Towards automated clinical coding

Finneas Catling[a,*], Georgios P. Spithourakis[a], Sebastian Riedel[a]

[a] *University College London, Gower Street, London, UK. WC1E 6BT*

Abstract

Background. Patients' encounters with healthcare services must undergo clinical coding. These codes are typically derived from free-text notes. Manual clinical coding is expensive, time-consuming and prone to error. Automated clinical coding systems have great potential to save resources, and realtime availability of codes would improve oversight of patient care and accelerate research. Automated coding is made challenging by the idiosyncrasies of clinical text, the large number of disease codes and their unbalanced distribution.

Methods. We explore methods for representing clinical text and the labels in hierarchical clinical coding ontologies. Text is represented as term frequency-inverse document frequency counts and then as word embeddings, which we use as input to recurrent neural networks. Labels are represented atomically, and then by learning representations of each node in a coding ontology and composing a representation for each label from its respective node path. We consider different strategies for initialisation of the node representations. We evaluate our methods using the publicly-available Medical Information Mart for Intensive Care III dataset: we extract the history of presenting illness section from each discharge summary in the dataset, then predicting the International Classification of Diseases, ninth revision, Clinical Modification codes associated with these.

Results. Composing the label representations from the clinical-coding-ontology nodes increased weighted F1 for prediction of the 17561 disease labels to 0.264-0.281 from 0.232-0.249 for atomic representations. Recurrent neural network text representation improved weighted F1 for prediction of the 19 disease-category labels to 0.682-0.701 from 0.662-0.682 using term frequency-inverse document frequency. However, term frequency-inverse document frequency outperformed recurrent neural networks for prediction of the 17561 disease labels.

*Corresponding author

URL: f.catling@ucl.ac.uk (Finneas Catling), g.spithourakis@cs.ucl.ac.uk (Georgios P. Spithourakis), s.riedel@cs.ucl.ac.uk (Sebastian Riedel)

Conclusions. This study demonstrates that hierarchically-structured medical knowledge can be incorporated into statistical models, and produces improved performance during automated clinical coding. This performance improvement results primarily from improved representation of rarer diseases. We also show that recurrent neural networks improve representation of medical text in some settings. Learning good representations of the very rare diseases in clinical coding ontologies from data alone remains challenging, and alternative means of representing these diseases will form a major focus of future work on automated clinical coding.

Keywords: Clinical coding, recurrent neural networks, hierarchical representation learning, knowledge representation, natural language processing, machine learning

---

1. Introduction

Encounters with patients in general practice, hospitals and other healthcare services are recorded in myriad ways. Many of the resultant data are highly-structured. However, the *narrative* of how a patient came to be in contact with healthcare services and of what happened thereafter is almost always recorded as free text. Free text is highly expressive and efficient, and it is thus enduringly popular with the busy healthcare professionals who record patient information [1].

A tension exists between the needs of healthcare professionals using data from individual patients at the point of care, and of those seeking insight into patient populations as a whole for purposes of research, quality improvement and administration. These latter purposes favour structured data which are straightforwardly amenable to statistical analysis. *Clinical coding* addresses the tension by assigning standardised codes to patient encounters, after having interpreted the data associated with them. All of the popular coding ontologies have a hierarchical structure.

Clinical coding is currently performed manually, and hospitals typically employ a large number of full-time staff for this purpose. Manual clinical coding is time-consuming, with many hospital trusts in the UK only aiming to complete clinical coding several weeks after patient discharge [2]. Even if the efficiency of manual coding increased significantly, there is no realistic prospect that it could be used to assign clinical codes in close to realtime. There is also a wealth of evidence to suggest that manual coding is prone to error [2, 3, 4, 5, 6].

A system which performs accurate, automated clinical coding would have great potential to save resources, against the backdrop of a National Health Service (NHS) facing unprecedented financial pressure [7]. Were the predicted clinical codes available in near-realtime, this could facilitate greater analytics capability and improve oversight of patient care. Near-realtime availability of codes would be a huge advantage to recruiters for clinical research trials searching for specific subgroups of patients, and would accelerate the cycles of audit and quality improvement projects. Studies of healthcare-related predictive models demonstrated improved model performance where patient notes were used as model input in addition to physiological variables [8, 9]. Clinical codes might be expected to similarly improve the decision support models which are currently used in clinical practice.

The idiosyncrasies of medical language are a barrier to automated clinical coding. Free-text clinical notes are formatted *ad hoc* to suit their author's current aims and are rife with obscure vocabulary, non-standard syntax and ambiguous abbreviations. They are typically typed hurriedly and, thus, contain many spelling and grammatical errors. Many possible synonyms exist for clinical concepts, and these are often used interchangeably. Negation is used very frequently, and negating expressions are often placed distantly from the negated concept [10]. In many cases, the main clinical concept under discussion is felt to be obviously implied, but it is not mentioned explicitly. In addition, clinical notes convey the subjective perspective of a healthcare professional — who

is themselves delivering care within an institution with its own peculiarities of medical practice — rather than the objective reality of a patient's condition [11].

Another long-standing barrier to automated coding has been the scarcity of hospitals using electronic health records (EHRs), which both prohibits automated coding at institutions still using paper records and limits the amount of training data available, even at hospitals that use EHRs. UK Government policy mandates that the NHS will be 'paper free at the point of care' by 2020 [12], so it is hoped that this barrier will rapidly be removed.

A more persistent challenge is the label-space problem: popular disease ontologies contain tens of thousands of labels, and their distribution is highly imbalanced in most datasets, with many absent labels for rare diseases. Some previous studies of automated clinical coding models adopt toy ontologies, consisting of the k most-frequent labels, and discard training examples with other labels. This approach would be unacceptable in real healthcare environments, where many rare diseases are potentially reversible but have serious sequelae when neglected. More promising approaches to the label-space problem exploit the structure of the underlying disease ontology and use this to learn better representations of individual labels.

In this study, we build and evaluate systems for automated clinical coding which mitigate the above challenges. In doing so, we explore methods for representing both clinical text and the labels in hierarchical clinical coding ontologies.

1.1. Related work

Several rule-based systems which mimic the approach of human clinical coders have been proposed [13, 14]. However, these are labour-intensive to develop and maintain, typically grow to become highly complex and unpredictable, and perform poorly on unconstrained corpora [15].

Other studies manually engineer features of clinical documents and use these, paired with their respective labels, as input to supervised classification models. Classifiers including naive Bayes, boosting, k-nearest-neighbours, support vector machines (SVMs) and Bayesian ridge regression have been considered [16, 17, 18, 19]. Generic features such as bag of words (BoW) counts [20] and term frequency-inverse document frequency (TF-IDF) weights [21, 22] are commonly used. Other features are healthcare-specific, including similarity scores between the input document and labels in a disease ontology or the metadata associated with those labels [23, 24, 25]. It is relatively straightforward to derive features using external medical knowledge and include these in the document representation, but it is more difficult to ensure that the model will learn to use these feature in the manner intended. It is also challenging to manually specify a compact feature set that captures the richness of the document text.

An alternative to manual feature engineering is representation learning directly from data. Recurrent neural networks (RNNs) are intuitively appealing for learning representations of sequential data. In particular, the long short-term memory (LSTM) and gated recurrent unit (GRU) variants improve representation of long sequences by avoiding the vanishing gradients observed with

earlier RNNs [26, 27]. GRUs are particularly attractive, as they produce similar performance to LSTMs whilst using a simpler design with fewer trainable parameters [28]. LSTMs and GRUs have been used to represent sequential healthcare data, including multivariate time series [29], text documents [30] and serial encounters with healthcare services [31, 32, 33].

In clinical coding, the structure of relevant knowledge is explicitly specified by the hierarchical relationships in disease ontologies. Several studies adopt model architectures which reflect this structure. One approach trains a binary SVM for each node in an ontology, with each classifier learning only from training examples classed as positive by its parent classifier [34, 35, 36, 37, 38]. A framework has been described for feedforward neural network training which is regularised so as to incorporate tree-based priors derived from disease ontologies [39]. Another approach represented each leaf in a disease ontology as a learnt convex combination of the leaf embedding and its ancestor nodes' embeddings. Subsequent analysis revealed that larger weights were assigned to nodes lower in the hierarchy for common diseases and to higher nodes for rarer diseases [32].

2. Methods

This study focuses on clinical coding tasks which equate to single-label multi-class classification of text documents. Each label corresponds to a path through an ontology structured as a directed singly-connected graph, i.e. a tree. The different models considered differ mainly in the way they represent the documents and the labels. Here, we present a general approach that should extend to a variety of clinical text data and ontologies in a straightforward fashion.

2.1. Document representation

We aim to represent each document as a feature vector that summarises the document's content. We denote the study vocabulary as $\boldsymbol{v} = \{w_1, w_2, \ldots, w_n\}$, which is used to convert the sequence of tokens in a document to a vector of token indexes $\boldsymbol{d}$. A variety of methods were considered for deriving a good document representation from $\boldsymbol{d}$.

