiments and how this data is preprocessed. The compared deep learning methods are described in Section 4. These methods are evaluated on the datasets in different settings and all findings are reported in Section 5. The most important findings will be discussed in Section 6. Finally, we conclude with some recommendations for future research.

2. Related Work

The most prominent and more recent advancements in categorizing medical reports with standard codes will be described in this section.

2.1. Traditional Models for ICD Coding

Larkey and Croft [3] are the first to apply machine learning techniques to ICD coding. Different techniques including a *k*-nearest neighbor classifier, relevance feedback, and a Bayesian classifier are applied to the texts of inpatient discharge summaries. The authors found that an ensemble of models yields the best results. At that time, and still later, one has experimented with rule-based pattern matching techniques, which are often expressed as regular expressions (see, e.g., in [4]). Farkas et al. [5] have proposed a hybrid system that partially relied on handcrafted rules and partially on machine learning. For the latter, the authors compare a decision tree learner with a multinomial logistic regression algorithm. The system is evaluated on the data from the CMC Challenge on Classifying Clinical Free Text Using Natural Language Processing, support vector machines (SVMs) were also a popular approach for assigning codes to clinical free text (see, e.g., in [6] who evaluate a SVM using *n*-gram word features on the MIMIC-II dataset). A systematic overview of earlier systems for automated clinical coding is found in [7]. The authors of [8] show that datasets of different sizes and different numbers of distinct codes demand different training mechanisms. For small datasets, it is important to select relevant features. The authors have evaluated ICD coding performance on a dataset consisting of more than 70,000 textual Electronic Medical Records (EMRs) from the University of Kentucky (UKY) Medical Center tagged with ICD-9 codes. Integrating feature selection on both structured and unstructured data is researched by the authors of [9] and has proven to aid the classification process. Two approaches are evaluated in this setting: early and late integration of structured and unstructured data, the latter yielding the better results. Documents are tagged with ICD-9 and ICD-10 medical codes.

2.2. Deep Learning Models for ICD Coding

More recently, and following a general trend in text classification, deep learning techniques have become popular for ICD coding. These methods learn relevant features from the raw data and thus skip the feature engineering step of traditional machine learning methods. Deep learning proved its value in computer vision tasks [10], and rapidly has conquered the field of text and language processing. Deep learning techniques also have been successfully applied to Electronic Health Records (EHR) [11]. In the 2019 CLEF eHealth evaluation lab, deep learning techniques had become mainstream models for ICD coding [12].

A deep learning model that encompasses an attention mechanism is tested by the authors of [13] on the MIMIC-III dataset. In this work, a Long Short-Term Memory network (LSTM) is used for both character and word level representations. A soft attention layer here helps in making predictions for the top 50 most frequent ICD codes in the dataset. Duarte et al. [14] propose bidirectional GRUs for ICD-10 coding of the free text of death certificates and associated autopsy reports. Xie et al. [15] have developed a tree-of-sequences LSTM architecture with an attention mechanism to simultaneously capture the hierarchical relationships among codes. The model is tested on the MIMIC-III dataset. Huang et al. [16] have shown that deep learning-based methods outperform other conventional machine learning methods such as a SVM for predicting the top 10 ICD-9 codes on the MIMIC-III dataset, a finding confirmed by Li et al. [17], who have confirmed that ICD-9 coding on the MIMIC-II and MIMIC-III datasets outperforms a classical hierarchy-based SVM and a flat SVM. This latter work also shows that convolutional neural networks (CNNs) are successful in text classification given their capability to learn global features that abstract larger stretches of content in the documents. Xu et al. [18] have implemented modality-specific machine learning models including unstructured text, semistructured text, and structured tabular data, and then have used an ensemble method to integrate all modality-specific models to generate the ICD codes. Unstructured and semistructured text is handled by a deep neural network, while tabular data are converted to binary features which are input as features in a decision tree learning algorithm [19]. The text classification problem can also be modeled as a joint label-word embedding problem [20]. An attention framework is proposed that measures the compatibility of embeddings between text sequences and labels. This technique is evaluated on both the MIMIC-II and MIMIC-III datasets but achieves inferior results to the neural models presented further in this paper. Zeng et al. [21] transfer MeSH domain knowledge to improve automatic ICD-9 coding but improvements compared to baselines are limited. Baumel et al. [22] have introduced the Hierarchical Attention bidirectional Gated Recurrent Unit model (HA-GRU). By identifying relevant sentences for each label, documents are tagged with corresponding ICD codes. Results are reported both on the MIMIC-II and MIMIC-III datasets. Mullenbach et al. [23] present the Convolutional Attention for Multilabel classification (CAML) model that combines the strengths of convolutional networks and attention mechanisms. They propose adding regularization on the long descriptions of the target ICD codes, especially to improve classification results on less represented categories in the dataset. This approach is further extended with the idea of multiple convolutional channels in [24] with max pooling across all channels. The authors also shift the attention from the last prediction layer, as in [23], to the attention layer. Mullenbach et al. [23,24] achieve state-of-the art results for ICD-9 coding on the MIMIC-III dataset. As an addition to these models, in this paper a hierarchical variant of each of them is constructed and evaluated.

Recently, language models have become popular in natural language processing. The use of Bidirectional Encoder Representations from Transformers (BERT) models, which uses a transformer architecture with multi-head attention, and especially BioBERT has improved the overall recall values at the expense of precision compared to CNN and LSTM models when applied in the ICD-10 coding task at CLEF eHealth in 2019 [25], a finding which we have confirmed in our experiments. Therefore, we do not report on experiments with this architecture in this survey.

Finally, Campbell et al. [26] survey the literature on the benefits, limitations, implementation, and impact of computer-assisted clinical coding on clinical coding professionals. They conclude that human coders could be greatly helped by current technologies and are likely to become clinical coding editors in an effort to raise the quality of the overall clinical coding process. Shickel et al. [11] review deep learning models for EHR systems by examining architectures, technical aspects, and clinical applications. Their paper discusses shortcomings of the current techniques and future research directions among which the authors cite ICD coding of free clinical text as one of the future challenges.

