Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática
2018

Vitor Manuel de Sousa Pereira

Codificação médica ICD-9-CM automatizada de relatórios clínicos de pacientes diabéticos

Automated ICD-9-CM medical coding of diabetic patient's clinical reports

Vitor Manuel de
Sousa Pereira

Codificação médica ICD-9-CM automatizada de relatórios clínicos de pacientes diabéticos

Automated ICD-9-CM medical coding of diabetic patient's clinical reports

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor José Luís Guimarães Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Sérgio Guilherme Aleixo de Matos, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president	Doutora **Ana Maria Perfeito Tomé** Professora Associada, Universidade de Aveiro
vogais / examiners committee	Doutor **Joel Perdiz Arrais** Professor Auxiliar, Faculdade de Ciências e Tecnologia da Universidade de Coimbra
	Doutor **Sérgio Guilherme Aleixo de Matos** Professor Auxiliar, Universidade de Aveiro

agradecimentos / acknowledgements

Ao Prof. José Luís Oliveira e ao Prof. Sérgio Matos. Aos meus pais.

Resumo

A atribuição de códigos ICD-9-CM a relatórios clínicos de pacientes é um processo dispendioso e cansativo, realizado por pessoal médico especializado e com um custo estimado de 25 mil milhões de dólares por ano nos Estados Unidos. É uma constante ambição de investigadores desenvolver um sistema que automatize esta atribuição. No entanto, o problema mantém-se irresoluto dadas as dificuldades inerentes em processar texto clínico não estruturado.

Este problem é aqui formulado como um de aprendizagem supervisionada *multi-label* em que a variável independente é o texto do relatório e a dependente os vários códigos ICD-9-CM atribuídos. São investigadas diferentes variações de dois modelos baseados em redes neurais, o *Bag-of-Tricks* e a Rede Neural Convolucional (RNC). Os modelos são treinados no subconjunto de pacientes diabéticos dos dados MIMIC-III.

Os resultados mostram que uma RNC com três níveis convolucionais em paralelo obtém avaliações F_1 de 44.51% para códigos de cinco dígitos e 51.73% para códigos abreviados de três dígitos. Além disto, é mostrado que a combinação de vários classificadores binários num só, com o método de relevância binária, produz uma melhoria de 7% em relação ao seu equivalente *multi-label*, num problema de classificação limitado aos onze códigos mais comuns nos dados.

Abstract

The assignment of ICD-9-CM codes to patient's clinical reports is a costly and wearing process manually done by medical personnel, estimated to cost about $25 billion per year in the United States. To develop a system that automates this process has been an ambition of researchers but is still an unsolved problem due to the inherent difficulties in processing unstructured clinical text.

This problem is here formulated as a multi-label supervised learning one where the independent variable is the report's text and the dependent the several assigned ICD-9-CM labels. Different variations of two neural network based models, the Bag-of-Tricks and the Convolutional Neural Network (CNN) are investigated. The models are trained on the diabetic patient subset of the freely available MIMIC-III dataset.

The results show that a CNN with three parallel convolutional layers achieves F_1 scores of 44.51% for five digit codes and 51.73% for three digit, rolled up, codes. Additionally, it is shown that joining several binary classifiers, with the binary relevance method, produces an improvement of almost 7% over its multi-labeling equivalent in a restricted classification task of only the eleven most common labels in the dataset.

Table of Contents

List of Figures

List of Tables

Acronyms

BoT Bag-of-Tricks

CNN Convolutional Neural Network

CRF Conditional Random Fields

EHR Electronic health record

ICD-9-CM International Classification of Diseases, 9th Version, Clinical Modification

LSTM Long Short-Term Memory

MIMIC-III Medical Information Mart for Intensive Care III

NER Named-Entity Recognition

PMI Pointwise Mutual Information

POS Part-of-speech

RNN Recurrent Neural Network

SVM Support Vector Machine

UMLS Unified Medical Language System

Chapter 1

Introduction

Over the past decade, significant advancements have been made in data production and collection, as well as in the development of analysis techniques to leverage all this information. Accumulating and processing these voluminous datasets — coined *big data* — is allowing every industry to make better informed, data-driven decisions. Health care is no exception. Medical data was estimated to be growing at the annual rate of 48% and is expected to reach an astonishing 2 314 exabytes by 2020 [1].

Trailing this growth of data, at a much less pronounced rate but nonetheless important, is the increase in national health expenditures. In Portugal, health expenditures have been increasing at least since 2016 [2]; in the United States, they reached 17.8% of the Gross Domestic Product in 2015 [3]. It is undoubtedly in the interest of governments and health institutions to invest in digital health and make the most out of their data, if for nothing else, to reduce costs. Most importantly, researchers have indicated that such digital health ventures, if successful, could prove to be immensely beneficial in enhancing the efficiency of care delivery, especially in regards to patient safety and quality [4, 5].

Increasingly, health data is generated by the patients themselves using devices like sensor enabled smartphones, at-home genetic testing kits and all sorts of wearables. It is, although, in electronic health records, produced by health professionals and generally written in free-flowing text where most information resides. Fully using these records, despite potential benefits, remains an underexplored topic mostly because of the unique challenges processing this type of data presents.

Electronic health records capture the state of a patient across time and are invaluable in diagnosis. Besides this, they play a significant part in the reimbursement process of health institutions by insurance companies. Medical codes, like ICD-9-CM, assigned to patient's reports after treatment serve as a justification for carrying out the said treatment. Failure in correctly assigning these codes is a liability for healthcare institutions: a missing code represents a loss of revenue and an extra one can constitute fraud. Assigning and correcting these codes is done manually by specialized medical personnel, called a medical coders, and is estimated to cost about $25 billion per year in the United States [6].

It is the aim of this thesis to develop a system which automates the assignment of ICD-9-CM codes to clinical reports. This is achieved by developing a statistical classification model which approximates a function parameterizing the relation between a report's textual contents and its assigned codes. The used classifiers are based on artificial neural networks, which have shown to be capable of handling large volumes of relatively messy data [7].

The classifier is developed on the diabetic patient subset of the MIMIC-III dataset. By constraining the system to this subset, its diseases, terminology and patterns, it is made less versatile but the lesser variability sets it up to produce more accurate results.

Chapter 2 explores the challenges in processing free-flowing clinical text and the methods researchers have developed to meet this challenge. **Chapter 3** takes a close look at the MIMIC-III dataset and presents the used neural network models as well as their results. **Chapter 4** briefly summarizes the accomplished and presents some suggestions for further work.

Chapter 2

Literature Review

This chapter will serve as an introduction to clinical text mining and its state-of-the-art techniques.

2.1 Text Mining

As the volume of data stored in electronic health records (EHR) keeps increasing, manual analysis of this data, done by professionals, to extract relevant information becomes a challenging task. As such, a focus point for scientists and information technology professionals has been to develop methods which ease and automate this kind of analysis.

Text mining is the field of data mining which concerns itself with the extraction of valuable information from unstructured sources. This process mostly works by finding interesting patterns and trends in text.

2.1.1 Clinical Text

Most research in biomedical text mining focuses on extracting information from scientific literature. Two compelling reasons can justify this focus. First, large swathes of data, in the form of scientific articles, are available to researchers who have access to journals such as *Cell* and *Nature*. Even more data is accessible to everyone through preprint archival projects such as arXiv and bioRxiv. Second, authors aim for accurate, concise, clear and objective language in writing their articles [8]. This, together with the fact that they follow a predictable structure, — "Introduction", "Prior Work", "Methods", "Results", "Conclusion" — results in texts which are readier for mining.

Clinical text mining offers no such advantages. In the United States, hospitals host patient's health records in local servers running proprietary software from different providers that are largely incompatible, preventing institutions from exchanging patient data among themselves [9]. This makes large amounts of patient data inaccessible to medical practitioners and researchers alike. In Portugal, a state funded EHR manager, the "Registo de Saúde Electrónico", was introduced in 2010 but is underutilized as public and private hospitals and clinics choose to keep running their already installed systems making the situation very much alike to that of the US [10].

Furthermore, patient EHR data is highly private so before being made available to researchers it needs to be anonymized by the institution that owns the data. Legislation like

the Health Insurance Portability and Accountability Act, in the US, and the General Data Protection Regulation, in the EU, regulate how the data is to be anonymized and make sure serious violations result in serious fines. In practice, this means that, even if data anonymization is automated, manual compliance checking, by legal professionals, is usually required. This is yet one more disicentive for institutions to make the data available.

The structure and content of clinical text is also not neatly defined like in scientific literature. Meystre et al. sum up the challenges in mining clinical text when compared to biomedical scientific literature [11].

First, a single patient's record has several types of report. These can be the patient's description, his and his family's medical history, remarks made by the physician during an examination, etc. The length of the reports also varies greatly e.g. writing a couple of vital signs during a routing check-up vs. the patient's full medical history during the first consultation. Related to length is the structure of the reports. EHR software is usually form-based ("symptoms", "suggested causes", "suggested treatment", …) which could result in thematically well defined short sentences for each of the form's fields. However, the clinician might opt to ignore the form's structure and use the free-form data entry field, if it is available, or, worst case, misuse one of the other form entries, if it is not, to introduce the report's data.

The clinician also appropriates his writing style and language depending on if the text is going to be read by other practitioners. A report, written by a radiologist, which accompanies an x-ray is composed for clear communication. In contrast, notes written by a physician during a consultation are not. These can be ungrammatical, ignoring capitalization and punctuation rules, and composed of short, telegraphic phrases where shorthand abounds — homonyms, acronyms, abbreviations and local dialectal shorthand phrases. Furthermore, the shorthand can be overloaded i.e. the same shorthand unit takes a different meaning depending on surrounding context. According to the 2015AA version of UMLS, the abbreviation "RA" can stand for 25 meanings as distant as "rheumatoid arthritis", "radioactive" and "ragweed antigen" [12]. Liu et al. show that 33% of the time acronyms in UMLS are highly ambiguous even in context [13]. Lexical, morphological and spelling errors also proliferate in clinical text. Ruch and Gaudinat report that about up to one in every five sentences in a corpus of discharge summaries, surgical reports and laboratory results has a spelling error [14]. It is not uncommon to find the acronyms themselves misspelled [11]. Text can also be interrupted by values copy-pasted from laboratory results, vital signs or drug dosages which can confuse standard text mining algorithms like sentence segmentation.

All these characteristics make clinical text particularly challenging for information extraction tasks.

2.1.2 Clinical Text in Portuguese

Text mining clinical text in Portuguese presents yet additional challenges. First and foremost, much less research work has been done in text mining in Portuguese than in English. Second, Portuguese is a morphologically very rich language. While nouns and adjectives have 4 forms (two genders, either masculine or feminine, and two numbers, either singular or plural), verbs have 66 different forms (two numbers, three persons and five modes each with a variable number of tenses) [15].

A recently instituted change added more variability to written Portuguese. The "Acordo Ortográfico da Língua Portuguesa de 1990" is an international treaty finalized in 1990, ratified by the Portuguese parliament in 2008 and its use made mandatory in 2015 which introduced

several orthographic changes into the Portuguese language. These changes can be summarized as such:

- Capitalization changes such as months and weekdays being written in lowercase and forms of address with optional capitalization ("doutor" vs. "Doutor").
- Removal of "mute" consonants (i.e. when they are written but not pronounced). For example, "acção" becomes "ação".
- Graphic accent usage. Words which were previously accentuated now are not. E.g. "pára" vs. "para" and "pêlo" vs. "pelo".
- Significant changes in regards to the usage of the hyphen. Some words like "pára-quedas" lose the hyphen (and the graphic accent) and become "paraquedas". With other words like "cor-de-rosa" and "cor de rosa" there does not seem to be a consensus[1].

The alterations to the language were not put into action without contest and several high profile dissenters [16]. The result of these changes is that some people choose to write according to the treaty, some choose to ignore it and some freely mix pre and post-treaty orthography.

Nevertheless, some pioneering researchers have done work in extracting information from clinical text written in Portuguese. Pereira uses EHR from children with epilepsy to develop a system which aids doctors in choosing the course of treatment by recommending medications and procedures [17]. Castro uses machine learning text mining techniques to classify patient's allergy records according to SNOMED CT terminology [18]. Ferreira et al. develop MedInX, an information extraction system which accurately maps free text discharge reports onto a structured representation using natural language processing [19].