TF-IDF representation. A document may be represented as TF-IDF weights, where the jth weight is obtained as

$$\mathrm{tf}(\boldsymbol{d})_j = \sum_{k=1}^{|\boldsymbol{d}|} \mathbb{I}(v_j = d_k) \tag{1}$$

$$\mathrm{df}(\boldsymbol{v})_j = \sum_{k=1}^{\mathrm{D}} \mathbb{I}(v_j \in \boldsymbol{d^k}) \tag{2}$$

$$\mathrm{idf}(\boldsymbol{v})_j = \log\left(\frac{1+\mathrm{D}}{1+\mathrm{df}(\boldsymbol{v})_j} + 1\right) \tag{3}$$

$$\mathrm{tfidf}(\boldsymbol{d})_j = \mathrm{tf}(\boldsymbol{d})_j \times \mathrm{idf}(\boldsymbol{v})_j \tag{4}$$

where D is the number of documents in the collection and $\mathbb{I}$ is the indicator function, such that $\mathbb{I}(q)$ is 1 when q is true and is 0 otherwise. The addition of 1 in Equation 3 prevents zero weights for tokens which occur in every document.

Mean-embedding representation. Alternatively, each token in the study vocabulary may be assigned a vector embedding. This yields embeddings $\boldsymbol{E^w} \in \mathbb{R}^{|\boldsymbol{v}| \times \mathrm{E}}$, where E is the dimensionality of each embedding. The embeddings are used to transform a document into a matrix $\boldsymbol{X} \in \mathbb{R}^{|\boldsymbol{d}| \times \mathrm{E}}$. A simple vector representation of this document may then obtained by calculating the mean of the word embeddings which comprise it. A document representation as the mean word embedding is henceforth denoted as $\bar{\boldsymbol{x}}$, where its jth element is obtained as

$$\bar{x}_j = \frac{1}{|\boldsymbol{d}|} \sum_{t=1}^{|\boldsymbol{d}|} X_{tj} \tag{5}$$

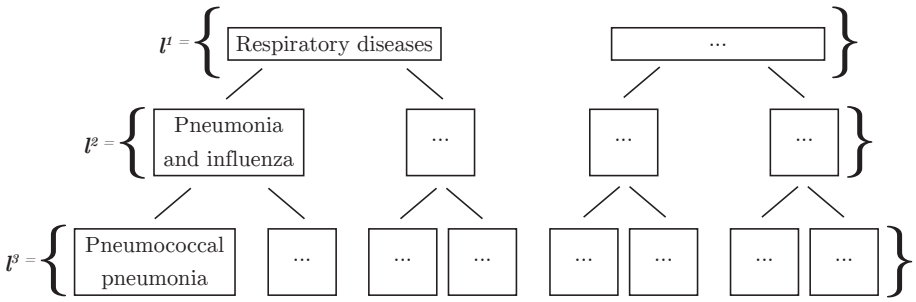

Figure 1: Example disease ontology, indicating the sets of nodes indexed by $\boldsymbol{l^1}$, $\boldsymbol{l^2}$ and $\boldsymbol{l^3}$.

GRU representation. A separate representation of the document may be obtained by feeding its word embeddings sequentially into a GRU, and making use of the outputs. The GRU output $\boldsymbol{o_t}$ at timestep t, given an input $\boldsymbol{X_{t:}}$ and the previous output $\boldsymbol{o_{t-1}}$, is obtained as

$$\boldsymbol{z_t} = \sigma\big(\boldsymbol{W^z}[\boldsymbol{X_{t:}}, \boldsymbol{o_{t-1}}] + \boldsymbol{b^z}\big) \tag{6}$$

$$\boldsymbol{r_t} = \sigma\big(\boldsymbol{W^r}[\boldsymbol{X_{t:}}, \boldsymbol{o_{t-1}}] + \boldsymbol{b^r}\big) \tag{7}$$

$$\tilde{\boldsymbol{o}}_{\boldsymbol{t}} = \tanh\big(\boldsymbol{W^o}[\boldsymbol{X_{t:}}, \boldsymbol{r_t} \odot \boldsymbol{o_{t-1}}] + \boldsymbol{b^o}\big) \tag{8}$$

$$\boldsymbol{o_t} = \boldsymbol{z_t} \odot \boldsymbol{o_{t-1}} + (\mathbf{1} - \boldsymbol{z_t}) \odot \tilde{\boldsymbol{o}}_{\boldsymbol{t}} \tag{9}$$

where $[,]$ indicates vector concatenation, $\odot$ indicates pointwise multiplication and $\mathbf{1}$ is a vector where every element equals 1.

A bidirectional GRU design was adopted to compensate for small backpropagated gradients from the long input sequences. The input sequence was fed the sequence through two GRUs, once in unmodified and once in reversed form, and the outputs were concatenated at each timestep. Also for reasons of gradient preservation, the mean of the concatenated GRU outputs was used in preference to the final concatenated output. A document representation as the mean output of a bidirectional GRU is henceforth denoted as $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$.

2.2. Label representation

We aim to represent each label as a feature vector that captures the label's meaning. Let $\boldsymbol{l^k}$ index the set of final nodes in all possible paths of length k through a tree-structured ontology, beginning at the root node. $\boldsymbol{l^0}$ indexes just the root node and is ignored. Figure 1 demonstrates this in an example ontology. A variety of representations were considered for the labels in each $\boldsymbol{l^k}$.

Atomic representation. Each label can be represented atomically, i.e. with no information was shared between label representations. Given a document representation $\boldsymbol{h}$, the probability distribution over the labels is obtained as

$$p(\boldsymbol{l^k} \mid \boldsymbol{h}) = \text{softmax}(\boldsymbol{W^k h} + \boldsymbol{b^k}) \tag{10}$$

That is, each label is represented as a row of a learnt weight matrix $\boldsymbol{W^k}$ and a corresponding element in a learnt bias vector $\boldsymbol{b^k}$.

Where $\boldsymbol{l^n}$ indexes the terminal labels in an ontology, predictions for the higher levels of labels can be obtained using the *separate-model* strategy, wherein we learn a separate $\boldsymbol{W^k}$ and $\boldsymbol{b^k}$ for each $\{\boldsymbol{l^k} : 0 < k < n\}$. Alternatively, predictions for the higher levels of labels can be obtained using the *truncated-terminal* strategy, wherein we learn only $\boldsymbol{W^n}$ and $\boldsymbol{b^n}$, predict $p(\boldsymbol{l^n} \mid \boldsymbol{h})$ then truncate it as per the ontology's tree structure. For example, the prediction for the ith label in the penultimate level of labels is obtained as

$$p(l_i^{n-1} \mid \boldsymbol{h}) = \sum_{l_j^n \in \text{children}(l_i^{n-1})} p(l_j^n \mid \boldsymbol{h}) \tag{11}$$

Mean-embedding representation. As an alternative to atomic representation, each node in the ontology may be assigned a vector embedding. This yields embeddings $\boldsymbol{E^n} \in \mathbb{R}^{\mathrm{N} \times \mathrm{F}}$, where N is the number of nodes in the ontology and F is the dimensionality of each embedding. A representation of each label is then composed from the nodes traversed in the path corresponding to that label. The embeddings are used to transform the indices $\boldsymbol{n}$ of the traversed nodes into a matrix $\boldsymbol{Z} \in \mathbb{R}^{|\boldsymbol{n}| \times \mathrm{F}}$. This strategy has the advantage of sharing information between representations of labels with common ancestor nodes. Analagously to Equation 5, a simple representation $\bar{\boldsymbol{z}}$ of a label is obtained as the mean of the node embeddings that comprise its path.

GRU representation. Similarly to the methods described in Section 2.1, a separate representation $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ is obtained by feeding a label's node embeddings sequentially into a GRU. As these sequences are much shorter than in the case of text documents, the final output from a unidirectional GRU is used as the label representation. The architecture of the models which represent the documents as $\bar{\boldsymbol{x}}$ or $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$, and the labels as $\bar{\boldsymbol{z}}$ or $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$, is summarised in Figure 2. The probability distribution over the labels is calculated as shown in Equation 10, but with $\boldsymbol{W^k}$ being composed row-wise using $\bar{\boldsymbol{z}}$ or $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ for each label, rather than being learnt directly.

Various techniques were considered for initialisation of $\boldsymbol{E^n}$. In the *random* strategy, each embedding was drawn from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The *pretrained* strategy augments the random strategy by substituted pretrained embeddings for terminal nodes in the ontology. The *composed* initialisation strategy began with embeddings for the terminal nodes derived using either the *random* or *pretrained* strategies. The embeddings for the nodes in the layer above were each calculated as the sum of the embeddings for their child nodes. The summation was carried out for the higher layers in the ontology in sequence, until an embedding was derived for each node. Finally, all node embeddings were scaled to zero mean and unit variance across all embedding dimensions.