2.3. Hierarchical Models for Classification

In this paper, we foresee several mechanisms to exploit the hierarchical taxonomy of ICD codes in a deep learning setting, in other words we exploit the known dependencies between classes. Although this is a rather novel topic, hierarchical relationships between classes have been studied in traditional machine learning models. Deschacht et al. [27] have modeled first-order hierarchical dependencies between classes as features in a conditional random field and applied this model to text classification. Babbar et al. [28] study error generalization bounds of multiclass, hierarchical classifiers using the DMOZ hierarchy and the International Patent Classification by simplifying the taxonomy and selectively pruning some of its nodes with the help of a meta-classifier. The features retained in this meta-classifier are derived from the error generalization bounds. Furthermore, hierarchical loss functions have been used in non-deep learning approaches. Gopal et al. [29] exploit the hierarchical or graphical dependencies among class labels in large-margin classifiers, such as a SVM, and in logistic regression classifiers by adding a suitable regularization term to their hinge-loss and logistic loss function, respectively. This regularization enforces the parameters of a child classifier to be similar to the parameters of its parent using a Euclidean distance function, in other words, encouraging parameters which are nearby in the hierarchy to be similar to each other. This helps classes to leverage information from nearby classes while estimating model parameters. Cai and Hofmann [30] integrate knowledge of the class hierarchy into a structured SVM. Their method also considers the parent–child relationship as a feature. All parameters are learned jointly by optimizing a common objective function corresponding to a regularized upper bound on the empirical loss. During training it is enforced that the score of a training example with a correct labeling should be larger than or equal to the score of a training example of an incorrect labeling plus some loss or cost. It is assumed that assignment of confusing classes that are "nearby" in the taxonomy is less costly or severe than predicting a class that is "far away" from the correct class. This is realized by scaling the penalties for margin violation. A similar idea is modeled in a deep learning model for audio event detection [31]. These authors propose the hierarchy-aware loss function modeled as a triplet or quadruplet loss function that favors confusing classes that are close in the taxonomy, over ones that are far away from the correct class. In [32], an hierarchical SVM is shown to outperform that of a flat SVM. Results are reported on the MIMIC-II dataset. In a deep neural network setting, recent publications on hierarchical text classification outside the medical field make use of label distribution learning [18], an hierarchical softmax activation function [33], and hierarchical multilabel classification networks [34].

Recent research shows the value of hierarchical dependencies using hierarchical attention mechanisms [22] and hierarchical penalties [34], which are also integrated in the training of the models surveyed in this paper.

If the target output space of categories follows a hierarchy of labels—as is also the case in ICD coding—the trained models efficiently use this hierarchy for category assignment or prediction [32,35,36]. During categorization the models apply a top-down or a bottom-up approach at the classification stage. In a top-down approach parent, categories are assigned first and only children of assigned parents are considered as category candidates. In a bottom-up approach, only leaf nodes in the hierarchy are assigned which entail that parent nodes are assigned.

In the context of category occurrences in hierarchical target spaces, a power-law distribution is described in [37]. Later, the authors of [38] have addressed this phenomenon quantitatively deriving a relationship in terms of space complexity for those kind of distributions. They have proved that hierarchical classifiers have lower space complexity than their flat variants if the hierarchical target space satisfies certain conditions based on, e.g., maximum branching factor and the depth of the hierarchy. The hierarchical variants discussed in this survey are of different shape than those discussed in these works, layer-based instead of node-based, and do not suffice the necessary conditions for these relationships to apply.

2.4. Models Relevant for This Survey

The experiments reported in Section 5 are carried out starting with and expanding the models described in [23,24]. These models are evaluated against common baselines, partly inspired by other models e.g., the GRU form [22]. For all models, the state-of-the-art, and the baselines, a hierarchical version is constructed using the principles explained in [34]. This hierarchical version duplicates the original model for each layer in the corresponding ICD taxonomy (ICD-9 or ICD-10). These are then trained in parallel. Furthermore, the weights in these networks are influenced by the weights of neighboring layers via the addition of a hierarchical loss function. This loss function penalizes hierarchical inconsistencies that arise when training the model. This leads to a clear comparison between all tested models among themselves as well as with their hierarchical variants.

3. Materials

3.1. ICD-9 Datasets

The publicly available MIMIC-III dataset [2] is used for ICD-9 code predictions. MIMIC-III is an openly accessible clinical care database. For this research, following the trends of previous related work, the patient stay records from the database are used. Every hospital admission has a corresponding unique HADM-ID. In the MIMIC-III database, some patients have also an added Addendum to their stay. Based on earlier studies, records of only those patients who have discharge summaries linked are selected. The addendum is concatenated to the patient's discharge summary. Analogous to the work in [24], out of the the original database, three sub-datasets are extracted. These datasets are used for the experiments and allow for evaluation in different settings. The sub-datasets are the following.

- **Dis-50** consists of a selection of the discharge summaries from the MIMIC-III dataset (11,369 out of 52,726) for the classification of Top-50 ICD-9 codes. We use the publicly available split [23] for training (8066), testing (1729), and development (1574) of the models.
- **Dis** describes the full label setting where all Diagnostic (6918) and Procedural (2011) ICD-9 codes are used. This leads to a total of 8929 Unique codes on the 52,726 discharge summaries. We again use the publicly available split for training (47,723), testing (3372), and development (1631) of the models.
- **Full** extends the **Dis** dataset with other notes regarding the patient (radiology notes, nursing notes, etc.) in addition to the discharge summaries. This dataset, contains almost thrice the number of tokens for training. We use the same test, train, development split as used in the **Dis** dataset.

3.2. ICD-10 Dataset

The **CodiEsp** corpus [39] consists of 1000 clinical cases, tagged with various ICD-10 codes by health specialists. This dataset is released in the context of the CodiEsp track for CLEF ehealth 2020. The dataset corresponding to the subtask of classifying diagnostic ICD codes is used. The original text fragments are in Spanish but an automatically translated version in English is also provided by the organizers, this version is used in this research. The publicly available dataset contains a split of 500 training samples, 250 development samples, and 250 test samples. In total, the 1000 documents comprises of 16,504 sentences and 396,988 words, with an average of 396.2 words per clinical case. The biggest hurdle while training with this dataset is the size and consequently the small number of training examples for each category present. Figure 4 gives a sorted view of all categories present in the training dataset and the amount of examples tagged with that specific category.

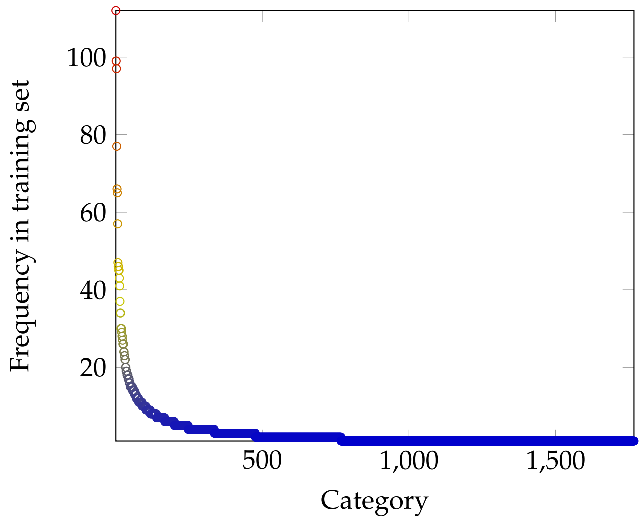


Figure 4. Category frequencies of CodiEsp training dataset.