2.1.3 Overview of Text Mining

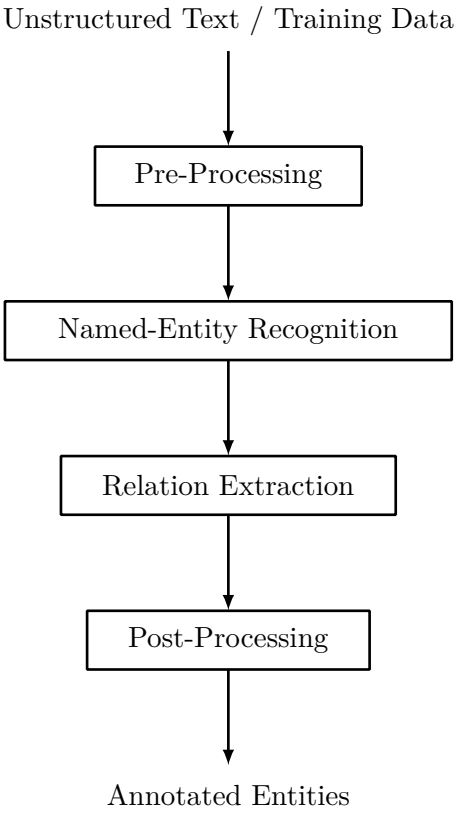
Figure 2.1 illustrates two variants of conceptual clinical text mining systems. Campos et al. offer in [20] a description of the illustrated steps.

Pre-processing is the first stage of the pipeline. The input **unstructured text** is split into sentences which are further split into meaningful units called tokens. The software which does this kind of processing is called a tokenizer. Choosing a good tokenization method is critical as all further processing will be executed on the tokens. An example used with biomedical information is SPECIALIST NLP [2].

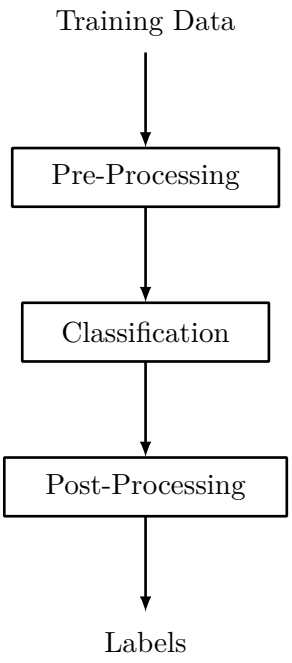
Following this, there is a need to normalize the tokens, reducing them to a canonical form. *Stemming* replaces inflected or derived words with their word stem — a form to which suffixes can be attached. "Children", "childish" and "childless" will be reduced to the stem form "child". A more complex approach is *lemmatization*. Stemmers might output words which have no actual meaning. For example "argues" and "argument" become "argu". Lemmatizers use part-of-speech information to find the token's root form — its lemma. In the previous example, the lemma would be "argue". POLARIS is a lexical database which can be used to generate lemmas for words in Portuguese [21].

In English, words like "the", "be" "this", "that", etc. prove to be non-informative during further processing. Other languages have equivalent non-informative words. These can be

[1] https://ciberduvidas.iscte-iul.pt/consultorio/perguntas/acerca-de-cor-de-rosa-e-cor-de-laranja/29892
[2] https://lexsrv3.nlm.nih.gov/Specialist/Home/index.html



(a) Relation extraction pipeline.



(b) Classification pipeline.

Figure 2.1: Conceptual overviews of the steps taken by a text mining system. They do not represent the actual data flow of an hypothetical implementation just the steps it might take.

discarded during a process called stopword removal where a pre-made list of these words is used.

The next step, **named-entity recognition**, is the basic building block for most information extraction tasks. Here, biomedical entities deemed pertinent are identified and classified. These might be genes, chemicals, drugs and diseases. To do this, rule-based, dictionary-based and, more recently, machine learning-based methods are used. When the last are used, it might also be necessary to do feature engineering and the existence of a large corpus of pre-annotated **training data**. Alternatively, in a **classification** system, a statistical model is developed to assign each of the datum in the pre-annotated training corpus to a pre-defined category. A practical example is the automated assignment of UMLS codes to clinical reports.

After the named-entities are identified, there is still the need to establish relations among them and other words in the sentence. This is done in the **relation extraction** phase. An invaluable tool in this step are syntactic parsers which output dependencies between all words in a sentence. This information can be combined with the entities obtained in the previous step and input into a machine learning model.

As these methods are not perfect, **post-processing** their results may be necessary. Uninformative, incorrect and unwanted annotations can be discarded. Correct annotations can be manually extended and be made more precise [22]. Labels need to be mapped to their actual value from whatever intermediate representation might have been used.

Named-entity recognition, relation extraction and classification are the main focus of this thesis and are discussed, at length, in following sections.

2.2 Named-Entity Recognition

Named-entity recognition (NER) in the biomedical domain is considered particularly challenging compared to other domains due to inconsistencies in entity naming. A certain drug can be referred to by its chemical name, generic name, brand name, a standard or nonstandard abbreviation.

NER can be said to be a two step task: *term extraction* and *term classification*. In term extraction, terms which might be of interest in text are determined, in term classification, a label is attributed to these terms.

Most recent work in biomedical NER focuses on the genetics subdomain. These tools focus on extracting proteins, genes and mutations from text. Systems such as BANNER [23] and BABELOMICS [24] focus on gene extraction while tmVar [25] takes a step further and focuses on gene mutations. This interest can possibly be explained by the usefulness of this kind of tool in oncology [26].

Following the trend in general NER, biomedical NER systems have evolved from handcrafted rule-based systems, to dictionary-based, to machine learning-based ones [22, 27]. Handcrafted rule-based systems have largely fallen out of favor as machine learning-based ones have proven more accurate [22]. Dictionary-based systems are still popular due to their usefulness in detecting domain specific terminology and, in recent projects, are used in tandem with machine learning models in hybrid systems.

2.2.1 Dictionaries

In a dictionary-based approach, a correspondence is made between a word in the text and another word in a lexical database of relevant words. This is an efficient way of detecting

domain specific terminology which is abundant in clinical text.

Several of these lexical databases are publicly available. A significant example is the Unified Medical Language System (UMLS) [28]. UMLS aims to aggregate the variety of ways in which concepts are expressed and collect them in a central Metathesaurus of controlled vocabularies in the biomedical domain. A second module of UMLS, the Semantic Network, assigns categories to the concepts from the Metathesaurus and establishes links among them. This gives its users an easy way to establish ontological relations among biomedical entities, like cause-effect. A tool, MetaMap[3], was developed to extract Metathesaurus concepts from text.

Other databases like The Pharmacogenetics Knowledge Base (PharmGKB) [29] and The Online Mendelian Inheritance in Man (OMIM) [30] also exist.

Attaining an exact match between a word found in text and another in the database might not always be possible. Approximate string matching methods can be used to establish a correspondence between similar words. A common technique is to use the Levenshtein distance algorithm to obtain a similarity metric between words [31].

As these databases can have millions of entries (UMLS as of 2017 has 13 897 048 concepts) doing a linear search over them is a lingering process. The storage of these entries in a data structure such as the Suffix Tree, which enable efficient string searching, is a practical requirement [32].

A fair amount of post-processing might be necessary when dictionaries are used. A word in the text can match several entities in the database which mandates a disambiguation process. A method for disambiguation is to collect other entities which occur in the same context as the ambiguous one and check their ontological proximity.

Katona and Farkas, in SZTE-NLP use MetaMap and the Illinois NER Tagger[4] to extract UMLS entities from text [33]. Rocktäschel et al. developed ChemSpot which uses dictionaries combined with a machine learning method to extract names, drugs, abbreviations, molecular formulas and chemical entities from text [34]. Usié et al. developed a similar system which combines dictionary matching with regular-expressions [35].

Dictionary-based approaches have some drawbacks. The previously mentioned case of drug naming illustrates one. Matching non-standard abbreviations is challenging as these are unlikely to be present in the database so a term which would otherwise be extracted during manual analysis is ignored. Ideally, lexical databases should receive periodic updates but, even so, unknown relevant terms which were meanwhile introduced can show up in text and they will also be ignored. Machine learning techniques aim to ameliorate these shortcomings.

2.2.2 Machine Learning and Feature Engineering

The type of machine learning technique which is dominant in NER is *supervised learning* [27]. Supervised learning algorithms involve observing several examples, called training data, of a random vector $\boldsymbol{x}$ and an associated label vector $\boldsymbol{y}$ then learning to predict $\boldsymbol{y}$ from $\boldsymbol{x}$, usually by estimating the probability distribution $p(y|\boldsymbol{x})$ [36]. This task is usually computationally expensive and requires the existence of an hand annotated corpus for use as training data.

The examples are the input of the machine learning model. These examples are a collection of features. *Feature engineering* is the step where the data scientist chooses these features.

[3] https://metamap.nlm.nih.gov/

[4] https://github.com/CogComp/cogcomp-nlp/tree/master/ner

The simplest of these is the pre-processed token itself. Developing other features is time-consuming and required domain expertise as they are task specific. Other commonly used features in clinical text mining are:

- The label attributed to the previous token. This is a simple way to get a model to develop correlation.
- Bag-of-words features which can be used to obtain the frequency of the token and other words which co-occur with it.
- Orthographic features like whether the word is capitalized or all capital and contains digits or special characters, like an hyphen, are also used.
- Morphological features that deal with the token's structure. It is determined if the token contains certain prefixes ("*anti*-inflammatory") and suffixes ("edemat*ous*"). Character *n*-grams can be developed to perform a similar function to prefixes and suffixes but in the middle of the tokens.
- Grammatical and syntactic features. The most common is the assignment of a part-of-speech (POS) class to the token such as a noun, verb, determiner or adjective, if it is singular or plural, etc. Shallow parsing, also called chunking, can use these POS classifications and assign the tokens to groups (such as groups of nouns).
- Supplementary dictionary-based features. Using a dictionary of well-known entities in combination with the machine learning features has been shown to be advantageous [37].
- Semantic categories of words, defined in ontology such as UMLS, are also useful. For example a correspondence between "allergies" and "pathologic function", enables the model to develop semantic relations.

Besides being chosen, features can also be learned. Word embedding is a feature learning technique where words in the corpus are assigned to vectors and words which share a context are nearer in vector space. The current state-of-the-art word embedder is Facebook's fastText [38].

The described features are merely some of the available to be fed to the model, depending on the task at hand.

If term extraction and classification are not happening in a single step, it is also necessary to settle on a representation which identifies which tokens are part of an entity which should be classified and which are not. The standard representation is the BIO encoding. The beginning token of an entity is identified with a B and following tokens with an I. Tokens which are not part of an entity are identified with an O. A more elaborate encoding, BMEWO, also distinguishes the ending token of a entity with an E, middle tokens with a M and adds a new tag for single token entities, W.

2.2.3 Conditional Random Fields

A supervised learning technique which is used by the great majority of biomedical NER systems [23, 25, 34, 39] are *Conditional Random Fields* (CRFs).

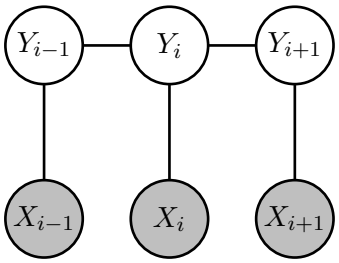


Figure 2.2: A first-order chain-linear CRF. The gray circles are the input words and are not generated by the model.

CRFs are undirected graphical models, introduced by Lafferty et al. for sequence prediction [40]. When the graph is a tree (as is usually the case in sequence prediction) the joint probability distribution over the label sequence $\boldsymbol{y}$ given input $\boldsymbol{x}$ takes the form of Equation 2.1

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})} \exp \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_j f_j(y_{i-1}, y_i, \boldsymbol{x}, i) \tag{2.1}$$

where the outer sum is over each example i in the sentence $\boldsymbol{x}$ and the inner one over each feature function f_j. A feature function takes as input the previous label, the current label, the examples $\boldsymbol{x}$ and the position i of the current example[5]. The output of the feature function can be a real number but is usually binary.