2.3. Training and evaluation

The parameters $\boldsymbol{\theta}$ for each model are learnt so as to minimise cross-entropy loss. Having obtained $p(\boldsymbol{l^k} \mid \boldsymbol{h})$ for a training sample as shown in Equation 10,

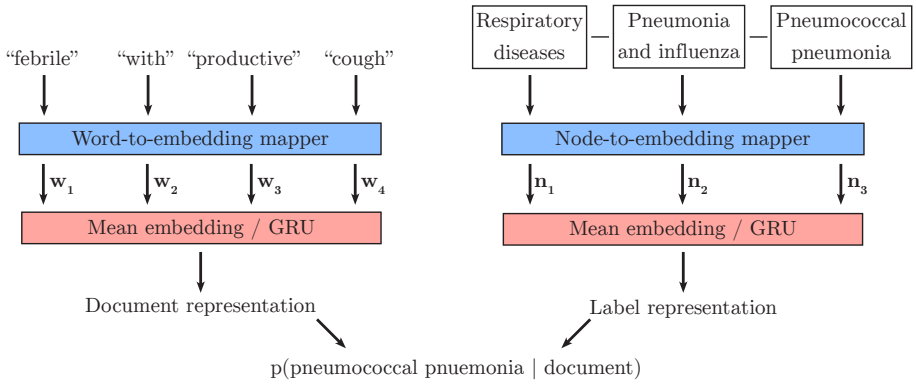

Figure 2: Overview of the architecture for models which represent the documents as $\bar{\boldsymbol{x}}$ or $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$, and the labels as $\bar{\boldsymbol{z}}$ or $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$. We show how a probability estimate for a disease label is obtained given an input document.

cross-entropy loss for that sample is calculated as

$$L(\boldsymbol{\theta}) = -\log\big(p(l^k_* \mid \boldsymbol{h})\big) \tag{12}$$

where $*$ indexes the true label in $\boldsymbol{l^k}$. After training, $p(\boldsymbol{l^k} \mid \boldsymbol{h})$ is calculated for each sample in a held-out test dataset. The indexes of the maximum calculated probability for each sample are concatenated to form the test label predictions $\hat{\boldsymbol{y}}$. Point estimates of model performance were obtained by calculating precision, recall and f1 scores using $\hat{\boldsymbol{y}}$ and the true label indexes $\boldsymbol{y}$. The scores were also calculated on 10000 bootstrap samples from $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$, and the results were used to obtain 95% confidence intervals for the point estimates.

Several averaging methods were considered for extending the scores to multiclass classification [40]. Models are evaluated using weighted-average scores in this study, following the logic that common diagnoses should usually be considered in preference to rare ones, in order that fewer patients are misdiagnosed. Where $\boldsymbol{t^k}$ indexes the subset of labels in $\boldsymbol{l^k}$ which occur at least once in $\boldsymbol{y}$ or $\hat{\boldsymbol{y}}$, and $f(\boldsymbol{y_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}})$ indicates the score of interest (precision, recall or f1) calculated for the ith label, the *weighted* average is obtained as

$$\text{weighted}(f, \boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{|\boldsymbol{y}|} \sum_{i \in \boldsymbol{t^k}} |\boldsymbol{y_i}| f(\boldsymbol{y_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}) \tag{13}$$

Accuracy score and micro-averaged scores are identical to weighted recall in the multiclass setting, so need not be reported separately. Macro-averaging is avoided, as it is problematic where insufficient data exist to learn good representations of some labels, e.g. rare diseases. The classifier that maximises macro f1 score in these circumstances would simply predict the rare labels frequently [41].

3. Dataset

The Medical Information Mart for Intensive Care III (MIMIC-III) dataset contains deidentified numeric and free-text data from patient admissions to the Beth Israel Deaconess Medical Center in Boston, Massachusetts, between 2001 and 2012 [42]. This study considers the 55172 free-text discharge summaries provided in the dataset, and their associated primary International Classification of Diseases, ninth revision, Clinical Modification (ICD-9-CM) codes. MIMIC-III is publicly available, and the Institutional Review Boards at Massachusetts Institute of Technology and the Beth Israel Deaconess Medical Center approved the use of the data for research.

3.1. Ontology

ICD-9-CM is a disease ontology used from 1979 until 2015 to code patient encounters throughout the USA [43]. Each admission in MIMIC-III is assigned multiple ICD-9-CM codes by human clinical coders, one of which is identified as the primary reason for admission.

ICD-9-CM contains four levels of hierarchy, excluding the root node. The first (*chapter*) level divides disease into 19 categories, most of which correspond to body systems. These are further subdivided into 150 categories at the second (*sub-chapter*) level by affected anatomical structures or types of pathology. The third (*major*) level consists of 1234 diseases or narrow disease categories, e.g. 'viral pneumonia'. Some paths through the ICD-9-CM hierarchy are 3 nodes in length, ending at the majors level, but the majority include one of the 16327 nodes from the fourth level. These fourth-level nodes provide greater detail, e.g. 'pneumonia due to parainfluenza virus'. In this study, the 17561 final nodes in all the possible (3- and 4-node) paths through ICD-9-CM are referred to collectively as the *terminal labels*.

3.2. Data adaptation

Very commonly in clinical practice, a patient's initial clinical presentation is ambiguous. There is accompanying uncertainty regarding their underlying diagnosis, and treatment is often initiated for a range of possible diagnoses then refined on the basis of further information over the following days. Healthcare professionals' documentation early in the clinical encounter reflects this uncertainty; the underlying diagnosis may be underspecified or not explicitly mentioned altogether. Realtime clinical coding using this text therefore requires a code *prediction* rather than a code *extraction* model. We adapt the MIMIC-III discharge summaries to favour such a model, by omitting the explicit lists of diagnoses they contain and instead considering only the history of presenting illness (HoPI) section. This section describes the patient's predominant symptoms and significant findings on physical examination at presentation to hospital, the results of initial investigations and the initial treatment offered. We are unaware of any studies that have adapted the MIMIC-III dataset in this way, and so are unable to present any relevant previous results for comparison purposes.

Pt was in USOH, awaiting R THR, collapsed while celebrating a funeral mass, was down for 1 min prior to EMS arrival, found to be pulseless, atrial activity noted on stips [*sic*] but only occasional wide qrs complexes, could not transcut pace, got atropine and calcium gluc, went to [**First Name4 (NamePattern1) 46**] [**Last Name (NamePattern1) **], was intubated for protection, K 6.6, HCO3 13, and Cr 2.7. Got kayexylate [*sic*], bicarb gtt, lasix, and extubated. ECG w/RBBB, LAD, LAFB, and sig PR delay so sent here for pacer. R IJ pacer wire screwed in but still temporary. Transferred to [**Hospital1 18**] for permanent pacer and further managment [*sic*].

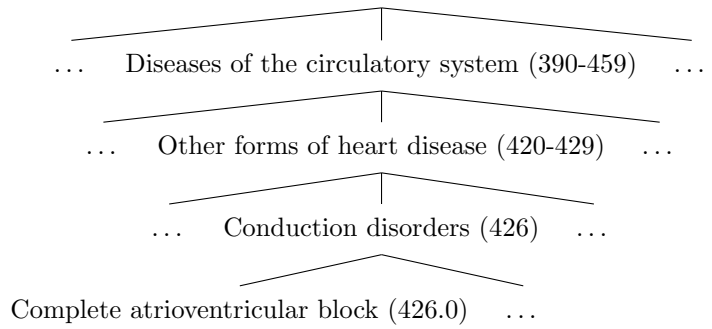

Figure 3: Example HoPI document and corresponding path through ICD-9-CM.

Figure 3 shows an example HoPI document and its associated ICD-9-CM code. Many of the idiosyncrasies of clinical text discussed in Section 1 are evident. In particular, non-standard abbreviations ('transcut', 'gluc', 'sig') and slang terms ('was down', 'got', 'screwed in') are used, and there are several spelling errors. The correct terminal label could be confidently predicted by a human coder based on the results of the 'stips' [*sic*], i.e. the initial electrocardiogram, but is not explicitly stated. For other HoPI documents, it would be very challenging or impossible to accurately predict the terminal ICD-9-CM label, even for an specialist doctor. For example, a patient may present to hospital with non-specific signs of sepsis, with the responsible pathogen only later identified by the hospital's microbiology laboratory. However, it would be much more feasible to accurately predict labels at the chapter (first), sub-chapter (second) or major (third) levels of ICD-9-CM in such cases.

3.3. Preprocessing

Access to MIMIC-III, version 1.4 was gained with permission of its curators, after completion of the prerequisite data-protection training. All available discharge summaries (55177 records) were isolated from the free-text notes table of the database, excluding any records marked as containing errors or without

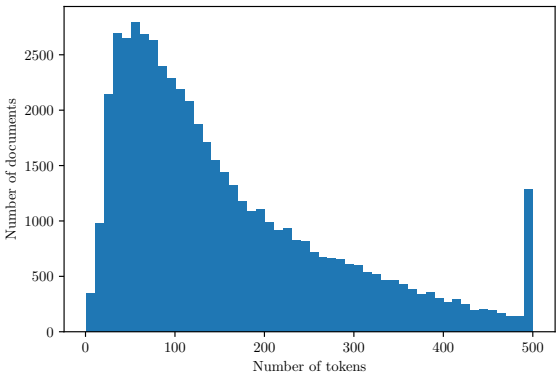

Figure 4: Number of tokens in each HoPI document, after data preprocessing.

any associated ICD-9-CM codes (5 records, 0.0%). The remaining records were merged with the primary ICD-9-CM codes for the corresponding admissions. The records were split randomly into training (38588 records, 69.9%), validation (5536 records, 10.0%) and testing (11048 records, 20.0%) folds. Where multiple discharge summaries existed for a single admission, the splitting process ensured they were allocated to the same fold.

Each discharge summary was tokenised using the Stanford Tokenizer [45]. Text in MIMIC-III is supplied with potential patient identifiers substituted with special sequences, examples of which are visible in the example HoPI document shown in Figure 3. These special sequences were identified and replaced by the first token in the sequence, e.g. '[**Hospital1 18**]' was replaced by 'Hospital1'.