There are in total 1767 different categories spread out over only 500 training documents. Every document is labeled with on average 11.3 different categories and each category is on average represented by 3.2 training examples. Even the top 50 most frequently occurring categories have only between 15 and 112 corresponding positive training documents. Therefore, tests for this dataset are conducted on these 50 categories. Table 1 gives an overview of statistics for all discussed training datasets. The specifics for the corresponding development and test sets are similar. Displayed statistics for the Dis and the Full dataset are the same since the only difference lies in larger text fragments, resulting in 72,891 unique tokens for the Full dataset compared to 51,917 for Dis. There are no differences concerning the labels.

Table 1. Dataset specifics overview.

Dataset	#Training Docs	#Labels	Avg. #Labels/Doc	Avg. #Training Docs/Label
Dis-50	8067	50	5.7	920
Dis	47,724	8922	15.9	85
Full	47,724	8922	15.9	85
CodiEsp	500	1767	11.3	3.2

3.3. Preprocessing

The preprocessing follows the standard procedure described in [23], i.e., tokens that contain no alphabetic characters are removed and all tokens are put to lowercase. Furthermore tokens that appear in fewer than three training documents are replaced with the "UNK" token. All documents are then truncated to a maximum length of 2500 tokens.

4. Methods

In this section, all tested models will be discussed in detail. First, a simple convolutional and a recurrent baseline commonly used in text classification are described. Then, two recent state-of-the-art models in the field of ICD coding are explained in detail. These models are implemented by the authors following the original papers and are called **DR-CAML** [23] and **MVC-(R)LDA** [24], respectively. We discuss in detail the attention mechanisms and loss functions of these models. Afterwards, as a way of handling the hierarchical dependencies of the ICD-codes, we propose various ways of their integration in all models. This is based on advancements in hierarchical classification as inspired by [34].

All discussed models have for each document i as input a sequence of word vectors x^i as their representation and as output a set of ICD-codes y^i.

4.1. Baselines

The performance of all models will be evaluated and compared against two simple common baselines used for handling sequential input data (text). These models are, respectively, based on convolutional and recurrent neural principles.

4.1.1. Convolutional

The baseline convolutional neural network model, or **CNN**, consists of a 1D temporal convolutional neural layer. This convolutional layer consists of different kernels, which are filters with a specific pattern that are tested against all sequences of the input data with the same length. This is followed by a (max-)pooling layer, to reduce the data size by only remembering the maximum value over a certain range. More formally, for an input x and a given 1D kernel f on element s of the input sequence, the convolutional and pooling operation can be defined as follows.

$$F_1(s) = (x \circledast f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-i} \quad (1)$$

$$F_2(s) = max_{i=0}^{l-1} F_1(s-1) \quad (2)$$

The amount of filters k and l in the convolutional and pooling layer, respectively, as well as their sizes are optimizable parameters of this model. For both layers a stride length, i.e., the amount by which the filter shifts in the sequence, can be defined leading to a trade-off between output size and observability of detailed features.

4.1.2. Recurrent

As the recurrent neural network baseline, two common approaches are considered.

BiGRU

The **GRU**, or Gated Recurrent Unit, is a gating mechanism in recurrent neural networks. It is the mechanism of recurrent neural networks allowing the model to "learn to forget" less important fragments of the data and "learn to remember" the more important fragments with respect to the learning task. More formally, consider an input vector x_t, update gate vector z_t, reset gate vector r_t, and output vector h_t at time t. The respective values can be calculated as follows.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{3}$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{4}$$
$$h_t^* = tanh(W_h \cdot [t_t \times h_{t-1}, x_t] + b_h) \tag{5}$$
$$h_t = (1 - z_t) \times h_{t-1} + z_t \times h_t^* \tag{6}$$

This leads to weight matrices W_z, W_r, and W_h to train as well as biases b_z, b_r, and b_h, σ stands for the sigmoid activation function. **BiGRU** is the bidirectional variant of such a model that processes the input data front-to-back and back-to-front in parallel.

BiLSTM

An **LSTM**, or Long Short-Term Memory neural network model, is very similar to a GRU but replaces the update gate with a forget gate and an additional output gate. This way it usually has more computational power than a regular GRU, but at the expense of more trainable parameters and more chance of overfitting when the amount of training data is limited [40–42]. Formally, consider again an input vector x_t and a hidden state vector h_t at time t. Activation vectors for the update gate, forget gate, and output gate are, respectively, represented by z_t, f_t, and o_t. These states relate to each other like follows.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{7}$$
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{8}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$
$$c_t^* = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{10}$$
$$c_t = (f_t) \times c_{t-1} + z_t \times c_t^* \tag{11}$$
$$h_t = o_t \times tanh(c_t) \tag{12}$$

This again leads to weight matrices W_z, W_f, W_o, and W_c to train as well as biases b_z, b_f, b_o, and b_c with σ being the sigmoid activation function. **BiLSTM** is the bidirectional variant of a regular LSTM, analogous to BiGRU, which is the bidirectional variant of GRU.

4.2. Advanced Models

This subsection describes the details of recent state-of-the-art models presented in [23,24] in the way they are used for the experiments in Section 5.

4.2.1. DR-CAML

DR-CAML is a CNN-based model adopted for ICD coding [23]. When an ICD code is defined by the WHO, it is accompanied by a label definition expressed in natural language to guide the model towards learning the appropriate parameter values of the model. For this purpose, the model employs a per-label attention mechanism enabling it to learn distinct document representations for each label. It has been shown that for labels for which there are very few training instances available, this approach is advantageous. The idea is that the description of a target code is itself a very good training example for the corresponding code. Similarity between the representation of a given test sample and the representation of the description of a target code gives extra confidence in assigning this label.

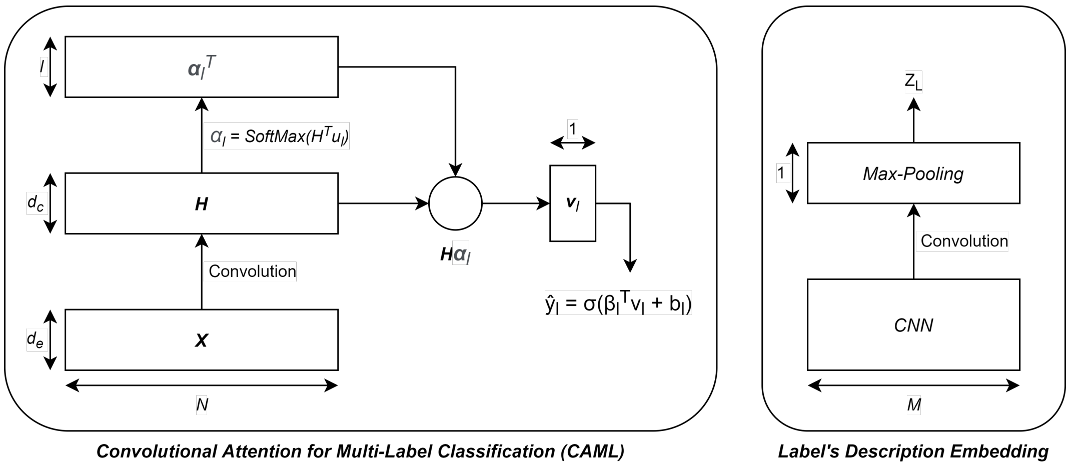
In general, after the convolutional layer, DR-CAML employs a per-label attention mechanism to attend to the relevant parts of text for each predicted label. An additional advantage is that the per-label attention mechanism provides the model with the ability of explaining why it decided to assign each code by showing the spans of text relevant for the ICD code.