It is the data scientist's job to provide these feature functions as they are highly dependent on context. An example of these can be seen in Equation 2.2 to model token sequences like "abdominal pain", "chest pain", …

$$f_j(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} 1, & \text{if } y_{i-1} = \text{ANATOMICAL and } x_i = \text{“pain”} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

$Z(\boldsymbol{x})$ is defined in Equation 2.3 and is used to normalize the exponentiation.

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \boldsymbol{Y}} \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_j f_j(y_{i-1}, y_i, \boldsymbol{x}, i) \tag{2.3}$$

An estimation for the parameters $\boldsymbol{\lambda}$ is obtained by maximizing the conditional log-likelihood objective function (Equation 2.4) using an iterative algorithm such as gradient ascent.

$$\lambda_{ML} = \arg\max_{\lambda} \sum_{i=1}^{m} \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\lambda}) \tag{2.4}$$

Abacha and Zweigenbaum use a CRF with lexical and morphosyntactic features combined with semantic feature obtained using MetaMap [41].

In a recent project, ChemSpot, a CRF is combined with a dictionary-based method to form a sort of ensemble learning method to annotate natural language text. The input text is ran through the CRF and the dictionary independently [34]. The output of the system is

[5]By restricting the feature function's input to a single previous label, a first-order linear-chain CRF is implemented but this need not be the case.

the merging of all annotations produced by both methods, the CRF being preferred in cases where there is an overlap. Building a similar system can be greatly simplified by using the Neji framework [42].

2.2.4 Support Vector Machines

Another supervised learning used by several NER systems [43, 44] is the *Support Vector Machine* (SVM).

An SVM solves binary classification problems by tracing an hyperplane which maximizes the distance between examples which belong to different classes (positive and negative labels). This hyperplane is nothing more than a linear function $\boldsymbol{w}^T\boldsymbol{x} + b$ which implies that the data must be linearly separable to obtain reasonable performance.

The key innovation in SVMs was introduced by Boser et al. when it was suggested to apply the kernel trick so as to make it possible to learn a nonlinear decision boundary by tracing the hyperplane in a transformed feature space [45].

The kernel trick is the observation that the linear function used by SVM's can be re-written as

$$\boldsymbol{w}^T\boldsymbol{x} + b = b + \sum_{i=1}^{m} \alpha_i \boldsymbol{x}^T\boldsymbol{x}^{(i)} \tag{2.5}$$

where $\boldsymbol{\alpha}$ is the vector of coefficients. Writing the hyperplane this way enables the replacement of the dot product with a kernel function $k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x}^{(i)})$ where $\phi(\boldsymbol{x})$ is a feature function. This gives us the general form of SVM classifiers in Equation 2.6. A commonly used kernel is the Gaussian kernel.

$$f(\boldsymbol{x}) = b + \sum_{i=1}^{m} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) \tag{2.6}$$

Such form enables computing the classifier using convex optimization techniques which are guaranteed to converge efficiently [36].

There is still the issue of SVMs being binary classifiers and sequence labeling being a multi-class problem. This can be solved by combining several binary SVMs in a one-vs-all or pairwise fashions [43]. In one-vs-all, the i-th SVM is trained with examples of the i-th class being the positive label and all other classes the negative. In pairwise, all classes are combined in pairs. The first class of the pair becomes the positive label and the second class the negative. All SVMs vote on a label and the label with the most votes is chosen as the final output.

2.2.5 Recurrent Neural Networks

As previously mentioned, CRFs and SVMs require feature functions to be provided by the data scientist. Other kinds of machine learning methods also usually require significant feature engineering. This is time consuming and the scientist requires domain knowledge. Even so, the selected features might prove not to be the most effective.

A collection of machine learning methods called *deep learning* make use of a structure inspired by biological neural networks called an *artificial neural network*. The main component of these networks is the *artificial neuron*. The output of these neurons is defined as $\boldsymbol{y} =$

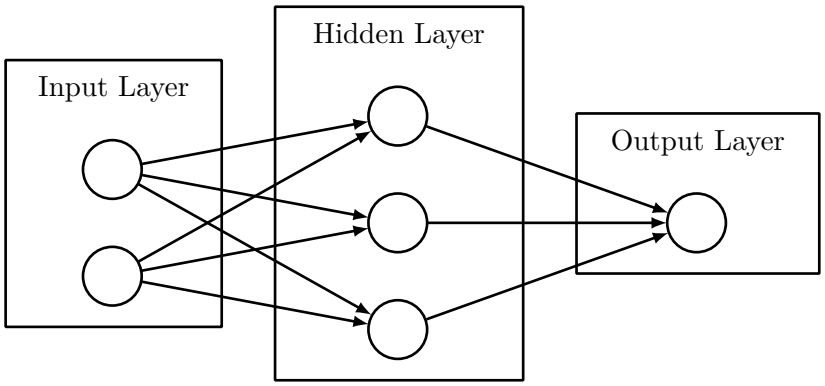


Figure 2.3: A graph representation of a 3-layer feedforward densely connected neural network. The vertices are the neurons and the edges the connections established among them. Each connection has a vector of weights $\boldsymbol{w}$ and biases $\boldsymbol{b}$ which are not shared with other neurons.

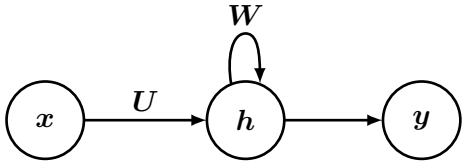


Figure 2.4: A recurrent neural network. The edge flowing from $\boldsymbol{h}$ to itself indicates weight sharing: weights from the previous input example are passed on to the next one.

$\sigma(\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b})$ where σ is a non-linear activation function such as the rectified linear unit $\sigma(\boldsymbol{x}) = \max(0, \boldsymbol{x})$. Figure 2.3 shows a simple neural network.

In supervised learning tasks, these neural networks do something akin to principal component analysis by assigning larger weights to the neurons whose output improves performance therefore disposing of the need to do heavy manual feature engineering.

Recurrent Neural Networks (RNNs) are a family of neural networks which specialize in processing sequences of values $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}$. These neural networks use parameter sharing to apply the model to sequences of different lengths and generalize across them. Sharing the weights across all words in the sentence enables the network to learn the rules of the text only once and with fewer parameters [36]. Figure 2.4 shows an RNN. The connections from the input and from the hidden layers of the previous word are parameterized by $\boldsymbol{U}$ and $\boldsymbol{W}$ respectively which are optimized during the training phase.

By combining two RNNs, one which moves forward from the start of the sequence and another which moves backwards from the end, a bidirectional RNN is obtained. This structure enables the joint RNN to combined dependencies from previous and following inputs. Such is the strategy employed by Mavromatis for disease extraction [46].

Most modern RNN implementations use the Long Short-Term Memory (LSTM) cell [47] to compute a neuron's output and in the biomedical domain this is no exception [48, 49, 50]. These cells are probably one of the most significant development in RNNs. See Olah [51] and Goodfellow et al. [36] for in-depth analysis.

RNNs can also be combined with other machine learning methods. A powerful combination is the LSTM-CRF which is a bidirectional LSTM RNN with a CRF as the output layer [48, 49]. This hybrid system besides capturing dependencies through the hidden layers also

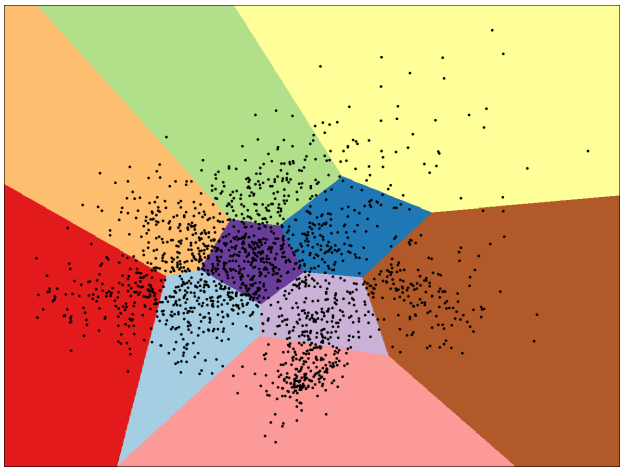


Figure 2.5: Generic clustering example. The dots could represent the tokens and the different colors the labels to which they were assigned e.g. “symptom”, “drug”, “disease”, etc.

captures label dependencies at the CRF layer. Unanue et al. show that pre-trained word embedding can improve the performance of this hybrid system [50].

A significant disadvantage of RNNs is that they experience very slow training. Because the output for an element in a sequence, besides depending on its input, is also a function of the previous element's hidden layer, $y_t = f(x_t, h_{t-1})$, RNNs can not be parallelized like other neural networks.

2.2.6 Unsupervised Learning

An already stated disadvantage of supervised learning methods is that they require a pre-annotated (usually hand-annotated) corpus to use during the model's training stage. *Unsupervised learning* techniques experience features but not the supervision signal i.e. by observing examples from the random vector $\boldsymbol{x}$ they learn the probability distribution $p(\boldsymbol{x})$ [36]. These methods are not common in NER, much less in biomedical NER. A reason for this is that they are not as versatile as supervised learning techniques because they can not do term extraction. They can, although, still do term classification.

Clustering, as used by Elsner et al., is one such approach [52]. A representation of this technique is illustrated in Figure 2.5.

The aim of clustering is to partition the examples $\boldsymbol{x}$ into k clusters according to some similarity metric using an algorithm like spherical k-means. Lin and Demner-Fushman use a variant of clustering called hierarchical agglomerative clustering where each entity starts with its own cluster that can then merge with another entity's cluster when a common UMLS hypernym is shared among them [53]. Apirin and ibuprofen would be merged into the anti-inflammatory cluster which could be merged into the drug cluster.

Zhang and Elhadad use a customized approach where they collect “seed terms” from UMLS and assign them to classes such as “disease” [54]. A shallow syntactic parser is used to analyze the sentences. Each word is attributed a part-of-speech label and wanted labels — nouns — are kept. The similarity between these nouns and the seed terms is computed, through frequency of co-occurrence, to get their classification.

2.3 Relation Extraction

After using NER to extract relevant entities from text, the next step is to establish relations among those entities and other words in the sentence. For example, a connecting between a symptom and its cause as well as how confident the physician is in the association and the suggested treatment can be extracted.

A simple, rule-based, approach is to collect certain cue words and search for them in the vicinity of the named-entity using regular expressions. E.g., to determine the severity of a symptom, adjectives for intensity can be searched for [55]. This does not, however, make the system resilient to new vocabulary.

Following the trend in NER, more elaborate techniques use machine learning methods so much of the detailed in the NER section can now be applied to relation extraction.

Pathak et al. combine a dictionary with a CRF which uses bag-of-words, stem value and other feature functions [56]. SVMs prove to be common in the most effective relation extraction systems [57].

Other techniques are discussed in the following subsections.

2.3.1 Pointwise Mutual Information

In linguistics, a collocation is a sequence of words which co-occur somewhat frequently. Computing these collocations is therefore a way to extraction relations between terms based on their linguistic properties such as adjectives being followed by nouns.

Pointwise Mutual Information (PMI) is a way of measuring the association between the terms and is defined as

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} \tag{2.7}$$

where the probabilities $p(x)$ and $p(x|y)$ can be approximated by counting the occurrences and co-occurrences of the words in the text.

PMI measures how much the probability of a certain co-occurrence $p(x, y)$ differs from what would be expected based on the probability of individual events $p(x)$ and $p(y)$, assuming independence. Good candidates for collocation have high PMI.

Brujin et al. use data from the Medical Literature and Retrieval System Online (MEDLINE) to calculate how related two concepts are by finding collocations using PMI [58].

2.3.2 Deep Parsers

Shallow parsing was briefly mentioned in Section 2.2.2 as way of collecting tokens classified with the same part-of-speech into groups.