The formatting of the MIMIC-III discharge summaries is standardised, allowing the HoPI sections to be extracted by means of rule-based pattern matching, e.g. the majority of HoPI sections begin with 'History of Present Illness' and end with three new lines. The pattern-matching rules were iteratively refined by inspection of random samples from the identified HoPI sections and the discharge summaries where no HoPI section had been identified. Discharge summaries identified as not containing a HoPI section or as containing an empty HoPI secion (2641 records, 4.9%) were dropped. On manual inspection, the majority of these records contained addenda to other discharge summaries rather than being the sole discharge summary associated with a patient admission. HoPI documents more than 500 tokens in length (1143 records, 2.2%) were truncated at 500 tokens. Figure 4 shows the distribution of token sequence lengths after truncation.

Each token present at least once in the training HoPI documents was extracted (92468 tokens), and the number of occurrences of each was counted. Tokens which occurred ≥ 5 times (19503 tokens) were retained, and comprised the study vocabulary. Each token in the study vocabulary was assigned a unique integer ID, with a further unique integer ID being assigned for out-of-vocabulary tokens, yielding the vocabulary index $\boldsymbol{v}$. Each HoPI document was converted

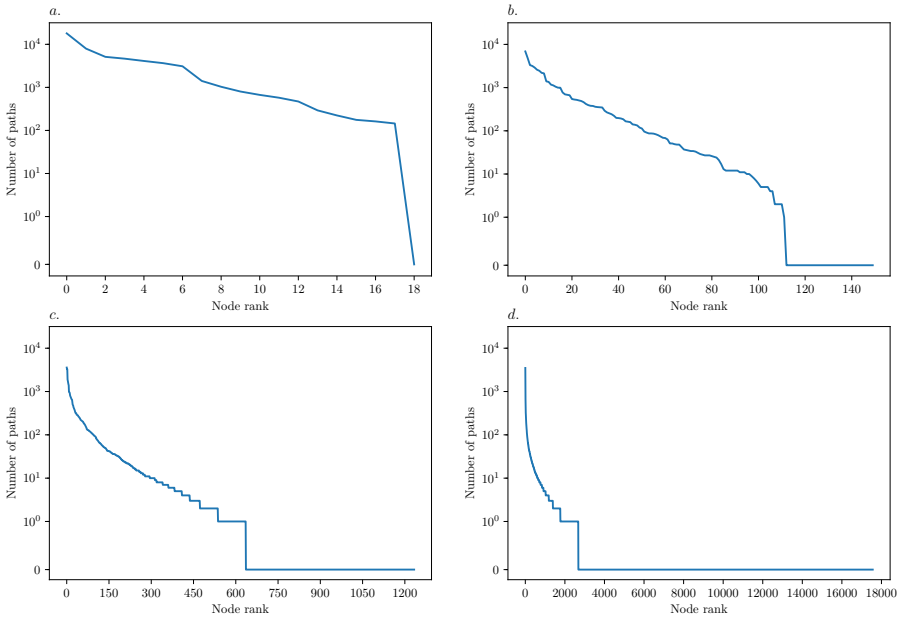

Figure 5: Number of times each ICD-9-CM node is traversed in the path to the terminal node for each document in the dataset. Plots *a*, *b*, *c* and *d* show nodes at the chapter (first), sub-chapter (second), major (third) and terminal (fourth) levels respectively.

to a 1-dimensional array of integers using this index.

Each node in the ICD-9-CM hierarchy was assigned a unique ID. Labels at the chapter (level 1), sub-chapter (level 2), major (level 3) and terminal levels of ICD-9-CM were each mapped to the sequence of IDs corresponding their ancestral node ordering. The distribution of node usage in the dataset is shown for each level of the ICD-9-CM hierarchy in Figure 5. The classes are imbalanced at all four levels of hierarchy, and become progressively more imbalanced at the deeper levels. At the deepest (terminal) level, only 2675 (15.2%) of 17561 available labels are used at least once, and the most-frequent label is used for 3499 (6.7%) records.

4. Experiments and results

We trained models to predict the primary ICD-9-CM code assigned to each HoPI document in the MIMIC-III dataset. We made four separate predictions for each document, each considering labels at a different level in the ICD-9-CM hierarchy. Higher-level labels in ICD-9-CM (i.e. those above the terminal level) were predicted using both the separate-model and truncated-terminal strategies. After training, we evaluated the performance of each model on the held-out testing fold.

4.1. Baseline models

The tfidf($\boldsymbol{d}$)-atomic models represented the HoPI documents as TF-IDF weights, and used these as input to a multinomial logistic regression classifier. This corresponds to an atomic representation of the labels. As each document representation is large — the same dimensionality as the study vocabulary — in these models, only the subset which occurred at least once in the training data were represented. Optimisation occurred via the stochastic average gradient method, using a maximum of 1500 iterations [46]. These models were implemented using *scikit-learn 0.18.1* [47].

We used l2 regularisation to avoid overfitting. For the chapter (level 1), sub-chapter (level 2) and major (level 3) models, 50 candidate values for the l2 coefficient were evaluated, logarithmically spaced between 10^{-2} and 10^{2}. Due to the length of training times (several days) for the terminal label-prediction model, a 32-value subset of these candidates was evaluated. The l2 coefficients which maximised weighted recall scores on the validation fold were used in the final models.

4.2. Neural models

These models represented the HoPI documents using word embeddings, as either $\bar{\boldsymbol{x}}$ or $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$. The $\bar{\boldsymbol{x}}$-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic models both used atomic representations of the ICD-9-CM labels. The $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model used $\bar{\boldsymbol{z}}$ for label representation. The $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ model used $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ for label representation with a 100-hidden-unit GRU. The node embeddings were initialised using the random strategy in the main experiments.

Each token in the study vocabulary was assigned a 200-dimensional pretrained embedding, where available. The pretrained embeddings were derived in a previous study, using a skip-gram model trained on a large text corpus which combines English Wikipedia and the PubMed and PubMed Central databases [48, 49]. We hypothesise that these embeddings capture the meaning of their tokens in a variety of technical domains, and in the biomedical domain in particular. Tokens without a pretrained embedding were assigned vectors drawn from independent zero-mean Gaussian distributions. Token embeddings were fixed during model training.

In the $\bar{\boldsymbol{x}}$-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic models, representations were learnt for all labels regardless of whether they occurred in the training data. Given

the large number of ICD-9-CM nodes not present in the dataset, it was felt to be unlikely that the node embeddings could be learnt satisfactorily during model training. The node embeddings were therefore fixed during training, except in models using $\bar{\boldsymbol{z}}$ to represent each label, as these representations would otherwise have no learnt component.

The random, random-composed, pretrained and pretrained-composed initialisation strategies were compared in a separate experiment, also using the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ model. The pretrained embeddings for the terminal nodes were derived in a previous study, using a dataset of ICD-9-CM codes from the health insurance claims of 4 million people. The authors partitioned the longitudinal data for each persion by time interval, randomly shuffled codes within partitions, then derived code embeddings by using each partition as an input 'sentence' to a skip-gram model [50]. Terminal nodes without a pretrained embedding were assigned vectors drawn from a standard Gaussian distribution.

The GRU weights used Glorot uniform initialisation [51]. The GRU biases were initialised as $\mathbf{1}$. $\mathbf{0}$ was used as the initial GRU output. The forward and backward GRUs used for document representation both contained 50 hidden units, meaning that $\overleftrightarrow{\text{GRU}}(\boldsymbol{X}) \in \mathbb{R}^{100}$. To match this dimensionality, $\bar{\boldsymbol{x}}$ was calculated using $\hat{\boldsymbol{E}}^{\boldsymbol{w}} \in \mathbb{R}^{|\boldsymbol{v}| \times 100}$, which was obtained from the pretrained word embeddings $\boldsymbol{E}^{\boldsymbol{w}} \in \mathbb{R}^{|\boldsymbol{v}| \times 200}$ as

$$\hat{\boldsymbol{E}}^{\boldsymbol{w}} = \tanh(\boldsymbol{E}^{\boldsymbol{w}}\boldsymbol{W}^{\boldsymbol{w}} + \boldsymbol{b}^{\boldsymbol{w}}) \tag{14}$$

where + denotes broadcasted addition, and $\boldsymbol{W}^{\boldsymbol{w}}$ and $\boldsymbol{b}^{\boldsymbol{w}}$ are learnt jointly with the other model parameters. Previous clinical coding studies use an equivalent strategy for dimensionality reduction [31].

In models using $\bar{\boldsymbol{z}}$ to represent each label, node embeddings $\boldsymbol{E}^{\boldsymbol{n}} \in \mathbb{R}^{\text{N} \times 100}$ were used in order to match the dimensionality of $\bar{\boldsymbol{z}}$ and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$. In models using $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ to represent each label, node embeddings $\boldsymbol{E}^{\boldsymbol{n}} \in \mathbb{R}^{\text{N} \times 300}$ were used as dictated by the pretrained embedding size, with dimensionality reduction accomplished by the $\overrightarrow{\text{GRU}}$ itself.

Each model was trained over multiple epochs using the Adam method [52]. Optimizer parameters were set as suggested in [52], with the exception of the learning rate which was increased to 0.003 based on validation fold results. At the end each epoch, cross-entropy loss was calculated on the validation fold. Where validation loss was lower than in all previous epochs, the model weights were saved. If lowest validation loss failed to improve for four consecutive epochs, training was halted. The neural models were implemented using *TensorFlow 1.2.0* [53].