DR-CAML consists of two modules: one for the representation of the input text, and the other for the embedding of the label's description as is visualized in Figure 5. The CAML module has a CNN at the base layer which takes a sequence of the embeddings of the text tokens as input and consequently represents the document as the matrix H. Then, the per-label attention mechanism applies. Attention in this context means learning which parts of some context (the label description vectors) are relevant for a given input vector.

After calculating the attention vector α using a softmax activation function, it is applied as a product with H. With h_l, the vector parameter for label l, the vector representation for each label is computed as

$$v_l = \sum_{n=1}^{N} \alpha_{l,n} h_n \tag{13}$$

Given the vector representation of a document and the probability for label l, $\hat{y}_l$ can be obtained as shown in Figure 5.



α_l^T
l
$\alpha_l = SoftMax(H^T u_l)$
d_c
H
Convolution
d_e
X
N
$H\alpha_l$
1
v_l
$\hat{y}_l = \sigma(\beta_l^T v_l + b_l)$
Convolutional Attention for Multi-Label Classification (CAML)
Z_L
1
Max-Pooling
Convolution
CNN
M
Label's Description Embedding

Figure 5. DR-CAML (after [23]).

The CNN modules on the left hand side try to minimize the binary cross entropy loss. The second module is a max-pooling CNN model which produces a max-pooled vector, z_l, by getting the description of code l. Assuming n_y is the number of true labels in train data, the final loss is computed by adding a regularization term to the base loss function. The loss function is explained in more detail in Section 4.2.3.

4.2.2. MVC-(R)LDA

MVC-LDA and **MVC-RLDA** can be seen as extensions of DR-CAML. Similar to that model, they are based on a CNN architecture with a label attention mechanism that considers ICD coding as a multi-task binary classification problem. The added functionality lies in the use of parallel CNNs with different kernel sizes to capture information of different granularity. MVC-LDA, the top module in Figure 6, is a multi-view CNN model stacked on an embedding layer. MVC-RLDA reintroduces the per-label attention mechanism introduced in the previous subsection.

In general, the multi-view CNNs are constructed with four CNNs that have the same number of filters but with different kernel sizes. This convolutional layer is followed by a max-pooling function across all channels to select the most relevant span of text for each filter. A separate attention layer for each label comes next, helping the model to attend to relevant parts of a document for each label. A linear layer with weight vector V_j is implemented for the j^{th} label and CV_j is the attention for the input C and label j. This attention vector CV_j is the output of a dense layer with softmax activation function leading to a relative weighting of the input elements C.

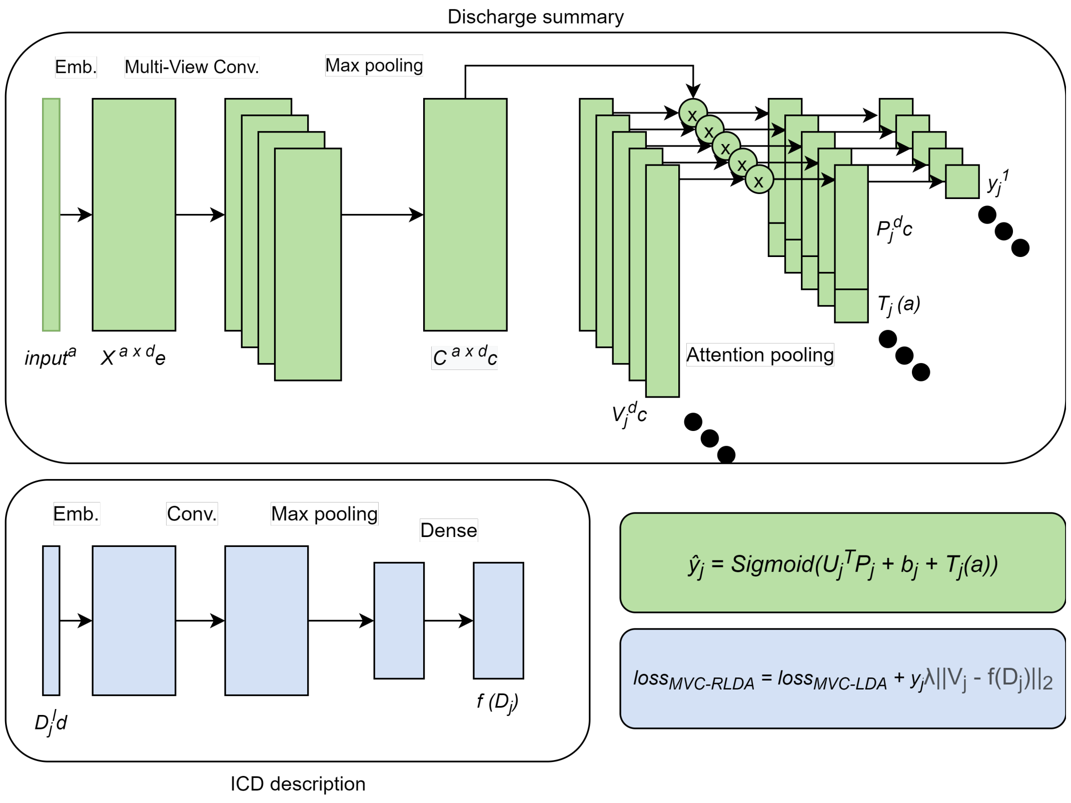


Figure 6. MVC-(R)LDA (after the work in [24]).

Then, the pooled outputs of the attention layer are computed as

$$P_j = C^T(CV_j). \tag{14}$$

At the end, a dense layer is used for each label. The length of an input document is also encoded into the output layers with embedding function

$$T_j(l) = Sigmoid(K_j l + d_j), \tag{15}$$

to decrease the problem of under-coding to some extent. This is done as in [24], which showed a statistically significant Pearson's correlation between the input length and the number of ground truth codes. Therefore, the model can derive an underlying bias that, on average, shorter input documents represent a lower amount of categories.

Parameters a, K_j, and d_j in the length embedding function respectively represent the input length, and the layer's weight and bias, respectively, for a given label j. The prediction y_j for class j is then computed as

$$y_j = Sigmoid(U_j^T P_j + b_j + T_j(a)). \tag{16}$$

Similar to DR-CAML, this model tries to minimize the binary loss function. Adding the label description embedding to MVC-LDA, the lower part of Figure 4 leads to MVC-RLDA whose loss function includes an extra weighted term as a regularizer. It guides the attention weights to avoid overfitting. In addition, this regularization forces the attention for classes with similar descriptions to be closer to each other. The loss function is again explained in more detail in Section 4.2.3.