A more powerful form of syntactic parsing, *deep parsing*, besides assigning parts-of-speech to each word, also builds a dependency tree with the relations among the words in the sentence (Figure 2.6). This reveals the sentence's semantic structure and, input into a machine learning model, functions as a powerful semantic feature. The tool to build these dependency trees is called a dependency parser. PALAVRAS is a significant example of a syntactic analyzer for the Portuguese language [59]. The current state-of-the-art dependency parser is Google's SyntaxNet [6].

[6] https://github.com/tensorflow/models/tree/master/research/syntaxnet

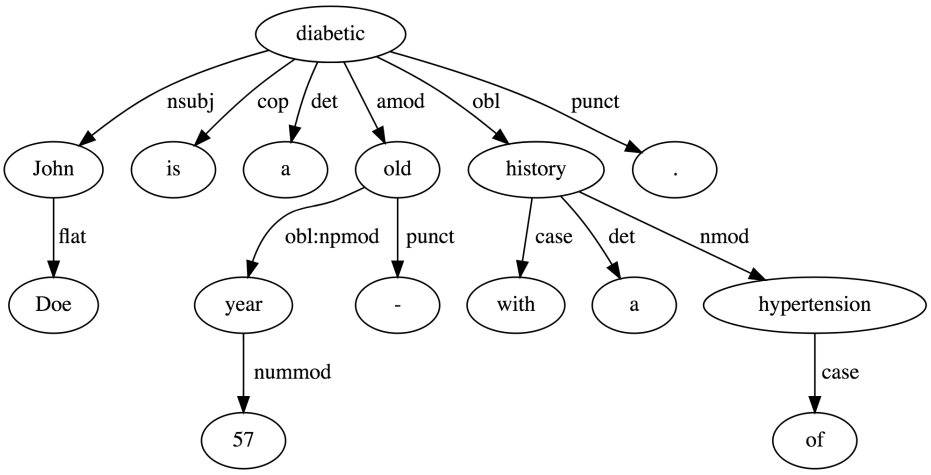


Figure 2.6: Dependency tree obtained using Google's SyntaxNet for "John Doe is a 57 year-old diabetic with a history of hypertension".

The winning entry for the Task 2 of SemVer's 2014 Task 14 [60], UTH-CCB [61], uses a dependency tree, obtained using the Stanford Parser[7], as one of the features input to an SVM classifier.

2.3.3 Convolutional Neural Networks

Sahu et al. [62] in a novel approach use another type of artificial neural network called a *Convolutional Neural Network* (CNN). CNNs specialize in modeling grid-like data such as images but they can be used with any kind of data such as text. What sets them apart from other neural networks is the usage of the *convolution* operation in at least one of their layers. Sahu et al. define the convolution as

$$s(t) = \boldsymbol{w} \cdot \boldsymbol{x}^{i:i+c-1} + \boldsymbol{b} \tag{2.8}$$

where the notation $\boldsymbol{x}^{i:i+j}$ represents the concatenation of feature vectors. This is similar to computing a linear activation in a regular neural network but with shared weights and biases, much like RNNs.

The difference is that the output of the convolution operation is a function of a small number of neighboring members of the input while in RNNs the output is a function of the previous members of the output. This makes RNNs better at generalizing for textual data but CNNs offset this disadvantage by experiencing much faster training due to their reduced number of parameters.

2.4 Classification

Yet another task in clinical text mining is statistical classification. This can be generally defined as the problem of assigning one or more labels to an observation. A classical example of this type of task is classifying e-mails as "spam" or "not spam".

[7] https://nlp.stanford.edu/software/lex-parser.shtml

A plethora of useful information can be obtained by classifying clinical text. Rajkomar et al. use three different models — a neural network, a LSTM and a time-aware neural network — to predict inpatient mortality, unplanned readmission, length-of-stay and diagnoses using EHR data from 216,221 patients from two different hospitals [7]. Garla et al. train an an SVM and a Laplacian SVM on a corpus of CT, MRI and Ultrasound reports labeled for the presence of malignant liver lesions that require follow up [63].

Medical coding is another task where text classification can be useful. Health facilities send patient data to insurance companies, for billing purposes, encoded in pre-established taxonomies like ICD, CPT and HCPCS. Assigning these codes to patient reports is done manually by medical personnel and automating this menial task is an ambition of researchers [64, 65].

Much like relation extraction, classification is mostly done with the already introduced machine learning techniques. Two of the mentioned publications do, although, introduce new concepts.

2.4.1 Semi-Supervised Learning

Semi-supervised learning is a mixture of supervised and unsupervised learning where an algorithm, besides using examples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l \in \boldsymbol{X}$ with associated labels $y_1, \ldots, y_l \in \boldsymbol{X}$, is also input unlabeled examples $\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+u} \in \boldsymbol{X}$. To usefully use the unlabeled data, it is required to make an assumption about the structure of the underlying data distribution. Chapelle et al. lists four assumptions: the smoothness assumption, the cluster assumption, the manifold assumption and transduction [66].

The Laplacian SVM, used by Garla et al., adopts the manifold assumption and adapts the SVM to semi-supervised learning by defining a kernel function which uses unlabeled data [67]:

$$k'(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) - k_{\boldsymbol{x}}(I + MK)^{-1} M k_{\boldsymbol{x}^{(i)}} \tag{2.9}$$

where M is based on the Laplacian matrix L:

$$M = \frac{\gamma I}{\gamma A} L^p \tag{2.10}$$

and $k_{\boldsymbol{x}}$ is a vector of kernel evaluations of $\boldsymbol{x}$ against all labeled and unlabeled training data.

The parameters γI and γA specify a trade-off between regularization and deformation, weakening or strengthening the manifold assumption by shaping the feature space to reflect the manifold's geometry.

2.4.2 Attention

The already presented LSTM cells used in RNNs aim to solve the problem of capturing long-term dependencies in sequences of various lengths. The recently developed mechanism of "attention" solves the same problem, although with much simpler modeling, by establishing a more direct dependence between states of the model at different points in time.

Rajkomar et al. learn attribution logits $\boldsymbol{\alpha}$, convert them to into weights B_i using the softmax function and then use them to recalculate the weighed embeddings $E = \sum_j B_j E_j$.

The calculation of attribution logits is done by Rajkomal et al. in a very problem specific way but Raffel and Ellis show a more general form:

$$\alpha_i = a(h_t) \tag{2.11}$$

where a is a learnable function which can be thought of as computing the scalar importance of h_t [68].

As the authors suggest, this mechanism permits the modeling of long term dependencies, much like LSTM cells in RNNs, but with an added advantage. Assuming that $h_t = f(x_t)$, no dependency exists between hidden states which means that the model does not need to be trained sequentially, like an RNN, but can be trained in parallel like any other feed-forward neural network. A disadvantage which it does have, although, is that it requires yet another training process, this time for $\boldsymbol{\alpha}$. Additionally, Raffel and Ellis conclude that temporal order is discarded by the model, making the method inappropriate for some tasks.

Chapter 3

Classification of Clinical Reports

This chapter analyses the dataset, applied methods and their results.

3.1 Dataset

3.1.1 MIMIC-III

The Medical Information Mart for Intensive Care III (MIMIC-III) is a large, freely available database of anonymized health-related data associated with patients of the intensive care unit (ICU) of the Beth Israel Deaconess Medical Center between 2001 and 2012 [69].

MIMIC-III is composed of several tables which host several types of data from admissions and transfers to patient information and test results. For the purposes of this thesis only two are used: the notes table (`NOTEEVENTS`) which contains reports input by clinicians and nurses into the EHR at each patient admission and the diagnosis table (`DIAGNOSIS_ICD`) which contains ICD-9-CM codes, associated with each admission, generated for billing purposes at the end of each patient's hospital stay. Furthermore, diagnosis entries, and their associated notes, are only kept for patients who have been diagnosed with diabetes. This filtering adds up to a total of 406 190 reports which will be further reduced as indicated ahead in Section 3.1.3.

Before being made available to researchers, protected health information is removed through a thoroughly evaluated deidentification system based on dictionary look-ups and patterns matching with regular expressions [70]. Text which was changed during the deidentification process is surrounded with the pattern "[** **]".

Each report in the notes table is assigned to one of fifteen categories:

- ECG
- Nutrition
- Radiology
- Nursing
- Discharge Summary
- Rehab Services
- Consult
- Echo
- Physician
- General
- Respiratory
- Social Work
- Nursing/Other
- Pharmacy
- Case Management

These categories are heterogeneous in their structure and language, much like already described in Section 2.1.1, which makes developing a technique for text classification that

```
[**2157-8-16**] 9:51 AM
UNILAT LOWER EXT VEINS LEFT    Clip # [**Clip Number (Radiology) 106670**]
Reason: Evaluate for DVT
Admitting Diagnosis: BILATERAL ADNEXAL MASS
_____________________________________________________________________
[**Hospital 2**] MEDICAL CONDITION:
75 year old woman with ovarian cancer, post op with left calf tenderness
REASON FOR THIS EXAMINATION:
Evaluate for DVT
_____________________________________________________________________
                              FINAL REPORT
@@@@@@@@@@@@@@@ is a revision of a previously signed report @@@@@@@@@@@@@@@@

INDICATION: 75-year-old woman with ovarian cancer, postop with left calf
tenderness.  Evaluate for DVT.

COMPARISON: None.

FINDINGS: The common femoral, femoral and popliteal veins on the left
demonstrate normal compressibility, augmentation, flow and respiratory
variability. The vessels at the trifurcation are also compressible and
appear patent. The posterior tibial amd peroneal veins are visualized and
are compressible.

IMPRESSION: No evidence of DVT within the left leg.
```

Figure 3.1: Radiology report example. The pattern "[** **]" indicates the original text was replaced with an anonymized equivalent which keeps the text coherent. Note the lack of rich text formatting, usage of medical slang — "UNILAT", "postop" — and a misspelling — "amd".

generalizes to all of them particularly challenging. Figures 3.1 and 3.2, slightly formatted but otherwise copied verbatim, illustrate the differences between two categories of report.

3.1.2 ICD-9-CM

The disease labels associated with each report use the ICD-9-CM taxonomy. The International Statistical Classification of Diseases and Related Health Problems (ICD) is a disease classification system maintained by the World Health Organization. Although, at the time of this thesis' publication, the most recent version is ICD-10, MIMIC-III uses the ninth revision of the codes with the *Clinical Modification* (CM). These codes are composed of three primary

```
Sinus rhythm. A-V conduction delay. Left atrial abnormality. Atrial and
ventricular ectopy.  Low limb lead voltage. Right bundle-branch block.
Compared to the previous tracing of [**2183-4-17**] atrial and ventricular
ectopy are recorded. The rate has increased. Otherwise, no diagnostic
interim change.
TRACING #3
```

Figure 3.2: ECG (Electrocardiography) report example. Reports which belong to this category are usually short.

digits and either one or two optional digits which convey extra information. The *Clinical Modification* augments ICD-9 with additional morbidity detail and with codes for surgical, diagnostic and therapeutic procedures.

For the classification task, two different representations of the codes are considered. A **regular** one with codes which range from three to five digits and a **rolled up** version where all codes are limited to their first three (primary) digits.

The used MIMIC-III subset has 4006 different regular codes and 779 different rolled up codes. Each of the codes is used in labeling at least one report. Procedure codes are not used. The dataset has a very low label cardinality — defined as the average number of codes per report — of 17 for regular and 15 for rolled up labels. Label densities — define as the number of codes per report divided by the number of unique codes, averaged over the full dataset — are 0.0042 and 0.0198.

Figure 3.3 shows the number of occurrences for grouped ICD-9-CM codes. The group with the second most occurrences is "Endocrine, Nutritional and Metabolic Diseases, and Immunity Disorders", which includes diabetes. This is preceded by "Diseases of the Circulatory System", such as vascular and heart diseases. The overrepresentation of these very serious diseases, surpassing even diabetes, is attributed to the fact that the data was obtained from an ICU.

For each report, each code also has an associated sequence number which loosely ranks the relevancy of the codes to the patient's stay.

Given the very high dimension of both the inputs — reports — and outputs — ICD-9 labels, whether rolled or not — this classification task can be considered to be a part of the eXtreme Multi-label Learning discipline.