Given the distribution of HoPI document lengths, and that recurrent neural networks require all sequences within each mini-batch to be the same length, training times were substantially decreased by a bucketing approach. Documents were sorted in length order, then bucket boundaries were calculated such that all but the final bucket (containing the longest documents) contained approximately 4000 documents. Within each bucket, shorter documents were

zero-padded to the length of the longest document. At the start of each training epoch, each bucket was populated with its documents in shuffled order, and documents were fed to the model from a single random bucket per training step in mini-batches of 128. Where fewer than 128 documents remained in a bucket, these were discarded for the remainder of the epoch. A mini batch size of 128 was used as this produced optimal performance in a previous similar study [44]. To ensure consistent results during model selection and evaluation, the entire validation and test folds were fed to the model as single batches.

At training time, 30% dropout was applied to the word embeddings, and to the node embeddings where they were used. As proposed previously [54], a random dropout mask was generated for each input sequence and applied consistently at each timestep. We used l2 regularisation to avoid overfitting the $\bar{\boldsymbol{x}}$-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic models. 9 candidate values for the l2 coefficient (0, and 8 values logarithmically spaced between 10^{-7} and 10^{0}) were evaluated for each model. The l2 coefficients which minimised cross-entropy loss on the validation fold were used in the final models.

4.3. Results

Results for prediction of the chapter (level 1), sub-chapter (level 2) and major (level 3) labels using the separate-model strategy are shown in Tables 1, 3 and 5 . Results for prediction of the terminal labels are shown in Table 7. $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic outperformed tfidf($\boldsymbol{d}$)-atomic when predicting the chapter (level 1) labels. Conversely, tfidf($\boldsymbol{d}$)-atomic outperformed $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic when predicting the terminal labels. The $\bar{\boldsymbol{x}}$-atomic models were not competitive. Relative to $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic, $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ performed similarly when predicting the labels higher in the ICD-9-CM hierarchy, but were superior when predicting the terminal labels. Results for two additional baseline classifiers are included for comparison purposes. These predict the most-frequent label, and randomly from a uniform distribution over all labels, respectively.

Results for prediction of the chapter (level 1), sub-chapter (level 2) and major (level 3) labels using the truncated-terminal strategy are shown in Tables 2, 4 and 6. At all 3 levels, tfidf($\boldsymbol{d}$)-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic performed significantly worse than with the separate-model strategy. In contrast, $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ performed similarly to the separate-model strategy.

Results when using different node initialisation strategies for $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ are shown in Tables 8, 9, 10 and 11 for the chapter (level 1), sub-chapter (level 2), major (level 3) and terminal labels, respectively. These results all use the separate-model strategy. All initialisations produced roughly equivalent performance when predicting the chapter (level 1), sub-chapter (level 2) and major (level 3) labels. Random and pretrained initialisations were exactly equivalent when predicting the chapter (level 1) and sub-chapter (level 2) labels, as pretrained embeddings were not used at these levels in either strategy. Random initialisation outperformed pretrained-composed initialisation when predicting the terminal labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.081 (0.075 - 0.087)	0.189 (0.174 - 0.204)	0.060 (0.056 - 0.065)
Most-frequent	0.176 (0.168 - 0.184)	0.118 (0.112 - 0.125)	0.344 (0.335 - 0.353)
tfidf($\mathbf{d}$)-atomic	0.672 (0.662 - 0.682)	**0.673 (0.660 - 0.682)**	**0.694 (0.685 - 0.702)**
$\bar{\mathbf{x}}$-atomic	0.620 (0.610 - 0.630)	0.620 (0.609 - 0.631)	0.646 (0.637 - 0.655)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	**0.691 (0.682 - 0.700)**	**0.686 (0.677 - 0.696)**	**0.702 (0.694 - 0.711)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.691 (0.682 - 0.701)**	**0.692 (0.682 - 0.702)**	**0.705 (0.696 - 0.713)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.688 (0.679 - 0.697)**	**0.684 (0.674 - 0.695)**	**0.701 (0.692 - 0.710)**

Table 1: Main results for chapter (level 1) label prediction, using the separate-model strategy and reporting weighted-average scores across labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.118 (0.111 - 0.124)	0.186 (0.174 - 0.198)	0.111 (0.105 - 0.117)
Most-frequent	0.176 (0.168 - 0.184)	0.118 (0.112 - 0.125)	0.344 (0.335 - 0.353)
tfidf($\mathbf{d}$)-atomic	0.638 (0.627 - 0.647)	0.664 (0.653 - 0.674)	0.650 (0.640 - 0.658)
$\bar{\mathbf{x}}$-atomic	0.611 (0.601 - 0.620)	0.632 (0.621 - 0.643)	0.617 (0.608 - 0.627)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	0.664 (0.655 - 0.673)	0.665 (0.656 - 0.676)	0.675 (0.666 - 0.684)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.689 (0.680 - 0.698)**	**0.696 (0.686 - 0.706)**	**0.696 (0.687 - 0.705)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.686 (0.677 - 0.695)**	**0.688 (0.679 - 0.698)**	**0.692 (0.683 - 0.700)**

Table 2: Main results for chapter (level 1) label prediction, using the truncated-terminal strategy and reporting weighted-average scores across labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.013 (0.011 - 0.016)	0.051 (0.039 - 0.064)	0.009 (0.008 - 0.011)
Most-frequent	0.032 (0.029 - 0.035)	0.018 (0.016 - 0.020)	0.134 (0.127 - 0.141)
tfidf($\mathbf{d}$)-atomic	**0.537 (0.527 - 0.547)**	**0.536 (0.525 - 0.548)**	**0.562 (0.552 - 0.571)**
$\bar{\mathbf{x}}$-atomic	0.447 (0.437 - 0.457)	0.455 (0.442 - 0.468)	0.490 (0.481 - 0.499)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	**0.549 (0.539 - 0.559)**	**0.550 (0.537 - 0.563)**	**0.573 (0.564 - 0.582)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.540 (0.530 - 0.550)**	**0.549 (0.536 - 0.561)**	**0.569 (0.559 - 0.578)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.544 (0.533 - 0.553)**	**0.549 (0.538 - 0.561)**	**0.564 (0.554 - 0.573)**

Table 3: Main results for sub-chapter (level 2) label prediction, using the separate-model strategy and reporting weighted-average scores across labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.023 (0.020 - 0.027)	0.061 (0.049 - 0.073)	0.021 (0.018 - 0.023)
Most-frequent	0.032 (0.029 - 0.035)	0.018 (0.016 - 0.020)	0.134 (0.127 - 0.141)
tfidf($\mathbf{d}$)-atomic	0.490 (0.479 - 0.500)	0.518 (0.505 - 0.530)	0.519 (0.510 - 0.529)
$\bar{\mathbf{x}}$-atomic	0.425 (0.415 - 0.435)	0.445 (0.433 - 0.458)	0.462 (0.452 - 0.471)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	0.500 (0.489 - 0.510)	0.507 (0.489 - 0.518)	0.528 (0.518 - 0.537)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.543 (0.532 - 0.552)**	**0.559 (0.542 - 0.570)**	**0.565 (0.556 - 0.575)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.538 (0.528 - 0.548)**	**0.542 (0.532 - 0.554)**	**0.554 (0.544 - 0.563)**

Table 4: Main results for sub-chapter (level 2) label prediction, using the truncated-terminal strategy and reporting weighted-average scores across labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.002 (0.001 - 0.003)	0.018 (0.006 - 0.037)	0.001 (0.001 - 0.002)
Most-frequent	0.009 (0.008 - 0.011)	0.005 (0.004 - 0.006)	0.071 (0.066 - 0.076)
tfidf($\mathbf{d}$)-atomic	**0.409 (0.399 - 0.419)**	**0.410 (0.398 - 0.422)**	**0.454 (0.445 - 0.464)**
$\bar{\mathbf{x}}$-atomic	0.322 (0.313 - 0.331)	0.326 (0.314 - 0.340)	0.371 (0.362 - 0.381)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	**0.408 (0.398 - 0.418)**	**0.394 (0.384 - 0.406)**	**0.449 (0.440 - 0.459)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.419 (0.410 - 0.429)**	**0.416 (0.404 - 0.427)**	**0.461 (0.451 - 0.470)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.419 (0.409 - 0.429)**	**0.412 (0.402 - 0.425)**	**0.455 (0.445 - 0.464)**

Table 5: Main results for major (level 3) label prediction, using the separate-model strategy and reporting weighted-average scores across labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.006 (0.005 - 0.008)	0.023 (0.015 - 0.033)	0.005 (0.004 - 0.006)
Most-frequent	0.009 (0.008 - 0.011)	0.005 (0.004 - 0.006)	0.071 (0.066 - 0.076)
tfidf($\mathbf{d}$)-atomic	0.378 (0.368 - 0.387)	0.391 (0.376 - 0.402)	0.424 (0.414 - 0.433)
$\bar{\mathbf{x}}$-atomic	0.302 (0.293 - 0.311)	0.307 (0.296 - 0.319)	0.352 (0.343 - 0.361)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	0.369 (0.360 - 0.379)	0.363 (0.349 - 0.374)	0.410 (0.401 - 0.419)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.416 (0.406 - 0.426)**	**0.421 (0.404 - 0.431)**	**0.454 (0.445 - 0.464)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.411 (0.402 - 0.421)**	**0.405 (0.396 - 0.418)**	**0.442 (0.432 - 0.451)**

Table 6: Main results for major (level 3) label prediction, using the truncated-terminal strategy and reporting weighted-average scores across labels.