4.2.3. Loss Function

The loss functions used to train DR-CAML and the multiview models MVD-(R)LDA are calculated in the same way. The general loss function is the binary cross entropy loss $loss_{BCE}$. This loss is extended by regularization on the long description vectors of the target categories, visualized in Figure 6 on the lower right corner.

Given N different training examples x^i. The values of $\hat{y}_l$ and max-pooled vector z_l can be calculated as represented in Figure 5 by getting the description of code l out of all L target codes. In this figure, and the following formulas, β_l is a vector of prediction weights and v_l the vector representation for code l. Assuming n_y is the number of true labels in the training data, the final loss is computed by adding regularization to the base loss function as

$$\hat{y}_l = \sigma(\beta_l^t v_l + b_l) \tag{17}$$

$$loss_{BCE}(X) = -\sum_{i=1}^{N}\sum_{l=1}^{L} y_l \log(\hat{y}_l) + (1 - y_l)\log(1 - \hat{y}_l) \tag{18}$$

$$loss_{Model}(X) = loss_{BCE} + \lambda \frac{1}{n_y} \sum_{i=1}^{N}\sum_{l=1}^{L} \|z_l - \beta_l\|_2 \tag{19}$$

4.3. Modeling Hierarchical Dependencies

In this section, we investigate the modeling of hierarchical dependencies as extensions of the models described above. A first part integrates the hierarchical dependencies directly into the structure of the model. This leads to **Hierarchical models**, which are layered variants of the already discussed approaches. The second way hierarchical dependencies are explicitly introduced into the model is via the use of a hierarchical loss function to penalize hierarchical inconsistencies across the model's prediction layer.

4.3.1. Hierarchical Models

Hierarchical relationships can be shaped directly into the architecture of any of the described models above. The ICD-9 taxonomy can be modeled as a tree with a general ICD root and 4 levels of depth, as already described in Section 1. This leads to a hierarchical variant of any of the models. In this variant, not 1 but 4 identical models will be trained, one for each of the different layers in the ICD hierarchy (corresponding to the length of the codes).

Such an approach is presented in [34] and is adapted to the target domain of ICD categories. An overview of the approach is given in Figure 7.

The input for each layer is partially dependent on an intermediary representation from the previous layer as well as the original input through concatenation of both. Layers are stacked from most to least specific or from leaf to root node in the taxonomy. Models corresponding to different layers will then rely on different features, or characteristics, to classify the input vectors. This way the deepest, most advanced representations, can be used for classifying the most abstract and broad categories. On the other hand, for the most specific categories, word level features can directly be used to make detailed decisions between classes that are very similar.

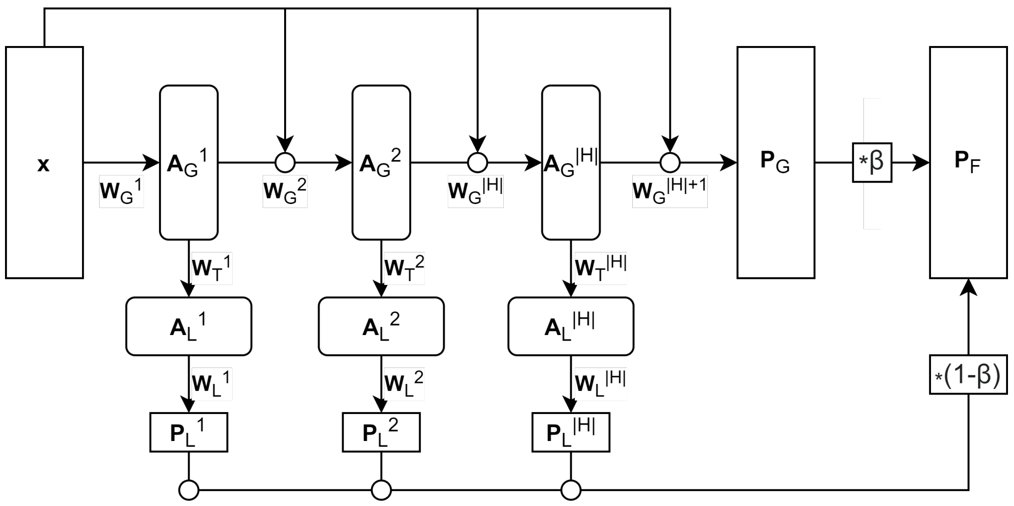


Figure 7. Overview of hierarchical variant of a model, inspired by [34].

4.3.2. Hierarchical Loss Function

To capture the hierarchical relationships in a given model, the loss function of the above models can be extended with an additional term. This leads to the definition of a **Hierarchical loss function** ($loss_H$). This loss function penalizes classifications that contradict the inherent ICD hierarchy. More specifically, when a parent category is not predicted to be true, none of its child categories should be predicted to be true. The hierarchical loss between a child and its parent in the tree is then defined as the difference between their computed probability scores, with 0 as a lower bound. More formally, for the entire loss function $loss_{H\_Model}$ for a category of layer X, combining the regular training loss $loss_{Model}$ described above and the hierarchical loss $loss_H$, is calculated as follows,

$$P(X) = Probability(X == True) \tag{20}$$

$$Par(X) = Probability(Parent(X) == True) \tag{21}$$

$$L(X) = True\,label\,of\;\; X(0\,or\;\;1) \tag{22}$$

$$loss_H(X) = Clip(P(X) - Par(X), 0, 1) \tag{23}$$

$$loss_{H\_Model}(X) = (1-\lambda)loss_{Model}(X) + \lambda loss_H(X) \tag{24}$$

which leaves a parameter λ to optimize the loss function (parameter λ is optimized over the training set).

5. Results

5.1. MIMIC-III

Results are displayed for five different models. First, results for the two baseline models, **CNN** and **BiGRU**, are shown. Because in most of the experiments, the BiGRU models performed at least on par with their BiLSTM variants, we only report the results of **BiGRU** as a recurrent neural network baseline. The reason for this good performance of GRU models compared to LSTM models most likely resides in the amount of available training data for various target categories. Then, we report on three more advanced models as discussed in the Method section: **DR-CAML**, **MVC-LDA**, and **MVD-RLDA**. Different hyperparameter values are considered and tested on the development set of MIMIC-III the setting giving the highest average performance on the development set is reported in Table 2.

Table 2. Optimal hyperparameter values obtained on the development set of the MIMIC-III dataset.

	CNN	BiGRU	DRCAML	MVC-LDA	MVC-RLDA
# of filters	500	-	50	6,8,10,12	6,8,10,12
Filter Sizes	4	-	10	70	90
λ	-	0.0005	-	-	0.0005
Lr	0.003	0.003	0.0001	0.001	0.001
Batch size	16	16	16	4	4
Seq. length~	2500	2500	2500	10000	10000

For all these models using their optimal hyperparameter settings, the average performance is reported in terms of Micro F1, Macro F1, Micro AUC (ROC), and Precision@X. For models that are only evaluated on the top 50 most frequent categories in the training data, results are displayed in Table 3. This experiment is then repeated over all categories, which leads to the results in Table 4. Last, Table 5 gives the results of training the models on all labels for the Full dataset.