3.1.3 Preprocessing

Preprocessing starts with a simple four step process which transforms each report into a sequence of tokens:

1. Punctuation characters, except for the apostrophe, are converted to whitespace
2. Digits are replaced by the character 'd'
3. All characters are converted to lowercase
4. The report is split at whitespace characters

As already noticed in Section 2.1.1, clinical text is prodigal of misspellings that can interfere with a classifier's correctness. Baumel et al. suggest a simple vocabulary reduction step as a partial remedy [65]. Tokens that occur less than five times in the whole corpus are eliminated. These eliminated tokens are then mapped to their most similar token of the ones which were kept. The most similar token is defined as the one which minimizes the Levenshtein distance string metric:

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + \alpha \end{cases} & \text{otherwise} \end{cases} \tag{3.1}$$

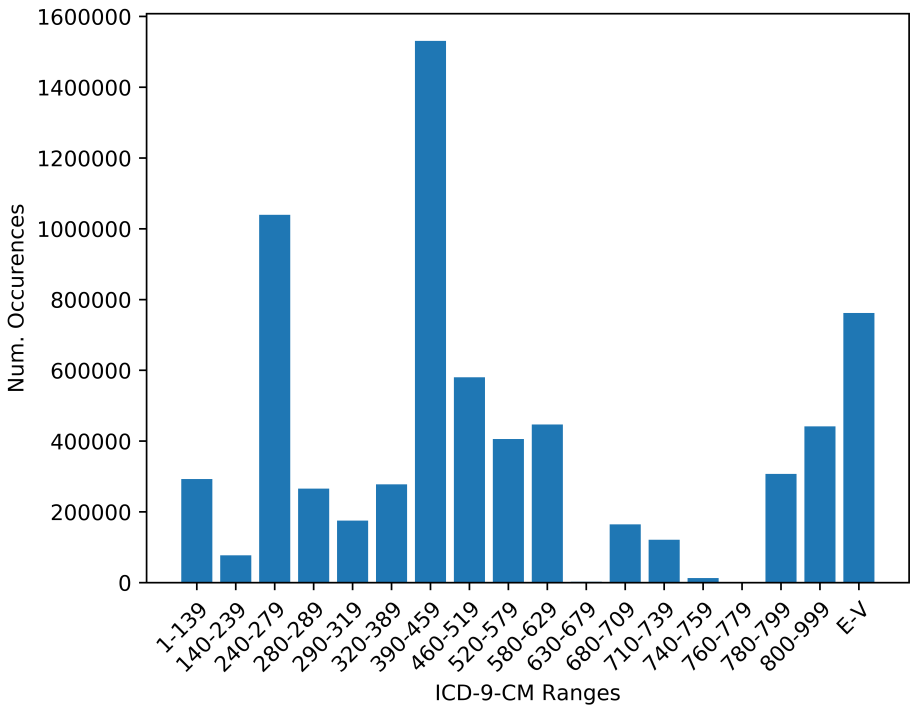


Figure 3.3: Number of occurrences for standardly grouped ICD-9-CM codes on the MIMIC-III subset. A reference for the grouping can be obtained from http://healthpolicy.ucla.edu/Documents/Spotlight/IDC9%20Groupings.pdf (visited on 30/03/2018).

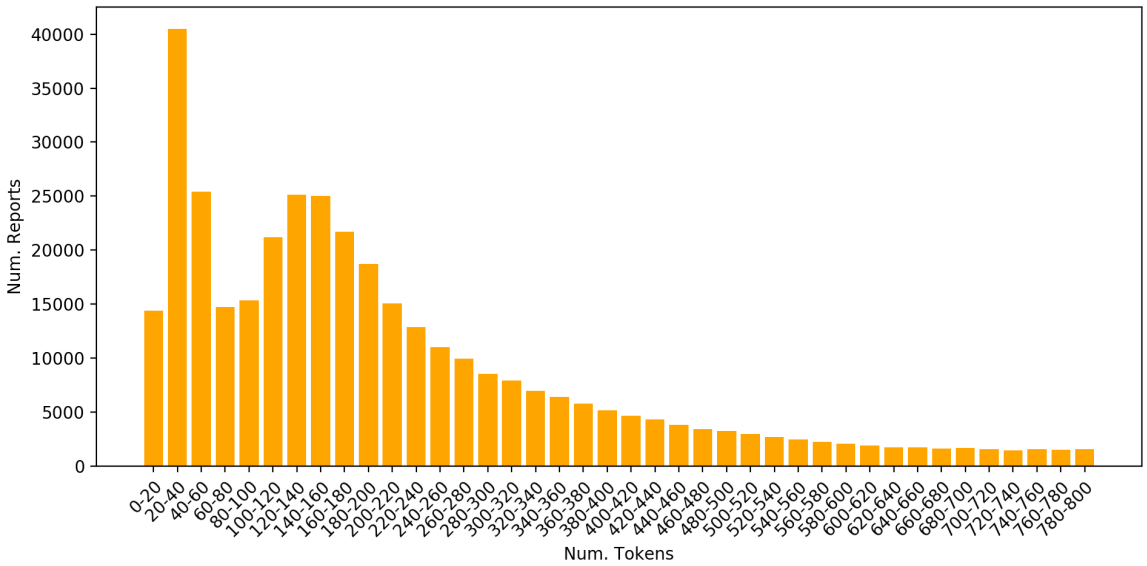


Figure 3.4: Report lengths in tokens for reports with sizes which range from 0 to 800. Most reports have from 20 to 40 tokens. Reports longer than 800 tokens are progressively fewer (not shown).

where a and b are the tokens, of length $|a|$ and $|b|$, and α is a binary variable which is 1 if $a_i = b_j$ and 0 otherwise. The function is applied as $\text{lev}_{a,b}(|a|, |b|)$. The *python-Levenshtein* package was used[1].

The assumption for this step is that tokens which occur infrequently are misspellings of other tokens which occur more frequently. This does not always hold true but good results were observed, with the added advantage that the vocabulary was reduced from 162 784 to 53 229 unique tokens.

Most text classification tasks opt to use word embeddings to represent tokens. *Word2vec*'s skip-gram neural network model was input the corpus and used to generate three-hundredth dimensions word embeddings. The concrete implementation of word2vec used was the one from the *gensim* library[2]. Pre-trained word embeddings were also experimented with but produced worse results due, most likely, to not containing enough domain specific terminology.

Further preprocessing was necessary before inputting the data into the models. The *Keras* library, used to develop the models, expects fixed size inputs. As indicated, the lengths of the reports vary greatly, from 1 to 8513 tokens and, due to memory limitations, it would be infeasible to resize all sequences to the maximum length.

Figure 3.4 shows the number of reports for some length ranges. The majority of the reports has a relatively short length. It was chosen to keep only reports which have more than 9 and less than 2200 tokens. By doing this, 98.38% of the reports are kept unaltered and can fit in working memory while 6 567 are eliminated. The average length for the reports after this step is 309 tokens.

Table 3.1 summarizes the preprocessed dataset.

[1] https://github.com/ztane/python-Levenshtein/
[2] https://radimrehurek.com/gensim/models/word2vec.html

Num. of used records	399 623
Num. of regular labels	4 006
Num. of rolled up labels	779
Regular label cardinality	16.99
Rolled up label cardinality	15.39
Regular label density	0.0042
Rolled up label density	0.0198
Num. of unique tokens	53 229
Avg. num. of tokens per report	309.06

Table 3.1: Preprocessed dataset description.

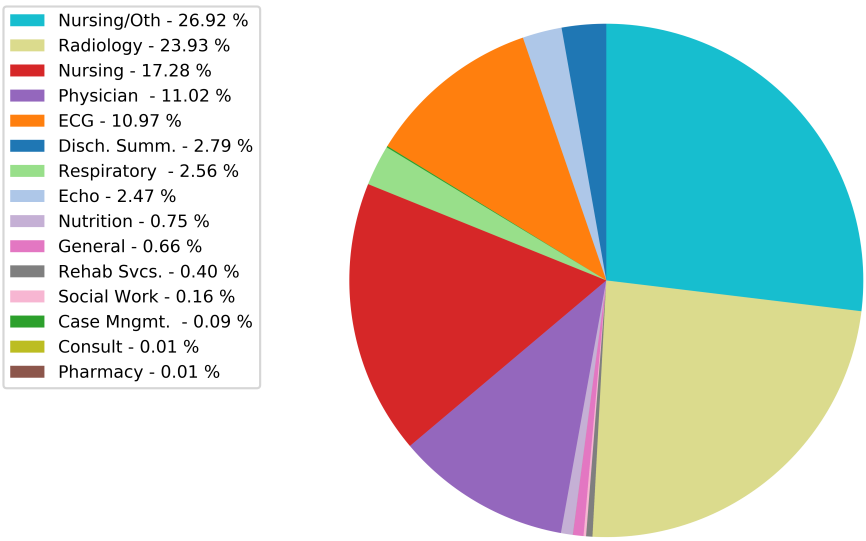


Figure 3.5: Distribution of reports per category. The low number of reports for categories like Case Management and Consult is attributed to the fact that such event are rare in an ICU context.

3.1.4 Additional Analysis

Several statistics of the dataset were already calculated mostly to enable an informed preprocessing of the dataset. Here, further analysis is presented.

The distribution of these reports, per category, can be seen in Figure 3.5. Once again, great variability is observed. Nursing and Nursing/Other reports make up 44.19% of the total reports. This proliferation of nursing reports is in accordance to the guidelines of the American Association of Critical-Care Nurses which mandates periodic patient assessment from every few minutes for extremely unstable patient to at least every 2 to 4 hours for very stable patients [71]. Nursing reports are followed by all kinds of medical exams (Echo, ECG, Respiratory and Radiology) which make up 39.94% of the total reports. Noticeably, the Radiology category makes up almost one quarter of the total reports. As the single most commonly requested radiographic examination in ICUs is the portable chest radiograph, this is not surprising [72].

ICD-9-CM Code	Description	%Reports
250	Diabetes mellitus	100.00%
427	Cardiac dysrhythmias	46.64%
518	Other diseases of lung	45.53%
428	Heart failure	45.19%
584	Acute kidney failure	44.27%
401	Essential hypertension	43.47%
276	Disorders of fluid electrolyte and acid-base balance	38.74%
414	Other forms of chronic ischemic heart disease	37.40%
285	Other and unspecified anemias	35.10%
272	Disorders of lipoid metabolism	34.52%
585	Chronic kidney disease	27.05%

Table 3.2: The eleven most common rolled up ICD-9-CM codes.

Category											
Discharge Summary	250	401	414	272	427	428	276	285	584	V45	V58
Echo	250	414	428	427	401	272	584	276	285	518	585
ECG	250	428	427	414	401	272	584	276	285	518	403
Case Management	250	518	584	276	427	285	428	585	038	995	272
Respiratory	250	518	584	427	276	285	428	401	995	272	038
Nursing	250	584	518	427	276	428	285	272	401	414	585
Physician	250	518	584	276	427	428	285	272	401	585	414
Nutrition	250	518	584	276	427	285	428	401	272	995	585
Pharmacy	250	584	518	276	995	427	038	285	272	401	599
General	250	518	584	427	276	428	585	285	272	401	414
Social Work	250	584	276	518	427	285	428	585	401	414	272
Rehab Services	250	427	518	584	401	272	414	276	428	285	E87
Consult	250	427	518	428	285	276	585	401	584	403	E93
Radiology	250	401	427	584	518	428	276	414	272	285	V58
Nursing/Other	250	518	428	427	401	584	414	038	285	276	995

Table 3.3: The eleven most common rolled up ICD-9-CM codes per category. The codes are sorted left to right where the leftmost label — 250 — is the most common.

Tables 3.2 and 3.3 show the eleven most common labels globally (for the whole dataset) and per category. Unsurprisingly, the code 250 — diabetes mellitus — is always the most common. Whether comparing the global labels with the categories or the categories among themselves, it can be verified that, even though their arrangement changes, most of the ICD-9-CM codes are shared. If outliers were to be named they would be Case Management, Pharmacy and Consult with a mere two codes which do not show up in the global ones. This breaks the heterogeneity pattern, observed up to this point, by showing that, at least regarding the most common codes, the categories present some similarities.