Model	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
Random	0.001 (0.000 - 0.001)	0.012 (0.000 - 0.036)	0.000 (0.000 - 0.001)
Most-frequent	0.009 (0.008 - 0.010)	0.005 (0.004 - 0.005)	0.069 (0.064 - 0.074)
tfidf($\mathbf{d}$)-atomic	**0.262 (0.253 - 0.270)**	**0.252 (0.239 - 0.261)**	**0.324 (0.315 - 0.333)**
$\bar{\mathbf{x}}$-atomic	0.190 (0.182 - 0.198)	0.181 (0.172 - 0.192)	0.251 (0.242 - 0.259)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-atomic	0.240 (0.232 - 0.249)	0.222 (0.215 - 0.233)	0.295 (0.287 - 0.304)
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\bar{\mathbf{z}}$	**0.272 (0.264 - 0.281)**	**0.252 (0.243 - 0.262)**	**0.331 (0.323 - 0.340)**
$\overleftrightarrow{\text{GRU}}(\mathbf{X})$-$\overrightarrow{\text{GRU}}(\mathbf{Z})$	**0.271 (0.263 - 0.280)**	**0.252 (0.244 - 0.263)**	**0.320 (0.311 - 0.329)**

Table 7: Main results for terminal label prediction, reporting weighted-average scores across labels.

Initialisation	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
random	0.688 (0.679 - 0.697)	0.684 (0.674 - 0.695)	0.701 (0.692 - 0.710)
random-composed	0.696 (0.687 - 0.705)	0.693 (0.684 - 0.703)	0.707 (0.699 - 0.716)
pretrained	0.688 (0.679 - 0.697)	0.684 (0.674 - 0.695)	0.701 (0.692 - 0.710)
pretrained-composed	0.687 (0.677 - 0.696)	0.684 (0.674 - 0.694)	0.703 (0.694 - 0.711)

Table 8: Results for chapter (level 1) label prediction, using different node embedding initialisations for the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ model and reporting weighted-average scores across labels.

Initialisation	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
random	0.544 (0.533 - 0.553)	0.549 (0.538 - 0.561)	0.564 (0.554 - 0.573)
random-composed	0.546 (0.536 - 0.556)	0.546 (0.536 - 0.558)	0.567 (0.558 - 0.577)
pretrained	0.544 (0.533 - 0.553)	0.549 (0.538 - 0.561)	0.564 (0.554 - 0.573)
pretrained-composed	0.548 (0.538 - 0.558)	0.550 (0.538 - 0.561)	0.571 (0.562 - 0.580)

Table 9: Results for sub-chapter (level 2) label prediction, using different node embedding initialisations for the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ model and reporting weighted-average scores across labels.

Initialisation	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
random	0.419 (0.409 - 0.429)	0.412 (0.402 - 0.425)	0.455 (0.445 - 0.464)
random-composed	0.418 (0.408 - 0.427)	0.412 (0.403 - 0.425)	0.453 (0.443 - 0.462)
pretrained	0.420 (0.411 - 0.430)	0.416 (0.403 - 0.426)	0.456 (0.446 - 0.465)
pretrained-composed	0.415 (0.406 - 0.425)	0.409 (0.399 - 0.422)	0.452 (0.442 - 0.461)

Table 10: Results for major (level 3) label prediction, using different node embedding initialisations for the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ model and reporting weighted-average scores across labels.

Initialisation	F1 (95% CI)	Precision (95% CI)	Recall (95% CI)
random	**0.271 (0.263 - 0.280)**	**0.252 (0.244 - 0.263)**	0.320 (0.311 - 0.329)
random-composed	0.265 (0.257 - 0.274)	0.246 (0.239 - 0.257)	0.319 (0.310 - 0.328)
pretrained	0.267 (0.258 - 0.276)	0.251 (0.241 - 0.262)	0.326 (0.317 - 0.335)
pretrained-composed	0.252 (0.244 - 0.262)	0.234 (0.224 - 0.243)	0.315 (0.306 - 0.324)

Table 11: Results for terminal label prediction, using different node embedding initialisations for the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ model and reporting weighted-average scores across labels.

5. Discussion

This study demonstrate superior performance of models using a node-path, rather than an atomic, representation of the terminal labels. This validates the hypothesis that exploiting hierarchically-structured medical knowledge — in this case, the ICD-9-CM tree — to learn shared representations of ancestral nodes produces better representations of diagnoses. We explore the reasons for this in Section 5.3. No single document representation method produced superior performance in all models.

5.1. Probabilistic predictions

Figure 6 shows the probability assigned to the most-probable terminal label for each test document, as predicted by the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model. The empirical distribution of the probabilities is continuous and smooth. Labels predicted with high confidence are likely to be correct, and labels predicted with low confidence are likely to be incorrect. These properties correspond with human intuitions regarding probability, and demonstrate that the probabilities predicted by the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model are useful in addition to its label predictions. For example, were the model used in a clinical coding recommender system, a threshold on the probabilities could be set to determine when codes are suggested to the end user.

We inspected test records (10 records, 0.09%) where the terminal $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model made its most-confident incorrect predictions. In 2 records, the 'true' label was obviously erroneous and the model had predicted the correct label. For example, the model predicted the label 'single liveborn, born in hospital, delivered without mention of caesarean section' for the HoPI containing the text

> Baby... born... on the day of admission after the mother presented there in preterm labour, which rapidly progressed to spontaneous vaginal delivery.

The erroneous 'true' label was 'single liveborn, born before admission to hospital'. This suggests that the model has potential applications as a error checker for the work of human coders.

In 4 records, the model predicted a label which was plausible, but which was not designated as the primary label by the human coders contributing to MIMIC-III. For example, from a HoPI describing a patient with an intracerebral haemorrhage, the model predicted the label 'intracerebral haemorrhage' rather than the true label 'unspecified intracranial haemorrhage'. This suggests that some of the apparent errors made by our model are in fact due to arbitrary labelling decisions in the setting of a degenerate ontology.

In 2 further records, the model predicted a plausible label for the HoPI document which was contradicted by the unabridged discharge summary. For example, the model predicted the label 'mitral valve disorders' for the HoPI document containing the text

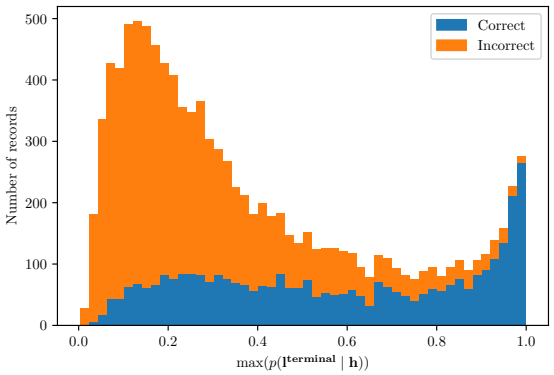

Figure 6: Histogram of the largest predicted probability $\max\big(p(\mathbf{l}^{\mathbf{terminal}} \mid \mathbf{h})\big)$ by the terminal $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model for each test record. Separate strata are shown for correct and incorrect predictions.

> ... known mitral valve prolapse diagnosed... ten years ago... patient came to preadmission testing... for elective mitral valve

The true label was 'mitral valve insufficiency *and* aortic valve insufficiency'. This suggests that some of the errors made by our models resulted from our dataset adaptation process, which produced some HoPI documents which underspecify their label.

We also inspected test records (10 records, 0.09%) where the terminal $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model made its least-confident predictions. 8 records had completely uninformative HoPI documents, containing no clinical information. A low-confidence prediction is appropriate in this context. The 2 remaining records were made very challenging by containing important-but-rare tokens, which were treated as out-of-vocabulary.

5.2. Document representation

The tfidf($\boldsymbol{d}$)-atomic models provided a competitive baseline. Manual inspection of predictions from these models suggests that TF-IDF succeeds by explicitly representing keywords which are associated with a specific diagnosis. However, prediction errors also resulted from this keyword approach. For example, the terminal tfidf($\boldsymbol{d}$)-atomic model predicts the label 'Human immunodeficiency virus' for the document containing the text

> ... past medical history significant for HIV and AIDS defining illnesses including PCP in the past... [presented to hospital with] abdominal pain and found to have elevated amylase and lipase...

The true label, 'Acute pancreatitis', which describes the reason for the patient's *current* presentation to hospital, was correctly predicted by the terminal $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic model.

Representing the HoPI documents as $\bar{\boldsymbol{x}}$ precludes representation of keywords, as it is unlikely that information about individual words is well preserved in the mean word embedding of a long document. $\bar{\boldsymbol{x}}$ also retains some of the disadvantages of TF-IDF, in that it fails to represent word order. It is notable that $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$ did not outperform TF-IDF in all models, especially given the recent dominance of neural models in fields such as machine translation. This is likely to result from the relatively-small size of our dataset, which constrains size of parameters such as number of hidden units, limiting expressive power. Our choice of word embeddings may have contrained the performance of the neural models: the documents in English Wikipedia, PubMed and PubMed Central are stylistically divergent from the text notes in MIMIC-III, and embeddings derived from the former may require further training before application to the latter. The long text sequences in this dataset also make neural representation challenging; the problem of vanishing gradients necessitated using the mean GRU output as the text representation, which is likely to smooth over some of the discriminative features of individual outputs.