Table 3. Results of flat models on top 50 most frequent categories of the MIMIC-III Dis-50 dataset.

		Micro F1	Macro F1	Micro AUC	P@5
Dis-50	**CNN**	63.42	59.74	91.57	62.33
	GRU	63.49	55.72	91.79	61.72
	DR-CAML	69.64	64.56	93.90	65.39
	MVC-LDA	69.07	64.17	93.69	65.15
	MVC-RLDA	69.53	64.85	93.77	64.91

Table 4. Results of flat models on all labels of the MIMIC-III Dis dataset.

		Micro F1 Proc	Micro F1 Diag	Micro F1 Both	Micro AUC	P@5
Dis	**CNN**	51.01	40.80	42.58	84.38	59.48
	GRU	53.88	40.86	43.40	85.36	61.59
	DR-CAML	48.99	59.03	50.47	89.45	68.28
	MVC-LDA	59.75	51.60	53.03	90.02	69.77
	MVC-RLDA	58.84	50.74	52.10	89.77	68.71

Table 5. Results of flat models on all labels of the MIMIC-III Full dataset.

		Micro F1 Proc	Micro F1 Diag	Micro F1 Both	Micro AUC	P@5
Full	**CNN**	46.19	41.19	42.18	83.96	57.53
	GRU	46.50	37.74	39.64	83.19	55.00
	DR-CAML	57.90	41.40	49.94	89.42	67.16
	MVC-LDA	58.12	50.70	51.97	89.93	68.53
	MVC-RLDA	57.61	49.67	50.97	89.68	67.60

This experiment is repeated for the hierarchical variants of all described models. This time, only results on the top 50 most frequent target categories are reported in Table 6. As hierarchical models introduce a large number of additional intermediate categories, the target space is too large to train these hierarchical variants in a full category setting.

Table 6. Results of hierarchical models obtained on the MIMIC-III Dis-50 dataset.

		Micro F1	Macro F1	Micro AUC	P@5
Dis-50	**CNN**	61.70	53.79	90.62	59.87
	GRU	62.38	54.88	91.77	60.03
	DR-CAML	67.68	63.74	93.47	63.48
	MVC-LDA	65.21	60.06	92.29	62.41
	MVC-RLDA	65.43	61.22	92.34	61.73

To assess the importance of the different components of the highest performing model on MIMIC-III Dis, an ablation study is conducted. The multi-view and the hierarchical component are added and the regularization on long descriptions of the target ICD-codes is removed while all other components stay the same. The difference in performance is measured and visualized in Figure 8.

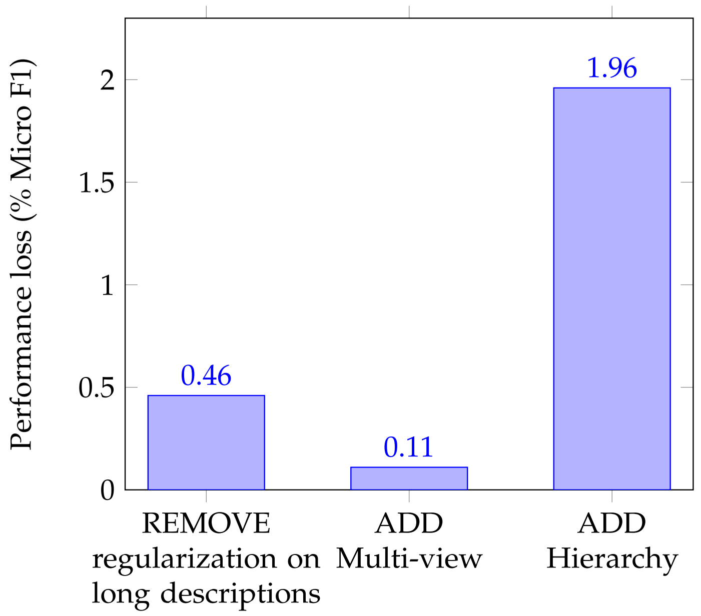


Figure 8. Ablation study on highest performing model (hierarchical MVC-RLDA) for MIMIC-III dataset.

5.2. CodiEsp

Similar experiments are carried out on the CodiEsp dataset, while only using the top 50 most frequent codes. The same hyperparameter settings are used as in Table 2. Results are visualized in Tables 7 and 8. Results for the full target space are not reported, as 90% of the target categories would only have 5 or less positive training examples. Furthermore, this would lead to a target space of 1767 different categories with only 500 training examples in total, which makes training of a decent model unfeasible.

Table 7. Results of flat models on top 50 codes obtained on the CodiEsp dataset.

		Micro F1	Macro F1	Micro AUC	P@5
CodiEsp	**CNN**	12.52	6.17	49.35	7.96
	GRU	11.54	11.03	50.54	7.68
	DR-CAML	9.58	8.24	48.63	7.96
	MVC-LDA	10.84	6.23	49.26	4.17
	MVC-RLDA	11.52	6.67	48.01	3.70

Table 8. Results of hierarchical models for top 50 codes obtained on the CodiEsp dataset.

		Micro F1	Macro F1	Micro AUC	P@5
CodiEsp	**CNN**	12.44	6.18	53.08	2.84
	GRU	11.87	11.50	50.14	7.68
	DR-CAML	10.35	3.97	53.61	5.59
	MVC-LDA	13.00	2.79	60.76	11.94
	MVC-RLDA	13.92	4.21	56.38	8.72

To assess the importance of the different components of the highest performing model on CodiEsp, an ablation study is conducted. The multi-view, the hierarchical component, and the regularization on long descriptions of the target ICD-codes are each removed while all other components stay the same. The difference in performance is measured and visualized in Figure 9.

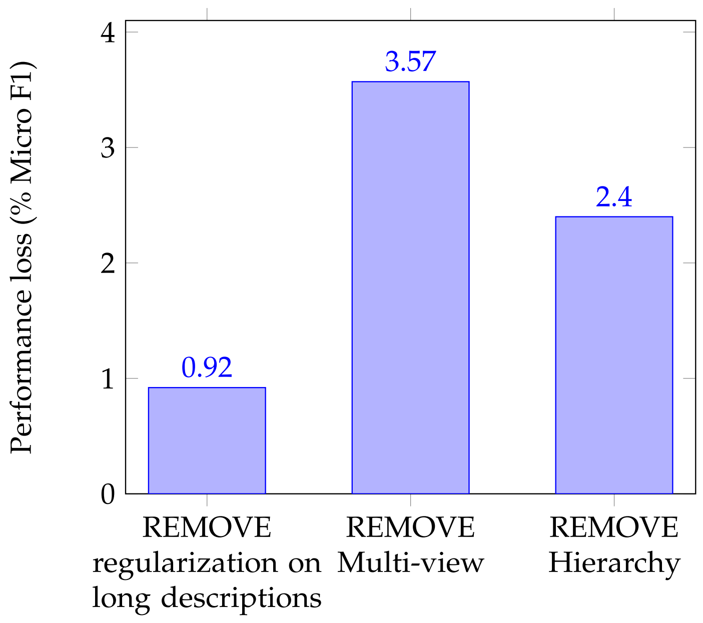


Figure 9. Ablation study on highest performing model (hierarchical MVC-RLDA) for CodiEsp dataset.