Another pattern that shows up, related to this homogeneity, is that the Nursing, Physician, General and Social Work categories, not only share the exact same labels, but also a very similar disposition.

Table 3.2 also shows the percentage of reports which are classified with a each of the eleven most common codes. It can be seen that most of these codes have a somewhat balanced distribution with the most imbalanced being 585 at 27.05% positive targets.

3.2 Methods

3.2.1 Setup

To better understand how this problem can be solved and determine the most advantageous form of classification, two different methods are experimented with:

- Multi-label classification of the reports. Here, the classifier must learn to predict *all* ICD-9-CM labels assigned to each report. Separate classifiers are trained to predict labels in their regular and rolled up forms. For both forms, two variations of these models are considered. One where the model uses just the tokenized reports and another where, besides using the reports, it also takes as input the category of the report.

- Binary classification of reports for each of the most common rolled up labels (see Table 3.2). Here, each classifier is trained to predict *a single* label, contrasting with the previous task where a classifier predicts multiple labels. Both baseline and category augmented classifiers are also used.

Developing these classifiers mandates adopting a validation strategy to assess how well they generalize to an independent dataset. A popular approach with neural network based models is to split the dataset into training, test and validation subsets. The training set contains the examples the classifier uses to optimize its parameters; the validation set is used to measure the training effectiveness and to support regularization techniques, like early stopping; and the test set is used to provide an unbiased evaluation of the trained model. A disadvantage of this technique is that, by picking a fixed test set, the evaluation could be biased i.e. if a another test set was then picked it might produce significantly different results.

An alternative is to use *k-fold cross-validation*. In this technique, the dataset is split into k more or less equal-sized subsamples. A single subsample is used for testing while the remaining $k - 1$ are used for training and validating. This process is then repeated k times with each of the subsamples being used exactly once for testing. This means that k different results are obtained which can then be averaged to produce a final estimation, making this method a more resilient approach than the previously presented. Its disadvantage, that is non trivial when used with neural networks which typically experience long training times, is

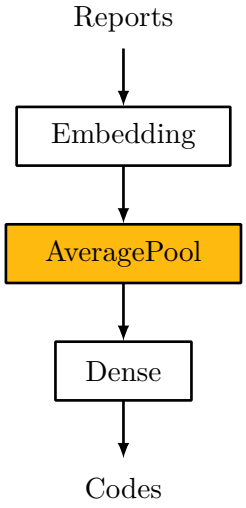


Figure 3.6: Baseline Bag-of-Tricks.

that the whole training process must also happen k times. A k of 5 was chosen and 10% of the $k-1$ subsamples are used for validation.

The library and classifier architectures used impose some constraints on how the data is to be represented. Reports are encoded as a sequence of indexes which map to tokens in the known vocabulary; categories are encoded as a 1-hot vector where the category assigned to the report is positive; labels are encoded as a k-hot vector, where all labels assigned to the report are positive.

As already mentioned, the Keras library was used to develop the classifiers [73]. Keras provides a high-level interface to the lower-level neural network library Tensorflow [74], greatly easing development.

3.2.2 Bag of Tricks

The first method is a simple linear model based on Armand et al.'s Bag of Tricks (BoT) [75]. BoT uses a very similar architecture to Mikolov et al.'s Continuous Bag-of-Words [76] but is not limited to words and can classify arbitrary type labels.

The baseline version of this classifier can be seen in Figure 3.6. The embedding layer transforms the report's tokens into high dimension word embeddings. These embeddings are then averaged into a fixed dimension vector before being fed into a dense/fully-connected layer. The size of the dense layer varies according to the task: 4 006 for regular multi-label classification, 779 for rolled up label classification and 1 for binary classification.

The significant departure from Armand et al. is in the usage of the sigmoid activation function (Equation 3.2) at the dense layer's outputs instead of the softmax function. Softmax squashes the layer's inputs into a probability distribution which would not be appropriate in a multi-label scenario as labels are not mutually exclusive e.g. both "diabetes" and "heart arrhythmia" can be assigned to a single report. The sigmoid function, instead, squashes each neuron's input into a value in the range $[0-1]$ so that each neuron outputs a Bernoulli probability distribution independent of all other neurons. To determine if the neuron's label

is assigned, a fixed threshold of 0.5 is used.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

The classifier is trained by minimizing the binary/binomial cross-entropy loss function as is usually the case in multi-label and binary classification scenarios:

$$L(y, p) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ij} \log p_{ij} \tag{3.3}$$

where i indexes the report and j the label, y is the correct label and p the prediction.

The *Adam* optimizer [77] is used and examples are fed into the classifier in batches of size 32. The classifier uses no other regularization except for early stopping with a minimum delta of 0.0001 and a patience of 2 epochs. Choosing the Adam optimizer, instead of the more traditional Stochastic Gradient Descent, enables the adaptive tuning of the classifiers hyperparameters during training [36].

The main advantage of this method, when compared to the CNN, is that, due to its simplicity, it experiences very fast training. However, its disadvantage is significant: linear classifiers do not share parameters among features and labels. This is particularly consequential in textual data as it means word order is ignored and multi-word expressions are not established.

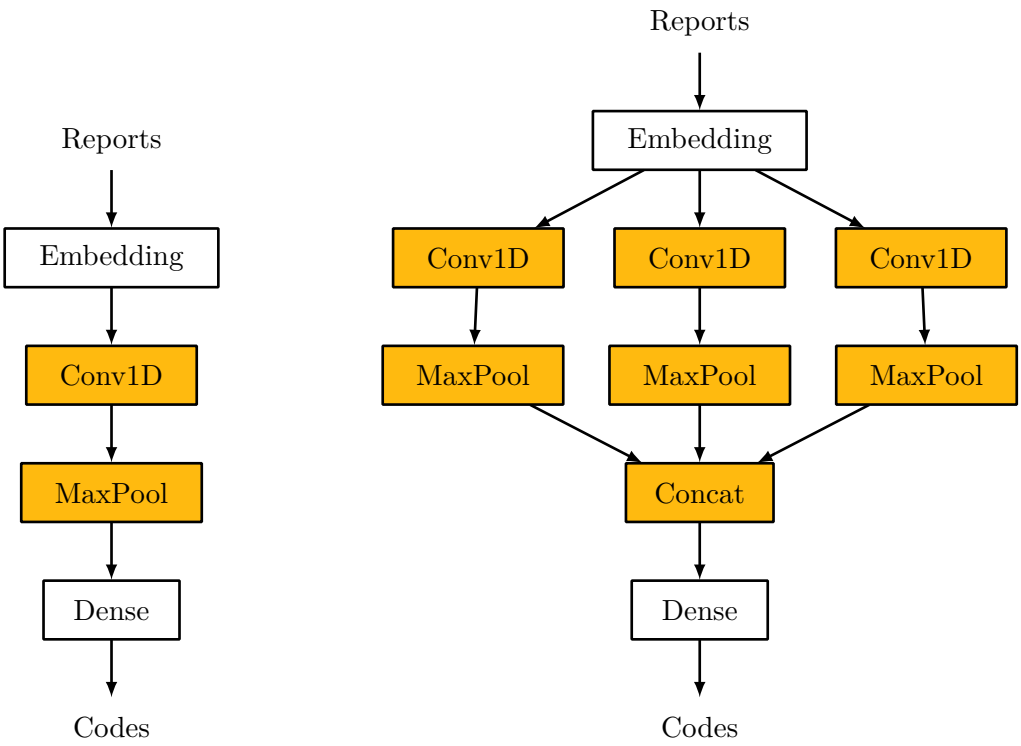
3.2.3 Convolutional Neural Network

The Convolutional Neural Network (CNN), by using parameter sharing, is able to address BoT' main problem. This comes at the cost of significantly slower training.

This classifier's baseline architecture is very similar to BoT' as can be seen in Figure 3.7a. Instead of being averaged, the word embeddings are input into a typical CNN three-stage computation. First, a one-dimensional convolution layer with filter size 250 and kernel size 3 outputs a set of linear activations. Next, each linear activation is run through a (non-linear) rectified linear unit. A final max pooling layer helps to make the obtained representation invariant to small translations in the input [36].

GoogLeNet, with its Inception modules, showed that using parallel convolutions with different filter sizes followed by concatenation layer produced state-of-the-art results in image classification [78]. An architecture inspired by these Inception modules, seen in Figure 3.7b, is experiment with. Its filters sizes, from left to right, are 2, 3 and 4. The "concat" operation merely concatenates its input vectors. All other hyperparameters are shared with the BoT classifier.

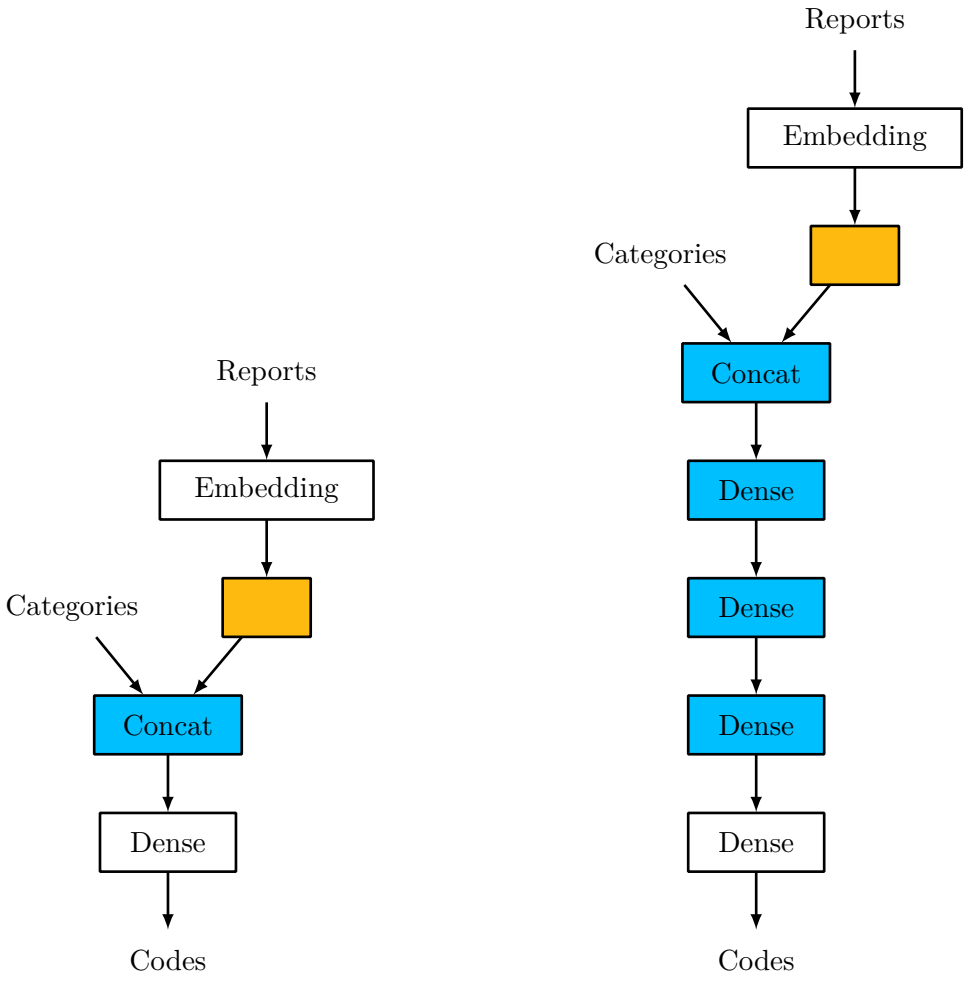
There was yet no mention on how the classifiers will use report category information. Figure 3.8 shows the new architecture which is used with both BoT and CNN models. An additional input is added for the category 1-hot vectors which is then concatenated with the classifier specific layer's output (see yellow color coded layers in Figures 3.6, 3.7a and 3.7b). Four variations of the category augmented architecture were tested, the shallower uses no additional dense layers and the deeper uses three additional ones. All the added dense layers, colored in blue, have output size 64 and a rectified linear unit activation.