5.3. Label representation

The performance of the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ models on the higher levels of labels in ICD-9-CM does not significantly degrade with the truncated-terminal strategy. This suggests that these models often learn to predict the correct ancestral node path even where they predict the incorrect terminal label — a task which is never explicitly optimised for in the truncated-terminal strategy. In contrast, the performance of the tfidf($\boldsymbol{d}$)-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic models on the higher levels of labels in ICD-9-CM *does* significantly degrade with the truncated-terminal strategy. These results suggest that our strategy for enforcing an implicit representation of the ICD-9-CM hierarchy in our models (by of representing the labels as $\bar{\boldsymbol{z}}$ or $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$) is successful.

Figure 7 demonstrates how test-fold accuracy of the terminal $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ models depends on the rarity of labels during training. Unsurprisingly, both models perform better on labels encountered frequently during training than those encountered rarely. However, the majority of the *relative improvement* in accuracy of the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model versus the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic model is evident when testing is limited to rarer labels. This supports the hypothesis that the shared nodal embeddings in $\bar{\boldsymbol{z}}$ and $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ allows better representation of rare labels in particular. In intuitive terms: when a model is shown a patient with a rare disease (e.g. pneumonia caused by an unusual pathogen) it performs better if it can access more-general existing knowledge (e.g. about respiratory diseases, and respiratory infections in particular) to help make the diagnosis. Conversely, when a model is shown a patient with a common disease it already has sufficient specific knowledge to make a good diagnosis, so using more-general knowledge is less helpful.

The similar performance of the $\bar{\boldsymbol{z}}$ and the $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ label representations is surprising, given that the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$ document representation easily outper-

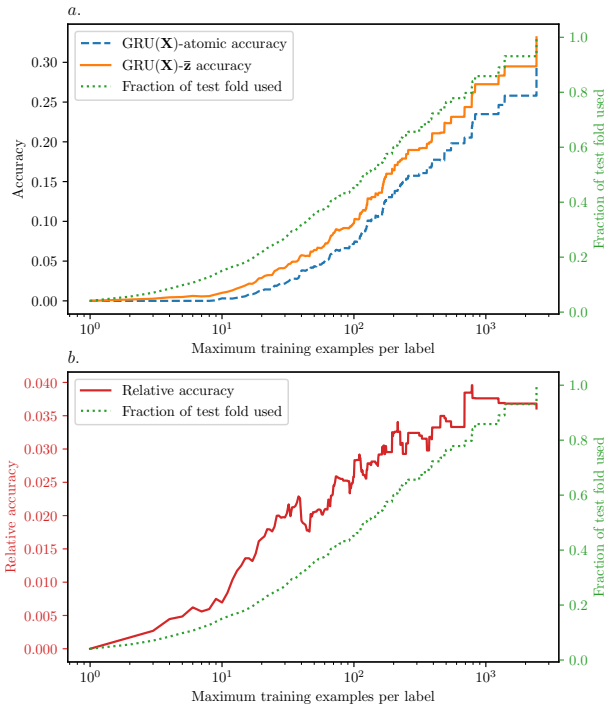

Figure 7: Plot a shows accuracy of the terminal $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-atomic and $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ models on the test fold. Plot b shows relative accuracy of these same models, i.e. $\text{accuracy}\big(\overleftrightarrow{\text{GRU}}(\boldsymbol{X})\text{-}\bar{\boldsymbol{z}}\big) - \text{accuracy}\big(\overleftrightarrow{\text{GRU}}(\boldsymbol{X})\text{-atomic}\big)$. At the left of the plots, only labels which occur rarely during model training are used to calculate the accuracy scores. At the extreme right of the plots, all labels in the test fold are used to calculate the accuracy scores.

formed $\bar{\boldsymbol{x}}$. One possible explanation is that multiplying $\bar{\boldsymbol{z}}$ with a document representation results in the mean of the per-node scores for that document. $\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ transforms its constituent node embeddings more than $\bar{\boldsymbol{z}}$ and so is unlikely to preserve per-node scoring. This also suggests that, in contrast to the tokens in a document, nodes in a path through ICD-9-CM may be commutative. The combined GRUs in the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\overrightarrow{\text{GRU}}(\boldsymbol{Z})$ models are highly expressive, and optimising with respect to their multiplied output may not produce stable training. Yet another, simpler explanation is that the $\overleftrightarrow{\text{GRU}}(\boldsymbol{X})$-$\bar{\boldsymbol{z}}$ model learnt a better node embedding matrix during training than the fixed node embeddings provided to the other models.

5.4. Node embedding initialisation

Pretrained nodes embeddings failed to improve model performance in this study. This may be due to the method used to train the embeddings. Whilst temporal co-occurrence is an intuitively appealing basis for learning relationships between codes in different ontologies, it is less relevant in a studies such as ours, which use only a single ontology with a strong pre-existing structure. Co-occurrence is certainly not a reliable indicator of similarity between diseases. For example, essential hypertension and type 2 diabetes mellitus co-occur frequently but have distinct pathophysiologies. Contrary to our results, other studies have demonstrated that relationships learnt from co-occurrence between codes improve performance of models which incorporate knowledge of ontology structure [39].

5.5. Limitations of this study

A 2014 audit of 8990 patient episodes across 10 NHS trusts found a mean error of 8.8% in manual coding of the primary diagnosis [2]. Our best results for terminal label prediction — the equivalent task — show performance well below this. However, it should be noted that our abridgement of each discharge summary as described in Section 3.2 makes terminal label prediction much more challenging, so comparison with the performance of human clinical coders on easier datasets is not straightforward. Our modification of the discharge summaries produced a new dataset, favouring code prediction rather than extraction. A corollary of working with this new dataset is that no previous results exist for us to compare ourselves to. In addition, only a sample of the extracted HoPI documents have been manually inspected, and the remaining records are likely to contain some noise, e.g. very short sequences of tokens, or documents which are irrelevant to their assigned ICD-9-CM code.

This study criticises manual clinical coding for being prone to error, but then adopts a supervised learning approach using manually-assigned clinical codes as the training data. It is expected that, where these errors occur randomly, they will be 'smoothed over' when our models are trained on a large dataset. Where errors occur systematically, they are likely to be reflected in the trained models. However, systemic errors are usually confined to individual institutions, and we hope that training on large datasets sourced from multiple institutions

will ameliorate this problem. Some models in this study were trained using *scikit-learn 0.18.1*, whilst others were trained using *TensorFlow 1.2.0*, which necessitated the use of different optimisers, training schedules and objectives during hyperparameter tuning. Unified training and tuning pipelines, using *TensorFlow 1.2.0* only, would make comparison of model performance more straightforward.

5.6. Future work

A document representation which retains the specificity of TF-IDF and the generalisability of word embeddings is likely to improve performance. A simple way to achieve this is concatenation of the two representations. Given the degraded performance of the atomic models using the truncated-terminal strategy, as compared to the separate-model strategy, improved terminal label prediction may be achieved by a hierarchical ensemble of atomic models. However, our strategy for implicit representation of the ICD-9-CM hierarchy obviates the need for this. It would be instructive to train our existing model architectures on a novel dataset and compare performance to that reported here. The possibility of knowledge transfer between datasets has been demonstrated in other clinical coding studies [31], and merits further exploration.

We aim to release our modification of the MIMIC-III dataset to the wider research community, in order to encourage progress on tasks which predict clinical codes from clinical text. Further curation of the modified dataset is necessary prior to release, and section 5.1 demonstrates that high- and low-confidence predictions by our models could be used to partially automate this process, by identifying noise in the labels and HoPI documents.

Purely data-driven approaches to clinical coding, and more broadly to clinical diagnosis, are hindered by the large number of rare diseases. 'One-shot' learning techniques are likely to improve performance in rare disease classification, but are unlikely to capture exhaustive representations of diseases with protean manifestations, or diseases which present differently in different patient populations. A large body of existing knowledge is contained within medical education and research literature. Furthermore, the extent of the knowledge about a particular disease need not correspond to that disease's incidence. For instance, a single-gene disease may be rare in humans, yet extremely well-characterised by study of genetic model organisms in the laboratory [55].

Incorporating existing knowledge from scientific papers and textbooks offers a potential solution to the problem of rare diseases. This motivates development of novel natural language processing techniques to automate knowledge extraction from these. Defining an objective function that captures the essence of this knowledge-extraction task is challenging. Another important focus for future work is integration of prior knowledge into models in a manner that properly accounts for the strength of evidence this knowledge provides, which depends on the study methodologies used.

5.7. Conclusion

This study demonstrates that hierarchically-structured medical knowledge can be incorporated into statistical models, and produces improved performance during automated clinical coding. This performance improvement results primarily from improved representation of rarer diseases. We also show that recurrent neural networks improve representation of medical text in some settings. Learning good representations of the very rare diseases in clinical coding ontologies from data alone remains challenging, and alternative means of representing these diseases will form a major focus of future work on automated clinical coding.

Authors' contributions

Finneas Catling conceived of the study, and all authors contributed to the study design. Finneas Catling performed the data analysis and drafted the manuscript. Georgios P. Spithourakis and Sebastian Riedel supervised the project and revised the manuscript critically for important intellectual content. All authors approved the final version of the manuscript prior to submission.

Acknowledgements

This research was supported by the Farr Institute of Health Informatics Research and an Allen Distinguished Investigator award.