6. Discussion

A comparison between the results displayed in Tables 3 and 4 shines a light on the value of the multiview component. The micro F1 scores of the five models are in similar relationship to each other in both tables, except for the two multiview models. They outperform CAML in the full label setting of the MIMIC-III Dis dataset, where they show very similar behavior to CAML in a top-50 category setting, where for each of the categories a decent amount of training samples is available. When the target space increases and more categories have fewer training examples, the added granularity of having multiple kernel sizes in the MVC-(R)LDA model pays off. Table 5 shows results for models trained on the Full dataset. The best performing model (MVC-LDA, 58.12%) gets outperformed by the best performing model for all labels on Dis (MVD-LDA, 59, 75%). The addition of the information in other medical documents than just discharge summaries thus seems to complicate instead of facilitate the classification process.

Furthermore, comparing Tables 3 and 6, where the influence of the hierarchical parameter can be assessed in a top-50 category setting, reveals a shift in the opposite direction. While in general, the modeling of the hierarchical relationships hurts the classification process for all categories, it hinders the multiview models the most. This time, DR-CAML is clearly the best performing model. Adding multiview and simultaneously modeling the hierarchical relationships between the target categories tend to make the model overfit on the training data.

Looking at Tables 7 and 8, it is clear that the lack of a sufficient amount of training data in CodiEsp (about 100 times less than for the Dis dataset) for most categories led to lower performance of all models on this dataset. For the flat variants of the models, a regular CNN even outperforms the more complex models. As the amount of training data is low, the added complexity of the latter models hinders them generalizing well for unseen data. Comparing the results in both tables also leads to the conclusion that in contrast to the results on MIMIC-III, on average the hierarchical component increases the classification performance based on Micro F1 on CodiEsp. Where the information embedded in the ICD taxonomy is redundant and even counteracting the performance for the larger MIMIC-III dataset, it is leveraged when there is a lack of information in the training data itself, which is the case for CodiEsp.

Last, Figures 8 and 9 display the relative importance of the long description regularization, the multi-view. and the hierarchy for the top performing model on both the Dis and CodiEsp datasets. For the Dis dataset, not using the hierarchy is by far the most important component. The regularization on long descriptions still adds 0.46% and the multi-view almost does not influence the results. For CodiEsp, it shows that the multi-view component has the biggest influence, followed by the hierarchy, whose importance on this smaller dataset is already shown previously.

7. Conclusions

In this paper, we have surveyed the current methods used for classification of clinical reports based on ICD codes using neural networks. We have combined the techniques already present in the literature and assessed the relative importance of all present components. Combining a convolutional framework with self-attention as well as regularizing, the loss function with attention on the long descriptions of target ICD codes proved to be valuable. Furthermore, a hierarchical objective was integrated in all presented models. Its added value lies especially in a setting with low amounts of available training data. Last, extending the dataset with the information present in other medical documents introduced too much noise into the data, hindering the performance of the tested models.

Concerning future research directions, it would be valuable to test the techniques on a ICD-10 or ICD-11 dataset of larger size. This would give better insights into which performance these models could achieve in current hospital settings. On a similar note, tackling the problem of lack of data by finding a way to combine the available training data from different datasets (e.g., MIMIC-III and CodiEsp) and different ontologies (e.g., ICD-9, ICD-10, and MeSH) could further improve the classification performance of all models. Last, it would be interesting to investigate the use of hierarchical descriptions as an addition to the loss function, giving another use for the information inherently present in the ICD taxonomy.

Author Contributions: Conceptualization, E.M. and M.-F.M.; Methodology, E.M. and M.-F.M.; Software, E.M., A.A. and A.K.; Validation, E.M., A.A. and A.K.; Formal Analysis, E.M. and M.-F.M.; Investigation, E.M.; Resources, A.K.; Data Curation, A.K.; Writing-Original Draft Preparation, E.M. and M.-F.M.;Writing-Review Editing, E.M., A.A., A.K. and M.-F.M..; Visualization, E.M.; Supervision, M.-F.M.; Project Administration, A.A. and M.-F.M.; Funding Acquisition, M.-F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the VLAIO SBO project, grant number 150056. M.-F.M. is co-funded by the ERC Advanced Grant CALCULUS H2020-ERC-2017-ADG 788506.

Conflicts of Interest: Authors declare no conflict of interest.