(a) Baseline

(b) Three parallel convolutional layers

Figure 3.7: Convolutional Neural Network models.



(a) Shallower architecture.

(b) Deeper architecture.

Figure 3.8: Category augmented model architectures. New layers are colored in blue. The yellow labelless layer is a placeholder for the classifier specific layers.

3.2.4 Evaluation

Before presenting the results, it is necessary to discuss the metrics that are used to evaluate the model's performance.

Evaluation metrics make use of four base definitions, called the confusion matrix, which measure the totality of outcomes:

- **True Positive (TP)**: Positive prediction matches the target.
- **False Positive (FP)**: Positive prediction does not match the target.
- **True Negative (TN)**: Negative prediction matches the target.
- **False Negative (FN)**: Negative prediction does not match the target.

The simplest of evaluation metrics is the *accuracy* which measures how often the model is correct:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

As already determined in Section 3.1.4, the eleven most common codes have a somewhat balanced distribution so this metric is useful in the binary classification task. It is not, although, in the multi-label classification task because label density is very low. A classifier could easily achieve very high accuracy by always predicting negative while completely failing to predict positive targets.

Instead, two other metrics are traditionally used. *Precision*, which measures how often a positive prediction is correct:

$$\text{precision} = \frac{TP}{TP + FP} \tag{3.5}$$

and *recall*, which measures how often positive targets are predicted as positive:

$$\text{recall} = \frac{TP}{TP + FN} \tag{3.6}$$

Another useful metric is the F_1 Score which combines precision and recall, through harmonic averaging, into a single metric:

$$F_1 = \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3.7}$$

3.3 Results

3.3.1 Multi-Label Classification Task

The displayed results were obtained by averaging the results obtained for each of the testing subsamples of the 5-fold cross validation.

Table 3.4 shows the results for the multi-labeling sub-task for both regular and rolled ICD-9-CM codes. The CNN with three parallel convolutional layers achieved best precision, recall and, consequently, F_1 score in both code formats.

Rolled up F_1 scores are an average of 12% higher than regular ones. This is to be expected as, by truncating codes to three digits, label sparsity is reduced and each code has more examples. All CNN based models are better than BoT based ones. The worst performing

Model	Regular			Rolled		
	Precision	Recall	F_1	Precision	Recall	F_1
BoT Baseline	66.25	8.61	15.24	75.91	18.89	30.25
BoT Baseline w/ Cat. 0 Dense	66.50	8.59	15.21	75.82	19.13	30.54
BoT Baseline w/ Cat. 1 Dense	67.99	11.17	19.18	74.72	22.24	34.28
BoT Baseline w/ Cat. 2 Dense	69.27	12.46	21.12	73.67	24.09	36.31
BoT Baseline w/ Cat. 3 Dense	69.03	13.08	21.99	75.02	24.21	36.59
CNN Baseline	73.97	25.88	38.13	77.73	35.13	48.38
CNN Baseline w/ Cat. 0 Dense	74.59	25.53	38.03	77.60	35.34	48.56
CNN Baseline w/ Cat. 1 Dense	71.96	17.66	28.35	76.06	26.50	39.28
CNN Baseline w/ Cat. 2 Dense	73.98	18.10	29.08	75.60	29.19	42.11
CNN Baseline w/ Cat. 3 Dense	73.07	19.62	30.93	76.34	30.02	43.07
CNN 3-Conv1D	**76.07**	**31.46**	**44.51**	**79.82**	**38.26**	**51.73**

Table 3.4: Results for the multi-label classification sub-task.

CNN, CNN Baseline w/ Categories 2 Dense, achieved a 7% higher F_1 score, in the regular codes, when compared to the best performing BoT, BoT Baseline w/ Categories 3 Dense. For rolled up codes, the difference is less significant at a 3% improvement.

Category augmented architectures managed to improve the baseline BoT model's F_1 score by a maximum of 6%, mostly through increased recall. It can be observed that deeper models marginally improved results. This once again confirms the empirical observation that deeper models result in better generalization [36]. Interestingly, not only did category augmentation fail to improve the baseline CNN model, it decreased its F_1 score by a maximum of 9%.

Another observation is that category augmented models with no additional dense layers produced no change when compared to the baseline ones. This can be justified as the model having the category information but not being able to learn how to use it.

Even though CNN based models performed the best, it came at the cost of a remarkable increase in training time, when compared to BoT. The baseline CNN took roughly 5x more time training than the BoT baseline. The CNN with triple convolution was more than twice slower than the baseline CNN at training. This paints a somewhat different picture making the BoT based model more attractive for applications where long training times are infeasible.

As far as the author is aware, no other publication exists where the classification of the MIMIC-III diabetic subset (or any other subset of similar magnitude) is attempted. Other authors opt to limit themselves to a single category of report. Baumel et al. use 52 139 reports belonging to the Discharge Summary category with 6 527 regular and 1 047 rolled up unique labels [65]. Their CNN's F_1 scores are similar to this thesis', which are only 2.59% and 4.26% lower for regular and rolled up labels. Curiously, Baumel et al.'s BoT model (CBOW in the paper) performed much better than this thesis' baseline model with 14.78% and 13.05% higher F_1 scores.

3.3.2 Binary Classification Task

Due to lack of time, it was opted to use a training, test and validation subsets for the binary classification sub-task, instead of 5-fold cross-validation. The allotted percentages for each subset were 70%, 20% and 10% of the full dataset, respectively. It was also chosen, for the same reasons, to train just four models.

		BoT Baseline		BoT Baseline w/ Cats. 3 Dense		
Code	Precision	Recall	F_1	Precision	Recall	F_1
427	67.76	47.69	55.98	72.68	50.04	59.27
518	74.64	59.33	66.11	75.00	67.95	71.30
428	65.83	49.20	56.31	66.00	63.39	64.67
584	70.12	40.53	51.37	67.96	56.26	61.56
401	61.11	22.56	32.96	63.81	39.36	48.69
276	62.58	24.10	34.80	62.62	38.77	47.89
414	74.87	35.71	48.35	73.31	54.55	62.46
285	63.85	11.78	19.89	69.35	24.57	36.28
272	63.11	8.92	15.63	58.97	33.88	43.03
585	73.36	19.93	31.35	79.66	22.67	35.29
Average	67.72	31.98	41.28	68.94	45.14	53.04
		CNN Baseline		CNN Baseline w/ Cats. 3 Dense		
Code	Precision	Recall	F_1	Precision	Recall	F_1
427	70.26	73.31	71.76	72.66	72.09	72.37
518	82.07	66.35	73.38	81.70	70.90	75.92
428	72.09	68.96	70.49	76.96	62.59	69.04
584	69.30	67.32	68.30	77.35	59.97	67.56
401	61.39	58.66	60.00	68.80	60.13	64.18
276	62.15	66.04	64.04	72.11	57.57	64.02
414	76.95	62.16	68.77	79.88	59.59	68.26
285	78.43	17.01	27.96	85.10	37.34	51.90
272	61.23	59.77	60.49	69.16	46.31	55.48
585	79.49	45.23	57.65	79.36	49.64	61.08
Average	71.34	**58.48**	62.28	**76.31**	57.61	**64.98**

Table 3.5: Results for the binary classification sub-task.

	Multi-Label			Binary		
Model	Precision	Recall	F_1	Precision	Recall	F_1
BoT Baseline	68.46	35.29	46.57	68.68	33.81	45.31
BoT Baseline w/ Cats. 3 Dense	68.47	42.85	52.71	68.68	46.94	55.77
CNN Baseline	71.06	50.15	58.80	69.89	**59.94**	64.53
CNN Baseline w/ Cats. 3 Dense	66.78	53.25	59.25	**75.70**	58.80	**66.19**

Table 3.6: Multi-label vs. binary classifier results for the most common labels.

Table 3.5 shows the results for this sub-task. The most common code, 250, is not modeled as it is guaranteed to be assigned to all reports. The category augmented baseline CNN model achieved best average performance with an F_1 score of 65%.

The general results mirror the ones obtained in the multi-labeling sub-task with CNN based models performing the best. Augmenting the baseline BoT model with category information once again proved to be worthwhile increasing the average F_1 score in 11.76% due mostly to a 13.16% increase in recall.

Contrary to what was verified in the multi-labeling sub-task, augmenting the CNN with category information managed to improve its average performance, albeit less significantly, by increasing the F_1 score in 2.7% through an increase of 4.97% in precision. It can be seen, though, that such improvement was not uniform across all of the ten codes with some having lower F_1 score (see 272).

An additional test was made to compare the multi-label and binary classifiers. The test consists of the multi-label classification of the ten most common labels (once again ignoring 250). The trained multi-label classifiers are used as is, except all labels which are not in the most common are ignored. For the binary classifiers, the *binary relevance* method is used [79]. This method simply consists of combining the ten distinct binary models into a single multi-label classifier.

Table 3.6 shows the results for this experiment. Multi-label and binary BoT based models performed similarly with the category augmented binary version achieving a 3% higher F_1 score due to a 4% increase in recall. CNN based models once more were the most accurate. Binary CNN models performed better than multi-label ones with F_1 scores 5.73% higher for the baseline model and 6.94% for the category augmented version. Noticeably, the category augmented CNN binary model shows an almost 9% increase in precision when compared to the multi-label one.

A final observation was made when developing this experiment. It can be seen in the multi-label CNN results of Table 3.6 that the category augmented version performed marginally better than the baseline one with a 0.45% increase in F_1 score. What is surprising about this is that it contradicts what is observed in Table 3.4 where augmenting the CNN baseline with category information produces much worse results. This leads to the conclusion that training multi-label category augmented CNNs on less sparse labels (see Table 3.2) could be fruitful.

Chapter 4

Conclusion

There has been a progressive amassing of patient medical data over the past decade. Accompanying this collection of data is the development of analysis techniques to fully leverage this information and use it to enhance the efficiency of care delivery. In this thesis is developed a statistical classification model which is capable of assigning an arbitrary number of ICD-9-CM codes to clinical reports in a diabetic patient's electronic health records. The ambition in developing a system like this is to eliminate the costly and wearisome process of manually assigning these codes, done by specialized medical personnel, which is currently practiced.

To solve this classification problem, it was formulated as a multi-label supervised learning task to be ran on the diabetic patient subset of the MIMIC-III dataset using artificial neural network based models. F_1 scores of 44.51% and 51.73% were obtained for regular and rolled up labels, respectively. As far as the author knows, no other publications address a similar overarching problem but when compared to other, more localized, experiments, the results appear to be good.

Furthermore, it was explored whether using a collection of binary models, instead of a single multi-label one was a better alternative. This experimentation was conducted on the restricted subset of the ten most common labels. It was shown that the best binary classifiers produced an almost 7% improvement in F_1 score over its multi-label equivalent.

The undertaken research highlights two avenues for further research.

First, the development of new classifiers. Despite the triple convolution classifier being the one with the best results, a category augmented version of it was not trained due to incredibly long training times. Doing so is certainly worthy. Given the improvements deeper architectures showed in the BoT classifier, deeper yet architectures should be experimented with. Besides these, whole new models, like an LSTM based neural network should be tried. All classifiers use early stopping as the single regularization technique. Trying other methods such as the L1 Loss or more sophisticated ones, like Dropout, could improve results at the cost of even longer training times.

Second, the leveraging of more features of the MIMIC-III dataset. While the reports are used, the model does not have any information regarding their temporal relation. Adding this information can enable it to develop associations between diseases. For example, if a report is classified with diabetes, the model can discern that following reports have a higher probability of being classified with diseases which are complications of diabetes like kidney and foot damage. It was also mentioned that the codes associated with each report have a sequence number assigned. Finding a way to factor it into the models and the evaluation

method could be useful in better appropriating them to the task at hand.

Bibliography

[1] *Stanford Medicine 2017 Health Trends Report*. Stanford University School of Medicine, 2017.

[2] Direção-Geral do Orçamento. *Conhecer o Orçamento do Estado*. 2018. URL: https://online.dgo.pt/DadosCidadao/Orcamento_CG.Entrada.aspx (visited on 04/28/2018).

[3] Thomas E Price, Anne Schuchat, and Charles J Rothwell. *Health, United States, 2016*. Centers for Disease Control and Prevention, 2016.