Statement on conflicts of interest

Declarations of interest: none.

Summary table

What was already known about this topic

- Manual clinical coding is expensive, time-consuming and prone to error
- Automated clinical coding has great potential to save resources, improve oversight of patient care and accelerate research
- Automated coding is made challenging by rare diseases and the idiosyncrasies of clinical text

What this study added to our knowledge

- Hierarchically-structured medical knowledge can be incorporated into statistical models
- Hierarchical representation of diseases improves automated clinical coding, primarily by improving representation of rare diseases
- Recurrent neural networks improve clinical text representation in some settings

References

[1] S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead, K. B. Johnson, Data from clinical notes: a perspective on the tension between structure and flexible documentation, J. Am. Med. Inform. Assoc. 18 (2) (2011) 181–186.

[2] Capita Health and Wellbeing Limited, The quality of clinical coding in the NHS, Tech. rep. (Sep. 2014).

[3] C. W. Cipparone, M. Withiam-Leitch, K. S. Kimminau, C. H. Fox, R. Singh, L. Kahn, Inaccuracy of ICD-9 codes for chronic kidney disease: A study from two practice-based research networks (PBRNs), J. Am. Board Fam. Med. 28 (5) (2015) 678–682.

[4] K. J. O'Malley, K. F. Cook, M. D. Price, K. R. Wildes, J. F. Hurdle, C. M. Ashton, Measuring diagnoses: ICD code accuracy, Health Serv. Res. 40 (5 Pt 2) (2005) 1620–1639.

[5] C. Benesch, D. M. Witter, Jr, A. L. Wilder, P. W. Duncan, G. P. Samsa, D. B. Matchar, Inaccuracy of the international classification of diseases (ICD-9-CM) in identifying the diagnosis of ischemic cerebrovascular disease, Neurology 49 (3) (1997) 660–664.

[6] W.-Q. Wei, P. L. Teixeira, H. Mo, R. M. Cronin, J. L. Warner, J. C. Denny, Combining billing codes, clinical notes, and medications from electronic health records provides superior phenotyping performance, J. Am. Med. Inform. Assoc. 23 (e1) (2016) e20–7.

[7] R. Robertson, L. Wenzel, J. Thompson, A. Charles, Understanding NHS financial pressures: how are they affecting patient care?, Tech. rep., The King's Fund (Mar. 2017).

[8] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, P. Szolovits, Unfolding physiological state: Mortality modelling in intensive care units, KDD 2014 (2014) 75–84.

[9] K. L. Caballero Barajas, R. Akella, Dynamically modeling patient's health state from electronic medical records: A time series approach, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, ACM, New York, NY, USA, 2015, pp. 69–78.

[10] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, B. G. Buchanan, Evaluation of negation phrases in narrative clinical reports, Proc. AMIA Symp. (2001) 105–109.

[11] D. Agniel, N. Benik, K. Borner, N. Brown, D. Halsey, I. Kohane, D. O'Donnell, G. Weber, Healthcare system dynamics (Jun. 2016).

[12] E. Parkin, A paperless NHS: electronic health records, Tech. rep., House of Commons Library (Apr. 2016).

[13] S. Pereira, A. Névéol, P. Massari, M. Joubert, S. Darmoni, Construction of a semi-automated ICD-10 coding help system to optimize medical and economic coding, Stud. Health Technol. Inform. 124 (2006) 845–850.

[14] K. Crammer, M. Dredze, K. Ganchev, P. P. Talukdar, S. Carroll, Automatic code assignment to medical text, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, BioNLP '07, Association for Computational Linguistics, Stroudsburg, PA, USA, 2007, pp. 129–136.

[15] M. Marcus, New trends in natural language processing: statistical natural language processing, Proceedings of the National Academy of Sciences 92 (22) (1995) 10052–10059.

[16] J. Medori, C. Fairon, Machine learning and features selection for semi-automatic ICD-9-CM encoding, in: Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents, Louhi '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 84–89.

[17] I. Goldstein, A. Arzrumtsyan, O. Uzuner, Three approaches to automatic assignment of ICD-9-CM codes to radiology reports, AMIA Annu. Symp. Proc. (2007) 279–283.

[18] W. B. C. Leah Larkey, Automatic assignment of ICD9 codes to discharge summaries, citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.816.

[19] L. V. Lita, S. Yu, R. S. Niculescu, J. Bi, Large scale diagnostic code classification for medical patient records, in: IJCNLP, 2008, pp. 877–882.

[20] J. Patrick, Y. Zhang, Y. Wang, Developing feature types for classifying clinical notes, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, BioNLP '07, Association for Computational Linguistics, Stroudsburg, PA, USA, 2007, pp. 191–192.

[21] P. Nigam, Applying deep learning to ICD-9 multi-label classification from medical recordsAccessed: 2017-7-12.

[22] L. Lefebure, ICD-9 coding of discharge summariesAccessed: 2017-7-12.

[23] D. Arifoğlu, O. Deniz, K. Aleçakır, M. Yöndem, CodeMagic: Semi-Automatic assignment of ICD-10-AM codes to patient records, in: Information Sciences and Systems 2014, Springer, Cham, 2014, pp. 259–268.

[24] J. Brauer, Clinical entity recognition for icd-9 code prediction in clinical discharge summariesAccessed: 2017-7-12.

[25] R. Weegar, A. Casillas, A. D. de Ilarraza, M. Oronoz, A. Pérez, K. Gojenola, The impact of simple feature engineering in multilingual medical NER, ClinicalNLP 2016 (2016) 1.

[26] S. Hochreiter, J. Schmidhuber, Long Short-Term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN Encoder-Decoder for statistical machine translation, arXiv [cs.CL]arXiv:1406.1078.

[28] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, Lille, France, 2015, pp. 2342–2350.

[29] Z. C. Lipton, D. C. Kale, C. Elkan, R. Wetzell, Learning to diagnose with LSTM recurrent neural networks, arXiv [cs.LG]arXiv:1511.03677.

[30] R. Chalapathy, E. Z. Borzeshi, M. Piccardi, Bidirectional LSTM-CRF for clinical concept extraction, arXiv [stat.ML]arXiv:1611.08373.

[31] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, J. Sun, Doctor AI: Predicting clinical events via recurrent neural networks, arXiv [cs.LG].

[32] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, J. Sun, GRAM: Graph-based attention model for healthcare representation learning, arXiv [cs.LG]arXiv:1611.07012.

[33] E. Choi, A. Schuetz, W. F. Stewart, J. Sun, Using recurrent neural network models for early detection of heart failure onset, J. Am. Med. Inform. Assoc. 24 (2) (2017) 361–370.

[34] Y. Zhang, A hierarchical approach to encoding medical concepts for clinical notes, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop, HLT-SRWS '08, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, pp. 67–72.

[35] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, N. Elhadad, Diagnosis code assignment: models and evaluation metrics, J. Am. Med. Inform. Assoc. 21 (2) (2014) 231–237.

[36] M. Subotin, A. R. Davis, A system for predicting ICD-10-PCS codes from electronic health records, Proc BioNLP.

[37] S. Boytcheva, Automatic matching of ICD-10 codes to diagnoses in discharge letters, Proceedings of the Workshop on Biomedical.

[38] W. Ning, M. Yu, R. Zhang, A hierarchical method to automatically encode chinese diagnoses through semantic similarity estimation, BMC Med. Inform. Decis. Mak. 16 (2016) 30.

[39] Z. Che, D. Kale, W. Li, M. T. Bahadori, Y. Liu, Deep computational phenotyping, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, ACM, New York, NY, USA, 2015, pp. 507–516.

[40] scikit-learn developers, 3.3. model evaluation: quantifying the quality of predictions, [Online; accessed 6-August-2017] (2017).
URL http://scikit-learn.org/stable/modules/model_evaluation.html

[41] Z. C. Lipton, C. Elkan, B. Narayanaswamy, Thresholding classifiers to maximize F1 score, arXiv [stat.ML]arXiv:1402.1892.

[42] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, MIMIC-III, a freely accessible critical care database, Sci Data 3 (2016) 160035.

[43] N. C. for Health Statistics, Icd-9-cm official guidelines for coding and reportingAccessed: 2017-7-31.

[44] S. Ayyar, O. Bear Don't Walk IV, Tagging patient notes with ICD-9 codesAccessed: 2017-7-12.

[45] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60.
URL http://www.aclweb.org/anthology/P/P14/P14-5010

[46] M. Schmidt, N. Le Roux, F. Bach, Minimizing finite sums with the stochastic average gradient, arXiv [math.OC]arXiv:1309.2388.

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[48] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, S. Ananiadou, Distributional semantics resources for biomedical text processing, Proceedings of LBM.

[49] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 3111–3119.

[50] Y. Choi, C. Y.-I. Chiu, D. Sontag, Learning Low-Dimensional representations of medical concepts, AMIA Jt Summits Transl Sci Proc 2016 (2016) 41–50.

[51] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[52] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv [cs.LG]arXiv:1412.6980.

[53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL http://tensorflow.org/

[54] Y. Gal, Z. Ghahramani, A theoretically grounded application of dropout in recurrent neural networks, arXiv [stat.ML]arXiv:1512.05287.

[55] H. Chial, Rare genetic disorders: Learning about genetic disease through gene mapping, SNPs, and microarray data, Nature Education 1 (1) (2008) 192.