References

1. Larkey, L.; Croft, W.B. *Automatic Assignment of ICD9 Codes To Discharge Summaries*; Technical Report; University of Massachusetts: Amherst, MA, USA, 1995.
2. Johnson, A.E.W.; Pollard, T.J.; Shen, L.; Lehman, L.w.H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 1–9. [CrossRef] [PubMed]
3. Larkey, L.S.; Croft, W.B. Combining classifiers in text categorization. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, 18–22 August 1996; (Special Issue of the SIGIRForum); pp. 289–297.
4. Goldstein, I.; Arzumtsyan, A.; Özlem, U. Three approaches to automatic assignment of ICD-9-CM codes to radiology reports. In *Proceedings of the AMIA Annual Symposium*; American Medical Informatics Association: Washington, DC, USA, 2007; pp. 279–283.
5. Farkas, R.; Szarvas, G. Automatic construction of rule-based ICD-9-CM coding systems. *BMC Bioinform.* **2008**, *9* (Suppl. 3), S10. [CrossRef] [PubMed]
6. Marafino, B.J.; Davies, J.M.; Bardach, N.S.; Dean, M.L.; Dudley, R.A. N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *J. Am. Med. Inform. Assoc.* **2014**, *21*, 871–875. [CrossRef] [PubMed]
7. Stanfill, M.; Williams, M.; Fenton, S.; Jenders, R.; Hersh, W. A systematic literature review of automated clinical coding and classification systems. *J. Am. Med. Inform. Assoc. JAMIA* **2010**, *17*, 646–651. [CrossRef] [PubMed]
8. Kavuluru, R.; Rios, A.; Lu, Y. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artif. Intell. Med.* **2015**, *65*, 155–166. [CrossRef]
9. Scheurwegs, E.; Cule, B.; Luyckx, K.; Luyten, L.; Daelemans, W. Selecting relevant features from the electronic health record for clinical code prediction. *J. Biomed. Inform.* **2017**, *74*, 92–103. [CrossRef]
10. Leo, M.; Furnari, A.; Medioni, G.G.; Trivedi, M.M.; Farinella, G.M. Deep Learning for Assistive Computer Vision. In Proceedings of the Computer Vision—ECCV 2018 Workshops—Part VI, Munich, Germany, 8–14 September 2018; pp. 3–14.
11. Shickel, B.; Tighe, P.; Bihorac, A.; Rashidi, P. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 1589–1604. [CrossRef]
12. Kelly, L.; Suominen, H.; Goeuriot, L.; Neves, M.; Kanoulas, E.; Li, D.; Azzopardi, L.; Spijker, R.; Zuccon, G.; Scells, H.; et al. Overview of the CLEF eHealth Evaluation Lab 2019. In *International Conference of the Cross-Language Evaluation Forum for European Languages*; Springer: Cham, Switzerland, 2019; pp. 322–339.
13. Shi, H.; Xie, P.; Hu, Z.; Zhang, M.; Xing, E.P. Towards automated ICD coding using deep learning. *arXiv* **2017**, arXiv:1711.04075.
14. Duarte, F.; Martins, B.; Pinto, C.S.; Silva, M.J. Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text. *J. Biomed. Inform.* **2018**, *80*, 64–77. [CrossRef]
15. Xie, P.; Xing, E. A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; ACL: Melbourne, Australia, 2018.
16. Huang, J.; Osorio, C.; Sy, L.W. An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes. *Comput. Methods Prog. Biomed.* **2019**, *177*, 141–153. [CrossRef]
17. Li, M.; Fei, Z.; Zeng, M.; Wu, F.; Li, Y.; Pan, Y.; Wang, J. Automated ICD-9 coding via a deep learning approach. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *16*, 1193–1202. [CrossRef] [PubMed]
18. Xu, K.; Lam, M.; Pang, J.; Gao, X.; Band, C.; Mathur, P.; Papay, F.; Khanna, A.K.; Cywinski, J.B.; Maheshwari, K.; et al. Multimodal machine learning for automated ICD coding. In *Proceedings of the 4th Machine Learning for Healthcare Conference*; Doshi-Velez, F., Fackler, J., Jung, K., Kale, D., Ranganath, R., Wallace, B., Wiens, J., Eds.; PMLR: Ann Arbor, MI, USA, 2019; Volume 106, pp. 197–215.
19. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
20. Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; Carin, L. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; ACL: Melbourne, Australia, 2018.

21. Zeng, M.; Li, M.; Fei, Z.; Yu, Y.; Pan, Y.; Wang, J. Automatic ICD-9 coding via deep transfer learning. *Neurocomputing* **2018**, *324*, 43–50. [CrossRef]
22. Baumel, T.; Nassour-Kassis, J.; Elhadad, M.; Elhadad, N. Multi-Label Classification of Patient Notes: A Case Study on ICD Code Assignment. In Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, Hilton, NO, USA, 2–7 February 2018.
23. Mullenbach, J.; Wiegreffe, S.; Duke, J.; Sun, J.; Eisenstein, J. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*; ACL: New Orleans, LA, USA, 2018.
24. Sadoughi, N.; Finley, G.P.; Fone, J.; Murali, V.; Korenevski, M.; Baryshnikov, S.; Axtmann, N.; Miller, M.; Suendermann-Oeft, D. Medical code prediction with multi-view convolution and description-regularized label-dependent attention. *arXiv* **2018**, arXiv:1811.01468.
25. Amin, S.; Neumann, G.; Dunfield, K.; Vechkaeva, A.; Chapman, K.A.; Wixted, M.K. MLT-DFKI at CLEF eHealth 2019: Multi-label classification of ICD-10 codes with BERT. In Proceedings of the Working Notes of CLEF 2019—Conference and Labs of the Evaluation, Forum, Lugano, Switzerland, 9–12 September 2019.
26. Campbell, S.; Giadresco, K. Computer-assisted clinical coding: A narrative review of the literature on its benefits, limitations, implementation and impact on clinical coding professionals. *Health Inf. Manag. J.* **2019**, *49*, 183335831985130. [CrossRef] [PubMed]
27. Deschacht, K.; Moens, M. Efficient hierarchical entity classifier using conditional random fields. In Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge@COLING/ACL 2006, Sydney, Australia, 22 July 2006; pp. 33–40.
28. Babbar, R.; Partalas, I.; Gaussier, É.; Amini, M. On flat versus hierarchical classification in large-scale taxonomies. In Proceedings of the Advances in Neural Information Processing Systems 26: Proccedings of the 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 1824–1832.
29. Gopal, S.; Yang, Y. *Recursive Regularization for Large-Scale Classification with Hierarchical and Graphical Dependencies*; Association for Computing Machinery: New York, NY, USA, 2013.
30. Cai, L.; Hofmann, T. Hierarchical document categorization with support vector machines. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004.
31. Jati, A.; Kumar, N.; Chen, R.; Georgiou, P. Hierarchy-aware loss function on a tree structured label space for audio event detection. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6–10.
32. Perotte, A.; Pivovarov, R.; Natarajan, K.; Weiskopf, N.; Wood, F.; Elhadad, N. Diagnosis code assignment: Models and evaluation metrics. *J. Am. Med. Inform. Assoc.* **2014**, *21*, 231–237. [CrossRef]
33. Mohammed, A.A.; Umaashankar, V. Effectiveness of hierarchical softmax in large scale classification tasks. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 1090–1094.
34. Wehrmann, J.; Cerri, R.; Barros, R. Hierarchical multi-label classification networks. In Proceedings of the Thirty-Fifth International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5075–5084.
35. Silla, C.N.; Freitas, A.A. A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov.* **2011**, *22*, 31–72. [CrossRef]
36. Kowsari, K.; Brown, D.E.; Heidarysafa, M.; Meimandi, K.J.; Gerber, M.S.; Barnes, L.E. HDLTex: Hierarchical Deep Learning for Text Classification. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications, Cancun, Mexico, 18–21 December 2017.
37. Yang, Y.; Zhang, J.; Kisiel, B. A scalability analysis of classifiers in text categorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 96–103.
38. Babbar, R.; Metzig, C.; Partalas, I.; Gaussier, E.; Amini, M.R. On power law distributions in large-scale taxonomies. *ACM Sigkdd Explor. Newsl.* **2014**, *16*, 47–56. [CrossRef]

39. Miranda-Escalada, A.; Gonzalez-Agirre, A.; Krallinger, M. CodiEsp Corpus: Spanish Clinical Cases Coded in ICD10 (CIE10)—eHealth CLEF2020. Available online: https://zenodo.org/record/3758054#.XxXGgy17E6h (accessed on 29 July 2020).
40. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
41. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
42. Kaiser, Ł.; Sutskever, I. Neural GPUs learn algorithms. *arXiv* **2015**, arXiv:1511.08228.