[4] Harlan M Krumholz. "Big data and new knowledge in medicine: the thinking, training, and tools needed for a learning health system". In: *Health Affairs* 33.7 (2014), pp. 1163–1170.

[5] Wullianallur Raghupathi and Viju Raghupathi. "Big data analytics in healthcare: promise and potential". In: *Health information science and systems* 2.1 (2014), p. 3.

[6] Richárd Farkas and György Szarvas. "Automatic construction of rule-based ICD-9-CM coding systems". In: *BMC bioinformatics*. Vol. 9. 3. BioMed Central. 2008, S10.

[7] Alvin Rajkomar et al. "Scalable and accurate deep learning for electronic health records". In: *arXiv preprint arXiv:1801.07860* (2018).

[8] Stephen M Griffies, William A Perrie, and Gaëlle Hull. "Elements of style for writing scientific journal articles". In: *Publishing Connect, Elsevier* (2013).

[9] Patrick Caldwell. "We've Spent Billions to Fix Our Medical Records, and They're Still a Mess. Here's Why." In: *Mother Jones* (Oct. 2015). URL: https://www.motherjones.com/politics/2015/10/epic-systems-judith-faulkner-hitech-ehr-interoperability/.

[10] Catarina Gomes. "Só 838 mil utentes se registaram na Plataforma de Dados da Saúde". In: *Público* (Sept. 2014). URL: https://www.publico.pt/2014/09/28/sociedade/noticia/so-838-mil-portugueses-se-registaram-na-plataforma-de-dados-da-saude-1670996.

[11] Stéphane M Meystre et al. "Extracting information from textual documents in the electronic health record: a review of recent research". In: *Yearb Med Inform* 35.8 (2008), pp. 128–144.

[12] Serguei Pakhomov, Ted Pedersen, and Christopher G Chute. "Abbreviation and acronym disambiguation in clinical discourse". In: *AMIA Annual Symposium Proceedings*. Vol. 2005. American Medical Informatics Association. 2005, p. 589.

[13] Hongfang Liu, Yves A Lussier, and Carol Friedman. "A study of abbreviations in the UMLS." In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 2001, p. 393.

[14] Patrick Ruch and Arnaud Gaudinat. "Comparing corpora and lexical ambiguity". In: *Proceedings of the workshop on Comparing corpora-Volume 9*. Association for Computational Linguistics. 2000, pp. 14–19.

[15] Teresa Gonçalves et al. "Analysing part-of-speech for portuguese text classification". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2006, pp. 551–562.

[16] Diogo Costa Leal. "Acordo ortográfico: Agora é obrigatório, mas não unânime". In: *JPN* (May 2015). URL: `https://jpn.up.pt/2015/05/13/acordo-ortografico-agora-obrigatorio-nao-unanime/`.

[17] Luís Miguel Oliveira Pereira. "Utilização de técnicas de text mining sobre registos clínicos de epilepsia em crianças, para auxílio ao diagnóstico e classificação". PhD thesis. Instituto Politécnico de Leiria, 2013.

[18] Mélanie Gonçalves Castro. "Utilização de Text Mining para Apoio à Classificação de Registos de Alergias e Reações Adversas". In: (2017).

[19] Liliana Ferreira, António Teixeira, and Joao Paulo Silva Cunha. "Medical information extraction in European Portuguese". In: *Handbook of Research on ICTs for Human-Centered Healthcare and Social Care Services*. IGI Global, 2013, pp. 607–626.

[20] Mourad Elloumi and Albert Y Zomaya. *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*. Vol. 23. John Wiley & Sons, 2013.

[21] José Gabriel Lopes, Nuno Cavalheiro Marques, and VJ Rocio. "Polaris: POrtuguese lexicon acquisition and retrieval interactive system". In: *The Practical Applications of Prolog* (1994), p. 665.

[22] David Campos, Sérgio Matos, and José Luís Oliveira. "Biomedical named entity recognition: a survey of machine-learning tools". In: *Theory and Applications for Advanced Text Mining*. InTech, 2012.

[23] Robert Leaman, Graciela Gonzalez, et al. "BANNER: an executable survey of advances in biomedical named entity recognition." In: *Pacific symposium on biocomputing*. Vol. 13. Big Island, Hawaii. 2008, pp. 652–663.

[24] Fátima Al-Shahrour et al. "BABELOMICS: a systems biology perspective in the functional annotation of genome-scale experiments". In: *Nucleic acids research* 34.suppl_2 (2006), W472–W476.

[25] Chih-Hsuan Wei et al. "tmVar: a text mining approach for extracting sequence variants in biomedical literature". In: *Bioinformatics* 29.11 (2013), pp. 1433–1439.

[26] Fei Zhu et al. "Biomedical text mining and its applications in cancer research". In: *Journal of biomedical informatics* 46.2 (2013), pp. 200–211.

[27] David Nadeau and Satoshi Sekine. "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[28] Olivier Bodenreider. "The unified medical language system (UMLS): integrating biomedical terminology". In: *Nucleic acids research* 32.suppl_1 (2004), pp. D267–D270.

[29] Micheal Hewett et al. "PharmGKB: the pharmacogenetics knowledge base". In: *Nucleic acids research* 30.1 (2002), pp. 163–165.

[30] Ada Hamosh et al. “Online Mendelian inheritance in man (OMIM)”. In: *Human mutation* 15.1 (2000), p. 57.

[31] Vladimir I Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[32] Peter Weiner. “Linear pattern matching algorithms”. In: *Switching and Automata Theory, 1973. SWAT'08. IEEE Conference Record of 14th Annual Symposium on*. IEEE. 1973, pp. 1–11.

[33] Melinda Katona and Richárd Farkas. “SZTE-NLP: Clinical Text Analysis with Named Entity Recognition”. In: ACL. 2014.

[34] Tim Rocktäschel, Michael Weidlich, and Ulf Leser. “ChemSpot: a hybrid system for chemical named entity recognition”. In: *Bioinformatics* 28.12 (2012), pp. 1633–1640.

[35] Anabel Usié et al. “CheNER: a tool for the identification of chemical entities and their classes in biomedical literature”. In: *Journal of cheminformatics* 7.1 (2015), S15.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[37] Christoph M Friedrich et al. “Biomedical and chemical named entity recognition with conditional random fields: The advantage of dictionary features”. In: *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM 2006)*. Vol. 7. BioMed Central Ltd, London UK. 2006, pp. 85–89.

[38] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In: *arXiv preprint arXiv:1607.04606* (2016).

[39] Roman Klinger et al. “Detection of IUPAC and IUPAC-like chemical names”. In: *Bioinformatics* 24.13 (2008), pp. i268–i276.

[40] John Lafferty, Andrew McCallum, and Fernando CN Pereira. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”. In: (2001).

[41] Asma Ben Abacha and Pierre Zweigenbaum. “Medical entity recognition: A comparison of semantic and statistical methods”. In: *Proceedings of BioNLP 2011 Workshop*. Association for Computational Linguistics. 2011, pp. 56–64.

[42] David Campos, Sérgio Matos, and José Luís Oliveira. “A modular framework for biomedical concept recognition”. In: *BMC bioinformatics* 14.1 (2013), p. 281.

[43] Ki-Joong Lee et al. “Biomedical named entity recognition using two-phase model based on SVMs”. In: *Journal of biomedical informatics* 37.6 (2004), pp. 436–447.

[44] Zhenfei Ju, Jian Wang, and Fei Zhu. “Named entity recognition from biomedical text using SVM”. In: *Bioinformatics and Biomedical Engineering,(iCBBE) 2011 5th International Conference on*. IEEE. 2011, pp. 1–4.

[45] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM. 1992, pp. 144–152.

[46] George Mavromatis. “Biomedical Named Entity Recognition Using Neural Networks”. In: ().

[47] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[48] Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. "An Investigation of Recurrent Neural Architectures for Drug Name Recognition". In: *arXiv preprint arXiv:1609.07585* (2016).

[49] Chen Lyu et al. "Long short-term memory RNN for biomedical named entity recognition". In: *BMC bioinformatics* 18.1 (2017), p. 462.

[50] Iñigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. "Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition". In: *Journal of biomedical informatics* 76 (2017), pp. 102–109.

[51] Christopher Olah. *Understanding LSTM Networks*. 2015. URL: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/` (visited on 04/26/2018).

[52] Micha Elsner, Eugene Charniak, and Mark Johnson. "Structured generative models for unsupervised named-entity clustering". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2009, pp. 164–172.

[53] Jimmy Lin and Dina Demner-Fushman. "Semantic clustering of answers to clinical questions". In: *AMIA Annual Symposium Proceedings*. Vol. 2007. American Medical Informatics Association. 2007, p. 458.

[54] Shaodian Zhang and Noémie Elhadad. "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts". In: *Journal of biomedical informatics* 46.6 (2013), pp. 1088–1098.

[55] Sérgio Matos, José Sequeira, and José Luís Oliveira. "BioinformaticsUA: Machine Learning and Rule-Based Recognition of Disorders and Clinical Attributes from Patient Notes". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 422–426. URL: `http://www.aclweb.org/anthology/S15-2073`.

[56] Parth Pathak et al. "ezDI: A Supervised NLP System for Clinical Narrative Analysis". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 412–416. URL: `http://www.aclweb.org/anthology/S15-2071`.

[57] Özlem Uzuner et al. "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 552–556.

[58] Berry de Bruijn et al. "Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 557–562.

[59] Eckhard Bick. "The parsing system Palavras". In: *Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework* (2000).

[60] Noémie Elhadad et al. "SemEval-2015 Task 14: Analysis of Clinical Text". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 303–310. URL: `http://www.aclweb.org/anthology/S15-2051`.

[61] Jun Xu et al. “UTH-CCB: The Participation of the SemEval 2015 Challenge – Task 14”. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 311–314. URL: http://www.aclweb.org/anthology/S15-2052.

[62] Sunil Kumar Sahu et al. “Relation extraction from clinical texts using domain invariant convolutional neural network”. In: *arXiv preprint arXiv:1606.09370* (2016).

[63] Vijay Garla, Caroline Taylor, and Cynthia Brandt. “Semi-supervised clinical text classification with Laplacian SVMs: an application to cancer case management”. In: *Journal of biomedical informatics* 46.5 (2013), pp. 869–875.

[64] John P Pestian et al. “A shared task involving multi-label classification of clinical free text”. In: *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Association for Computational Linguistics. 2007, pp. 97–104.

[65] Tal Baumel et al. “Multi-Label Classification of Patient Notes a Case Study on ICD Code Assignment”. In: *arXiv preprint arXiv:1709.09587* (2017).

[66] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. “Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]”. In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[67] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. “Beyond the point cloud: from transductive to semi-supervised learning”. In: *Proceedings of the 22nd international conference on Machine learning*. ACM. 2005, pp. 824–831.

[68] Colin Raffel and Daniel PW Ellis. “Feed-forward networks with attention can solve some long-term memory problems”. In: *arXiv preprint arXiv:1512.08756* (2015).

[69] Alistair EW Johnson et al. “MIMIC-III, a freely accessible critical care database”. In: *Scientific data* 3 (2016), p. 160035.

[70] Ishna Neamatullah et al. “Automated de-identification of free-text medical records”. In: *BMC medical informatics and decision making* 8.1 (2008), p. 32.

[71] Suzanne M Burns. *AACN Essentials of Critical Care Nursing*. 3rd ed. McGraw-Hill Education, 2014.

[72] Ami N Rubinowitz, Mark D Siegel, and Irena Tocino. “Thoracic imaging in the ICU”. In: *Critical care clinics* 23.3 (2007), pp. 539–573.

[73] François Chollet et al. *Keras*. https://keras.io. 2015.

[74] Martín Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).

[75] Armand Joulin et al. “Bag of tricks for efficient text classification”. In: *arXiv preprint arXiv:1607.01759* (2016).

[76] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[77] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[78] Christian Szegedy et al. “Going deeper with convolutions”. In: Cvpr. 2015.

[79] Jesse Read et al. “Classifier chains for multi-label classification”. In: *Machine learning* 85.3 (2011), p. 333.